Outstanding Contributions to Logic 25

Alessandra Palmigiano
Mehrnoosh Sadrzadeh *Editors*

Samson Abramsky on Logic and Structure in Computer Science and Beyond

Springer

Outstanding Contributions to Logic

Volume 25

Editor-in-Chief

Sven Ove Hansson, Division of Philosophy, KTH Royal Institute of Technology, Stockholm, Sweden

Outstanding Contributions to Logic puts focus on important advances in modern logical research. Each volume is devoted to a major contribution by an eminent logician. The series will cover contributions to logic broadly conceived, including philosophical and mathematical logic, logic in computer science, and the application of logic in linguistics, economics, psychology, and other specialized areas of study.

A typical volume of Outstanding Contributions to Logic contains:

- A short scientific autobiography by the logician to whom the volume is devoted
- The volume editor's introduction. This is a survey that puts the logician's contributions in context, discusses its importance and shows how it connects with related work by other scholars
- The main part of the book will consist of a series of chapters by different scholars that analyze, develop or constructively criticize the logician's work
- Response to the comments, by the logician to whom the volume is devoted
- A bibliography of the logician's publications Outstanding Contributions to Logic is published by Springer as part of the Studia Logica Library.

This book series, is also a sister series to Trends in Logic and Logic in Asia: Studia Logica Library. All books are published simultaneously in print and online. This book series is indexed in SCOPUS.

Proposals for new volumes are welcome. They should be sent to the editor-in-chief sven-ove.hansson@abe.kth.se

Alessandra Palmigiano · Mehrnoosh Sadrzadeh
Editors

Samson Abramsky on Logic and Structure in Computer Science and Beyond

Springer

Editors
Alessandra Palmigiano
Department of Ethics, Governance and Society
Vrije Universiteit Amsterdam
Amsterdam, The Netherlands

Mehrnoosh Sadrzadeh
Department of Computer Science
University College London
London, UK

ISSN 2211-2758 ISSN 2211-2766 (electronic)
Outstanding Contributions to Logic
ISBN 978-3-031-24116-1 ISBN 978-3-031-24117-8 (eBook)
https://doi.org/10.1007/978-3-031-24117-8

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Introduction to the Volume

This volume is inspired by and revolves around Samson Abramsky's outstanding contributions to logic. The role of logic in Samson's work is actually rather subtle. As a computer scientist, his primary focus has been the mathematical foundations of computation. In particular, the central theme of his work has been the quest for *intelligible structure*—for "rhyme and reason" as he puts it in his scientific autobiography included in this volume. One can say that this quest for rhyme and reason is inherently logical in character, in a sense deeper than the mere elaboration of formal logical systems, their syntax and semantics—although of course that is a key tool in this quest.

This quest has led Samson through a striking number of areas: starting with the semantics of computation, leading to his celebrated work on domain theory in logical form, and then to game semantics, which had a major impact in transforming the semantic paradigm, as a novel form of intentional semantics. His work then took another major turn, into the semantics and structural aspects of quantum computation, leading to categorical quantum mechanics. This led in turn to his work on contextuality, in quantum mechanics and beyond. This situates the essence of non-classicality in quantum mechanics as lying in the fact that its predictions generate data which is *locally consistent, but globally inconsistent*. The mathematical language of sheaf theory is used to articulate this—and allows the same phenomena to be observed in a wide range of situations beyond the quantum realm. His most recent work addresses the fundamental fault line in computer science between structure and power—the structural aspects studied in semantics, and the issues of expressiveness and complexity. His work on game comonads offers a new approach to bridging this gap.

Samson's work on these topics has involved a wide range of logics and logical methods, but also the extensive use of categorical methods, and a broad spectrum of other mathematical tools, including probability theory, convex geometry and linear programming, Hilbert spaces, lattice theory, topology, sheaves and cohomology. He is always searching for the logic and intelligible structure of the concepts he is studying. Logic and category theory are essential tools, but he is ready to use whatever other mathematics seems called for.

This same breadth of vision has led Samson to carry the insights and methods developed in the semantics of computation well beyond any narrow remit and into other fields: these include the foundations of quantum mechanics, linguistics, social choice theory and game theory, as well as a number of broad-ranging conceptual and philosophical discussions.

These considerations have led to the title of our volume:

Samson Abramsky on Logic and Structure in Computer Science and Beyond.

The volume has been structured according to the range of topics mentioned above. Part I, on Domains and Duality, contains chapters which relate to Samson's work on domain theory in logical form. Part II is on the theme of Game Semantics, with chapters which connect to Samson's pioneering and fundamental contributions in this field. Part III, on Contextuality and Quantum Computation, gathers contributions relating to Samson's work on contextuality in quantum mechanics and beyond. Part IV, on Game Comonads and Descriptive Complexity, addresses Samson's ongoing work on relating structure and power, and bringing disparate parts of computer science together.

Samson's broad and diverse contributions overflow any simple classification, and this is reflected by the final two parts of the volume. Part V, on Categorical and Logical Semantics, contains work relating to Samson's extensive contributions ranging over these topics; while Part VI, on Probabilistic Computation, reflects Samson's interest in the foundations of probability and its relation to logic, arising from his work in the foundations of quantum mechanics.

By using logic to bring out new meaning and systematic connections, and with his elegant and powerful vision, Samson has extraordinarily sharpened and expanded the meaning and potential of the logical toolbox itself. Through his contributions to fields spanning from game theory to the foundations of quantum physics, his leadership and ability to bring people from different communities together, Samson has also built a unique and vibrant community of researchers who have enthusiastically agreed to contribute to the present volume, which we hope will help to further propel Samson's ideas, vision and style into the future.

Alessandra Palmigiano
Mehrnoosh Sadrzadeh

Contents

Chapter 1
Logical Journeys: A Scientific Autobiography

Samson Abramsky

Abstract A short scientific biography emphasising the main phases of Abramsky's research: duality theory and domains in logical form, game semantics, categorical quantum mechanics, the sheaf-theoretic approach to contextuality, and game comonads and a structural view of resources and descriptive complexity.

Keywords Duality theory · Domains in logical form · Game semantics · Categorical quantum mechanics · Contextuality · Game comonads · Resources and coresources

1.1 Beginnings

High school mathematics did not work for me—in fact, it was something of a disaster. I experienced it as a bunch of arbitrary rules and tricks, with no underlying rhyme or reason. Science was no more appealing, for similar reasons, with an added penalty of lists of facts to be learnt. It was with a sense of relief that I turned to literature and history in my final year of high school. I entered Cambridge University to study English.

That lasted a week. The prospect of marching through the great works of English Literature, writing an essay a week, did not inspire me. Instead, I switched to a degree in Philosophy.[1] Philosophy has great-sounding problems which it promises to address: free will verses determinism, the nature of knowledge and existence, the foundations of everything. However, I found the actual outputs of Philosophy much less appealing, and the methods[2] unconvincing.

[1]Cambridge largely adhered in those days to single-subject degrees: "do one thing, and do it well". I believe it still does.

[2] At that time, in Cambridge in the 1970s, a blend of Wittgenstein, Oxford ordinary-language philosophy, and American analytical philosophy, heavily influenced by Quine and Davidson.

S. Abramsky (✉)
Department of Computer Science, University College London, London, UK
e-mail: samson.abramsky@cs.ox.ac.uk

A. Palmigiano and M. Sadrzadeh (eds.), *Samson Abramsky on Logic and Structure in Computer Science and Beyond*, Outstanding Contributions to Logic 25,
https://doi.org/10.1007/978-3-031-24117-8_1

It was, though, through Philosophy that I found logic, with mathematics lying beyond that. I was beginning to understand that there *was* rhyme and reason in mathematics, if it was presented in the right way. But there was no immediate path into these subjects from Philosophy within the confines of the Cambridge degree system. As I came to the end of my degree, I needed an escape route. I found it by taking the Diploma in Computer Science at Cambridge. Computing was a young, exciting field, although this was still the dinosaur era of mainframes, with the industry dominated by the mainframe manufacturers ("IBM and the Seven Dwarfs"). After the Diploma, I joined G.E.C. Computers Ltd in London,[3] working as an operating system developer. The company was relatively small, and in global terms a backwater, but it was a great opportunity to work directly on concurrent programming.

It was at this time that I came across Robin Milner's early work on modelling concurrent processes, which would soon develop into CCS. This was a nice example of serendipity. A couple of graduates from Edinburgh joined the company, and one of them was on the mailing list to receive the Edinburgh Computer Science technical reports—the orange booklets that some may remember. Being an inveterate browser, I noticed one of these on his desk, and picked it up—it was *Concurrent Processes and their Syntax*, and a new world opened up to me. I realised that the kind of problems I was grappling with on a practical level could be approached in a scientific manner, using a range of mathematical tools which at the time I was largely ignorant of.

Triggered in part by this experience, I applied to do a Ph.D., and started at Queen Mary College (as it then was), supervised by Richard Bornat, in 1978. My initial research topic was programming languages for distributed systems, which led to the Pascal-m project [1,2].[4] But before too long, I had gravitated (or sunk) to the foundational level, studying programming language semantics. Some early work on concurrency[5] attracted the attention of Robin Milner, who visited me at Queen Mary during a trip to London, and later invited me to give a talk at Edinburgh. Thus began my long-standing contacts with the theory group in Edinburgh, which became the Laboratory for the Foundations of Computer Science (LFCS) a few years later.

My first significant publications in this area were on the semantics of non-deterministic programs [3,4]. More importantly, during this period I was making up the deficit in mathematical background I had recognised when first encountering Robin Milner's work. I was relatively isolated intellectually. I had to learn by studying the literature. Above all, reading the papers of Robin Milner and Gordon Plotkin played a formative role. What appealed to me in their work was the harmonious interplay of conceptual and mathematical analysis. The mathematics acquired a direction and narrative from its use in understanding fundamental notions of computation; while the computational modelling achieved depth and rigour through its

[3] This was in Borehamwood. One of the buildings on that site had been owned by Elliot Brothers, where Tony Hoare had worked in the previous decade.

[4] References to my papers are to numbered items in the list of publications included in this volume. To avoid having a long bibliography appended to a short autobiographical essay, I will not give citations for the works of the many other authors whom I will mention.

[5] An unpublished 1981 QMC technical report, now available on the arXiv:1406.1965.

articulation as elegant mathematics. Not only did I acquire a great deal of technical information, I also absorbed, as much as I could, the point of view, the intellectual vision, the scientific perspective and methodology.

I might, perhaps, have learned more if I had been a student in one of the top research centres, working directly with a leader of the field. But perhaps it would have taken me longer to form my own style, my own approach.

Queen Mary appointed me to a Lectureship in 1980, just two years after I had started my Ph.D.[6] This was a great boon to me, and meant that I avoided the stressful years of moving from one post-doc to another which is the typical experience of an early-career researcher. Still, my intellectual isolation and the high teaching loads at Queen Mary led me to move to Imperial College in 1983. Together with Tom Maibaum, I started a Theory and Formal Methods group which would grow into a thriving activity, with over 30 members.

It was at Imperial College, almost immediately following my arrival, that I embarked on what was to be my first extended research program, which corresponds to the first part of this volume, on Domains and Duality.

1.2 Domains and Duality (1983–91)

Part of the process of becoming a researcher in the mathematical sciences is that one falls under the spell of some mathematical theory. One becomes a native speaker in the language of that theory, and a participant in its ethos, the tacit knowledge and attitudes that live "between the lines" of the formal texts. This is an essential prerequisite for developing a distinctive style of one's own. In my case, this first formation was in programming language semantics, and in particular domain theory, as the mathematical foundation for denotational semantics.

For many researchers, categories of domains are merely particular examples of models of certain type theories, which provide convenient settings for fixpoint semantics of recursion. By contrast, I saw domain theory as a *mathematical theory of information*, viewed qualitatively and topologically rather than quantitatively. The key idea was of partial information, reflecting the possibility of divergence in computation, and the partial order of information increase. I tried to lay out some of the ideas underlying this point of view in my contribution to a previous volume in this series [88].

The centrepiece of my work in domains and duality was *Domain theory in logical form* (DTLF). The first draft of this was written in 1983, the conference version appeared in LiCS 1987 [12], and the journal version in the LiCS special issue of Annals of Pure and Applied Logic in 1991 [18]. The paper won a LiCS Test of Time award in 2007. It formed the central chapter of my Ph.D.thesis,[7] the other main

---

[6] Which I had still not completed—in fact, this appointment meant that I was never to complete the thesis I had originally envisaged.

[7] My thesis—the one I eventually wrote—appeared in 1987, "which was rather late for me".

contributions being chapters on the lazy λ-calculus [15] and on a domain equation for bisimulation [19]. While both of these contained substantial ideas in their own right, they also played the role of worked examples for the general theory expounded in DTLF.

The key idea in DTLF was that one could use Stone duality as an organising principle for understanding the relationship between denotational semantics and program logic—understanding programs as points in a semantic space, or via their properties.

We start with an expressive metalanguage for denotational semantics, containing type constructions for sum, product, function space and powerdomain, and allowing full use of recursive types. On the spatial side of the duality, this can be interpreted in a suitable category of domains—in [18], this was taken to be SFP (countably based profinite domains). To capture domains "in logical form", a propositional theory is associated with each type in the metalanguage. Each type construction is interpreted as an operation on theories, so e.g. for function types $\sigma \to \tau$, $\mathcal{L}(\sigma \to \tau) = \mathcal{L}(\sigma) \to_{\mathcal{L}} \mathcal{L}(\tau)$, where on the right hand side we have the function space construction on theories applied to the theories inductively assigned to σ and τ. Recursive types are handled simply as inductive definitions of theories.

The meaning of terms in the metalanguage, which will denote programs, is then axiomatised in terms of their action as predicate transformers: a "Hoare triple" $\phi\{t\}\psi$ is interpreted denotationally as $[\![\phi]\!] \subseteq [\![t]\!]^{-1}([\![\psi]\!])$.

In this way, given a semantic specification of a programming language in the metalanguage, we can read off a syntactic presentation of the corresponding domain logic, unpacking the complex recursive types which may be involved. Particular examples where this methodology was used effectively included the lazy λ-calculus [15], synchronization tree models of concurrency [19], and the finitary non-well-founded sets [77].

The core result is a form of completeness theorem showing that the lattices presented by the axiomatization are exactly the Stone duals of the domains given by the standard denotational semantics. This then means that the corresponding logic is sound and complete.

The critical tension points in carrying through this program were in the axiomatization of function types and powerdomains. In particular, a key property used in the axiomatisation of the function space is preservation of coprimes, as later observed and shown to arise in other important situations as well by Mai Gehrke and Sam van Gool. Other continuations of this work can be found e.g. in work by Achim Jung and Drew Moshier, and Clemens Kupke, Alexander Kurz and Yde Venema. More generally, one can say that much of the work in coalgebraic logic over the past two decades is in a similar spirit.

1.3 Game Semantics (1992–2000)

By the time the journal version of DTLF appeared in 1991, I had already embarked on a new voyage. My main focus over the next decade was to be the semantics of interaction, and above all, game semantics. One of the factors leading to this was the arrival of Girard's Linear logic on the scene, as a disruptive influence, questioning settled assumptions, and "rousing me from my dogmatic slumbers".[8] Linear logic carved the world up more finely that traditional type theory and denotational semantics. In particular, it decomposed the standard exponential (function type construction) $A \Rightarrow B$ into the linear implication and the exponential modality: $A \Rightarrow B = !A \multimap B$. This "splitting the atom of (logic and) computation"[9] opened up the prospect of resource-sensitive logic and semantics. There was a quest for a semantics of linear logic which would properly reflect the dynamic intuitions behind it. Andreas Blass's game semantics for linear logic was a promising step in this direction, but there were some issues with it, as elaborated in [66]. This led to my first work on game semantics [22,32], done in 1992 with Radha Jagadeesan, who had joined me as a post-doc the year before. There were several key contributions of this paper, which I regard as the real start of game semantics as a paradigm in the semantics of programming languages:

- The formalization of games and strategies in terms of trace semantics, inspired by concurrency theory and process algebra, with composition of strategies defined in terms of "parallel composition plus hiding".
- Games and strategies were shown to organize themselves into categories with a rich type structure.
- The identification (and naming) of copy-cat strategies as key elements of the paradigm, interpreting identity morphisms, and more generally the structural morphisms which witnessed "dynamic tautologies".
- The focus on constraints on stategies—in [22,32], history-freeness—leading to strong definability results. In [22,32] the key notion of *full completeness* was identified: a model is fully complete if every morphism between interpretations of types comes from, *i.e.* is the denotation of, a proof. In other words, the semantic functor is full.

The obvious limitation of [32] was that it gave no account of the exponentials: the issue was how to marry these with history-freeness. It was clear that the stakes were high: if we could find a solution to this problem, then, given the success with full completeness, it seemed that a solution to the full abstraction problem for PCF, commonly seen as the outstanding problem in programming language semantics, should be at hand. Radha and I worked on this for several months: perhaps the most sustained period of focussing on a single technical question I have experienced. Eventually, we got there, with a formulation of taking plays modulo an equivalence. Initially, this

---

[8] Cf. Kant on Hume.

[9] *Interaction: splitting the atom of computation* was the title of my inaugural lecture at Edinburgh, given in 1996.

equivalence was defined via a permutation group acting on the exponential indices embedded in a move [34], somewhat reminiscent of nominal sets, then simplified to a given equivalence relation subject to some axioms [56]. The game semantics applied naturally to the multiplicative-exponential fragment of (intuitionistic) linear logic, which was sufficient to soundly interpret PCF. So we had a model. Now we had to confront the question: was it fully abstract? By this point, Pasquale Malacaria had joined us, having just completed his Ph.D. in Paris. The three of us—Radha, Pasquale and myself, the "AJM team"—worked together with great intensity and a sense of excitement, leavened by good humour and friendship, in one of the highpoints of my scientific life. We found the decomposition argument, which mapped any strategy onto a PCF term, thus yielding the key definability result. So we had a model which completely characterized the sequential processes which could be generated by PCF programs—which were typical of functional programming languages in general.

At this culminating moment of some two years of strenuous efforts, we learned that we were not alone. Martin Hyland and Luke Ong had developed a rather different approach to handling exponential types in game semantics, in terms of dialogue games and innocent strategies. This different construction, however, led to exactly the same fully complete model of PCF. These two constructions were announced together in 1993, and conference versions appeared in 1994. Later, we learned that Hanno Nickau had produced essentially the same construction as Hyland and Ong in his thesis. The journal versions of the AJM and HO papers were submitted together to Information and Computation in 1996, and appeared side-by-side in 2000.[10] In 2017, all six authors jointly received the Alonzo Church award for this work.

Back in 1993, our work encountered some resistance. The full abstraction problem as originally posed had asked for an *extensional* fully complete model.[11] However, programs were interpreted by *strategies* rather than by functions in the game semantics—the model was intensional. We could recover an extensional model, more specifically the fully abstract extensional model, [12] by quotienting the games model by a suitable observational preorder. But had we solved the full abstraction problem as originally intended?

This provoked a certain amount of debate at the time. What gave room for such a debate was that the "full abstraction problem" was not a clear-cut mathematical problem.[13] After all, Robin Milner, in the very paper which posed the full abstraction problem for PCF, had shown the existence of such a model by a term construction, *i.e.* built out of syntax, and had proved its uniqueness. The problem was to find a semantically natural and informative way of obtaining this construction. Digging a little deeper, one could say this asked for a naturally described semantic universe

[10] The long delay between submission and publication was caused purely by editorial inaction. The reviews we eventually received were perfunctory.

[11] It was not formulated in this way, but the requirement of capturing the observational preorder amounted to asking for an ω-algebraic cpo-enriched extensional model such that every compact element was PCF-definable.

[12] Milner had shown the uniqueness of this model in his seminal paper on full abstraction.

[13] One might say, channelling Bill Shankly: it was more interesting than that.

which gave rise to a cartesian closed category (ccc) which contained the fully abstract model as a full sub-ccc. But what is “natural”?

An obvious question arose: could one do better than the games model? That is, was there a direct, syntax-independent way of defining the extensional fully abstract model? Achim Jung and Alley Stoughton proposed a necessary condition for the existence of such a construction: the fully abstract model of Finitary PCF, the version of PCF over the booleans and without recursion, would be effectively presentable, and the observational preorder on terms of Finitary PCF would be decidable. Any hope for such a direct construction was therefore ruled out by Ralph Loader’s remarkable result that the observational preorder for Finitary PCF is undecidable.[14]

Another notable contribution was by Peter O’Hearn and John Riecke. Building on earlier work by Kurt Sieber and by Achim Jung and Jerzy Tiuryn, they gave an elegant construction of the fully abstract model for PCF using Kripke logical relations.

In my view, however, the decisive advantage of game semantics was precisely that it established a new semantic paradigm, rather than trying to “save the appearances” of an old one. In this new paradigm, programs are interpreted as strategies, so that the meaning of a program is given by its potential interactions with the environment. Composition of strategies is given by playing them off against each other in a given locus of interaction (the “Cut formula”), so that each actualises part of the environment of the other.[15] This refines the simple composition of functions. From the point of view of purely functional programming, all this additional structure may seem like a detour to allow sequentiality to be captured.

I saw things differently. There was a rich semantic space of strategies as interactive processes. To obtain the fully complete models for PCF, one had to *constrain* this space of strategies. There were two main kinds of constraint: innocence or history-freedom, whereby the locality of the information available to a subterm in a purely functional computation was captured; and well-bracketing, capturing the “stack discipline” of properly nested function calls and returns. Relaxing one or other of these constraints led to models for non-functional features. In particular, relaxing innocence led to models of local state; while relaxing well-bracketing led to models of non-local control. Moreover, these models were fully abstract “on the nose”, without any need for taking quotients. Thus what has been called the “Abramsky cube” [49] emerged, and by extension, a world of possibilities for modelling a wide range of computational features. This was to lead to a second phase of development of game semantics.

Interlude: a change of scene In 1995, I moved from Imperial College to Edinburgh, as the first incumbent of the Chair of Theoretical Computer Science established when

[14] On the first occasion I met Ralph, in Henk Barendregt’s office at CWI, shortly after his earlier result on the undecidability of λ-definability in the full type hierarchy over a finite set, I suggested to him that he prove this result for Finitary PCF, which I felt in my bones must be true. Of course, as Leonard Cohen observed, “feelings come and go”. By contrast, Loader’s theorem stands as a high point of the field. His departure from research soon after obtaining this result was a real loss to our subject.

[15] This makes essential use of the Player/Opponent duality of two-person games.

Robin Milner left Edinburgh for Cambridge. I joined LFCS, which was the largest and most prominent theory group in the U.K. The period ahead was to be one of considerable organizational change at Edinburgh, in which the School of Informatics was formed out of several previously separate Departments and Institutes.

The move more or less coincided with the transition from the first phase of game semantics, mainly focussed on the full abstraction result for PCF and extensions to richer functional languages, to a second phase, in which a much wider landscape of computational features would be explored. My last student at Imperial was Guy McCusker, whose thesis work on game semantics for a functional metalanguage had won the Kleene Award at LiCS. His thesis won the BCS/CPHC Distinguished Dissertation award. I had an extended collaboration with Guy during my time in Edinburgh, with work on local state [41,52] and call by value [46]. Our paper with Kohei Honda on game semantics for general references in LiCS 1998 [47] won the Test-of-Time award in 2018. My first student in Edinburgh, Jim Laird, made an important contribution in his thesis to studying non-local control from the game semantics perspective. Many others contributed to this program of developing game semantics as a very flexible and powerful paradigm for constructing highly structured, fully abstract semantics for languages with a wide range of computational features. Besides those already mentioned, these included nondeterminism, concurrency and probability. In many cases, game semantics has yielded the first, and often still the only, semantic construction of a fully abstract model for the language in question. In each case, finding the right conditions on strategies to characterize the semantic universe yields an analysis of the computational feature. One can see in this the beginnings of a *fine structure theory of processes*. Moreover, the original full completeness result for PCF was a keystone in the whole mathematical development.[16]

Two other strands of work in game semantics began towards the end of my time in Edinburgh. One was on algorithmic game semantics, initiated by Dan Ghica and Guy McCusker. This was a trend I saw the importance of, and strongly encouraged [62]. The other was on concurrent games, in joint work with Paul-André Melliès [53].

Postlude: another change of scene In 1999, Tony Hoare and Robin Milner, probably the two most eminent computer scientists in the U.K., retired from their chairs at Oxford and Cambridge respectively. I was appointed to the Christopher Strachey Chair of Computing at Oxford in 2000.[17]

This move coincided with a major shift in emphasis of my research. I would continue to do work on game semantics [67,68,69,72,91], and several of my students in Oxford would write theses in the area, notably Nikos Tzevelekos,[18] Matthijs Vákár, and Norihiro Yamada. But after a decade of intensive work on game semantics and allied topics, I was open to new challenges, and willing to look in new directions.

---

[16] One way of proving completeness for an extended language was by a factorization theorem, which reduced this to full completeness for PCF [41,49].

[17] I was actually the first incumbent of the chair under this name: Tony had held it as the James Martin Professor.

[18] Nikos also won the Kleene award.

1.4 Categorical Quantum Mechanics (2001–2009)

I had two main ideas in mind when contemplating a new research direction.

- One was that theoretical computer science, and in particular the semantics of programming languages, had introduced or greatly developed several important methodological ideas, which were ripe for application much more widely in the sciences. Perhaps the most important of these was compositionality. For some discussion of this, see [92].
- Another was on the mathematical side, to make links between the logical and semantical approach, and more mainstream mathematical ideas. This has long been a preoccupation of model theory and descriptive set theory within mathematical logic, but different kinds of connections seemed possible stemming from the Computer Science perspective. There was some early work with Prakash Panangaden and Rick Blute, looking at classical analytical structures as part of a generalized theory of "weighted" relations [50]. I made some attempts to go further in this direction during my first couple of years in Edinburgh, but these did not immediately lead anywhere.

In the end, both of these found expression in my work on foundations of quantum mechanics and quantum information and computation, which began following my move to Oxford. These two areas are closely linked. Quantum computation is premised on the idea of *quantum advantage*: that there are information processes which can be performed better, or only performed at all, using quantum resources, as compared with purely classical resources. The source of quantum advantage is thus to be found in the non-classicality of quantum mechanics—exactly the features which render it problematic from a foundational point of view.

The study of quantum foundations and quantum information and computation draws on a wide range of mathematics: linear and multilinear algebra, operator algebras, probability theory, convex geometry and linear and semidefinite programming, as well as logic and algebra. From the computational point of view, quantum mechanics challenges the basic assumptions on which our models of classical information processing are based—and does so with the authority of our best-confirmed physical theory. Thus the quantum realm offered everything I was looking for, and more.

My starting point was my previous work on geometry of interaction and interaction categories (of which more later). The ideas started to take shape in collaboration with Bob Coecke, who I recruited as a post-doc in 2001. This led to the formulation of categorical quantum mechanics, and its application to the derivation of teleportation and other quantum protocols [71,70,73,79]. This work showed that methods and concepts developed in theoretical computer science can be applied very directly in quantum information, and the foundations of quantum mechanics itself. The axiomatic foundations of quantum mechanics can be recast in the language of category theory, providing a rigorous basis for high-level methods for reasoning about quantum protocols and quantum computational models. The categorical methods are accompanied by a very intuitive diagrammatic calculus, based on the string diagram representation of monoidal categories. These diagrams make the information flow in quantum protocols such as teleportation very explicit, and amenable to structured, high-level calculation. This greatly enhances the attractiveness and potential applicability of the approach, and provides the basis for automated tool support. There have been recent applications to optimizing quantum compilers. The diagrammatic methods have also influenced other work in quantum foundations, notably of Lucien Hardy, and of D'Ariano, Chiribella and Perinotti in their axiomatic reconstruction of quantum mechanics.

Categorical quantum mechanics and its subsequent developments, such as the ZX-calculus, has led to an extensive research activity, and given rise to two textbooks. There has also been an impact on industry, particularly the quantum computing company Quantinuum. Other off-shoots include work on compositional distributional semantics in computational linguistics.

After establishing the basic categorical quantum mechanics paradigm, my subsequent work in this area included relating no-cloning and no-deleting to fundamental consistency results in categorical logic [94], and studying free constructions of compact closed categories and related structures [76]. With Ross Duncan, I related categorical quantum mechanics to categorical logic and type theory [80]. With Chris Heunen, I explored the extension of categorical quantum mechanics to the infinite-dimensional setting [103], and the relationship to operational theories [123]. I also developed connections with knot theory and Temperley-Lieb algebra [84], and in turn made connections between these and "planar λ-calculus", a topic which later would be developed further by Noam Zeilberger.

While categorical quantum mechanics has proved very fruitful, it essentially provides an elegant and convenient diagrammatic formalism, underpinned by general categorical foundations, for (mostly finite dimensional) quantum mechanics. What was most interesting to me was to try to understand the non-classicality inherent in quantum mechanics, expressed most characteristically in terms of its non-locality and contextuality. I wanted to study more deeply the mathematical and logical structures at the root of this non-classicality. This led me to a new research agenda, organized around a new mathematical language—that of sheaf theory.

1.5 Contextuality (2010–Present)

A cardinal principle I have followed is that if contextuality is to be applied meaningfully as a property of quantum mechanics, it must be *defined* completely independently of quantum mechanics. Moreover, if it is defined as generally as possible, we can see where it applies more broadly, in non-quantum situations.

My first step in this direction was prompted by a paper by Adam Brandenburger and Noson Yanofsky, in which they carefully define a notion of probabilistic empirical models independently of quantum mechanics, and study general properties of these models, and of their realizations by hidden variables. This subsumes much of the previous discussion of hidden variables and no-go theorems in the literature on quantum foundations.

My first insight in [106] was that their analysis carries over to a purely relational, possibilistic setting, without probabilities. In this form, it has a direct logical formulation. This has been very attractive to logicians, and led to several cross-over works in which logical methods have been applied to quantum foundations. I also studied the connection between probabilistic and possibilistic models in some depth in [106],[19] and introduced the notion of quantum realization of an empirical model. This led to the formulation of a number of natural computational problems. The complexity of some of these was obtained via their logical description, but the decidability of determining whether an empirical model has a quantum realization of arbitrary dimension was left open. This was subsequently to be resolved (in the negative) by the spectacular results of Slofstra, and Ji, Vidick, Wright and Yuen.

Many of the ideas and themes which would occupy me over the next decade were initiated in this paper.[20] However, something important was missing. The setting of this paper is that of *Bell scenarios*, where n agents each has a number of available measurement settings, and they operate independently of each other. This provides a suitable setting for the study of non-locality. However, it does not capture the more general situations studied under the heading of contextuality, where measurement bases can overlap in more complicated ways. This more general setting is needed for the Kochen-Specker theorem and its many off-shoots.

Adam Brandenburger raised this with me during a meeting in Copenhagen in 2010. This led to an intense collaboration, resulting in [99], which introduced the "sheaf-theoretic approach" to contextuality, which has opened up a wealth of new insights.

We take contextuality in its broadest sense to arise where we have a family of data which is *locally consistent*, but *globally inconsistent*. Since sheaf theory is the mathematical language for talking about passages from local to global and their obstructions, it is a canonical choice of mathematical formalism to use in describing these situations.

In the probabilistic case, contextuality refers to the situation where, instead of having a probability measure defined on a single sample space, we have a family of overlapping sample spaces, each with its own probability measure. Such families arise naturally in quantum mechanics, where each of the sample spaces corresponds to a family of compatible observables, which can be performed together in a physically realizable experimental set-up—a *context*. The corresponding probability distribution for that context has direct empirical or observable content.

[19] And later, with Shane, Rui and Kohei, in [125].

[20] Which was written in 2010. The publication date of the journal article, as is often the case, is misleading.

These measures will be *locally consistent*—they will have consistent marginals on their overlaps—but *globally inconsistent*—there will be no joint distribution on the global space of all observables which marginalizes to yield the empirically accessible probabilities. This failure of global consistency corresponds to the impossibility of a classical hidden variable model.

In [99], a general mathematical setting using tools from category theory and sheaf theory was developed for studying contextuality and non-locality. This was able to unify the many forms of argument which have appeared in the literature, including those essentially using probabilities (as in the use of Bell inequalities to prove Bell's theorem), and those of a logical form, using possibilities, such as the Kochen-Specker, GHZ and Hardy paradoxes. This led to a strict hierarchy of strengths of contextuality, generalizing and distinguishing the various kinds of arguments found in specific constructions in the literature, and exemplified by

$$\text{Bell} \; < \; \text{Hardy} \; < \; \text{GHZ}$$

The top level was named strong contextuality in [98]; it has played a crucial role in subsequent developments.

One of the fascinating things about contextuality is that it has so many facets and different aspects. As well as probabilistic and quantative aspects, such as Bell and contextual inequalities and measures of contextuality, there are structural and qualitative features. Part of the power of the sheaf-theoretic approach is that it well adapted to bringing these clearly into view.

Cohomology An idea that I was keen to pursue was to use *cohomology* to detect contextuality. Cohomology is one of the major tools of modern mathematics, and I believe that potentially it can be a powerful weapon in computer science and logic. Although presheaves and sheaves have been widely used in the semantics of computation, there has been little or no use of cohomology. It seemed to me that contextuality, which is from the sheaf-theoretic perspective exactly about obstructions to the existence of global sections because of "logical twisting", called out for a cohomological description.

My quest for a way to realize this idea led to several months of intensive work—my hardest effort to find the right mathematical formulation of an idea since my work on the exponentials for game semantics. Eventually, I found a viable, and indeed very natural approach, using the Čech cohomology of an abelian presheaf associated with the presheaf of supports of an empirical model. I suggested to Shane Mansfield and Rui Soares Barbosa, then my students, that they work on this with me, and this led to our first joint paper [104], and the start of a long, very fruitful, and still continuing collaboration, and a lasting friendship.

This analysis was taken further in [120], written with Kohei Kishida and Ray Lal as well as Shane and Rui. As well as clarifying the cohomological construction, connections were made to logical paradoxes, and to All-versus-Nothing arguments, which are important in contextuality.[21] A proposed characterisation of All-versus-Nothing arguments in the Pauli n-group—the "AvN Triple Conjecture"—was later proved in a paper with Rui, my then student Giovanni Carù, and Simon Perdrix [130].

Databases and constraints The sheaf-theoretic view of contextuality allowed the remarkably close structural similarity with central notions in the theory of relational databases to be observed in [109]. These include connections between Vorob'ev's Theorem and database acyclicity. The relational formulation of non-locality in [106] is also related to robust constraint satisfaction in [108], written with Georg Gottlob and Phokion Kolaitis. The relationship between contextuality and valuation algebras, a general formalism for generic inference, was studied in a paper with Giovanni [135].

Hardy is almost everywhere It is shown in [124], written with my then student Carmen Constantin and Shengang Ying, that almost all n-qubit states—all but the product states and the maximally entangled states—admit a "Hardy paradox", *i.e.* a probability-free proof of non-locality. Moreover, the construction is effective, yielding an algorithm which given a state, produces measurements which give a witness for the non-locality proof.

Logical Bell inequalities The connection between structural and quantitative aspects was reinforced in subsequent work with Lucien Hardy on "logical Bell inequalities" [101]. Bell inequalities are a central method in quantum information. But what *are* Bell inequalities in general? We show that they all arise from purely logical consistency conditions on the underlying events. Moreover, this is shown in great generality, for all contextuality scenarios. This can be seen [138] as answering Boole's question on the "conditions of possible experience", whose connection to Bell inequalities had been observed by Itamar Pitowsky.

Contextual fraction and resources A striking recent development in physics has been the study of *resource theories*, with applications ranging from thermodynamics to quantum information and computation. Two important aspects of these theories are the identification of "free operations", which can be performed without consuming the resource, and "monotones", quantitative measures of the resource for which the free operations are monotonic.

Shane, Rui and I developed the first resource theory for contextuality [129], and derived a range of free operations, and an important measure of contextuality, the *contextual fraction*, which is a monotone for this theory. We used the duality theory of linear programming to relate the contextual fraction to maximum violation of Bell inequalities. We also found a form of *resource inequality* involving the contextual fraction which we have shown to be applicable in a wide range of computational situations, including measurement-based computation, non-local games, communication complexity, sequential transformations, computation with shallow circuits, and quantum random-access codes.

[21] They would also turn out to be important in a completely different context, as we will see later.

With Rui, Shane, Giovanni, Kohei and my former student Nadish de Silva, we characterized the minimum resources needed for strong non-locality in three-qubit systems [128].

Comonadic simulation With Shane, Rui and Martti Karvonen,we developed a structural approach to simulation, which allows *comparison* of resources [135]. To get a sufficiently general view of simulation, which allows e.g. for the use of classical randomness and classical information processing, we are led to introduce a comonad of measurement protocols, which feature adaptive behaviour—the result of one measurement can be used to determine the choice of a subsequent measurement. This comonadic view is shown to correspond exactly to the algebraic view provided by the free operations.

The quantum monad With Rui, Nadish, and Octavio Zapata, we introduced the *graded quantum monad* $\mathcal{Q}_d$ in [131]. This encapsulates the non-classical possibilities opened up by the use of quantum resources in a graded monad. Here the grade d gives the dimension of the Hilbert space providing the quantum resource. Kleisli maps $A \to \mathcal{Q}_d B$ are shown to subsume the quantum homomorphisms of Mančinska and Roberson, and the quantum constraint systems of Cleve and Mittal. Systematic connections are shown between: (1) quantum contextuality, a key non-classical ingredient of quantum mechanics; (2) non-local games, a major source of examples of quantum advantage in information processing; and (3) the existence of quantum solutions to classical constraint satisfaction problems.

Partial Boolean algebras and duality In current work with Rui [140], we have been revisiting partial Boolean algebras, the formalism used by Kochen and Specker in their seminal work on contextuality. We have related them to the sheaf-theoretic approach, and identified a logical exclusivity principle and studied its relationship to the probabilistic exclusivity principle, which has been considered in recent work on contextuality as a step towards closing in on the set of quantum-realisable correlations. We made partial progress towards the goal of a fully logical characterisation of the Hilbert space tensor product. We have also developed a duality theory between complete atomic partial Boolean algebras and a category of reflexive graphs and certain relations between them, generalising the standard duality between complete atomic Boolean algebras and sets.

1.6 Resources, Game Comonads, and Relating Structure to Power (2017–Present)

In 2016 I was co-organizer[22] of a very successful 4-month program at the Simons Institute for Theory of Computing on Logical Structures in Computation. This led to a number of new collaborations and research directions among the participants, and

[22] With Anuj Dawar, Phokion Kolaitis and Prakash Panangaden.

contributed to the emergence of new research communities, e.g. in Applied Category Theory and in logical aspects of quantum information.

An important effect for me happened right at the end of the program. I had been sharing an office with my co-organiser Anuj Dawar for four months. Following my work on the quantum monad with Rui, Nadish and Octavio, which had been done earlier in the program, some remarks Anuj had made a couple of months previously about a possible categorical formulation of pebble games began to resonate with me. I soon realised that there *was* a categorical construction there, but it was a comonad rather than a monad. I suggested to Anuj that we work together on elaborating this idea. This led to the paper [127] on the pebbling comonad with Anuj and his student Pengming Wang, which appeared in LiCS 2017.

This work suggested a wider agenda. There is a deep divide running through the foundations of computation, with two major organising principles, which we may refer to as Structure and Power:

Structure Compositionality and semantics, addressing the question of mastering the size and complexity of computer systems and software.

Power Expressiveness and computational complexity, addressing the question of how we can harness the power of computation and recognize its limits.

A striking feature of the current state of the art is that there are *almost disjoint communities* of researchers studying Structure and Power respectively, with very little in the way of common technical language or tools. In my opinion, this is a major obstacle to fundamental progress in Computer Science.

The ideas in [127] suggested a research program for relating Structure to Power in a substantial way. This was elaborated in a follow-up paper with Nihil Shah [134,143], and led to a joint project with Anuj. Contributors to this program include Dan Marsden, Luca Reggio, Tomáš Jakl, Tom Paine, Nihil Shah, Adam Ó Conghaile and Yoàv Montacute.

The key initial idea of this program as laid out in [127,143] is to encapsulate various forms of model comparison game, in which Spoiler tries to distinguish two structures, and Duplicator tries to show they are the same, as *resource-indexed comonads* on the category of relational structures. Intuitively, these comonads correspond to *co-resources*, which limit the ability of Spoiler to access the structure by bounding resources of some kind. Thus, if C_k is such a comonad, with resource index k, then to have a homomorphism

$$C_k A \to B$$

means that we only have to check the homomorphism conditions against limited—k-bounded—parts of the structure of A. This limitation of resources gives a syntax-free way of looking at limitations on exploring structures induced by considering only those properties definable in some logical language.

Some key features have emerged in the elaboration of this idea in subsequent work:

- This idea has proved to be extremely robust. It can be used to capture a wide range of model comparison games, and the resource-indexed equivalences induced by the corresponding logics. These include the quantifier rank fragments, the finite variable fragments, the modal, guarded, hybrid and bounded fragments, generalized quantifiers, and more.

- The basic idea of coresources $C_k A \to B$ seems at first blush restricted to forth-only equivalences, and hence to existential positive fragments. However, the ideas in fact extend smoothly to cover the full back-and-forth equivalences, using a refinement of the open maps formulation of bisimulation to *open pathwise embeddings*. Moreover, isomorphism in the coresource setting characterizes extensions of the logics with counting quantifiers.
- Coalgebras $A \to C_k A$ correspond to resource-bounded *structural decompositions* of A. The existence of such decompositions yields significant combinatorial invariants of A. This allows important combinatorial parameters such as *tree-width* and *tree-depth* to be recovered.
- The whole pattern of comonads, their coalgebras, and the corresponding resolutions into comonadic adjunctions, forms a robust template which recurs throughout finite model theory and descriptive complexity. This template has been axiomatized at a very general categorical level with Luca Reggio in [142]. This provides a new kind of axiomatic basis for model theory, finite and infinite.

We can regard the axiomatization in [142] as the culmination of a "first wave" of this research program. Results building on this initial phase of development are beginning to emerge. These include:

- General versions of model-theoretic results such as preservation theorems: Rossman's homomorphism preservation theorems, the van Benthem-Rosen theorem on bisimulation invariance, etc. (Reggio, Paine and Abramsky).
- Uniform proofs of preservation theorems in the finite and infinite cases: "model theory without compactness". (Abramsky and Marsden)
- Structural features of comonads (idempotence, bisimilar companions property), and their significance for computational tractability. (Abramsky and Reggio)
- Lovász-type theorems on counting homomorphisms (Dawar, Jakl and Reggio)
- Combinatorial parameters: concrete cases, and an axiomatic approach via density comonads (Abramsky, Jakl and Paine).
- Feferman-Vaught-Mostowski type theorems, with applications to Courcelle's theorem (Jakl, Marsden, and Shah).

An exciting recent development is the proposal by Adam Ó Conghaile of new cohomological approximation algorithms for constraint satisfaction and structure isomorphism. This makes use of the sheaf-theoretic and cohomological methods originally developed for quantum contextuality in [120].

There are hopeful signs of a growing interest in this broad research agenda. A successful workshop on *Structure meets Power* was held in association with LiCS 2021, and a second workshop on this theme will be held alongside ICALP 2022.

1.7 Supplements

Our narrative has focussed on the main themes of my research during successive periods. This is a neat story—too neat to cover everything. To achieve some narrative flow, it is necessary to simplify and select.

I will not attempt to describe all the other things I have worked on during these years. These include strictness analysis and abstract interpretation [7,8,9,10,16,20], testing equivalences, observational logic and quantales [11,24], a general Kahn principle for asynchronous networks [17], proofs as processes [33], specification structures [42], process realizability [54], realizability and full completeness [57,58,60], game theory and epistemic paradoxes [118,126], multi-player games [79,85], Arrow's theorem [117], computational semantics of thermodynamics [121], semantics of natural language [110], and Dependence logic [90].

I will, however, mention a couple of items in slightly more detail.

Firstly, the "game semantics" period, 1992–2000, was in fact a "semantics of interaction" period. In particular, there were two other strands which were an important part of my work during this period:

- **Interaction categories** The work on interaction categories [28,29,37] was a pioneering attempt to develop a categorical semantics of concurrency, which anticipated current ideas on process categories, string diagrams, and compact closed structure *inter alia*. In collaboration with my then students Simon Gay and Raja Nagarajan, we also pioneered (strong) types for concurrency [38,44,51]. This work had less impact when it appeared than I had hoped. Perhaps it was ahead of its time.
- **Geometry of Interaction and traced monoidal categories** The contributions here [21,22,31,39,63] emphasize the semantic and structural aspects of what is somewhat nebulously referred to as "Geometry of interaction". A wide range of examples was identified, in two main styles: "particle" (or coproduct-based) [22], and "wave" (or product-based) [21].[23] The diagrammatic understanding of composition as symmetric feedback, formalised via the trace, was also pioneered in this work. Conceptually, the Int construction of Joyal, Street and Verity amounts to interpreting linear higher order structure (more specifically, compact closed structure) in a first-order setting (traced monoidal categories). This can be extended to a full interpretation of combinatory logic, a universal model of computation, in terms of reversible automata, as shown in [88,61,78]. It can also be seen as a primitive form of untyped game semantics: strategies without games. There have been some very interesting further developments in the subsequent literature, e.g. in the work of Ichiro Hasuo and Naohiko Hoshino, and of Furio Honsell, Marina Lenisa, Alberto Ciaffaglione and Ivan Scagnetto.

[23] Note that these concrete constructions using different incarnations of the trace anticipated the paper by Joyal, Street and Verity on *Traced Monoidal Categories* by several years. I explained them at some length to André Joyal during LiCS 1993 in Montreal.

Secondly, I have also written quite a number of expository articles [30,40,62,93,98] and surveys [49,92,112], discursive articles [81,97,107,137], and even excursions back to, or at least near, the world of philosophy I left so long ago [88,114,144]. The influence of Johan van Benthem, in particular, was important in encouraging me to go in this direction. Whether this is to his credit or not I leave to the reader to judge. More recently, my former student Yoshihiro Maruyama has also encouraged me to relate my ideas to broader intellectual horizons.

1.8 People

I have deliberately written this account as a narrative of ideas and scientific contributions. In particular, I have avoided "personalia".[24] But of course, the people I have encountered along the way have been an essential part of the story. Many of them have been mentioned in the preceding pages. And there are many others!

In particular, the contributors to this volume are not only outstanding scientists, but my friends. The editors, Alessandra Palmigiano and Mehrnoosh Sadrzadeh, deserve special mention: they initially proposed the volume to Springer, and they have made it happen.

I have always found Wordsworth's lines on Newton rather haunting:

a mind forever
Voyaging through strange seas of Thought, alone.

Science can be a lonely endeavour at times, but I have been fortunate to have had many good companions during my journey. My heartfelt thanks goes to them all.

1.9 Envoi

History is "one damn thing after another". Biography is the projection of history onto a single person's timeline. Autobiography is the restriction of biography to the subject's experiences. What hope, then, for a scientific autobiography to be more than one damn theorem after another, or a succession of conferences, research trips, academic positions, a reconstituted c.v.?

I have tried to indicate the flow of ideas, the quest for rhyme and reason behind the scientific work I have done; which takes us back to our starting point. But the serpent has not quite swallowed its tail yet. In 2021, I started a new position as Professor of Computer Science at University College London, and a new research project on *Resources in Computation*.

The next chapter is still to be written …

[24] I have in fact written, and even published, a short autobiographical essay which is a sort of dual to this one.

List of Publications

Samson Abramsky
1982

1. S. Abramsky and S. Cook, “Pascal-m in Office Information Systems”, in *Office Information Systems*, N. Naffah, ed. (North Holland) 1982.

1983

2. S. Abramsky and R. Bornat, “Pascal-m: a language for the design of loosely coupled distributed systems”, in *Distributed Computing Systems: Synchronization, Control and Coordination*, Y. Paker and J.-P. Verjus, eds. (Academic Press) 1983, 163–189.
3. S. Abramsky, “Semantic Foundations for Applicative Multiprogramming”, in *Automata, Languages and Programming*, J. Diaz ed. (Springer-Verlag) 1983, 1–14.
4. S. Abramsky, “Experiments, Powerdomains and Fully Abstract Models for Applicative Multiprogramming”, in *Foundations of Computation Theory*, M. Karpinski ed. (Springer-Verlag) 1983, 1–13.

1984

5. S. Abramsky, “Reasoning about concurrent systems: a functional approach”, in *Distributed Systems*, F. Chambers, D. Duce and G. Jones, eds. (Academic Press) 1984, 307–319.

1985

6. S. Abramsky and R. Sykes, “SECD-M: a virtual machine for applicative multiprogramming”, in *Functional Languages and Computer Architecture*, J.-P. Jouannaud ed. (Springer-Verlag) 1985, 81–98.

1986

7. G. Burn, C. Hankin and S. Abramsky, “Strictness Analysis for Higher Order Functions”, *Science of Computer Programming* 7, (1986), 249–278.
8. S. Abramsky, “Strictness Analysis and Polymorphic Invariance”, in *Programs as Data Objects*, H. Ganzinger and N. Jones, eds. (Springer-Verlag) 1986, 1–23.
9. G. Burn, C. Hankin and S. Abramsky, “The Theory of Strictness Analysis for Higher Order Functions”, in *Programs as Data Objects*, H. Ganzinger and N. Jones, eds. (Springer-Verlag) 1986, 42–62.

1987

10. S. Abramsky and C. Hankin, “Introduction to Abstract Interpretation”, Ch. 1 in *Abstract Interpretation for Declarative Languages*, S. Abramsky and C. Hankin, eds. (Ellis Horwood) 1987, 9–31.

11. S. Abramsky, "Observation Equivalence as a Testing Equivalence", *J. Theoretical Computer Science* 53, (1987), 225–241.
12. S. Abramsky, "Domain Theory in Logical Form", in *Symposium on Logic in Computer Science*, (Computer Society Press of the IEEE) 1987, 47–53.

1988

13. D. Fuller and S. Abramsky, "Partial evaluation and MIX compilation in PROLOG", *New Generation Computing* 6, (1988), 119–143.

1989

14. S. Abramsky, "Observational Logic and Process Semantics (Abstract)", in *Logic at Botik 89*, A. Meyer and M. Taitslin, eds. (Springer-Verlag) 1989.

1990

15. S. Abramsky, "The Lazy λ-Calculus", in *Research Topics in Functional Programming*, D. Turner, ed. (Addison Wesley) 1990, 65–117.
16. S. Abramsky, "Abstract Interpretation, Logical Relations and Kan Extensions", *J. Logic and Computation*, 1(1) (1990), 5–41.
17. S. Abramsky, "A Generalized Kahn Principle for Abstract Asynchronous Networks", in *Symposium on Mathematical Foundations of Programming Language Semantics*, M. Mislove, ed. (Springer-Verlag) 1990, 1–21.

1991

18. S. Abramsky, "Domain Theory in Logical Form", *Annals of Pure and Applied Logic* 51, (1991), 1–77.
19. S. Abramsky, "A Domain Equation for Bisimulation", *J. Information and Computation*, 92(2) (1991), 161–218.
20. S. Abramsky and T. Jensen, "A Relational Approach to Strictness Analysis for Higher-Order Polymorphic Functions", in *ACM Symposium on Principles of Programming Languages*, (ACM Press) 1991, 49–55.

1992

21. S. Abramsky and R. Jagadeesan, "New Foundations for the Geometry of Interaction", in *Symposium on Logic in Computer Science*, (Computer Society Press of the IEEE) 1992, 211–222.
22. S. Abramsky and R. Jagadeesan, "Games and Full Completeness for Multiplicative Linear Logic", in *Foundations of Software Technology and Theoretical Computer Science*, R. Shyamsundar, ed. (Springer-Verlag) 1992, 291–301.

1993

23. S. Abramsky and C.-H. L. Ong, "Full Abstraction in the Lazy λ-calculus", *Information and Computation*, 105(2) (1993), 159–268.

24. S. Abramsky and S. J. Vickers, "Quantales, Observational Logic and Process Semantics", *Mathematical Structures in Computer Science*, (1993), vol. 3, 161–227.
25. S. Abramsky, "Computational Interpretations of Linear logic", *J. Theoretical Computer Science*, 111 (1993), 3–57.
26. A. Finkelstein, J. Kramer, S. Abramsky, K. Broda, S. Drossopoulou and S. Eisenbach, "An Integrated Engineering Study Scheme in Computing", *Computer Journal*, 36(4), (1993), 320–334.
27. I. Mackie, L. Roman and S. Abramsky, "An Internal Language for Autonomous Categories", *Journal of Applied Categorical Structures*, vol. 1, (1993), 311–343.
28. S. Abramsky, "Interaction Categories (Extended Abstract)", in *Theory and Formal Methods 1993*, G. L. Burn, S. J. Gay and M. D. Ryan, ed. (Springer-Verlag) 1993, 57–69.

1994

29. S. Abramsky, "Interaction Categories and Communicating Sequential Processes", in *A Classical Mind: Essays in honour of C. A. R. Hoare*, A. W. Roscoe, ed. (Prentice Hall International) 1994, 1–16.
30. S. Abramsky and A. Jung, "Domain Theory", in *Handbook of Logic in Computer Science*, S. Abramsky, D. Gabbay and T. S. E. Maibaum, eds., (Oxford University Press), 1994, 1–168.
31. S. Abramsky and R. Jagadeesan, "New Foundations for the Geometry of Interaction", *Information and Computation*, 111(1), (1994), 53–119.
32. S. Abramsky and R. Jagadeesan, "Games and Full Completeness for Multiplicative Linear Logic", *Journal of Symbolic Logic*, (1994), vol. 59, no. 2, 543–574.
33. S. Abramsky, "Proofs as Processes", *J. Theoretical Computer Science* vol. 135 (1994), 5–9.
34. S. Abramsky, R. Jagadeesan and P. Malacaria, "Full Abstraction for PCF (Extended Abstract)", in *Theoretical Aspects of Computer Software*, M. Hagiya and J. C. Mitchell, eds., (Springer-Verlag) 1994, 1–15.

1995

35. S. Abramsky and G. McCusker, "Games and Full Abstraction for the Lazy λ-calculus", in *Proceedings of the Tenth Annual Symposium on Logic in Computer Science*, D. Kozen, ed. (IEEE Computer Society Press), 1995, 234–243.
36. S. Abramsky and G. McCusker, "Games for recursive types", in *Proceedings of the 1994 Workshop on Theory and Formal Methods*, C. L. Hankin, ed. (Imperial College Press), 1995, 1–20.

1996

37. S. Abramsky, S. Gay and R. Nagarajan, "Interaction Categories and the Foundations of Typed Concurrent Programming", in *Proceedings of the 1994 Marktoberdorf Summer Sxhool on Deductive Program Design*, M. Broy, ed. (Springer-Verlag) 1996, 35–113.

38. S. Abramsky, S. Gay and R. Nagarajan, "Specification Structures and Propositions-as-Types for Concurrency", in *Logics for Concurrency: Structure vs. Automata—Proceedings of the VIIIth Banff Higher Order Workshop*, G. Birtwistle and F. Moller, eds. (Springer-Verlag) 1996, 5–40.
39. S. Abramsky, "Retracing some paths in process algebra", in *CONCUR '96: Concurrency Theory, 7th International Conference*, U. Montanari and V. Sassone, eds. (Springer-Verlag) 1996, 1–17.

1997

40. S. Abramsky, "Semantics of Interaction: an introduction to Game Semantics", in *Proceedings of the 1996 CLiCS Summer School, Isaac Newton Institute*, P. Dybjer and A. Pitts, eds. (Cambridge University Press) 1997, 1–31.
41. S. Abramsky and G. McCusker, "Linearity, Sharing and State: a fully abstract game semantics for Idealized Algol", in *Algol-like Languages*, P. O'Hearn and R. D. Tennent, eds. (Birkhauser) 1997, 317–348.
42. S. Abramsky and D. Pavlovic, "Specifying Processes", in *Proceedings of the International Symposium on Category Theory In Computer Science*, E. Moggi and G. Rosolini, eds., Springer Lecture Notes in Computer Science Vol. 1290, (Springer-Verlag), 1997, 147–158.
43. S. Abramsky, "Game Semantics for Programming Languages", in *22nd International Symposium on Mathematical Foundations of Computer Science*, I. Privara and P. Ruzicka, eds., Springer Lecture Notes in Computer Science Vol. 1295, (Springer-Verlag), 1997, 3–4.
44. S. Abramsky and S. Gay and R. Nagarajan, "A Type-theoretic Approach to Deadlock-freedom of Asynchronous Systems", in *Theoretical Aspects of Computer Software*, M. Abadi and T. Ito, eds., Springer Lecture Notes in Computer Science Vol. 1281, (Springer-Verlag), 1997, 295–320.
45. S. Abramsky, "Games in the Semantics of Programming Languages", in *Proceedings of the 11th Amsterdam Colloquium*, P. Dekker, M. Stokhof and Y. Venema, eds., (ILLC, Dept. of Philosophy, University of Amsterdam), 1997, 1–6.

1998

46. S. Abramsky and G. McCusker, "Call-by-value games", in *Proceedings of the Eleventh International Workshop on Computer Science Logic*, M. Nielsen and W. Thomas, eds., Springer Lecture Notes in Computer Science Vol. 1414, (Springer-Verlag), 1998, 1–17.
47. S. Abramsky and K. Honda and G. McCusker, "A fully abstract game semantics for general references", in *Proceedings of the Thirteenth International Symposium on Logic in Computer Science*, (Computer Society Press of the IEEE) 1998, 334–344.

1999

48. S. Abramsky, "Axioms for Definability and Full Completeness", in *Essays in Honour of Robin Milner*, G. Plotkin, M. Tofte and C. Stirling, eds., (MIT Press) 1999, 55–75.
49. S. Abramsky and G. McCusker, "Game Semantics", in *Computational Logic: Proceedings of the 1997 Marktoberdorf Summer School*, H. Schwichtenberg and U. Berger, eds. (Springer-Verlag) 1999, 1–56.
50. S. Abramsky, R. Blute and P. Panangaden, "Nuclear and trace ideals in tensored $*$-categories", *J. Pure and Applied Algebra* vol. 143 (1999), 3–47.
51. S. Abramsky, S. J. Gay and R. Nagarajan, "A Specification Structure for Deadlock-freedom of Synchronous Processes", *Theoretical Computer Science*, vol. 222 (1999), 1–53.
52. S. Abramsky and G. McCusker, "Full Abstraction for Idealized Algol with passive expressions", *Theoretical Computer Science* vol. 227 (1999), 3–42.
53. S. Abramsky and P.-A. Melliès, "Concurrent Games and Full Completeness", in *Proceedings of the Fourteenth International Symposium on Logic in Computer Science*, (Computer Society Press of the IEEE) 1999, 431–442.

2000

54. S. Abramsky, "Process Realizability", in *Foundations of Secure Computation: Proceedings of the 1999 Marktoberdorf Summer School*, F. L. Bauer and R. Steinbrüggen, eds. (IOS Press) 2000, 167–180.
55. S. Abramsky, "Concurrent Interaction Games", in *Millennial Perspectives in Computer Science*, J. Davies, A. W. Roscoe and J. Woodcock, eds. (Palgrave) 2000, 1–12.
56. S. Abramsky, R. Jagadeesan and P. Malacaria, "Full Abstraction for PCF", *Information and Computation* vol. 163 (2000), 409–470.
57. S. Abramsky and M. Lenisa, "A Fully Complete PER Model for ML Polymorphic Types", in *Proceedings of the Fourteenth International Workshop on Computer Science Logic*, P. Clote and H. Schwichtenberg, eds., Springer Lecture Notes in Computer Science Vol. 1862, (Springer-Verlag) 2000, 140–155.
58. S. Abramsky and M. Lenisa, "Axiomatizing Fully Complete Models for ML Polymorphic Types", in *Proceedings of the International Symposium on Mathematical Foundations of Computer Science*, M. Nielsen and B. Rovan, eds., Springer Lecture Notes in Computer Science Vol. 1893, (Springer-Verlag) 2000, 141–151.

2001

59. B. Thomsen and S. Abramsky, "A fully abstract denotational semantics for the calculus of higher-order communicating systems", *Theoretical Computer Science* vol. 254 (2001) 557–589.
60. S. Abramsky and M. Lenisa, Fully Complete Minimal PER Models for the Simply Typed λ-calculus, *CSL 2001* Conference Proceedings, Springer LNCS Vol. 2142, 2001, 442–457.

61. S. Abramsky, A Structural Approach to Reversible Computation, in *LCCS 2001: Proceedings of the International Workshop on Logic and Complexity in Computer Science*, edited by D. Beauquier and Y. Matiyasevich, LACL 2001, 1–16.

2002

62. S. Abramsky, Algorithmic Game Semantics: A Tutorial Introduction, in *Proof and System Reliability*, H. Schwichtenberg and R. Steinbrüggen, eds., Proceedings of the NATO Advanced Study Institute, Marktoberdorf, Kluwer Academic Publishers, 2001, 21–47.
63. S. Abramsky, E. Haghverdi and P. Scott, Geometry of Interaction and linear combinatory algebras, in *Mathematical Structures in Computer Science* vol. 12, 2002, 625–665.

2003

64. S. Abramsky and B. Coecke, Physical Traces: Quantum vs. Classical Information Processing, in *Electronic Notes in Theoretical Computer Science*, vol. 69, 2003, 1–26.
65. S. Abramsky and R. Jagadeesan, A Game semantics for Generic Polymorphism, in *Foundations of Software Science and Computational Structures, 6th International Conference, FOSSACS 2003*, A. D. Gordon, ed., Springer Lecture Notes in Computer Science vol. 2620, 2003, 1–22.
66. S. Abramsky, Sequentiality vs. concurrency in games and logic, *Mathematical Structures in Computer Science* vol. 13, 2003, 531–565.
67. S. Abramsky, D. R. Ghica, L. Ong and A. Murawski, Algorithmic Game Semantics and Component-Based Verification, in *Proceedings of SAVBCS 2003: Specification and Verification of Component-Based Systems*, Workshop at ESEC/FASE 2003, published as Technical Report 03-11, Department of Computer Science, Iowa State University, 2003, 66–74. Available at http://www.cs.iastate.edu/~leavens/SAVBCS/2003/papers/SAVCBS03.pdf.

2004

68. S. Abramsky, D. Ghica, L. Ong and A. Murawski, Applying Game Semantics to Compositional Software Modelling and verification, in *TACAS 2004: Tools and Algorithms for the Construction and Analysis of Systems, 10 International Conference*, Springer LNCS Vol. 2988, 2004, 421–435.
69. S. Abramsky, D. Ghica, A. Murawski, C.-H. Ong and I. Stark, Nominal games and Full Abstraction for the Nu-Calculus, in *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science: LICS 2004*, IEEE Computer Society, 150–159, 2004.
70. S. Abramsky, High-Level Methods for Quantum Computation and Information, in *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science: LICS 2004*, IEEE Computer Society, 410–414, 2004.
71. S. Abramsky and B. Coecke, A Categorical Semantics of Quantum Protocols, in *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science: LICS 2004*, IEEE Computer Society, 415–425, 2004.

2005

72. S. Abramsky and R. Jagadeesan, A Game Semantics for Generic Polymorphism, in *Annals of Pure and Applied Logic*, vol 133, 3–37, 2005.
73. S. Abramsky and B. Coecke, Abstract Physical Traces, in *Theory and Applications of Categories*, vol 14, 111–124, 2005.
74. S. Abramsky and M. Lenisa, Linear realizability and full completeness for typed lambda-calculi, in *Annals of Pure and Applied Logic*, vol 134, 122–168, 2005.
75. S. Abramsky, What are the fundamental structures of concurrency? We still don't know!, in *Algebraic process calculi: the first 25 years and beyond*, BRICS Notes Series NS-05-03, 1–5, June 2005.
76. S. Abramsky, Abstract Scalars, Loops, and Free Traced and Strongly Compact Closed Categories, in *Proceedings of CALCO 2005*, Springer Lecture Notes in Computer Science Vol. 3629, 1–31, 2005.
77. S. Abramsky, A Cook's Tour of the Finitary Non-Well-Founded Sets, in *We Will Show Them: Essays in honour of Dov Gabbay*, edited by Sergei Artemov, Howard Barringer, Artur d'Avila Garcez, Luis C. Lamb and John Woods, College Publications, Vol. 1, 1–18, 2005.
78. S. Abramsky, A Structural Approach To Reversible Computation, *Theoretical Computer Science* vol. 347(3), 441–464, 2005.

2006

79. S. Abramsky, Socially Responsive, Environmentally Friendly Logic, *Truth and Games: Essays in Honour of Gabriel Sandu*, Aho, Tuomo and Ahti-Veikko Pietarinen, eds., Acta Philosophica Fennica, Societas Philosophicas Fennica, Helsinki, 17–46, 2006.
80. S. Abramsky and R. Duncan, A Categorical Quantum Logic, *Mathematical Structures in Computer Science*, 16, 469–489, 2006.
81. S. Abramsky, What are the fundamental structures of concurrency? We still don't know!, in *Electronic Notes in Theoretical Computer Science*, 162, 37–41, 2006.

2007

82. S. Abramsky, Event Domains, Stable Functions and Proof Nets. Plotkin Festschrift, in *Electronic Notes in Theoretical Computer Science*, 172, 33–67, 2007.
83. S. Abramsky and B. Coecke, Physics from Computer Science. *International Journal of Unconventional Computing*, vol. 3:3, 179–197, 2007.
84. S. Abramsky, Temperley-Lieb algebra: From knot theory to logic and computation via quantum mechanics. In *Mathematics of Quantum Computing and Technology*, ed. G. Chen, L. Kauffman and S. Lomonaco. Taylor and Francis, 415–458, 2007.
85. S. Abramsky, Game Semantics for Multi-Agent Logics: A Compositional Game Semantics for Multi-Agent Logics of Partial Information. In *Texts in Logic and Games Vol. 1*, Amsterdam University Press, pages 11–48, 2007.

2008

86. S. Abramsky, Petri Nets, Discrete Physics and Distributed Quantum Computation. In Pierpaolo Degano, Rocco De Nicola and Jose Meseguer, editors, *Concurrency, Graphs and Models, Essays Dedicated to Ugo Montanari on the Occasion of His 65th Birthday*. Vol. 5065 of Lecture Notes in Computer Science, pages 527–543. Springer, 2008.
87. S. Abramsky, Types, Orthogonality and Genericity: some tools for communicating process architectures. In *Communicating Process Architectures 2008: WoTUG 31*, ed. P. H. Welch, S. Stepney, F. Polack, F. Barnes, A. McEwan, G. Stiles, J. Broenink and A. Sampson, pages 1–13. IOS Press, 2008.
88. S. Abramsky, Information, Processes and Games. In *Handbook of the Philosophy of Information*, ed. P. Adriaans and J. van Benthem, pages 483–549. Elsevier 2008.

2009

89. S. Abramsky and B. Coecke, Categorical Quantum Mechanics. In *Handbook of Quantum Logic and Quantum Structures: Quantum Logic*, ed. K. Engesser, D. Gabbay and D. Lehmann, pages 261–324, Elsevier 2009.
90. S. Abramsky and J. Väänänen, From IF to BI: a tale of dependence and separation. *Synthese*, 167:2, pages 207–230, 2009.
91. S. Abramsky and R. Jagadeesan, Game Semantics for Access Control. In Proceedings of MFPS 25, *Electronic Notes in Theoretical Computer Science*, Vol. 249, pages 135–156, 2009.
92. S. Abramsky, Logic and Geometry of Agents in Agent-Based Modeling. Encyclopedia of Complexity and Systems Science, pages 5286–5299, 2009.

2010

93. S. Abramsky, Logic and Categories as Tools for Building Theories. In Journal of Indian Council of Philosophical Research, Issue on Logic and Philosophy Today, Vol. 27,1 pages 277–304, 2010.
94. S. Abramsky, No-Cloning in Categorical Quantum Mechanics. In *Semantic Techniques in Quantum Computation*, ed. S. Gay and I. Mackie, pages 1–28, Cambridge University Press 2010.
95. S. Abramsky, Coalgebras, Chu Spaces and Representations of Physical Systems. In Proceedings of 25th Annual Symposium on Logic in Computer Science (LiCS), pages 411–420, IEEE Press, 2010.
96. S. Abramsky, From CSP to Game Semantics. In *Reflections on the Work of Tony Hoare*, C.B. Jones, A.W. Roscoe and K.R. Wood (eds.), pages 33–46, Springer 2010.
97. Samson Abramsky, Robin Milner's Work on Concurrency, in *Proceedings of the 26th Conference on the Mathematical Foundations of Programming Semantics, MFPS 2010, Ottawa, Ontario, Canada, May 6-10, 2010*, Michael W. Mislove and Peter Selinger (eds.), pages 5–10, ENTCS volume 265, 2010.

2011

98. S. Abramsky and N. Tzevelekos, Introduction to Category Theory and Categorical Logic. In *New Structures for Physics*, B. Coecke (ed). *Lecture Notes in Physics* Vol. 813, pages 3–94, Springer-Verlag 2011.
99. S. Abramsky and Adam Brandenburger, The Sheaf-Theoretic Structure Of Non-Locality and Contextuality. In *New Journal of Physics*, 13(2011), 113036, 2011.

2012

100. S. Abramsky, Big Toy Models: Representing Physical Systems As Chu Spaces. *Synthese* Volume 186 Issue 3, pages 697–718, 2012.
101. S. Abramsky and L. Hardy, Logical Bell Inequalities, *Phys. Rev. A* 85, 062114 (2012).
102. S. Abramsky and J. Zvesper, From Lawvere to Brandenburger-Keisler: interactive forms of diagonalization and self-reference. In *Coalgebraic Methods in Computer Science*, 11th International Workshop, CMCS 2012, D. Pattinson and L. Schröder (eds), pages 1–19, Springer LNCS 7399, 2012.
103. S. Abramsky and C. Heunen, H^*-algebras and nonunital Frobenius algebras: first steps in infinite-dimensional categorical quantum mechanics. In *Mathematical Foundations of Information Flow*, AMS Proceedings of Symposia in Applied Mathematics Vol. 71, S. Abramsky and M. Mislove (eds), pages 1–24, American Mathematical Society 2012.
104. S. Abramsky, S. Mansfield and R. Soares Barbosa, The Cohomology of Non-Locality and Contextuality, in *Proceedings of Quantum Physics and Logic 2011*, Electronic Proceedings in Theoretical Computer Science, vol. 95, pages 1–15, 2012.

2013

105. S. Abramsky, Coalgebras, Chu Spaces and Representations of Physical Systems, *Journal of Philosophical Logic* vol. 42, issue 3, pages 551–574, 2013.
106. S. Abramsky, Relational Hidden Variables and Non-Locality, *Studia Logica* vol. 101 issue 2, pages 411–452, 2013.
107. S. Abramsky, Two puzzles about computation, in *Alan Turing: his work and impact*, ed. S.B. Cooper and J. van Leeuwen, pages 53–57, Elsevier 2013.
108. S. Abramsky, G. Gottlob and P. Kolaitis, Robust Constraint Satisfaction and Local Hidden Variables in Quantum Mechanics, Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI), AAAI Press, pages 440–446, 2013.
109. S. Abramsky, Relational Databases and Bell's Theorem. In *In Search of Elegance in the Theory and Practice of Computation: Essays dedicated to Peter Buneman*, ed. V. Tannen, L. Wong, L. Libkin, W. Fan, W.-C. Tan and M. Fourman, Springer, pages 13–35, 2013.

2014

110. S. Abramsky and M. Sadrzadeh, Semantic Unification: a sheaf theoretic approach to natural language, in *Categories and Types in Logic, Language and Physics: Essays Dedicated to Jim Lambek*, Eds. Claudia Casadio, Bob Coecke, Michael Moortgat, Philip Scott, Springer 2014.
111. S. Abramsky and A. Brandenburger, An Operational Interpretation of Negative Probabilities and No-Signalling Models, in *Horizons of the Mind: A Tribute to Prakash Panagaden*, ed. F. van Breugel and E. Kashefi and C. Palamidessi and J. Rutten, Springer, pages 59–75, 2014.
112. S. Abramsky, Contextual Semantics: From Quantum Mechanics to Logic, Databases, Constraints, and Complexity, in Bulletin of the European Association for Theoretical Computer Science, Number 113, pages 137–163, June 2014.
113. S. Abramsky, Events in Context, in *Theoretical Computer Science*, Volume 548, pages 7–9, 2014.
114. S. Abramsky, Intensionality, Definability and Computation. In *Johan van Benthem on Logic and Information Dynamics*, ed. A. Baltag and S. Smets, Springer 2014.
115. S. Abramsky, A. Brandenburger and A. Savochkin, No-Signalling Is Equivalent To Free Choice of Measurements, in Proceedings 10th International Workshop on Quantum Physics and Logic (QPL 2013), ed. B. Coecke and M. Hoban, *Electronic Proceedings in Theoretical Computer Science* 171, pp. 1–9, 2014.
116. S. Abramsky and C. Constantin, A classification of multipartite states by degree of non-locality, in Proceedings 10th International Workshop on Quantum Physics and Logic (QPL 2013), ed. B. Coecke and M. Hoban, *Electronic Proceedings in Theoretical Computer Science* 171, pp. 10–25, 2014.

2015

117. S. Abramsky, Arrow's Theorem by Arrow Theory. In *Logic Without Borders*, ed. Å. Hirvonen, J. Kontinen, R. Kossak and A. Villaveces, Ontos, pp 15–29, 2015.
118. S. Abramsky and J. Zvesper, From Lawvere to Brandenburger-Keisler: Interactive forms of diagonalization and self-reference. *Journal of Computer and System Sciences* 81, pp. 799–812, 2015.
119. S. Abramsky, R. Jagadeesan, Radha and M. Vákár, Games for Dependent Types, Proceedings of International Colloquium on Automata, Languages, and Programming (ICALP 2015), Springer, pages 31–43, 2015.
120. S. Abramsky, R.S. Barbosa, K. Kishida, R. Lal and S. Mansfield, Contextuality, Cohomology and Paradox, CSL 2015, pages 211–228, 2015.
121. Samson Abramsky and Dominic Horsman, DEMONIC programming: a computational language for single-particle equilibrium thermodynamics, and its formal semantics. QPL 2015: 1-16 2015.
122. S. Abramsky, Contextual Semantics, in *Contextuality from Quantum Physics to Psychology*, ed. E. Dzhafarov, S. Jordan, R. Zhang and V. Cervantes, pages 23–50, World Scientific, 2015.

2016

123. S. Abramsky and C. Heunen, Operational Theories and Categorical Quantum Mechanics. In *Logical and Algebraic Structures in Quantum Computing*, ed. J. Chubb, A. Eskandarian and V. Harizanov, ASL Lecture Notes in Logic, Cambridge University Press, pages 88–122, 2016.
124. S. Abramsky, C. Constantin and S. Ying, Hardy is (almost) everywhere: nonlocality without inequalities for all entangled multipartite states, *Information and Computation* 250, pages 3–14, 2016.
125. S. Abramsky, R.S. Barbosa, K. Kishida and S. Mansfield, Possibilities determine the combinatorial structure of probability polytopes, *Journal of Mathematical Psychology* 74 (Special issue on Foundations of Probability), pages 58–65, 2016.

2017

126. S. Abramsky and V. Winschel, Coalgebraic analysis of subgame-perfect equilibria in infinite games without discounting. Mathematical Structures in Computer Science, 27:5, pages 751–761, 2017.
127. Samson Abramsky, Anuj Dawar, and Pengming Wang. The pebbling comonad in finite model theory. In Proceedings of the 32nd Annual IEEE Symposium on Logic in Computer Science, pages 1–12, 2017. URL: http://arxiv.org/abs/1704.05124.
128. Samson Abramsky, Rui Soares Barbosa, Giovanni Carù, Nadish de Silva, Kohei Kishida, and Shane Mansfield. Minimum quantum resources for strong nonlocality. In Proceedings of 12th Conference on Theory of Quantum Computation, Communication and Cryptography, pages 9:1–9:20, 2017. URL: http://arxiv.org/abs/1705.09312.
129. Samson Abramsky, Rui Soares Barbosa, and Shane Mansfield. Contextual fraction as a measure of contextuality. Phys. Rev. Lett., 119:050504, Aug 2017.
130. Samson Abramsky, Rui Soares Barbosa, Giovanni Carù, and Simon Perdrix. A complete characterisation of All-versus-Nothing arguments for stabiliser states. In Phil. Trans. Royal Society A 2017 375 20160385.
131. Samson Abramsky, Rui Soares Barbosa, Nadish de Silva, and Octavio Zapata. The quantum monad on relational structures. 42nd International Symposium on Mathematical Foundations of Computer Science, MFCS 2017, August 21–25, 2017 - Aalborg, Denmark, pages 35:1–35:19, 2017.
132. Samson Abramsky. Contextuality: at the borders of paradox. In *Categories for the Working Philosopher*, ed. Elaine Landry, Oxford University Press, pages 262–287, 2017.

2018

133. M. Vákár, R. Jagadeesan, and S. Abramsky, Game semantics for Dependent Types, *Information and Computation* vol. 261, pages 401–431, 2018.
134. Samson Abramsky and Nihil Shah, Relating Structure and Power: Comonadic Semantics for Computational Resources, 27th EACSL Annual Conference on

Computer Science Logic, (CSL) 2018, September 4–7, 2018, Birmingham, UK, pages 2:1–2:17, 2018.

2019

135. Samson Abramsky, Rui Soares Barbosa, Martti Karvonen and Shane Mansfield, A comonadic view of simulation and quantum resources, 34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2019, Vancouver, BC, Canada, June 24–27, 2019.
136. Samson Abramsky and Giovanni Carù, Non-locality, contextuality and valuation algebras: a general theory of disagreement. In *Philosophical Transactions of the Royal Society A*, vol. 377, number 2157, p. 20190036, 2019.

2020

137. Samson Abramsky, Whither Semantics?, *Theoretical Computer Science*, vol. 807, pages 3–14, 2020.
138. Samson Abramsky, Classical logic, classical probability, and quantum mechanics. In *Quantum, Probability, Logic: The Work and Influence of Itamar Pitowsky*, ed. Meir Hemmo and Orly Shenker, Springer, pages 1–17, 2020.
139. Norihiro Yamada and Samson Abramsky, Dynamic game semantics, Math. Struct. Comput. Sci., volume 30, number 8, pages 892–951, 2020.

2021

140. Samson Abramsky and Rui Soares Barbosa, The logic of contextuality, 29th EACSL Annual Conference on Computer Science Logic, CSL 2021, January 25-28, 2021, pages 5:1–5:18, LIPiCS volume 183, 2021.
141. Samson Abramsky and Dan Marsden, Comonadic semantics for guarded fragments, 36th Annual ACM/IEEE Symposium on Logic in Computer Science, (LICS) Rome, Italy, June 29–July 2, 2021, pages 1–13, IEEE, 2021.
142. Samson Abramsky and Luca Reggio, Arboreal Categories and Resources, 48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, July 12–16, 2021, Glasgow, Scotland (Virtual Conference), edited by Nikhil Bansal, Emanuela Merelli, and James Worrell, LIPiCS volume 198, pages 115:1–115:20, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.
143. Samson Abramsky and Nihil Shah, Relating Structure and Power: Comonadic semantics for computational resources (expanded version), Journal of Logic and Computation, Volume 31, Issue 6, pages 1390–1428, September 2021.

2022

144. Samson Abramsky, Nothing Will Come of Everything: Software Towers and Quantum Towers. In *From Electrons to Elephants and Elections*, pages 539–552, Springer, Cham, 2022.

Overview of the Contributions

Part A: Domains and Duality

Chapter 1, *Duality, Intensionality, and Contextuality: Philosophy of Category Theory and the Categorical Unity of Science in Samson Abramsky*, by Yoshihiro Maruyama, is really *sui generis* in this volume. It provides a wide-ranging and thought-provoking survey of the philosophical viewpoint and themes implicit, and occasionally explicit, in Abramsky's work, and provides a very stimulating starting point for the volume. It develops the idea of Categorical Unified Science, a re-conception of the 20th century philosophical project of Unity of Science in modern terms.

In Chap. 2, *Minimisation in Logical Form*,[25] by Nick Bezhanishvili, Marcello Bonsangue, Helle Hvid Hansen, Dexter Kozen, Clemens Kupke, Prakash Panangaden, and Alexandra Silva, the authors use duality as a setting for unifying two streams of work on understanding minimization algorithms for automata from a structural and conceptual point of view. In particular, Brzozowski's minimization algorithm, which appears like something of a magic trick in its concrete presentation, is given a very lucid reading in terms of duality. As an application, minimization algorithms are derived in some novel settings. Apart from this beautiful application of duality to automata theory, another feature of this work which is very appealing is its bringing together of Structure and Power, the latter in the form of a fundamental algorithmic construction. We shall return to this theme in our discussion of Part D of the volume.

Chapter 3, *A Cook's tour of duality in logic: from quantifiers, through Vietoris, to measures*, by Mai Gehrke, Tomas Jakl and Luca Reggio, is a wide-ranging study of a duality-theoretic approach to quantifiers in a variety of settings. In particular, the process of adding a layer of quantification in a step-wise fashion is analyzed using the Vietoris construction. This is related to model theory in general, and to the logic of words which is important in the study of automata and formal language theory. It is also related to probabilistic quantifiers and the structural limits studied by Ossona de Mendez and Nesetril in the context of combinatorics. This "Cook's tour" of duality[26] gives an inspiring vision of the possibilities which exist for further and deeper applications of the duality perspective in logic and computation.

In Chap. 4, *Stone duality for relations*, by Achim Jung, Alexander Kurz, and Drew Moshier, the theme is extending Stone duality to categories of spaces and relation, which opens up many new possibilities for applications. The theory developed in this chapter, hinging on a duality of spans and cospans, is very elegant, and is applied to a range of important examples.

Part B: Game Semantics

One of Abramsky's important contributions to game semantics is what has become known as the "Abramsky cube"; a hierarchy of constraints on strategies which lead to a landscape of game models in precise correspondence with a range of computational features. This underpins the wide range of full abstraction results which have been obtained using game semantics. Dan Ghica's paper on *The far side of the cube: An*

[25] The title alludes to Domain theory in logic form, items [12, 18] in the list of publications.

[26] Alluding to [77].

elementary introduction to game semantics uses the cube in a different and novel way, studying highly unconstrained strategies as an effective pedagogical device for presenting the ideas of game semantics in an elementary fashion.

Jim Laird and Guy McCusker, in *An axiomatic account of a fully abstract game semantics for general references*, analyze a games model of Abramsky, Kohei Honda, and McCusker (AHM) which gives a fully abstract model for general references—a high-point of game semantics, which received the LiCS Test-of-Time award in 2018. They show how the structure of this quite sophisticated model can be derived in an axiomatic fashion in the setting of sequoidal categories—an impressive *tour de force*. This builds on and extends the axiomatic approach to definability and full completeness results introduced by Abramsky in [48].

The AHM paper is also the starting point for Andrzej Murawski and Nikos Tzevelekos in *Deconstructing general references via game semantics*. They strengthen the results on visible factorisation, allowing them to prove universality for the model, and to give a much sharper analysis of so-called "bad variables", which are then actually used as a tool for constructing interesting program transformations.

The next two papers concern concurrent games defined on event structures. Concurrent games were introduced by Abramsky and Melliès in [53], and have been developed much further subsequently, in the work of Glynn Winskel, Pierre Clairambault and their coauthors. In *The Mays and Musts of Concurrent Strategies*, Simon Castellan, Clairambault and Winskel refine the bicategory of concurrent games and non-deterministic strategies studied in their previous work to treat may and must testing properties of strategies, allowing deadlock and divergence information to be accurately reflected in the semantics. This raises a number of technical challenges, to which interesting solutions are found.

In *A Tale of Additives and Concurrency in Game Semantics*,[27] Clairambault revisits the original problem motivating the introduction of concurrent games in [53], namely full completeness for Multiplicative-Additive Linear Logic. (MALL), and the issues arising from having classical dualities in a purely sequential framework [66]. He reformulates the models of MALL, and subsequent developments by Melliès, in the setting of current work on concurrent games based on event structures, giving a synthetic and unified presentation.

Finally, in *The game semantics of game theory*, Jules Hedges connects game semantics with game theory, in particular with the compositional form of game theory based on open games which he has been developing. Apart from the fact that both game semantics and game theory figure in Abramsky's work (the latter e.g. in [126]), another point of contact is the use he makes of Geometry of Interaction ideas, in particular the "wave" style geometry of interaction introduced in [21, 31].

Part C: Contextuality and Quantum Computation

In *Describing and Animating Quantum Protocols*, Richard Bornat (Abramsky's Ph.D. supervisor) and Raja Nagarajan (a former student) describe a quantum simulator for animating and exploring quantum security protocols. Apart from relating to Abramsky's interests in quantum computation, it also connects with his early work,

[27] The title alluding to [90].

joint with Bornat, on the Pascal-m programming language [1, 2]. An interesting twist is that what had seemed a mistaken design decision in Pascal-m, allowing mixed guards, re-emerges as a useful feature of the simulator.

The next group of papers relate to Abramsky's work on contextuality in quantum computation and beyond. The first of these, *Closing Bell, Boxing black box simulations in the resource theory of contextuality* by Rui Soares Barbosa, Martti Karvonen and Shane Mansfield, gives a very conceptual and insightful presentation of the sheaf-theoretic approach to contextuality pioneered by Abramsky and Adam Brandenburger in [99]. Particular emphasis is given to the resource theory of contextuality previously developed by these authors and Abramsky in [129, 135]. A striking new contribution is the introduction of a closed structure in a category of behaviours and simulations, allowing simulations of one behaviour by another to be regarded as behaviours in their own right. This implies a kind of reduction, novel in the study of resource theories, in which "free" operations can be seen simply as non-contextual behaviours. This paper, two of whose authors are former students and long-standing collaborators of Abramsky, is written in an amusing and evocative style.

The paper by Kohei Kishida on *Godel, Escher, Bell, Contextual Semantics for Logical Paradoxes* extends the connections between contextuality and logical paradoxes observed in [120] to cover a much wider range of the paradoxes discussed by logicians and philosophers. It shows that the sheaf-theoretic and topological view of paradoxes taken in [120] can be developed as a unifying framework for comparing and classifying paradoxes. It also links contextuality to a breakdown in compositionality of logical semantics.

Finally, the paper by Ehtibar Dzhafarov on *The Contextuality-by-Default View of the Sheaf-Theoretic Approach to Contextuality* makes a detailed comparison between the sheaf-theoretical approach to contextuality and his own Contextuality-by-Default (CbD) theory, which he has developed extensively with his collaborators. While these theories are formulated mathematically in very different terms, for probabilistic systems which are compatible or "consistently connected" they essentially coincide. Addressing this main point of difference, a "consistification" construction is introduced in the CbD setting to render a system consistently connected. A strength of the sheaf-theoretic approach is that non-probabilistic systems also fall under its scope. Another contribution of this paper is to describe a way of dealing with possibilistic systems in the CbD approach, using epistemic probabilities. The paper provides a basis for further comparisons and interactions between the two approaches, which should be to the benefit of both.

In *Putting paradoxes to work: contextuality in measurement-based quantum computation*, Robert Raussendorff describes recent work with his coauthors on using cohomology to give witnesses for contextuality, and relating this to measurement-based quantum computation. This gives a very attractive way of turning quantum paradoxes into quantum advantage in computation, mediated by cohomology. The use made of cohomology was in part stimulated by the work of Abramsky and coauthors in [104, 120], although the specific cohomology theory used in this work is different to the sheaf cohomology used in [104, 120]. However, Abramsky's student Sivert Aasnæss has related the cohomology of partial monoids used by Raussendorff and his coauthors to the sheaf cohomology, and shown that the latter covers all cases of cohomology detected by the former.

The last paper in this part is *Consistency, Acyclicity, and Positive Semirings* by Albert Atserias and Phokion Kolaitis. This paper generalises a celebrated result by Vorob'ev, which gives necessary and sufficient conditions for when any family of probability distributions defined on overlapping sets of variables always has a joint distribution. The condition is a purely combinatorial one defined in terms of the family of sets of variables, and is in fact equivalent to the well-studied notion of database acyclicity, as first observed by Rui Soares Barbosa, in his thesis written under Abramsky's supervision. Beeri at al. have a result analogous to Vorob'ev's for database instances. The authors provide a common generalisation of these results in terms of databases valued in positive semirings. Although the authors do not explicitly use sheaf-theoretic language, the result is very much in the spirit of the work by Abramsky and his coauthors on contextuality. The use of semiring-valued distributions on event presheaves as a common generalisation of probabilistic and possibilistic models appears already in [99]. The connection to databases is made explicitly in [109, 112, 122], and Barbosa proved one direction of the generalised Vorob'ev theorem over semirings in his thesis. The full equivalence established in the present paper is a powerful extension of this result, which further cements the connections between contextuality and database theory emphasised by Abramsky.

Part D: Game Comonads and Descriptive Complexity

The pebbling comonad paper by Abramsky, Anuj Dawar and Pengming Wang [127] pioneered what has developed into the game comonads paradigm, and led to an ongoing program of work under the banner of "Structure meets Power". In *Constraint Satisfaction, Graph Isomorphism, and the Pebbling Comonad*, Anuj Dawar describes the background in constraint satisfaction, finite model theory and descriptive complexity motivating this work. A particular feature of the pebbling comonad is that it provides a common setting for two important algorithmic approximation schemes: the (strong) local consistency notion of approximation of homomorphisms in constraint satisfaction, and the Weisfeiler-Leman approximations of structure isomorphism, widely studied in graph theory and finite model theory. Dawar asks if there is scope for refinement of these schemes, to obtain more powerful approximations. His student Adam Ó Conghaile has recently proposed such refinements, using ideas from the cohomological characterisation of contextuality in the sheaf-theoretic approach [104,120], thus making an unexpected connection between two apparently quite different branches of Abramsky's work. This is being pursued in current joint work by Abramsky, Barbosa, Ó Conghaile and Dawar.

The paper by Mikolaj Bojanczyk, Bartek Klin, and Julian Salamanca on *Monadic Monadic Second Order Logic* is very much in the spirit of Structure meets Power, although it is monads rather than comonads which are the main structural tool deployed here. There is a classical logical characterization of regular languages as those which are MSO definable. There are a number of generalizations of this result to data structures beyond finite words, such as trees and ω-words. This paper seeks a general setting for these results, more specifically for the implication from MSO-definability to recognisability. The idea is to replace specific data types by (algebras of) monads over sets. The situation turns out to be quite subtle: several classes of monads are shown to satisfy the desired implication, while others do not; many explicit counter-examples are provided. The paper raises the challenge of finding a general structure theory of monads, and characterisation of those which admit the key closure property needed to show that MSO definability is captured by recognisability.

The chapter by Jouko Väänänen on *The Strategic Balance of Games in Logic* takes a broader perspective on games in logic. It concerns three key forms of logical games: model comparison games, which form the basis of the game comonads mentioned above; evaluation games, corresponding to the Tarskian evaluation of a formula on a structure; and model existence games, which correspond to an abstract form of proof procedures. While these games are quite classical, the main point of the paper is to explicitly describe translations between them; this answers a question previously raised by Abramsky. It seems that there is much more to be said about the interaction of these forms of games, which lie at the heart of logic.

Part E: Categorical and Logical Semantics

In logic and linguistics, compositionality is the principle that meaning should be ascribed to expressions as a function of the meanings of their immediate syntactic constituents, thus allowing semantics to be defined by recursion on the parse tree. This is in tension with the view that the meaning of an expression cannot be determined in isolation, but only in the context of its use. In *Compositionality in Context* by Alexandru Baltag, Johan van Benthem, and Dag Westerstahl, this tension is resolved in the spirit of a quote taken from Abramsky [79] to the effect that adding suitable parameters to the semantic function allows the meanings of expressions to be made sensitive to their context, and hence defined compositionally. There is an extensive discussion of manifestations of compositionality in logic, linguistics and computer science, with many sharp and novel insights, often challenging the received wisdom on the subject. A concluding discussion on game semantics, with particular reference to Abramsky's work, suggests the importance of an "outside-in", interactive or co-recursive view of meaning. It is an interesting challenge to reconcile this with the more familiar recursive perspective.

In *Compact Inverse Categories*, Robin Cockett and Chris Heunen characterise compact inverse categories as semilattices of compact groupoids. This is a categorification of a structure theorem for commutative inverse monoids. They also connect this to the Baez-Lauda characterization of compact groupoids in terms of 3-cocycles. There are numerous connections between the structures studied in this paper and Abramsky's work: compact categories appear already in his work on interaction categories [28,37], and compact dagger categories in the work on categorical quantum mechanics [71, 73]. There are also connections to his work on geometry of interaction and traced monoidal categories [21, 22, 31, 39, 63], and reversible computation [88, 61, 78].

In *Reductive Logic, Proof-search, and Coalgebra: A Perspective from Resource Semantics*, Alexander Gheorghiu, Simon Docherty, and David Pym study Reductive Logic, a systematic approach to proof search. The logic BI of bunched implications is used as a case study. After developing logic programming for BI, coalgebraic semantics is used to formalise a reductive logic for BI, capable of capturing the control features of algorithmic proof search in a structural fashion. There are several points of contact with Abramsky's work, e.g. on the computational interpretation of linear logic [25], and also his work on how the propositional connectives of BI arise naturally in the team semantics of dependence logic [90].

Another approach to the analysis of proof search uses focussed sequent calculi, in which side-conditions on rules are used to enforce a strategy for proof search. In *Lambek-Grishin Calculus: Focusing, Display and Full Polarization*, Giuseppe Greco, Michael Moortgat, Valentin D. Richard, and Apostolos Tzimoulis introduce a focussed display calculus and polarized algebraic semantics for the Lambek-Grishin calculus, and prove soundness and completeness. The authors acknowledge inspiration from Abramsky's work on geometry of interaction [31], and game semantics for linear logic [32, 53]. A possible refinement of the work in this paper would be a categorical semantics, admitting a full completeness theorem in the sense of [32, 53].

A reflexive object in a closed category is one isomorphic to its internal hom: $R \cong [R \to R]$. A founding contribution to mathematical semantics was Dana Scott's construction of reflexive objects in categories of domains, to give semantics to the untyped λ-calculus. A linear version can be seen as underlying Girard's "geometry of interaction", as observed in [63], which showed how such a construction gives rise to a linear combinatory algebra. Just as realizability constructions over (partial) combinatory algebras give rise to extensional models of intuitionistic type theories, so linear realizability gives rise to extensional models of linear logic and type theory [57, 60, 74]. In *On strictifying extensional reflexivity in compact closed categories*, Peter Hines studies how to strictify reflexive objects in compact closed categories—*i.e.* to turn the isomorphism into an identity. He develops the general theory of such strictifications, and shows that two iconic algebraic structures, the Thompson group and the bicyclic monoid, must arise from the endomorphism monoids of such strictly reflexive objects. He also shows that there are non-trivial interactions between these two structures, derived from the Frobenius algebra identity, and connects this to the unitless Frobenius algebras of [103]. He then studies concrete examples arising from the traced monoidal category of partial injections.

In *Semantics for a Lambda Calculus for String Diagrams* by Bert Lindenhovius, Mike Mislove and Vladimir Zamdzhiev, the authors address the issue of the semantic foundations of high-level languages for quantum computation. The basic language they study is Benton's linear/non-linear calculus, seen as embedding a basic calculus of string diagrams in its monoidal part, as applicable e.g. to quantum circuits. They develop an axiomatic approach to models for this language, involving a canonical self-enrichment, and extend the language with general recursion. They exhibit concrete domain-theoretic models for the language, and prove soundness, and adequacy for the diagram-free fragment. This connects with Abramsky's work on several topics, including domain theory [30], computational interpretations of linear logic [25], and categorical quantum mechanics [71, 89].

In Abramsky's work on interaction categories [28, 29, 37], the methods of categorical semantics and type theory were applied to the dynamic realm of processes, under the guiding principle that "processes are relations extended in time". In *Retracing* *some paths in categorical semantics: From process-propositions-as-types to categorified reals and computers*,[28] Dusko Pavlovic gives a novel and insightful presentation of interaction categories from a logical perspective. He shows that the algebra of real numbers and vector spaces can be extracted from some canonical interaction categories, using his pioneering previous work on the coinductive nature of the reals.[29] He also emphasizes the issue of computability within the universe of processes. Addressing the "extrinsic" character of the notion of computability as applied to extensional objects such as sets or functions, as discussed in [114], he proposes an abstract structure for computability which is generally applicable within categorical semantics.

Part F: Probabilistic Computation

Statistical programming languages, which allow sampling from continuous distributions and conditioning posterior probabilities based on observations, pose challenges for semantics in the presence in higher-order functions. One approach, by Heunen, Kammar, Staton and Yang, is to extend the semantic setting from classical measure theory to quasi-Borel spaces. In *(Towards a) Statistical Probabilistic Lazy Lambda Calculus*, Radha Jagadeesan proposes an alternative approach, using a combination of open bisimulation and probabilistic simulation to stay within the scope of classical measure theory, while also inheriting coinductive proof methods for open bisimulation, and approximation techniques arising from the underlying domain-theoretic and metric space structure. This draws on Abramsky's work on the lazy λ-calculus [15, 23], and his work with Jagadeesan and Malacaria on game semantics [34, 56, 72, 91].

In *Multisets and Distributions, in Drawing and Learning*, Bart Jacobs uses categorical semantics to clarify some fundamental constructions in basic probability theory involving multiple copies of data, ordered or unordered. In particular, he looks at drawing from an urn, and learning distributions. A uniform treatment of urn models with ordered or unordered data, and with or without replacement, is given using iteration of Kleisli maps for a monad. For probabilistic learning, the semantic formalism is used to distinguish two forms of likelihood, also based on iteration, which lead to different forms of learning. These have not been clearly distinguished in previous literature. Although this study is not directed related to Abramsky's work, it is very much in a kindred spirit, exposing the fundamental mathematical structures arising in an important application area, and using this not only to clarify existing ideas, but to lead to new insights and directions.

Recalling the theme of "Structure meets Power" from Part D, machine learning is a major area of current activity where thus far there has been very little use of structural methods. In *Structure in Machine Learning*, Prakash Panangaden takes up this issue, addressing the challenges posed by Abramsky in [137]. He reviews the development of probabilistic programming languages, which are starting to influence at least some portion of the machine learning community. He then describes

[28] The title alludes to [39].

[29] Note that it was already observed in [37] that Aczel's universe of non-well-funded sets appears as the scalars in SProc, the interaction category of synchronous processes.

several cases in which there have been successful uses of some tools from semantics in machine learning, notably fixed point theorems and bisimulation metrics. To date, these have largely arisen from independent parallel evolution. He then describes a more direct application of semantic methods to reinforcement learning, using the Kantorovich metric on a space of distributions. This allows complex proofs of correctness for several algorithms to be reduced to simple coupling arguments. The further application of structural methods to machine learning remains a promising and highly challenging topic for future research.

Part I
Duality and Domains in Logical Form

Chapter 2
Duality, Intensionality, and Contextuality: Philosophy of Category Theory and the Categorical Unity of Science in Samson Abramsky

Yoshihiro Maruyama

Abstract Science does not exist in vacuum; it arises and works in context. Groundbreaking achievements transforming the scientific landscape often stem from philosophical thought, just as symbolic logic and computer science were born from the early analytic philosophy, and for the very reason they impact our global worldview as a coherent whole as well as local knowledge production in different specialised domains. Here we take first steps in elucidating rich philosophical contexts in which Samson Abramsky's far-reaching work centring around categorical science as a new kind of science may be placed, explicated, and articulated. We argue, inter alia, that Abramsky's work, as a whole, may be construed as demonstrating the categorical unity of science, or rather the sciences, in a mathematically rigorous, down-to-earth manner, which has been a salient feature of his work. At the same time we trace his intellectual history, leading from duality, to intensionality, and to contextuality, and place it in a broader context of philosophy beyond the analytic-continental divide, namely towards the reintegration of them as in the post-analytic tradition. Besides, we address issues in philosophy of category theory, such as the foundational autonomy of category theory and the (presumably two) dogmas of set-theoretical foundationalism, which Abramsky actually touch upon in one of his few philosophically inclined works. As to philosophy of category theory, we also address categorical structuralism as higher-order structuralism, categorical epistemology as elucidating higher-order meta-laws, and categorical ontology as allowing for reduction of ontological commitment via structural realism, the structuralist resolution of Benaceraff's dilemma, and the pluralistic multiverse view of science as opposed to the set-theoretical reductionist 'universe' view. We conclude by speculating about the existence of the Oxford School of (Pluralistic) Unified Science as opposed to the Vienna Circle of (Monistic) Unified Science and to the Stanford School of (Pluralistic) Disunified Science; Categorical Unified Science may potentially allow us to reconcile the two camps on the unity and disunity of science whilst doing justice to both of them. Categorical unity arguably allows for unification via epistemological and ontological networking, and via knowledge transfer thus enabled, rather than unification via the reduction

Y. Maruyama (✉)
School of Computing, Australian National University, Canberra, Australia
e-mail: yoshihiro.maruyama@anu.edu.au

A. Palmigiano and M. Sadrzadeh (eds.), *Samson Abramsky on Logic and Structure in Computer Science and Beyond*, Outstanding Contributions to Logic 25,
https://doi.org/10.1007/978-3-031-24117-8_2

of everything and every truth to a single foundationalist framework, whilst taking at face value disunity, plurality, and diversity, and their significance in science and in human civilisation as a whole.

Keywords Category theory · Unity of science · Unified science · Vienna circle · Disunity of science · Scientific pluralism · Stanford school

2.1 Introduction: One-Man Embodiment of Unity of Science

Samson Abramsky has been a leading figure in mathematical logic and theoretical computer science (especially in Euro style), and more recently, in foundations of physics as well. Phrasing his achievements in this way would only be partially correct, however. His work actually stretches over mathematics, physics, informatics, AI, economics, linguistics, psychology, and so fourth. In terms of striking breadth (as well as powerful problem solving capabilities as seen in his solution to the full abstraction problem for PCF), he is probably comparable to John von Neumann, who started his academic career in foundations of mathematics, and then moved on to quantum physics, mathematical economics, computer science, and so fourth. And thus Abramsky's academic path, from a broader perspective, looks fairly similar to von Neumann's.

Notwithstanding, an intriguing exception to the Abramsky-Neumann correspondence seemingly lies in their early days academic inclinations: Abramsky had a habit of writing poems, and started his undergraduate study with a literature scholarship, and soon thereafter changed his subject into philosophy, in which he read, for example, Ian Hacking, who is, to be interesting, in between the analytical and continental traditions of philosophy, and so his first degree is actually in philosophy; von Neumann's first degree, by contrast, is in chemical engineering, although his doctoral degree is in foundations of mathematics. Whereas von Neumann went rather straight to science, Abramsky took a more winding path in his early academic engagement; there were even "Some Paths Not Taken" as he called them, which would make the formative stage of his distinctive scholarship even more intriguing, thus posing an interesting challenge to the historian in future generations. According to what he once told, the dream of his youth was to become a literary writer rather than a mathematician or scientist; he indeed authored quite some poems himself. It would thus be of serious scholarly interest to explore a missing link between Abramsky's mathematical and literary works in order to shed new light on his intellectual history (whilst strictly avoiding a Whiggish view of history as often observed in non-historians' popular writings about prominent scholars, and whilst still admitting that "history is an unending dialogue between the present and the past", or between idealism and realism, as Edward Hallett Carr, (1961) says; by the way, Chimen Abramsky, one of Samson's relatives, has published an interesting book about E. H. Carr). Yet in this article, we shed light on his scientific work rather than his literary work.

It would be impossible to associate Abramsky with a single discipline as the primary focus of his work, though it might have been possible to call him a theoretical computer scientist with a logic background before he moved to foundations of physics around the beginning of this century. Today he could only be called a mathematical scientist in the broadest possible sense of the term; the same would apply to von Neumann as well. Yet what does not apply to von Neumann is arguably the conceptual unity of methodology as in Abramsky's work as we shall discuss extensively below. There are many scholars who know a lot about a variety of fields, and yet there are very few who can actually produce cutting-edge research results in diverse fields. Abramsky is one of the rare scholars who can do that better than virtually anyone else, neither being a superficial generalist (cf. too profound an armchair thinker who does no real job), nor being a narrowly minded specialist sticking to an established, single tradition for years. Put another way, depth coexists with diversity in his work. Interdisciplinarity can be an excuse for those who cannot dig deep down in one field; this never applies to Abramsky's work, which is definitely deep and diverse. As in the work of any great mathematician, theory building and problem solving are interlaced with each other in his work; indeed he achieved both at once in his building the novel paradigm of game semantics, and solving, by means of it, the long-standing, full abstraction problem for PCF in the semantics of programming languages.

It is a distinctive characteristic of von Neumann that his work often encompassed some sort of frontier spirit towards a 'new kind of science' (perhaps because any existing paradigm was not really satisfactory to him). The same would apply to Abramsky's work, which has contributed to the formation of new fields, such as game semantics and categorical quantum mechanics; it would even open up the possibility of a new kind of (pluralistic) unified science as we shall discuss below. The diversity of Abramsky's work never implies that there is no coherent unity in his work, but rather it is underpinned by his unifying methodology via category theory, the broad, or virtually universal, applicability of which has led to categorifications of different fields of science and beyond. This arguably makes an epistemological difference between Abramsky's transdisciplinary work and von Neumann's interdisciplinary work, which presumably lacked a unifying methodology, or there was then no mathematics available for such a purpose (it must be remarked that interdisciplinarity here only implies 'multiplicity' or 'plurality' whereas transdisciplinarity implies the 'unity of the multiplicity' or 'unity of the plurality' as shall be discussed in detail below; cf. the unity of the manifold as problematised in the continental tradition of philosophy). In light of this, Abramsky is more like Gottfried Wilhelm Leibniz (than like von Neumann), who envisaged *characteristica universalis* as the unifying language and methodology of science (or thinking in general). Leibniz is recognised as a "one-man embodiment of the unity of science" (Weingart, 2010). Abramsky, then, may be regarded as a one-man embodiment of the categorical unity of science; at least his work is rich enough to make such a speculation come into play.

The nature of Abramsky's work that attains the unity on top of plurality and diversity seems to have far-reaching consequences to our scientific image of the world today, on which we are facing severe problems such as the fragmentation of knowledge (as well as society). The science he has created is not merely an accumulation of profound technical results, but it does have rich philosophical implications. The principal aim of the present article lies in elucidating conceptual views surrounding his science, placing them in a broader context of philosophical thought, and thereby explicating and articulating their implications to our global worldview. It is not always possible to patch local knowledge together to form a global, coherent view, just as Abramsky characterizes contextuality as local consistency plus global inconsistency. Much the same idea is expressed in the 'end of grand narrative' thesis in contemporary continental philosophy and in continentally inclined Anglophone philosophy, such as Richard Rorty's, which combined the Quinean and Heideggerian traditions of philosophy, thus called post-analytic philosophy, which has recently been reproblematised in Michael Friedman's work on Ernst Cassirer, who was the last, integrative philosopher right before the emergence of the analytic-continental divide, namely the 'parting of the ways' in philosophy as Friedman calls it. According to the 'end of grand narrative' philosophy and its kin such as Rorty's philosophy, there is no global Truth but only local truths. The unity of science can be taken to be an antithesis to this scientific nihilism, i.e., nihilism about the unity of knowledge in general (including scientific knowledge in particular), which is arguably an instance of more global nihilism characterising (what might be called the disease of) modern civilisation or modernity per se. Overcoming modernism has been a great concern in intellectual history in different guises, sometimes equated with overcoming nihilism, not only in Western but also in Asian traditions of philosophy, such as the Kyoto School of Philosophy, especially Keiji Nishitani, (1990), a Kyoto School philosopher who problematised overcoming modernity as overcoming nihilism whilst having aimed at the 'construction of a unified worldview as the fundamental challenge of the contemporary era'. What Abramsky has shown to us in his colossal body of research outputs, diverse and yet networked by higher concepts and methodology for categorical unification, and what could be called Abramskyan science in general, may be conceived as paving the way for overcoming the scientific nihilism.

In the rest of the article as follows, we shall first discuss the unity versus disunity of science debate and the rôle of categorical foundations therein, and then move on to various philosophical issues concerning Abramsky's work, leading from duality, to intensionality, and to contextuality. The discussion shall be concluded with a perspective on the categorical unity of science as embodied through Abramsky's work on categorical foundations of science (rather than narrowly minded foundations of mathematics). Category theory has long been regarded as a structuralist approach to foundations of mathematics, an alternative to conventional set-theoretical foundations, supporting the pluralistic multiverse of categories as opposed to the monistic universe of set theory (note that category theory can be seen as monistic foundations as well, since it is a vast generalisation of set theory; more on this follows). Category theory can certainly give structuralist foundations of mathematics such that the 'myth of the given' in terms of Wilfrid Sellars' philosophy, such as the given elements of underlying sets of alleged 'structures' in set-theoretical structuralism (including Bourbaki's and model-theoretic structuralism), has been eliminated successfully. Yet at the same time, Abramsky's work tells us that category theory can even serve as foundations of science in general, rather than just mathematics, thus paving the way for categorical unified science as a form of pluralistic unified science, a new kind of unified science different from those in the past, such as the monistic reductionist unified science that the Vienna Circle once envisaged (to be precise, Neurath's unified science was significantly different from others', and closer to pluralistic unified science, although we do not intend to get into historical issues here, since we would need more nuanced extensive discussions to do so). Finally, as a disclaimer, this article only presents a conceptual perspective centring around Samson Abramsky's work within the limits and biases of the author, whether academic or personal, and is never intended to be the true picture of it (if there is any such thing; the true picture of whatever thing could be yet another kind of 'grand narrative' or 'Truth with capital T' as the continental philosopher calls it). The author's modest hope is just that the present article sheds some new light on philosophical aspects lurking behind Samson Abramsky's (or 'Samson the Dinosaur's) hardcore mathematical unified science.

2.2 Categorical Foundations and the Unity of Science

In this section we first take a look at some history and historiography of the unity and disunity of science, and then discuss categorical foundations of mathematics and categorical foundations of science in general, and their potential rôle in the unity versus disunity of science debate.

2.2.1 History and Historiography of Unity and Disunity of Science

The unity of science is a counter to the fragmentation of science. The fragmentation of science is not necessarily a recent issue. Hermann von Helmholtz, known as a Neo-Kantian thinker as well as physicist, was already aware of the tendency of fragmentation in the nineteenth century:

> No one could oversee the whole of science and keep the threads in one hand and find orientation. The natural result is that each individual researcher is forced to an ever smaller area as his workplace and can only maintain incomplete knowledge of neighbouring areas. (Helmholtz, 1896; translation by Weingart, 2010)

The same lament is pervasive today; it already existed in the nineteenth century German academia. Broadly speaking, the disciplinary order of science gradually emerged in the period from the late eighteenth to the early nineteenth century; the fragmentation of knowledge came to be recognised as a critical problem in the systematisation of knowledge in the late nineteenth century (see, e.g., Wittgenstein, 2010). The currently prevailing divide between science and humanities at least goes back to Dilthey's distinction between *Naturwissenschaft* and *Geisteswissenschaft* in the nineteenth century, which, in his view, aim at *Erklären* (i.e., explanation) and *Verstehen* (i.e., understanding), respectively (Dilthey, 1991). August Boeckh once made the sarcastic remark that "no problem was too small not to be worthy of a serious scientific analysis" (Daston, 1999; Weingart, 2010). Emil du Bois-Reymond also expressed a similar concern in the nineteenth century in the following manner:

> A thousand busy ants are producing daily countless details [...] only concerned to attract attention for a moment and obtain the best price for their goods [...] stream of discovery is split into ever more and ever more unimportant trickles. (du Bois-Reymond, 1886)

Recall that David Hilbert's well-known saying "We Must Know, We Will Know" was a counter to du Bois-Reymond's "We Will Never Know" arguing for the limitations of science as to "The Seven World Riddles", some of which were concerned with the nature of life and mind, and so comparable to the so-called Hard Problem of Consciousness, posed by David Chalmers, (1996) much more recently (and yet it actually dates back to Leibniz's mill argument, which problematises essentially the same thing as the Hard Problem).

Science is knowledge, and at the same time, activity. The fragmentation of science, therefore, is of two dimensions: i.e., the fragmentation of knowledge and the fragmentation of academia (in the sense of academic communities). The issue of fragmentation, if viewed from a broader perspective, is not just about academic knowledge and communities, but also about the modern society or the modern world as a whole. Unity has been debated at different stages of modern civilization since the Renaissance; quite some part of Kant's and Leibniz's early modern philosophy is devoted to the issue of how to attain the 'unity of the multiplicity', 'unity of the plurality', or 'unity of the manifold'. Modernity, arguably, has inclined towards disunity in both science and society. Indeed, the Renaissance had two consequences, the scientific revolution and the industrial revolution, which were intertwined with each other, both geared towards fragmentation and disunity in order to meet the modernist demand of efficiency. The fragmentation of science, when seen from such a point of view, is of positive value in knowledge production, just as the division of labour is of positive value in industrial production (there would be no sense in forcing mathematicians to produce philosophical knowledge; and vice versa). In the accelerating process of fragmentation in the modernised world, Leibniz eventually came to be recognised as a one-man-embodiment of the unity of science; his natural philosophy encompassing all that there is as a coherent whole was seen as a paradigmatic embodiment of the unity of science. Leibniz was thus reconceived of as representing the lost ideal of the unity of science:

> During the second half of the 19th century Leibniz was hailed as the 'one-man-embodiment of the unity of science' (Weingart, 2010)

Around the end of the nineteenth century, however, even philosophers begun to take a step towards the end of systematic philosophy in the Leibnizian tradition, one strand leading to analytic philosophy in the Anglophone world, which deliberately limits its scope on its own (as the early Wittgenstein says that "whereof one cannot speak, thereof one must be silent"), and another to deconstructive philosophy in continental Europe, which started in Nietzsche and Kierkegaard, and culminated in Heidegger and Derrida, through the Southwest School of Neo-Kantianism (especially, Rickert). Note that another Marburg School of Neo-Kantianism, together with British empiricism, significantly contributed to the formation of analytic philosophy. Both analytic and continental traditions, though in different manners, gave up systematic philosophy of unifying nature. This went hand-in-hand with the decline of Cartesian foundationalism, which was affected, on one hand, by the bankruptcy of Hilbert's programme in foundations of mathematics, i.e., the bankruptcy of epistemological justification of absolute truth, and on the other, by the bankruptcy of the Vienna Circle's programme on unified science. In the nineteenth century the unity of science was meant to be natural philosophy in Leibniz style, and around the middle of the twentieth century, the unity of science was reconceived of as the foundationalist reduction of science to logic and physics (which is basically the reduction of mathematical truth to 'truth by convention' and of empirical truth to 'sense-data', which is supposed to be immune to scepticism; here we are not concerned with differences between Carnap, Neurath, Ayer, etc.). We shall touch more upon the Vienna School in relation to the Stanford School arguing for the disunity of science, but before that let us take a closer look at the unity of mathematical science in particular, especially Bourbaki's programme on the unity of mathematics, which was involved in yet another, pragmatist dimension of the concept of unity.

Mathematical science around the end of the nineteenth century was not so fragmented yet; there were several mathematicians who can more or less play the unifying rôle, if not in science in general but in mathematical science, such as Poincaré and Hilbert. Yet during the first half of the twentieth century the fragmentation of knowledge in mathematics came to be problematised, especially by Bourbaki, a group of mathematicians based in France. Bourbaki aimed at the unity of mathematics, which was concerned with a stream-lined account of all mathematical structures, from simple to complex ones, in terms of sets and additional structures on them. Although they used some logic and set theory, their enterprise was more concerned with everyday mathematical practice per se than with epistemological foundations of mathematics. Especially, they aimed at the standardization of mathematical language or 'architecture of mathematics'; it would be fair to say that they indeed made quite some success in the standardization programme, a more pragmatic facet of unity. They problematise the unity of mathematics in the following manner:

> [I]t is legitimate to ask whether this exuberant proliferation makes for the development of a strongly constructed organism, acquiring ever greater cohesion and unity with its new growths [...] In other words, do we have today a mathematic or do we have several mathematics? Although this question is perhaps of greater urgency now than ever before, it is by no means a new one; it has been asked almost from the very beginning of mathematical science. Indeed, quite apart from applied mathematics, there has always existed a dualism between the origins of geometry and of arithmetic (Bourbaki, 1950)

They believed that mathematics is one, and not many, thus arguing for singular *mathématique* rather than plural *mathématiques* as the very title of their book series, *Éléments de Mathématique*, clearly show. To Bourbaki, mathematics is meant to be a single organism as a coherent whole. To be unfortunate (or fortunate), Bourbaki's programme seems to be the only case in which an attempt at unity led to some success (or did not go bankrupt). Note that some interpret French *Encyclopédie*, its British precursor *Cyclopaedia, or an Universal Dictionary of Arts and Sciences* by Ephraim Chambers, and its Japanese descendent *Interweaving a Hundred Sciences* by Amane Nishi, as attempts at the unity of science of some sort. Yet the unity of science, arguably, would be more than the classification of knowledge; put another way, the unified scientist would not be someone who just knows everything about every science (who would be just a polymath). They thus would not count as unity-of-science programmes.

Broadly speaking, the quest for unity, absolute foundations, and the like has mostly failed in the twentieth century, and the antifoundationalist tendency is pervasive in philosophy today. Foundationalism is now generally considered something bad in philosophy, including in particular philosophy of science and philosophy of mathematics (cf. Stewart Shapiro's *Foundations without Foundationalism*). The Vienna Circle's programme on the unity of science represented foundationalism in philosophy of science, and the Hilbert's programme foundationalism in philosophy of mathematics. In the late twentieth century, furthermore, philosophy of language faced the so-called Kripkenstein's sceptical paradox (i.e., Wittgenstein's paradox formulated by Kripke, 1982) on the (in)determinacy of meaning, and realist metaphysics faced Putnam's model-theoretic argument (Putnam, 1983), which is of sceptical nature as well. In this transformation of philosophical worldview, disunitism, antifoundationalism, and antirealism have had their driving force stronger than ever before. They may arguably be different manifestations of the modernist tendency of scepticism; Kurt Gödel says in his (never delivered) lecture notes "The modern development of the foundations of mathematics in the light of philosophy" as follows:

> [T]he development of philosophy since the Renaissance has by and large gone from right to left [...] Particularly in physics, this development has reached a peak in our own time, in that, to a large extent, the possibility of knowledge of the objectivisable states of affairs is denied, and it is asserted that we must be content to predict results of observations. This is really the end of all theoretical science in the usual sense. (Gödel, 1995)

According to Gödel, right-wing thought include metaphysics, rationalism, idealism, and theology, whereas left-wing thought include materialism, empiricism, scepticism, and nihilism. The shift from right to left in Gödel's terms implies the sceptical tendency since the Renaissance, and he illustrates it with what happened during the rise of quantum theory, which, in a sense, counts as a sceptical (or statistical) theory of reality, as Feynman-Mermin say, "Shut up and calculate!", suggesting that it would be useless to think of the ultimate picture of reality in quantum theory. Bell-type No-Go theorems suggest that we have to give up the classical (e.g., non-contextual) picture of reality, and they would arguably support the Gödel's view that quantum theory is a left-wing theory. Ordinary mathematicians could say, "Shut up and calculate!", in face of students obsessed with foundations of mathematics. The antifoundational tendency surely existed in physics as well as mathematics. Physics suffered from quantum paradoxes, just as mathematics suffered from logical paradoxes. There is thus a conceptual parallelism between logical and quantum paradoxes, both representing the sceptical tendency. Gödel's last words in the above passage, “This is really the end of all theoretical science”, sounds like French post-modernist thought. These are Gödel’s genuine words, however. Some commentators equate Gödel’s position with naïve realism, but he is not a simple realist as the lecture notes (Gödel, 1995) clearly show. Indeed, he attempts to reconcile the right thought (metaphysics, idealism, theology, etc.) and the left thought (positivism, materialism, scepticisim, etc.) via the transcendental philosophy of Kant and Husserl in the lecture notes. He illustrates the dynamic history of human thought through the conceptual shift of worldview leading from the metaphysics of substance to the antimetaphysics of different sorts. Gödel’s ‘right’ and ‘left’ are a matter of degree; for example, scepticism is an extreme left thought, compared with positivism, which is less lefty than scepticism. He also says: “Schopenhauer’s pessimism is a mixed form, namely a pessimistic idealism.” Hilbert’s transcendental finitism is a left position in mathematics, and yet at the same time, it is a foundation for the right, that is, the infinitary metaphysics of mathematics.

In the context of the unity of science debate, the sceptical tendency reached a peak in the late twentieth century within the Stanford School in philosophy of science, which, among others, included Peter Galison, Patrick Suppes, John Dupré, Nancy Cartwright, and Ian Hacking, known for propounding the disunity of science as an antithesis to the Vienna Circle’s unity-of-science movement. The Vienna Circle had two series of publications for their unified science project, i.e., *Einheitswissenschaft* in Europe and *International Encyclopedia of Unified Science* in the United States. The Stanford School had no such series, but some of their publications, such as *The Plurality of Science* by Suppes and *The Disunity of Science: Boundaries, Contexts, and Power* by Galison et al., clearly show their doctrine. Ironically, *International Encyclopedia of Unified Science* included Thomas Kuhn’s *The Structure of Scientific Revolutions*, and the Kuhnian tradition had contributed to the rise of the Stanford School, especially, Peter Galison, who was a continentally inclined historian and philosopher of science just as Kuhn was. Galison argues for the positive value of disunity:

> It is precisely the disunification of science that underpins its strength and stability. (Galison, 1999)

Science is strong and stable because of its disunity or diversity. Yet it is disunity with binding, that is, what Galison calls the ‘binding culture of science’ whilst referring to Peirce:

> It should not form a chain which is no stronger than its weakest link, but a cable whose fibres may be ever so slender, provided they are sufficiently numerous and intimately connected. (Peirce, 1974)

Galison argues that there is no single golden thread in the cable (rather than chain) of science:

> With its intertwined strands, the cable gains its strength not by having a single, golden thread that winds its way through the whole. No one strand defines the whole. (Galison, 1997)

In the reductionist approach to the unity of science, there is a golden thread, which is usually physics, combined with logic in the case of the logical empiricism of the

Vienna School. The Stanford School, on the other hand, presents a decentred view of the sciences, which are interlaced with each other, the strength and stability of scientific knowledge stemming from the strength and stability of this interlacing. In other words, the Vienna School gives a monist view of science, and the Stanford School a pluralist view of it. What is essential in the tenets of the Stanford School would be this decentred pluralism as emphasised by Suppes rather than disunitism per se, which is seemingly more like rhetoric to counter the Vienna School. It should be remarked that the Vienna School was not really monolithic, and Neurath had a more pluralist view than Carnap, with an emphasis on interconnections between different sciences rather than the foundationalist reduction of one science to another. Galison's view of disunity as the underpinning of strength and stability is relevant to the aforementioned issue of efficiency enhanced via fragmentation. He emphasises the importance of what he calls: a 'trading zone', a place where different kinds of scientists can work together despite fundamental differences in their epistemologies (such as epistemic criteria for justification and significance) and ontologies (such as objects posited); 'interlanguage', a simple language to facilitate interdisciplinary communication beyond epistemic differences; and 'boundary objects', those entities that exist beyond disciplinary boundaries and thus allow for ontic interdisciplinarity. All this enables local coordination despite global differences:

> Two groups can agree on rules of exchange even if they ascribe utterly different significance to the objects being exchanged; they may even disagree on the meaning of the exchange process itself. Nonetheless, the trading partners can hammer out a local coordination, despite vast global differences. In an even more sophisticated way, cultures in interaction frequently establish contact languages, systems of discourse that can vary from the most function-specific jargons, through semispecific pidgins, to full-fledged creoles rich enough to support activities as complex as poetry and metalinguistic reflection. (Galison, 1997)

Galison's disunity-of-science view may be seen as an underpinning of interdisciplinary studies; any of trading zones, interlanguage, and boundary objects can exist because science is disunified. And science is fruitful thanks to local coordination in its binding culture enabled by these derivatives of disunity.

Now, in what follows, we shall elucidate categorical foundations in light of the unity versus disunity debate, and argue for the categorical unity of science. Note that the disunity of science leads to the pluralist conception of objectivity in science as well (see, e.g., Daston & Galison, 2007).

2.2.2 Categorical Foundations and the Categorical Unity of Science

Abramsky's work on categorical foundations transgressing the boundaries of science tells us that category theory may count as foundations of science in general rather than just mathematics. Pluralism is inherent in category theory; there is no single golden category for all sciences but different categories for different sciences (or for different problems in different fields of different sciences). Set theory endorses global ontology and global epistemology: there is the absolute universe of sets and all mathematical knowledge is about sets. Category theory has its ontology and epistemology localised in different categories. There is no single universal language of categories, which are indefinitely extensible (in Michael Dummet's terms) to richer and richer (e.g., higher and higher) languages, whereas set theory mostly sticks to the language of Zermelo-Fraenkel set theory, though there are minor conservative extensions such as definitional extensions. Here we first discuss category theory as foundations of mathematics, and then move on to category theory as foundations of science or a new kind of unified science as embodied in Abramsky's work. Bourbaki endorsed set theory as foundations of mathematics and attained the unity of mathematics, especially in the sense of standardisation of mathematical language, as we have discussed above. Abrasmky has arguably endorsed category theory as foundations of science and thereby attained the unity of science in a manner coherent with scientific pluralism. The pluralist unity of science as explicated and articulated through category theory shall be elucidated further below; we take this to be the most fundamental aspect of category theory that makes it intrinsically differ from any other positions about the unity of science and foundations of science in general.

Foundations of mathematics have largely become an ordinary field of mathematics (or computer science) since Gödel's theorems; different fields in foundations of mathematics today mostly target at technical problems just as ordinary mathematics does. Antifoundationalism is prevailing in philosophy of mathematics as well as other fields of philosophy. And there might be no proper foundation of mathematics today; some would even argue that mathematical practice precedes foundations of mathematics, and the foundation of mathematics is mathematical practice per se. This is actually a plausible view supporting the autonomy of mathematics; mathematics had been quite well for a long time before the birth of foundations of mathematics without any foundations in the foundationalist sense. Even after the birth of foundations of mathematics, pure mathematicians have not been so much concerned with them, and from their perspective, mathematics has been autonomous on its own, foundational questions being merely marginal and tending to be just ignored in the mainstream pure mathematics community (some mathematicians even do not hesitate to assert that mathematical logic and foundations of mathematics are diseases in mathematics, which especially young people tend to be infected with; diseases certainly qualify as objects of scientific studies, though). Foundations of mathematics in good old days, when top mathematicians in mainstream pure mathematics such as Hilbert and von Neumann (and Weyl) seriously worked on them, do not exist any more these (not necessarily good) days.

Yet this is a highly simplistic view, and there is quite some foundational work within foundations of mathematics today. For example, partial realisations of Hilbert's programme, despite Gödel's theorems, have been achieved by Friedman-Simpson's reverse mathematics and by Coquand's predicative constructivist foundations (applying Joyal's duality theory for the finitist justification of commutative algebra and algebraic geometry and allowing us to embody Hilbert's idea of the elimination of ideal objects; see, e.g., Coquand, 2009). It is actually possible to argue that Gödel's theorems do not necessarily undermine Hilbert's programme in light of technical subtleties lurking behind Gödel's theorems. If logic is strong enough we can axiomatise the complete theory of arithmetic on it; Hilbert himself would not accept such infinitary logic as the epistemological basis of inquiry such as consistency proofs, though. If the arithmetisation of metamathematics (encoding of provability predicate in particular) is formulated in a certain manner the corresponding consistency statement is perfectly provable within the system concerned; for this reason, the second incompleteness is called intensional whereas the first incompleteness is called extensional. There are many mathematical methods to escape from the preconditions of Gödel's theorems. It would nevertheless be true that quite some mathematicians have lost their foundational interests since the discovery of Gödel's theorems, probably including von Neumann (for the philosophical significance of Gödel's theorems and their relationships with the frame problem in artificial intelligence, we refer the reader to Maruyama, 2018).

Category theory, nonetheless, was born after the discovery of Gödel's theorems, and it is supposed to be foundations of mathematics in some sense. Arguably the most traditional categorical foundation of mathematics is topos theory, which integrates logic and algebraic geometry, and another is univalent foundations (aka. homotopy type theory; see, e.g., Voevodsky et al., 2013), which integrate type theory and homotopy theory. Both of them are of foundational significance in a certain sense. And both of them are primarily intuitionistic or constructive/predicative, even though classical mathematics obtains as their axiomatic extensions. Yet at the same time, they allow us to understand intuitionism and constructivism within classical mathematics, just as Kleene's realisability played a similar rôle in the early days of foundations of mathematics. In Brouwer's intuitionism, for example, all functions from reals to reals are continuous, and this obviously contradicts the classical existence of discontinuous functions. Yet, for example, any computable function from reals to reals is always continuous as is well known, and such computational models allow us to understand intuitionism and constructivism within classical mathematics (this is close to the idea of realisability models including the effective topos and other realisability toposes).

Topos theory and univalent foundations play the same rôle of understanding intuitionism and constructivism in terms of classical mathematics such as algebraic geometry and homotopy theory; it is quite surprising that exotic foundational mathematics can be accounted for in terms of mainstream pure mathematics (which may possibly give evidence that the aforementioned alleged diseases of logicians are not really diseases even from the perspective of mainstream mathematics; the author would be quite happy with this because he is a logician himself). Topos theory is also useful in proving independence results. In both of topos and univalent foundations we can find an interplay between foundational and mainstream mathematics; it is something that did not exist before. From a more conceptual point of view we could even argue that logic and space are one and the same thing in categorical foundations, whether topos-theoretic or homotopy-type-theoretic. Topos theory unifies the (higher-order) intuitionistic notion of logic and the algebro-geometric notion of space. Homotopy type theory unifies the intuitionistic/constructive/predicative notion of type theory and the homotopy-theoretic notion of space. Both of them give the place where logic and space are one and the same thing. In general, category theory allows us to reconcile the algebraic and geometric aspects of mathematics, and this might be the reason why the success of universal algebra was limited, but category theory flourished across different fields of mathematics (broadly speaking, traditional universal algebra remains within the realm of algebra; categorial universal algebra nevertheless allows us to go beyond the limitation of traditional universal algebra).

The sense in which category theory is foundational, however, can be very different from the sense in which traditional set theory is foundational. Interestingly, Abramsky argues as follows (in an article on the conceptual significance of category theory published in a philosophy journal in the Indian tradition, where, as well as in other Asian traditions of philosophy, processes are metaphysically more fundamental than substances):

> Category theory has been portrayed, sometimes by its proponents, but more often by its detractors, as offering an alternative foundational scheme for mathematics to set theory. But this is to miss the point. What category theory offers is an alternative to foundational schemes in the traditional sense themselves. (Abramsky, 2010)

In the following let us attempt to articulate different facets of the notion of foundations; for example, foundations of mathematics and foundations of physics (such as quantum foundations) do not precisely share the same meaning of foundations, being highly different types of foundations as a matter of fact. Let us conceive of set theory as giving global foundations of mathematics, and category theory as providing local foundations of mathematics in particular and of science in general; category theory today is indeed working as a unifying language for different sciences as we shall discuss below. Put another way, set theory aims at reductive absolute foundations of mathematics, reducing everything and every truth to a single foundational universe, and category theory at structural relative foundations of mathematics, shedding light on structural interconnections between different scientific disciplines and thereby allowing for the networking and transfer of science and knowledge. Moreover we may think of yet another sort of foundations, that is, conceptual foundations. These distinctions make sense for science in general as well as mathematics in particular, although our primary focus is placed upon the elucidation of foundations of mathematics in particular. The principal differences between the three sorts of foundations may be articulated in the following manner:

- Global foundations, reductive foundations, or absolute foundations allow us to reduce everything and every truth to single foundational frameworks, providing absolute, domain-independent environments to work within. For example, in set theory, there is a single, absolute universe of sets, into which all sorts of mathematical entities and all sorts of mathematical truths are encoded and thus reduced, and there is nothing and no truth other than sets and set-theoretical truths in se-theoretical foundations of mathematics (if as in ZF set theory we do not allow for urelements). Both set theories (such as ZFC) and category theories (such as Lawvere's ETCS and Awodey-Voevodsky's Homotopy Type Theory or Univalent Foundations) can serve as global/reductive/absolute foundations of mathematics.

Yet category theory may moreover give another type of foundations for mathematics and science in general, namely local foundations, structural foundations, or relative foundations.

- Local foundations, structural foundations, or relative foundations give relative, domain-specific contextual frameworks to work within, allowing for no global absolute ontology or epistemology but only local structural ontologies and epistemologies across mathematics and the sciences. In category theory, there is no single universe as a global foundation of everything and every truth; there are only plural categories as local foundations. In local foundations one may change a framework according to structural focus (and see what remains invariant, and what does not). Broadly speaking, base change is a fundamental idea of category theory (cf. Grothendieck's relative point of view). Category theory indeed gives a variety of relative contexts to work within, such as ribbon categories for knot theory, model categories for homotopy theory, and dagger-compact categories for quantum mechanics and information, to name but a few. Whilst set-theoretical foundations are monistic (note that even the multiverse view of sets rests upon the primary existence of the absolute universe of sets), categorical foundations are inherently pluralistic and intrinsically supports the multiverse view of science and knowledge. The foundational significance of category theory in mathematics (in the sense of mathematical practice rather than any idealised formal systems) would arguably consist in its foundational rôle as relative foundations rather than absolute foundations.
- Conceptual foundations involve no foundationalism whatsoever, although the aforementioned two types of foundations do more or less. Conceptual foundations aim at explicating and articulating the nature of fundamental concepts in mathematics per se, namely mathematics in the sense of mathematical practice. It is surely debatable what concepts are truly fundamental, but fundamental concepts in mathematics would include space, number, continuum, infinitesimal, computation and so fourth. Conceptual foundations, unlike absolute foundations, are compelling enterprises for the working mathematician as well as the foundational mathematician. Foundations of mathematics and foundations of physics in the ordinary usage of the terms seem not to be based upon the same conception of foundation; the meaning of 'foundation' in the term 'foundations of physics' or 'quantum foundations' in particular seems to be closer to conceptual foundations rather than any of the above. Conceptual foundations would be the most plain or undogmatic kind of foundations, when compared with the other two types of foundations, namely global and local foundations, which are associated with specific philosophical views such as mathematical/scientific monism and mathematical/scientific pluralism, respectively. Ordinary mathematicians and scientists can be foundational researchers in this sense, whereas it would be difficult for them to be foundational in the other two senses.

The mathematician works on different types of objects, some of which are algebraic and others of which are geometric. In set theory, all that is translated or encoded into the underlying language of sets because there is nothing else in set theory (if it does not allow for urelements). The monist ontology of set theory provides a common absolute ground for all possible mathematical entities (and truths), and differences between different entities are captured as differences in additional structures on them. Every mathematical entity is a set in the first place, and we are nevertheless allowed to introduce additional structures on it, such as order, algebraic, and topological structures, rendering the same set to be different types of structures. Put another way, sets are prior to structures in set theory. In category theory, by contrast, structures are prior to sets, and structures may even have no underlying sets. Categorical structures are pure structures. Category theory allows us to express pure structures rather than structures on given substances, which are impure structures after all, rooted in the set-theoretical "myth of the given" in Wilfrid Sellars terms. There is no intrinsic reason that what is given to us is substance rather than structure. Substances are rather ideal entities like points in geometry, the existence of which indeed require indeterministic principles such as the axiom of choice and its weaker relatives (since points are prime ideals according to duality theory and algebraic geometry).

Let us remark on the foundational relationships between logic, set theory, and category theory; note that set theory in this article always means the standard ZF set theory. A common misconception about them is that set theory is required for the very formulation of category theory. Some notions of collection and operation are indeed required to express the concept of categories (such as Hom) in the first place. Yet this never means that set theory is necessary for foundations of category theory, just as saying "a forest is a collection of trees" does not make one commit herself to set theory. Set theory has the strong ontological commitment to the infinitary universe of sets that is generated transfinite-recursively. Category theory has no such ontological commitment to an infinitary universe of mathematical objects, and for the very reason, it cannot be that category theory is based upon set theory. Category theory can indeed be developed, for example, under highly weak finitist assumptions. It is happy to live in the modest finitist world, unlike set theory requiring the greedy transfinitist realm of far-reaching infinities. The logical essence of set theory arguably consists in the cumulative hierarchy of sets enabled by transfinite recursion, which category theory does not necessarily presuppose at all. The set theorist would build category theory on set-theoretical foundations of course, but that is not absolutely necessary at all. If it seems so, that is just a set-theoretical prejudice or ungrounded presupposition. Category theory is autonomous on its own, just as mathematics or mathematical practice is autonomous, and there is no essential need for set-theoretical foundations to justify the practice of it. In this sense, the requirement of set-theoretical justification is merely a set-theoretical prejudice or ungrounded presupposition, from which we have to be emancipated to realise the foundational autonomy of category theory. Abramsky accounts for the relationships between set theory and category theory as follows:

> [T]he required 'theory of collections and operations' is quite rudimentary. Certainly nothing like formal set theory is presupposed. In fact, the basic notions of categories are *essentially algebraic* in form [...] that is, they can be formalized as partial algebras [...] (Abramsky, 2010)

Note that category theory is essential algebraic in the formal technical sense (as well as in the informal conceptual sense). Set theory does not precede category theory, which is just an algebra with virtually no ontological commitment, in contrast with set theory with the strong ontological commitment to transfinite infinities. Yet at the same time, how about logic? Does logic precede category theory? We may need some logic in order to formalise the concept of categories, but this does not force us to conceive of logic as conceptually prior to category theory. Logic does not necessarily precede arithmetic whilst logic is required for the formalisation of arithmetic. Indeed, Hilbert argues, following the Kantian tradition of transcendental philosophy, that the finitist arithmetic precedes logic, and the finitist arithmetic is the condition of the possibility of "all scientific thinking" (Hilbert, 1983). Note that one essentially relies upon some arithmetic, such as recursion, in order to express logic. Hilbert, again following Kant, emphasises that mathematics is independent of logic:

> Kant taught — and it is an integral part of his doctrine — that mathematics treats a subject matter which is given independently of logic. (Hilbert, 1983)

He also says that "certain intuitive concepts and insights are necessary conditions of scientific knowledge" (Hilbert, 1983). The autonomy of category theory could be vindicated along the same lines, since categorical reasoning only requires certain intuitive symbols and diagrams (or it may only require pictures as in Bob Coecke's picturalist conception of category theory). For more details of Abramsky's strong counter to the "foundationalist critique of category theory", we refer the reader to (Abramsky, 2010), which is also a gentle introduction to category theory, and deserves more attention from both mathematical and philosophical points of view.

Now, in what follows, we discuss category theory as foundations of science rather than mathematics. Historically, category theory was first born in algebraic topology in the mid-twentieth century, and then turned out to be able to give foundations of mathematics in the late twentieth century. Categorical foundations of mathematics advanced hand-in-hand with categorical logic, and in this process, connections with computer science were discovered as well:

- It was Abramsky who played a leading rôle in the categorical understanding of computational processes and their intensional semantics.
- It surely counts as an application of category theory to computer science, but at the same time it is the fundamental science of what processes are. That is to say, he initiated the categorical science of processes as opposed to the traditional science of systems.
- It is rare that Abramsky publicly talks about philosophical issues, but he does so in at least two articles, Information, Processes and Games (Abramsky, 2008b) and Intensionality, Definability and Computation (Abramsky, 2014), both of which problematise the concept of process. The notion of process has been central in his work.
- It is general process science as opposed to general system science, encompassing different types of processes, such as computational processes, logical processes (e.g., deductions), and physical processes (e.g., time evolution). He even speculates about the Church-Turing thesis for processes.

- The process-theoretical connection with physics was explicated in his work on categorical quantum mechanics in the early twenty-first century, which has then seen a flourish of categorical science in different guises.
- Categorical quantum mechanics was developed in joint work with Bob Coecke. The categorical correspondence between logical, computational, and physical processes may be called the Abramsky-Coecke correspondence.
- Abramsky's more recent theory of contextual semantics even works across physics, computer science, logic, linguistics, and cognitive science, by sheaf theory as an abstract theory of how local information can(not) be patched together to form global information.

In such a way category theory has served as foundations of science rather than just mathematics. Abramsky has played a leading rôle in this process, and he has even addressed different issues in economics (Abramsky & Winschel, 2017), social choice theory (Abramsky, 2015), artificial intelligence (Abramsky et al., 2013), and so fourth, within the same categorical paradigm. Here let us focus upon logic, informatics, and physics, and their Abramsky-Coecke correspondence between them. Recall that a category is a network structure of objects and morphisms (and their composition), which can arise from different sciences other than mathematics. In mathematics, there is a category of algebras and homomorphisms, a category of spaces and continuous maps, and so on. There are different categories in different sciences:

- Physics: objects are physical systems (e.g., quantum systems as Hilbert spaces), and morphisms are physical processes (e.g., quantum processes as bounded operators). The category of (finite-dimensional) quantum systems (as used in quantum informatics) and processes forms a compact closed category, which is a special type of $*$-autonomous category, which, in turn, is a special type of monoidal closed category.
- Logic: objects are propositions, and morphisms are proofs or deductions (from one proposition to another). It is well known that the category of propositions and proofs in intuitionistic logic forms a bicartesian closed category. It is also known that the category of propositions and proofs in classical linear logic forms a $*$-autonomous category, sharing much the same structure as the category of quantum systems. The Abramsky-Coecke correspondence in a narrow sense is the correspondence between substructural logic and quantum physics. They have shown that the logic of quantum mechanics is substructural; this was an antithesis to the traditional conception of quantum logic in the work of von Neumann and Birkhoff, which they argue is 'non-logic' whereas theirs is 'hyper-logic' (Abramsky, 2008a; Coecke, 2007).
- Computer Science: objects are data types, and morphisms are (functional) programs. The category of types and programs in the standard typed λ-calculus forms a cartesian closed category, sharing almost the same structure as intuitionistic logic. The correspondence between logic and computation is called the Curry-Howard correspondence, a physical extension of which is what we call the Abramsky-Coecke correspondence. It is notable that categorical quantum mechanics was

naturally born from categorical linear logic and type theory; technically or categorically, they are not very different.

Category theory thus allows us to analyze analogies and disanalogies between different sciences in a systematic manner, and thereby enables knowledge transfer between different sciences. Categorical foundations of science make it transparent what is shared by different sciences and what is not. A case of knowledge transfer enabled by category theory may be seen in a (semi-)automated reasoning system for quantum mechanics and quantum computation, which has been developed by applying automated reasoning in logic and computer science to quantum mechanics through the categorical Abramsky-Coecke correspondence. Categorical quantum mechanics provides different axiomatisations of quantum mechanics, and would arguably count as a solution to the Hilbert's Sixth Problem; the axiomatisation of physics was envisaged by Hilbert after his axiomatisation of mathematics, and yet could not be realised for a long time.

These developments of categorical foundations of science, quite some of which are Abramsky's contributions and their descendants, pave the way for the categorical unity of science, and may lead us to a new kind of unified science, i.e., categorical unified science as pluralistic unified science. The situation of fragmentation is presumably even more severe in science today than in any past era, and it may lead us to some sort of scientific nihilism supporting the disunity of science. The search for unity, absolute foundations and the like mostly failed in the twentieth century as we discussed above. There is no first philosophy (Quine) or grand narrative (Lyotard), and there is no ultimate foundation of mathematics, let alone that of knowledge in general. Today we may live in the age of post-truth if truth is a 'grand narrative' of unifying nature. The global order of the sciences or knowledge therein may be a mirage as well as the global order of the world or the societies. Categorical foundations nonetheless give us some hope for the unity of the sciences, not in the old-fashioned Vienna School style, but in the modern categorical (Oxford) style, the nature of which shall be explored and elucidated in the following. Characteristics of the categorical unity of science may be summarised as follows:

- It supports scientific pluralism, the central tenet of the Stanford School, in accordance with the decentred view of the sciences. There is no privileged Archimedean vantage point upon which everything is grounded, and there is no epistemological or ontological centre of the network of the sciences thus conceived. There is no monistic reductionism involved.
- It allows the epistemological and ontological networking of the sciences that accounts for both structural analogies and disanalogies between different fields, and that facilitates knowledge transfer between different fields. It tells us higher laws unifying or interconnecting different laws in different sciences (cf. 'labyrinth of the multiplicity of possible particular laws' and their unification; see Kant, 2007).
- It even opens up the possibility of going beyond the Cartesian dualism; for example, categorical structure shared by logic, the laws of reason, and physics, the laws of nature, tells us commonalities between the realms of matter and of mind which

are separated in the Cartesian dualism (cf. double-aspect theory of information Chalmers, 1996, in which the physical and the phenomenal are derived from the fundamental reality of information, which is arguably structural in nature, and may be described in terms of category theory; the categorical informational view of reality may be called informational structural realism).

- It is 'unity from below', based upon down-to-earth scientific practice by different applied category theorists. The Vienna Circle attempted to invent or force 'unity from above' and failed after all. Unity dogmatically imposed upon scientific practice (by philosophers, policy makers, or whoever) often does not work and tend to collapse quite soon. There is no such revisionism involved in the categorical 'unity from below'.

The categorical unity of science, by itself, is not a normative, philosophical doctrine, but rather it is more like a descriptive concept to symbolise the scientific practice of categorical scientists as represented by Samson Abramsky. There would be no other approach to the unity of science that is supported by actual scientific practice rather than unrealistic philosophical fantasies. The categorical unity of science never implies that all scientists learn and use the methodology of category theory. There would arguably be no monolithic norm to unify different scientists with different backgrounds, different ontologies, and different epistemologies; philosophers such as Galison would rather argue that there should be no such thing in the first place. From such a pluralist point of view, the possibility of unified science lies in the pursuit of unity as a special science (since all sciences are special sciences); unified science, then, is just a new kind of special science to network the rest of the special sciences, and to give a unified, systematic understanding of how they interlace with each other. There is thus no revisionism involved in the pluralist conception of unified science. This is what is meant by the pluralist unity of science.

The problem of unity we face today may be characterized as a lack of the local-to-global correspondence: there is no global worldview to glue local worldviews in different sciences together into a coherent whole. The primary task of contemporary unified science, then, would be the global networking (or binding in Galison's terms) of local epistemologies and ontologies in different sciences that nevertheless involves no universalised absolute ontology or epistemology. Categorical unified science arguably serves as theoretical foundations to incarnate pluralistic unified science thus conceived (for category theory and the unity of science, see also (Maruyama, 2019a); some of the discussion above, especially the distinction between global, local, and conceptual foundations, is based upon or adapted from material developed in (Maruyama, 2019a), even though there are certain conceptual differences between the perspective developed here and the one in (Maruyama, 2019a), which reflect the author's changing perspective on philosophy of category theory).

2.2.3 Remarks on the Philosophy of Category Theory

Category theory has the remarkable potential to contribute to philosophy of mathematics, philosophy of science, and unified philosophy, integrative philosophy, or synthetic philosophy beyond the analytic-continental divide in various manners. Here we summarise several ideas concerning categorical structuralism, including categorical epistemology and categorical ontology, which have striking philosophical consequences.

Categorical structuralism is higher-order structuralism, and at the same time, structuralism without substance; set-theoretical structuralism is based upon substance metaphysics whilst categorical structuralism is based upon function or process metaphysics (cf. Cassirer's philosophy of substance and function with the priority of function over substance; Whitehead's process philosophy; Wataru Hiromatsu's philosophy arguing for the shift from object metaphysics to event metaphysics). In set-theoretical structuralism, such as Bourbaki's and model-theoretic structuralism, relata are prior to relations; in categorical structuralism, by contrast, relations are prior to relata. Category theory presents the purest notion of structure without any substance whatsoever, whereas the set-theoretical notion of structure is sort of impure in the sense that it presupposes substance, namely underlying sets and their elements, as being prior to structure. Categorical metaphysics is the metaphysics of functions or processes without any substantivalism involved.

Categorical structuralism is higher-order in the following two senses. First, categorical structuralist epistemology allows us to elucidate higher-order meta-laws. Categorical duality between the algebraic and topological worlds, for example, allows us to explicate the higher-order meta-laws that govern both algebraic and topological laws at once; duality between dualities as conceived in the category of dualities tells us about even higher-order laws, or in Banach-Ulam's terms, analogy between analogies. The Abramsky-Coecke correspondence between logic, physics, and computation through category theory tells us the existence of higher-order structural laws governing those three different realms. Second, categorical structuralist ontology gives the higher-order concept of structure such as the structure of structures; it even allows us to abstract, and thus structurally relativise, the relationships between substance and structure, as in the so-called concrete category theory (over general bases), which is actually even more abstract or general than ordinary category theory. Higher category theory instantiates the higher-order nature of categorical structuralism as well.

Categorical metaphysics gives several interesting suggestions in philosophy of science and philosophy of mathematics in particular; here we shall address three of them. Categorical ontology allows for reduction of ontological commitment via structural realism. The so-called indispensability argument tells that those abstract entities that are indispensable for our scientific theories must be taken to exist, so that we must take the existence of mathematical objects at face, thus committing ourselves to mathematical platonism. Yet we can still debate what are really indispensable for scientific theories. Category theory allows us to reduce the ontological commitment to objects into the ontological commitment to structures, which must still exist according to the indispensability argument, but are ontologically more modest than objects themselves. In a nutshell, categorical metaphysics provides lightweight ontology for mathematics and science; we shall discuss the categorical principle of ontology-lite in the context of categorical quantum mechanics as well.

Closely related to this is the categorical structuralist resolution of Benacerraf's dilemma, which tells that the standard referential semantics of mathematical language necessarily leads to the platonistic existence of abstract mathematical entities, and yet is incompatible with the causal account of knowledge, since we humans have no causal epistemic access to the realm of those abstract mathematical entities that do not exist in our universe. Brouwer's intuitionism does not suffer from Benacerraf's dilemma, since mathematical entities are mental entities in Brouwer's intuitionism and there is no problem with our minds having causal epistemic accesses to mental entities in our own brains. Brouwer's intuitionism nonetheless faces yet another type of challenges such as the objectivity of mathematical entities and the certainty of mathematical truths. This may be called as the dual of Benacerraf's dilemma or Benacerraf's dual dilemma. If one employs a heavy ontology of mathematics such as platonism, then one faces Benacerraf's dilemma; if one employs a light ontology of mathematics such as intuitionism and nominalism, then once faces Benacerraf's dual dilemma. All in all, this may be called Benacerraf's duality. Note that nominalism does not suffer from Benacerraf's dilemma, but does suffer from the problem of explaining the applicability of mathematics to the world, i.e., what Wigner called the unreasonable effectiveness of mathematics. How about categorical structuralism? Categorical structuralism arguably does not suffer from any of Benacerraf's dilemma and its dual dilemma. The structuralist reduction of mathematical ontology allows us to explain mathematics in terms of structures only, so that we have a better epistemic access to the objects of mathematics, namely structures, which are ontologically lighter than platonistic mathematical entities, and could be found either in the world or in the mind. Categorical structuralism does not suffer from Benacerraf's dual dilemma, either. The objectivity of mathematical objects consists in the objectivity and intersubjectivity of structures in the world and in the mind, respectively. Categorical structuralism allows us to account for the applicability of mathematics in terms of the structural mapping between mathematics and reality (cf. Pincock's structural mapping account of the applicability of mathematics). So the categorical metaphysics of mathematics allows us to resolve both Benacerraf's dilemma and Benacerraf's dual dilemma at once. We shall revisit this striking feature of categorical metaphysics in the context of categorical quantum mechanics below, which allows for lightweight ontology for quantum mechanics, or the 'many structures' interpretation, the structuralist interpretation of quantum theory, as opposed to the ordinary 'many worlds' interpretation, the substantivalist interpretation of quantum theory.

Benacerraf's dilemma and its dual dilemma can be understood in the general context of the fundamental tension between ontology and epistemology in philosophy. Stronger ontology such as naïve realism and platonism makes epistemology more difficult because of Benacerraf's dilemma on the epistemic access to abstract entities, and weaker ontology such as naïve antirealism and nominalism makes epistemology easier, but instead faces Benacerraf's dual dilemma on the objectivity, certainty, and applicability of entities and truths about them. Benacerraf's duality, therefore, is about the dual relationships between ontology and epistemology in general.

Category theory has recently been applied in philosophy of science, especially in the context of theory equivalence. There are a number of options to define theory equivalence in terms of category theory, such as categorical equivalence and Morita equivalence. Morita equivalence only works for theories formulated in terms of formal logic such as first-order logic, but categorical equivalence, of course, work for any categories. Yet it is actually highly non-trivial how to regards theories as categories. Especially, the notion of morphisms in categories significantly affects the resulting notion of theory equivalence. If we, for example, only considers logical connectives, we have to accept, for example, the surprising fact that the Peano arithmetic and the ZF set theory are equivalent as theories. If we consider finer structures, we can escape from such collapsing phenomena. These considerations lead us to subtle issues of both mathematical and philosophical significance, which shall be addressed in greater detail somewhere else.

There are also some other issues in philosophy of science that category theory can shed new light on. It is remarkable, among other things, that categorical duality allows us to show equivalence between the Newtonian absolute conception of space and the Leibnizian relational conception of space, or between the point-set conception of space and the point-free conception of space. The Leibnizian conception of space is epistemologically more certain than the Newtonian conception, since the existence of points, namely prime ideals as in duality theory and algebraic geometry, hinges upon the validity of indeterministic principles such as the axiom of choice and its weaker versions. The Leibnizian point-free conception of space allows us to constructivise topology, just as Martin-Löf-Sambin's formal topology does. Technically, some point-free theories of space such as topos theory and locale theory actually have the same consistency strengths as their classical counterparts, and yet it is possible to modify them so as to make them truly constructive in the sense of predicative constructive set theory or type theory (note that Martin-Löf type theory and homotopy type theory are strictly weaker than classical set theory ZF; their difference consists in predicativity/impredicativity, as toposes and locales are impredicative). It is even possible to give a predicative constructive foundation for Grothendieck's scheme theory in algebraic geometry. And this leads us to a predicative constructivist realisation or reconstruction of Hilbert's programme by Thierry Coquand and his collaborators (see, e.g., Coquand, 2009), in which Hilbert's elimination of ideal elements corresponds to the Leibnizianisation or relationalisation of space, since points are ideal elements to be thus eliminated. This is done on the basis of categorical duality; for example, Joyal's duality theory allows us to describe the spectrum of a ring in a completely point-free manner, so that points as ideal elements are successfully eliminated. At the same time it is nevertheless allowed to exploit the merits of point-based reasoning through the duality-theoretical equivalence between point-set space and point-free space. Categorical duality, therefore, plays a prominent rôle in contemporary developments of Hilbert's programme. Besides, there is the relativity of the notion of point in duality theory, which is interesting philosophically as well as mathematically (see, e.g., Maruyama, 2010; 2013a; 2020a). Philosophy of duality, or philosophy of categorical duality in particular, would deserve more attention in the philosophy community (for more on philosophy of duality, see, e.g., Maruyama, 2012; 2013a; 2013c; 2017; 2017, especially their concluding sections).

Philosophy has not been so structural as mathematics. Yet structural perspectives on philosophy can be fruitful as categorical duality allows us to regard the Newtonian concept of space and the Leibnizian concept of space as being structurally equivalent with each other. Another application of categorical duality or rather categorical logic is the unification of realist and antirealist conceptions of meaning as problematised by Michael Dummett. The referential or denotational view of meaning, which presupposes the existence of reality and makes sense of meaning via the picture-theoretic correspondence between language reality, categorically turns out to be equivalent with the inferential or autonomous of meaning, which has not outward reference to reality and makes sense of meaning via the internal use-theoretic structure of language per se, such as the structure of inferences to characterise the meaning of logical constants as in inferential proof-theoretic semantics as opposed to referential model-theoretic semantics. Mathematically, there are two rationales for this equivalence between the truth-theoretic theory of meaning and the proof-theoretic theory of meaning. On the one hand, there is categorical duality between semantics and syntax, and on the other, categorical logic gives the unified conception of logic in which proof-theoretic and truth-theoretic models are subsumed under one and the same thing notion of models and interpretations. There are syntactic/proof-theoretic categories/hyperdoctrines, interpretations with which correspond to proof-theoretic semantics, and at the same time, there are semantic/model-theoretic categories/hyperdoctrines, interpretations with which correspond to model-theoretic semantics. So the realist and antirealist theories of meaning are structurally equivalent with each other in the aforementioned two senses, namely in terms of both categorical duality and categorical logic (for more on this, see, e.g., Maruyama, 2016a; 2016b; 2017; 2013b; 2020b).

These resolutions of binary oppositions in philosophy would even articulate the broader significance of categorical structuralism in philosophy in general, rather than of science or mathematics in particular. The categorical structural philosophy in the sense articulated above would lead us to unified philosophy, integrative philosophy or synthetic philosophy, as opposed to analytic philosophy and continental philosophy, in which there is no debate between realist and antirealist worldviews, since they are just structurally equivalent with each other. This would have many significant consequences for our conception of philosophy. For example, scientific realism in analytic philosophy may not necessarily be incoherent with social constructivism in continental philosophy, since realism and antirealism can be equivalent with each other. Philosophical structuralism, moreover, would allow us to problematise different connections between the analytic and continental traditions of philosophy, thus transgressing the boundaries beyond the analytic-continental divide. Foucault's *Les Mots et les Choses* (Words and Things) could be structurally homomorphic with Quine's *Words and Objects*. Quine was strongly against Derrida's philosophy, and yet Quine's abandonment of the Cartesian goal of a first philosophy may be homomorphic with Derrida's deconstruction of logocentrism. Both Quine and Derrida, furthermore, argue for the indeterminacy of meaning on formal logical grounds (even if Derrida's mathematical understanding of logic is problematic). Davidson's "there is no such thing as a language" may be homomorphic with Derrida's "there is no outside-text." Quine's abandonment of the Cartesian goal of a first philosophy may also be homomorphic with Heidegger's *destruktion of cogito sum*, Lyotard's "end of grand narrative", or Foucault's archaeology of knowledge asserting "man is an invention of recent date." The structuralist philosophy would allow us to discover this sort of structural linkages between different philosophical traditions. This is the broad philosophical significance that categorical structuralism could have in the context of pure philosophy. The structuralist philosophy in this sense endorses the same kind of conceptual structuralism as the structuralist mathematics; structuralism in the ordinary sense of philosophy is not so structural after all. It is merely philosophy about structure. Yet the philosophical structuralism is structural philosophy or a structural way to do philosophy, thus elucidating the structural relationships between different philosophies. It is philosophy regarding philosophies as structures, explicating and articulating the structure of thought. This may also be called categorical philosophy as opposed to philosophy of category theory. In a nutshell, structural philosophy is different from philosophy of structure, problematising philosophies as structures rather than building philosophies on the basis of the notion of structures.

Let us finally mention the two dogmas of set theory, which are the following two forms of reductionism: the ontological foundationalist doctrine of element reductionism or elementism and the epistemological foundationalist doctrine of logical reductionism or formalism. The fundamental rôle of category theory in foundations of mathematics arguably consists in liberating mathematics from material set theory or from its 'pernicious idioms' as Wittgenstein calls them (if mathematics is a wild animal, set theory is like a jail confining that highly lively animal). Category theory presents the pluralistic multiverse view of science as opposed to the set-theoretical reductionist 'universe' view, thus emancipating mathematics and the sciences from the reductive foundationalist doctrine of set theory or the 'pernicious idioms of set theory' as Wittgenstein calls them.

All this is just a bird's-eye view of the philosophical significance of category theory from different angles. Some of these issues shall be revisited in the following discussion on duality, intensionality, and contextuality, which are all more or less concerned with philosophy of category theory and with philosophy of Samson Abramsky.

2.3 Duality, Intensionality, and Contextuality

In the last section we gave a global perspective on Abramsky's work, which opens up the possibility of categorical unified science, leading us to the categorical unity of the sciences. In the present section we give local perspectives on different issues he has addressed in his work. His major achievements so far include (and are not exhausted by) the following four themes. The present landscape of any of them simply would not have existed, or would have been very different, without Samson Abramsky's contributions to it.

1. Domain theory and the logic of observable properties.
2. Game semantics, full abstraction, GoI, and full completeness.
3. Categorical quantum mechanics and the Abramsky-Coecke correspondence.
4. Contextual semantics across the sciences (physics, informatics, and beyond).

These developments are not independent of each other. Domain theory allowed us to give compositional semantics of computation, but did not allow for full abstraction in particular. Game semantics provided an effective paradigm to solve the full abstraction problem. Categorical quantum mechanics technically benefited much from the study of categorical semantics in linear logic and its full completeness. The Abramsky-Coecke correspondence extends the Curry-Howard(-Lambek) correspondence in logic and computation to the realm of physics. Contextuality studies via sheaf theory (Abramsky & Brandenburger, 2011) complement (and might possibly be seen as an antithesis to) categorical quantum mechanics, leading to the general theory of logical Bell inequalities (Abramsky & Hardy, 2012), which unify the fault-tolerant feature of Bell-type theorems 'with inequalities' with the logically robust feature of Bell-type theorems 'without inequalities'.

From a philosophical point of view, the first theme involves the notion of duality, the second the notion of intensionality, and the fourth the notion of contextuality, the philosophical significance of the third being conceptualised in the Abramsky-Coecke correspondence, which would be the fundamental underpinning of general process theory as opposed to general system theory. In the following we touch upon philosophical issues related to these notions. Abramsky does not claim much about the philosophical significance of his work, which is nonetheless nontrivial, and at the same time it actually seems that there have often been conceptual ideas of broader philosophical significance underlying his technical achievements.

2.3.1 Duality Between Reality and Observables

Abramsky's early work, including his Ph.D. thesis (Abramsky, 1987), is well known for applying Stone duality to semantics of computation, in particular domain theory, having won the LICS Test-of-Time Award in 2007. He is also the first discoverer of the duality between modal logic and the coalgebras of the Vietoris hyperspace functor (Yde Venema suggested the author to call it Abramsky duality in Kyoto in 2011 when the author was wondering about how to name it whilst writing a paper on a general theory of coalgebraic dualities (Maruyama, 2012), which includes additional historical remarks about the coalgebraic duality for modal algebras). Let us place Abramsky dualities in the broader landscape of categorical dualities between the ontic (reality) and the epistemic (observables) in different fields of mathematical science:

	Ontic	**Epistemic**	
Algebraic Geometry	*Variety/Scheme*	*k-Algebra/Ring*	Hilbert-Grothendieck
Representation Th.	*Group*	*Representations*	Pontryagin-Tannaka
Galois Theory	*(Profinite) G-Set*	*Algebra Extension*	Galois
Topology	*Topological Space*	*Algebra of Opens*	Isbell-Papert
Logic	*Space of Models*	*Algebra of Theories*	Stone
System Science	*System (Coalg)*	*Behaviour (Alg)*	Abramsky
Program Semantics	*Denotations*	*Observable Properties*	Abramsky
Quantum Physics	*State Space*	*Observables*	von Neumann

The Abramsky duality of program semantics is the duality between the space of denotations of programs and the logic of observable properties of them. The Abramsky duality of system science is the duality between (coalgebraic) systems and (algebraic) modal logics to reason about their behaviour. Both of them look quite similar to the von Neumann duality of quantum physics. The state-observable duality seems to be ubiquitous in physics and computer science; one of the reasons for this may be that physics and computer science share the 'logic' of states and observables on systems. Duality is a traditional theme in philosophy as well as mathematics. There are different dualities between the ontic and the epistemic in different traditions of philosophy:

	Ontic	***Epistemic***	
Descartes	*Matter*	*Mind*	Cartesian Dualism
Kant	*Thing-in-itself*	*Appearance*	Idealism
Cassirer	*Substance*	*Function*	Logical Idealism
Heidegger	*Essence*	*Existence*	Analysis of *Dasein*
Whitehead	*Reality*	*Process*	Holism/Organicism
Wittgenstein	*World*	*Language*	Logical Philosophy
Searle	*Intentionality*	*Simulatability*	Philosophy of Mind
Dummett	*Truth*	*Verification*	Theory of Meaning

The aforementioned Gödelian shift from the 'right' to the 'left' may be compared with the shift from substance to function in Cassirer, the shift from essence to existence in Heidegger, the shift from reality to process in Whitehead, the shift from the world to language in Wittgenstein (aka. the Linguistic Turn Rorty, 1962; 1967, which may be encompassed by the broader notion of the Informational Turn as addressed below), and the shift from truth to verification in Dummett, which led to developments of what is now called proof-theoretic semantics. An analogous shift happened in the conception of logic and geometry, and may be called the Intensional Turn, in which Abramsky, especially his notion of full completeness and game semantics, played a major rôle, as we shall see below. What is not included in the above picture is

Lawvere's duality between the formal and the conceptual, which, too, may be seen as a manifestation of the duality between the epistemic and the ontic.

In the following, let us take a closer look at the duality landscape as presented above, especially the philosophy of duality across different traditions. Broadly speaking, duality emerges between realist and antirealist aspects of fundamental concepts. What is Meaning? The realist would assert meaning lies in the picture-theoretic correspondence between language and reality. The antirealist, by contrast, would argue that meaning consists within the fully autonomous/internal system/structure of language or linguistic practice. The dichotomy between the realist and antirealist conceptions of meaning is essentially homomorphic with the dichotomy between Wittgenstein's early and later conceptions of meaning, between Davidson's and Dummett's conceptions of meaning, or between model-theoretic and proof-theoretic semantics. It is also analogous to the dichotomy between symbolic compositional semantics and statistical distributional semantics in artificial intelligence and natural language processing, or between Chomsky's and Norvig's conceptions of meaning (see, e.g., Maruyama, 2019a and references therein). Wittgenstein's later conception of meaning, according to which "words are not a translation of something else that was there before they were" (Wittgenstein, 1970), is homomorphic with Derrida's conception of meaning, according to which "there is no outside-text." What is Truth? The realist would argue that it is constituted by the picture-theoretic correspondence of statements to states of affairs in reality or in the world. The antirealist, by contrast, would contend that truth has no outward reference to reality, fully given by the internal structure/system/coherency of statements or by a certain type of instrumentalist pragmatics. These may be compared with Russell and Bradley's conceptions of truth, respectively. What is Being? The realist would argue that it is static substance (as in Aristotle's metaphysics). The antirealist would contend that it is born via emergence, i.e., it emerges within a dynamic process, cognition, structure, network, environment, or context (as in Cassirer's functionalist metaphysics, Whitehead's process metaphysics, and Heidegger's existentialist metaphysics as opposed to Aristotle's essentialist metaphysics). What is Intelligence? The realist (such as John Searle) would maintain that genuine intelligence is something greater than behavioural simulation as in the Turing Test, vitally driven by the intentionality and other salient features of the human mind. The antirealist would content herself/himself that intelligence is adequately constituted by behavioural simulation or copycatting as in the Turing Test or the Chinese Room. What is Space? The realist (such as Newton) would contend that space is a collection of substantival points with no extension. The antirealist, by contrast, would argue that space is a structure of regions, properties, relations or information with no substantival (Leibniz, Husserl, and Whitehead would share this idea more or less, even though they are usually not classified antirealists). All this is manifestations of ontic-epistemic duality or formal-conceptual duality in Lawvere's terms; to be precise, duality deconstructs dualism as developed above by proving that the ontic and the epistemic, and thus the realist and the antirealist worldviews, are structurally equivalent with each other.

What is particularly significant, from a categorical point of view, in the duality landscape above is the Cassirer duality between substance and function and his genetic conception of knowledge along with what could be called the higher-order theory of concept formation, which may be seen as a precursor of higher-order categorical structuralism as elucidated above. Cassirer's early studies were concerned with Leibniz's philosophy, especially his relationalism, which influenced Cassirer's philosophy of the functional in his *Substance and Function*, the main focus of which was placed upon higher-order relationalism and the higher-order relational understanding of the sciences. Cassirer's higher-order relationalism or higher-order functionalism had finally evolved into his mature *Philosophy of Symbolic Forms* as global philosophy of culture subsuming the sciences as part of it. Leibniz's relationalism is conceptually akin to categorical structuralism, and category theory may be regarded as incarnating Leibniz's *characteristica universalis* in contemporary form. In light of Cassirer's philosophy, modernisation is functionalisation, the radical departure from substantivalism. In Cassirer's philosophy, the copy theory of knowledge is replaced by the genetic theory of knowledge:

> Myth and art, language and science, are in this sense shapings toward being; they are not simple copies of an already present reality, but they rather present the great lines of direction of spiritual development, of the ideal process, in which reality constitutes itself for us as one and many – as a manifold of forms, which are nonetheless finally held together by a unity of meaning. (Cassirer, 1955; translation by Friedman, 2000)

Reality, truth, or knowledge does not exist on its own in a static manner; rather, it is generated as "never-completed X" (cf. (Friedman, 2000)), which merely exists in the ideal limit of the generation process. This is essentially the same as the process of generating points (as prime ideals or ideal elements in Hilbert's terms) in point-free topology such as locale theory and formal topology; points constitute reality in topology. Cassirer extends this idea to the whole realm of human knowledge and culture, and the resulting philosophy is the philosophy of symbolic forms. Symbolic forms are shapings towards reality, finally held together by a unity of meaning into a coherent whole constituting functional reality. We could conceptually equate symbolic forms with categories, so that the philosophy of symbolic forms is the philosophy of categories; however, symbolic forms are arguably broader in its conceptual scope than categories per se, encompassing all the aspects of human civilisation.

If there is any single philosophy that is closest to the philosophy of category theory, it would be Cassirer's philosophy, which is arguably one of the earliest origins of structuralism or structural realism, and presumably the earliest origin of higher-order structuralism or higher-order structural realism. So let us elaborate upon Cassirer's philosophy in the following. Cassirer's shift from the copy theory to the functional theory is in conceptual parallel with the Linguistic Turn, the Informational Turn, and the Intensional Turn as shall be addressed in the next subsection. Cassirer's philosophy, from a broader perspective, regards modernisation as functionalisation (cf. Max Weber's conception of modernisation as disenchantment); this idea actually works in diverse contexts of modernisation, including mathematics, physics, and beyond. Structuralist mathematics is the functionalisation of traditional mathematics. Quantum physics is more functionalist than classical physics, which is basically substantivalist. Point-free topology functionalises the concept of space; functionalisation may also be called Leibnizianisation. The Leibnizianisation of space has been common in noncommutative geometry and quantum physics as well as point-free topology and topos theory; all this is underpinned by categorical duality, the duality-theoretic correspondence between point-set and point-free space. Quantum physics, from an epistemic point of view, could even be regarded as functionalising reality per se. Artificial intelligence is the functionalisation of the mind. Quite some part of modern science is functionalised, and Cassirer was aware of this functionalist tendency at a remarkably early stage. The logic of modernisation as functionalisation even applies to modern art. Paul Klee's famous saying goes as follows: "Art does not reproduce what we see; rather, it makes us see." Alberto Giacometti's similar saying is as follows: "The object of art is not to reproduce reality, but to create a reality of the same intensity." This is a highly functionalist idea, which is, at the same time, a highly modernist idea. The modernisation of mathematics and the sciences arguably went hand-in-hand with the modernisation of art or even culture in general. Cassirer, as the so-called last philosopher of culture, expressed the modernist tendency as the functionalist tendency in its essence, a prominent characteristic of which is the modernist departure from reality (to be precise, (the departure from the naïvely posited concept of reality). Rorty's, *Philosophy and the Mirror of Nature* (1970) presents an idea similar to the departure from reality, arguing for the abandonment of the mirror of nature, which is akin in its spirit to Klee's and Giacometti's conceptions of (modern) art.

Cassirer is regarded as an origin of both analytic structuralism and continental structuralism. Charles Parsons, a philosopher of mathematics in the analytic tradition of philosophy, mentions Cassirer as a structuralist: "Cassirer takes a position that could be called structuralist" (Parsons, 2014). Steve Lofts, a philosophy in the continental tradition, regards Cassirer as a structuralist, too:

> [T]he philosophy of symbolic forms is presented as a *type* of "structuralism" *avant la lettre* resulting from a fusion and critical transformation of Kant's transcendental critical philosophy, on the one hand, and Hegel's phenomenology of spirit, on the other. (Lofts, 2000)

Cassirer is not a structuralist with respect to a single subject such as mathematics or physics. That is to say, he is not a local structuralist. Cassirer is rather a global structuralist, his structuralism spanning a broad variety of scientific and cultural disciplines. Cassirer even expresses a structuralist idea closer to the philosophy of category theory that abandons the set-theoretical 'myth of the given' (in Sellars' terms), namely the element-based metaphysics of mathematics:

> The relational structure as such, not the absolute property of the elements, constitutes the real object of mathematical investigation (Cassirer, 1923)

Mathematical structuralism is confined to this claim. Yet Cassirer's structuralism is all-encompassing universal structuralism, so that the structuralist idea is extended to things in general beyond mathematical entities:

> [W]e can only reach the category of thing through the category of relation (Cassirer, 1923)

Cassirer endorses the categorically sounding 'relative point of view' that relative properties are "the first and positive ground of the concept of reality" (Cassirer, 1923). Reality is symbolically constructed from relative properties. Cassirer's idea of the symbolic construction of reality seems to be the philosophical counterpart of the topos-theoretic or locale-theoretic idea of the symbolic construction of space. Cassirer is a Neo-Kantian in the so-called Marburg School rather than the Southwest School (which Heidegger stemmed from; and Cassirer and Heidegger had the well-known Davos debate, in which young Carnap attended as well; see Friedman, 2000). Cassirer explicitly refers to Kant's philosophy as follows, which already had some flavour of structuralism (to be precise, epistemic structural realism):

> "Whatever we may know of matter," here, too, we can cite the Critique of Pure Reason, "is nothing but relations, some of which are independent and permanent and by which a certain object is given us" (Cassirer, 1923)

Cassirer's structuralism arguably originates in Göthe's morphology as well as Kant's philosophy. When Cassirer developed his essentially structuralist idea, the term *structuralism* was however not popular yet. Cassirer was nevertheless aware of the term *structuralism* as early as 1945 when he published *Structuralism in Modern Linguistics*. Cassirer's arguments in this article go as follows: "language is neither a mechanism nor organism"; "we must say that language is 'organic', but that it is not an organism"; "the program of structuralism developed by Brondal is, indeed, very near to Humboldt's ideas."

Let us summarise the conceptual genealogy of how human thought has led to functionalism, structuralism, and the like (note that functionalism in this article means Cassirer's and its relatives rather than functionalism in the computational theory of mind, which is subsumed under Cassirer's functionalism). Cassirer's shift from the substantival to the functional is analogous to the Linguistic Turn, which however only encompasses, in the usual understanding of the term, Frege, Russell, Wittgenstein, and the like on the analytic side, and Saussure, Derrida (and possibly Foucault) and the like on the continental side. Cassirer has no place in it, and his philosophy is, as a matter of fact, much broader than linguistic philosophy. Cassirer's turn may thus be called the Symbolic Turn, which subsumes the Linguistic Turn. Or it could be placed within the Informational Turn coming after the Linguistic Turn. Both Leibniz and Cassirer are informational philosophers, and their philosophies may be regarded as the philosophy of informational reality. Cassirer's symbolic construction of reality is actually an instance of the informational construction of reality, which has actually been done successfully in what is called information physics. Information physics indeed allows for informational reconstructions of quantum physics and some others, presenting the view that ultimate reality is informational, i.e., the ultimate constituents of reality is information rather than matter, energy, and so fourth. Information is arguably structure, and the view that information is the most fundamental elements of reality leads to the structuralist conception of reality. Symbol is information, and information, in turn, is structure. Any of these three things can play the rôle of fundamental building blocks of reality. The corresponding 'turns' may be called the Symbolic Turn, the Informational Turn, and the Structuralist Turn; yet these share a certain core idea.

Rorty popularised the Linguistic Turn view of (history of) philosophy. Cassirer is regarded today as potentially being able to reconcile the analytic and continental traditions of philosophy (see, e.g., Friedman, 2000). The same applies to Rorty, who understood both traditions, his philosophy being explicitly geared towards post-analytic philosophy integrating the two traditions. Rorty argues that philosophy transformed from philosophy of things (e.g., Aristotle) to philosophy of ideas (e.g., Kant), and from philosophy of ideas to philosophy of words (e.g., Wittgenstein):

> [T]he dialectical progression which in the course of three centuries has led philosophers away from things to ideas, and away from ideas to words (Rorty, 1962)

The shift from things to ideas is the so-called Copernican Turn. Simply speaking, ideas precede things. The shift from ideas to words is the Linguistic Turn. Simply speaking, words precede any of things and ideas. These may be seen as the virtually necessary transformations of philosophy according to Rorty:

> Given the initial Cartesian sundering of cognition from the object of cognition, it was only to be expected that epistemology should claim priority over metaphysics. Once this happened, it was only a matter of time before metaphysics shrivelled away and died. In due course, epistemology, finding itself unable to function as an autonomous discipline when deprived of metaphysical support, was pushed aside by its ungrateful child, linguistic analysis. (Rorty, 1962)

What would be the next turn? It is arguably the Informational Turn from words to information. Information precedes any of things, ideas, and words, thus being the fundamental building blocks of reality. The informational perspective on reality has been developed, for example, by David Chalmers; in his double aspect theory of information (Chalmers, 1996), the most fundamental informational reality governs both the physical and the phenomenal, which are two different types of informational reality, his position thus being called property dualism. Information physics, artificial intelligence, and computational biology are all consequences of the Informational Turn across the sciences (including humanities). What is happening after the Informational Turn is actually the return to things per se, namely the Realist Turn or the Neo-Realist Turn (cf. Marx Gabriel).

Let us recapitulate all the turns in the history of human philosophical thought as follows.

- The Copernican Turn from philosophy of things to philosophy of ideas, as represented, for example, by Kant.
- The Linguistic Turn from philosophy of ideas to philosophy of words, as represented, for example, by Wittgenstein.
- The Informational Turn from philosophy of words to philosophy of information, as represented, for example, by Chalmers and Floridi.
- The Realist Turn (aka. the Speculative Turn), a reactionary shift towards a new kind of realism, as represented by Meillassoux (known for speculative materialism/realism) and Ladyman (known for structural realism).

Even mathematics may be seen as following the same Rortyan pattern of development: mathematics shifted from things to infinities (ideas), and from infinities to symbols (words), which characterise axiomatic mathematics. There may be no informational turn in mathematics, but the return to concrete things has happened around this century (which has urged mathematicians to solve concrete problems via the abstract methods of axiomatic mathematics by Bourbaki, Grothendieck, and others). If we regard modernisation as disenchantment as Weber does, we could regard post-modernisation as reenchantment as Berman, (1953) and others do. Mathematics has been reenchanted during post-modernisation after the modernist disenchantment by abstract axiomatic mathematics; note that analytic philosophy is the disenchantment of traditional philosophy, and there is some reenchantment these days in philosophy as well. The return to the concrete could be compared with the movement of *musique concrète* in contemporary art. Contemporary art puts a strong emphasis on the contingencies of concrete reality, as seen in John Cage's (contingent) chance music (as opposed to Pierre Boulez's controlled chance music), which is conceptually comparable with the contemporary science of chance such as data science and machine learning. The contingency of Being has been a promiment theme in the philosophical tradition that leads from Heidegger and Wittgenstein to Derrida and Rorty. Process philosophy such as Whitehead's is relevant to contemporary reenchanted philosophy (see, e.g., Griffin et al., 1992 and Griffin, 2001); Cassirer's philosophy may count as process philosophy as well. In Asia, the Kyoto School of Philosophy, while facing the modernist crisis of the fragmentation of worldview, propounds the ideal of overcoming modernity by virtue of constructing a unified view of the world; their project would be an early case of reenchantment. They argue that we are facing the fragmentation of worldview in three different directions (that is, Human-, Nature-, and God-centered views), and we have to restore the lost unity of our worldview again in this time.

Infinitary mathematics is arguably the mathematics of ideas, and structuralist mathematics is arguably the mathematics of symbolic forms, which abandons the mirror of nature. Rorty regards the mirror of nature view as the original sin of philosophy:

> One can identify the original sin as the representative theory of perception, as the confusion of noetic being with being as such (Rorty, 1962)

The abandonment of the mirror of nature is a salient feature of modernism in general. According to Jeremy Gray in his *Plato's Ghost*:

> [M]odernism is defined as an autonomous body of ideas, having little or no outward reference, placing considerable emphasis on formal aspects of the work and maintaining a complicated – indeed, anxious – rather than a naive relationship with the day-to-day world. Gray, (2008)

The naïve correspondence between mathematics and the world is abandoned in axiomatic foundations of mathematics. The naïve correspondence between physics and reality is abandoned in quantum theory. There are a lot more cases of the modernist abandonment of the mirror of nature, for example, in modern art as mentioned above. Traditionally, modernisation is understood by Weber's concept of disenchantment, which is underpinned by the modernist propensity for the left in terms of Gödel,

(1995) (see also the above quotation on Gödel's 'right' and 'left'). Meillassoux's speculative materialism employs the logic of justifying the right (realist metaphysics) via the left (materialism). Structural realism and scientific metaphysics, too, employ the same logic, justifying the right (realist metaphysics) via the left (structure and science, respectively). Hilbert's transcendental finitism was an early instance of this logic, justifying the infinitary (right) via the finitary (left); the elimination of ideal elements is exactly the justification of the right (ideal elements) by virtue of the left (finitist methods). Put another way: when 'right' mathematics suffered from Russell's and other paradoxes, the 'left' foundation of mathematics was developed to save the 'right' mathematics from the crisis. Now a fundamental question arises: Why must the right have been justified via the left across the sciences? Realist metaphysics has suffered from the sceptic atmosphere of modernity since the Renaissance, and the naïve conception of reality is not tenable any more, both philosophically and scientifically, as Gödel's quote above illustrates the antirealist tendency of modernist thought in both science and philosophy. The indirect justification of the right via the left is the only way to save the right when the right cannot justify itself on its own. When realist metaphysics loses its direct foundation in face of scepticism, it is the only way to save reality to justify it upon the ground even the sceptic has to accept or presuppose. Note that this again follows the same pattern as Hilbert's transcendental argument.

Modernist (and post-modernist) disenchantment has eventually led us to the nihilistic end of various things: the end of grand narrative (Lyotard); the end of classical realism (Bell); the end of the first philosophy (Quine); the end of meaning (Derrida); the end of history (Fukuyama); and the end of God (Nietzsche). The end of classical realism is almost a fact in quantum physics, mathematically shown by Bell-type theorems as we shall revisit in the following discussion on contextuality. The end of humanity or post-humanity is relevant to the rise of artificial intelligence. Yet the idea itself has an older origin as in Nietzsche's notion of the last man and in Foucault's saying "Man is an invention of recent date." The present concept of human or individual is a social construct of the modern civilisation; the meaning of being human in the pre-modern society was different from that in the modernised society. The modernist conception of the human being begins and ends at some points of history. The problem of modernity may be regarded as the problem of nihilism as in Nietzsche's philosophy. Overcoming modernity, from this perspective, is overcoming nihilism. Along similar lines, Keiji Nishitani, a philosopher of the Kyoto School, propounds the overcoming of modernity via modernity, in particular the overcoming of nihilism via nihilism, via the synthetic integration of Western and Asian philosophies.

Let us finally come back to duality. Everything, from Truth and Meaning to Being and Mind, has dual facets as we have addressed above. Duality in turn unites them together, thereby telling us they are the two sides of the same coin. Duality thus conceived is a constructive canon to deconstruct dualism; this is the fundamental idea that differentiates duality from dualism. Realism and antirealism are grounded upon the ontic and the epistemic, respectively. The ultimate form of duality would then be duality between ontology and epistemology, as witnessed by Benacerraf's duality above. As Benacerraf's duality above suggests, the hard problem of philosophy in general is to give a stream-lined account of how the epistemic and the ontic can interact with each other, which is arguably the very task of duality; without suitable interactions between the ontic (reality) and the epistemic (cognition), there could not be any knowledge about the world whatsoever. Duality is arguably the fundamental basis of knowledge as enabled by interactions between the ontic and the epistemic. At the same time, knowledge about the world only emerges via interactions between the ontic and the epistemic. In light of this, duality is arguably the fundamental basis of knowledge in general.

Some of the above discussion is based upon or adapted from material in (Maruyama, 2019b) (even though there are some conceptual differences), which gives a more Wittgensteinian perspective on (some of) the issues addressed in this article. Note that the following Intensional Turn in logic, geometry and beyond may be regarded as a manifestation of the Informational Turn, which is more focused and thus more nuanced than the broader notion of the Informational Turn.

2.3.2 The Intensional Turn in Logic, Geometry, and Beyond

Domain theory gave compositional semantics of computational processes, and yet it was not fully abstract in order to capture observational equivalence between computational processes. Game semantics however changed the game, and gave fully abstract models of computational processes. Abramsky played a pivotal rôle in developments of game semantics and its application to full abstraction problems, having won another LICS Test-of-Time Award in 2018 for his joint work with Honda and McCusker, (1998). In these developments, he also propounded the concept of full completeness, together with Jagadeesan, (1994). Full completeness gives the intensional conception of completeness for logic, as opposed to the ordinary, extensional conception of it. The traditional, extensional conception of logic and completeness is only concerned with the structure of propositions and their consequence relation (i.e., the poset structure of propositions). The intensional conception of logic and completeness is concerned with the structure of proofs and their identity (i.e., the categorical structure of propositions and proofs; cf. Quine's dictum "no entity without identity").

The Intensional Turn may be elucidated in a broader context of science. The Intensional Turn in the conception of logic is comparable to the Grothendieck turn in the conception of space in geometry, according to which varieties (i.e., the zero loci of polynomials) were replaced by rings in order to capture, e.g., the difference between $x = 0$ and $x^2 = 0$, i.e., nontrivial nilpotent elements as Grothendieck emphasised. The objects of study have changed in algebraic geometry, from denotational to formal entities. Geometry thus made a departure from the notion of point. The same applies to noncommutative geometry, in which noncommutative spaces are supposed to be algebraic structures such as von Neumann algebras with no geometrical points presupposed. Formal topology in constructivism is along the same lines. The point-free conception of space is a consequence of the Intensional Turn; it is a modern analogue of Leibniz's relational space as opposed to Newton's absolute space. Philosophically, it is an instance of the shift from substance to function in Cassirer's terms, or the Gödelian shift from the 'right' to the 'left', which, Gödel argues, happened in quantum physics as well. The Intensional or Functional Turn may thus be observed across the sciences. It could even be observed in cognitive science. Behaviourism in cognitive science regards the mind as an extensional function characterised by inputs and outputs alone. It was eventually replaced by cognitive computationalism, which regards the mind as an intensional function or program. This shift in cognitive science would count as yet another instance of the Intensional Turn.

Abramsky remarks that the shift from set theory to category theory marks a Kuhnian paradigm-shift in mathematics:

> The basic feature of category theory which makes it conceptually fascinating and worthy of philosophical study is that it is not just another mathematical theory, but a way of mathematical thinking, and of doing mathematics, which is genuinely distinctive, and in particular very different to the prevailing set-theoretic style which preceded it. If one wanted a clear-cut example of a paradigm-shift in the Kuhnian sense within mathematics, involving a new way of looking at the mathematical universe, then the shift from the set-theoretic to the categorical perspective provides the most dramatic example we possess. (Abramsky, 2010)

From another angle, category theory may also be seen as marking the Intensional Turn in (foundations of) mathematics, in which the extensional, set-theoretical view was dominating for a while. Wittgenstein is actually an early precursor of those who disagree with the language of set theory:

> Mathematics is ridden through and through with the pernicious idioms of set theory. One example of this is the way people speak of a line as composed of points. A line is a law and isn't composed of anything at all. (Wittgenstein, 1974)

This is reminiscent of point-free topology, and indeed he was influenced by Brouwer, whose theory of continuum is an origin of point-free topology. Abramsky, (2010) presents many interesting arguments. For example, he gives a counterargument to Kunen's set-theoretical doctrine, i.e., "All abstract mathematical concepts are set-theoretic" and "All concrete mathematical objects are specific sets" (Kunen, 2009), saying, "This claim fails to ring true" (Abramsky, 2010), and elaborating on the reasons in detail. He also discusses the normative aspect of category theory. We recommend the reader to refer to (Abramsky, 2010) for his different views of category theory. Note that Wittgenstein, especially in his later philosophy, was inclined towards antiscientism, strongly against the formal conception of logic, as he says, "'Mathematical logic' has completely blinded the thinking of mathematicians and philosophers" (Wittgenstein, 1978). To Wittgenstein, "logic belongs to the natural history of mankind" (Wittgenstein, 1978). He is against formal logic, but not against logic per se.

The themes that Abramsky explicitly philosophises are rare, but they do include information, process, and intensionality (Abramsky, 2008b; 2014). He poses two intriguing puzzles in the beginning of (Abramsky, 2008b): "how can information increase in computation?" and "what is it that we are actually computing in general?". The first puzzle is closely related with what is called Hintikka's scandal of deduction in logic; if deduction is tautological (or analytical), it would not give us any new information. So the first one could be called Abramsky's scandal of computation in informatics. Regarding computation as normalisation via Curry-Howard, he points out that "much (or all?) of the actual usefulness of computation lies in getting rid of the hay-stack, leaving only the needle" (Abramsky, 2010). He also discusses the observer-relativity of information: computing from 3×5 to 15 can be informative to someone; at the same time, computing from 15 to 3×5 (i.e. prime factorisation) can also be informative to another. Computation per se may be considered observer-relative as Searle argues: "Computation exists only relative to some agent or observer who imposes a computational interpretation on some phenomenon" (Searle, 2002). He considers this a better alternative to the Chinese room argument. He says: "The Chinese room argument showed semantics is not intrinsic to syntax"; "But what this argument shows is that syntax is not intrinsic to physics" (Searle, 2002). Note that he identifies syntax with computation. Since even syntax or computation is not intrinsic to the physical machine, there cannot be any semantics or linguistic understanding of meaning intrinsic to it. The observer-relativity of computation implies antirealism about computation; computation is not part of reality, being purely epistemic. The observer-relativity of information would imply antirealism about information; information is not part of reality, being a purely epistemic property. Realism about information includes the Landauer's "Information is Physical" view.

Concerning the second question, he also asks, "What does the Internet compute?". It cannot be an extensional function. The Internet may be seen as a collection of many programs constituting it, each of which computes some extensional function characterised by its inputs and outputs, but as a whole it does not compute any function. What then? Let us have a closer look at his problematisation:

> The traditional conception of computation is that we compute an output as a function of an input, by an algorithmic process. This is the basic setting for the entire field of algorithms and complexity, for example. So *what* we are computing is clear — it is a function. But the reality of modern computing: distributed, global, mobile, interactive, multi-media, embedded, autonomous, virtual, pervasive, ... — forces us to confront the limitations of this viewpoint. [...] In much of contemporary computing [...] the *purpose* of the computing system is to exhibit certain behaviour. (Abramsky, 2010)

The Internet question may be compared with the following questions: what does nature compute?; what does the human being (or mind) compute? According to pancomputationalism, every system is a computational system. The purpose of the human being as a computational system may possibly be to exhibit some behaviour or process. There would be no single function it aims to compute. Likewise, the purpose of nature as a computational system may be to exhibit some behaviour or process. Yet some physicists, such as Seth Lloyd, argue that the universe is a gigantic quantum computer calculating (the time evolution of) its own wave function. Even if the Internet as part of the universe computes its own wave function as part of the whole function, that would not be the purpose of the Internet. The aim of Abramsky, (2010) is to raise the questions and then start the science of informatic processes and their dynamic interactions; he calls this method 'science from philosophy'. What is intriguing in these problems from a philosophical point of view is that the process may not have any type (domain and codomain in category theory). Types are actually ambiguous in the pure mathematician's daily practice as well. Process philosophy (Whitehead, Bergson, Rescher, etc.) is concerned with intensional 'becoming' rather than extensional 'being'; it is quite akin in its spirit to category theory. Yet, once process is formalised, it gets crystalised into a being (cf. time crystalised into the real line, which is no different from space). Mathematics of time and process can be extremely difficult from a philosophical point of view; this may account for the reason why Brouwer regarded mathematics as essentially living in the mind, and any description of it as being something incomplete. The relationships between process philosophy and category theory seem to be strong and deserve more attention and further inquiry; yet at the same time, they would be not so simple as it seems at first sight, as we have argued above. A major difficulty in elucidating the relationships between process philosophy and category theory, in a nutshell, consists in this: processes are dynamic in process philosophy, whilst they are static in category theory; it nevertheless would not be impossible to argue that processes as represented by category theory, rather than processes in category theory, are actually as dynamic as those in process philosophy.

Before moving on to categorical quantum mechanics, let us finally remark that the notion of duality plays a pivotal rôle in Abramsky's work after domain theory, especially in his intensional semantics:

> Who is the System? Who is the Environment? This depends on point of view. We may designate some agent or group of agents as the System currently under consideration, with everything else as the Environment; but it is always possible to contemplate a rôle interchange, in which the Environment becomes the System and vice versa. (This is, of course, one of the great devices, and imaginative functions, of creative literature.) This *symmetry* between System and Environment carries a first clue that there is some structure here; it will lead us to a key *duality*, and a deep connection to logic. (Abramsky, 2010)

From a categorical point of view, this is more like duality within categories (such as dual objects in categories) rather than duality between categories (the former is analogies in categories, and the latter analogies between categories; cf. analogies between analogies as discussed by Banach-Ulam). Duality in categories plays a crucial rôle in categorical quantum mechanics as well, in which states and effects (ket and bra in Dirac's terms) are dual to each other.

2.3.3 CQM, Reconstructionism, and Lightweight Ontology

Around the beginning of the twentieth-first century, Abramsky took a first step towards categorical foundations of physics. Categorical quantum mechanics (CQM for short) was developed by Abramsky and Coecke, (2004), its conceptual underpinning being the Abramsky-Coecke correspondence between logical, computational, and physical processes. It gives the categorical axiomatisation of quantum mechanics and information, and may be seen as giving a solution to the Hilbert's Sixth Problem:

> The investigations on the foundations of geometry suggest the problem: To treat in the same manner, by means of axioms, those physical sciences in which already today mathematics plays an important part (Hilbert, 1902)

Categorical quantum mechanics even gives a graphical axiomatisation of quantum mechanics and information as well as a symbolic axiomatisation in terms of substructural logic and type theory. It tells us that the logic of quantum mechanics is substructural due to the No-Cloning and No-Deleting properties of quantum states and information (note that contraction and weakening are about the cloning and deleting of logical information). From a mathematical point of view, therefore, there is a tight connection between categorical quantum mechanics and Abramsky's preceding work on linear logic (which is a substructural logic), geometry of interaction, and their categorical foundations. Categorical quantum mechanics was also a kind of antithesis to traditional quantum logic by von Neumann and Birkhoff, which was able to account for superposition in single systems, but not for entanglement in composite systems. Categorical quantum mechanics now looks like the process theory of everything, applied beyond physics, notably in natural language semantics and natural language processing (Coecke et al., 2010).

Categorical quantum mechanics may be placed in broader developments of information physics (i.e., informational reconstruction of physics):

- Wheeler (1990): "It from Bit"; "all things physical are information-theoretic in origin" (Weingart, 1989).
- Hardy (2001): "The usual formulation of quantum theory is based on rather obscure axioms"; "It is natural to ask why quantum theory is the way it is" (Hardy, 2001).
- Jozsa (2003): "a viewpoint that attempts to place a notion of information at a primary fundamental level in the formulation of quantum physics"; "In the spirit of Landauer's slogan "Information is physical!" we would declare "Physics is informational!" (Jozsa, 2003).
- Informational approaches have been pervasive since; for example, Chiribella et al. (2011) gives "Informational derivation of quantum theory" (Chiribella et al., 2023).

Categorical quantum mechanics may be seen as a categorical informational reconstruction of quantum mechanics. All this tells us a new approach to conceptual foundations of quantum theory, i.e., recontructionism as opposed to interpretationism. Conceptual problems of quantum theory may be resolved through natural reconstruction. There is no interpretation problem in classical physics and even in relativity theory, since the correspondence between physical reality and theoretical formalism is obvious there. What is wrong with quantum theory may be the Hilbert space formalism; von Neumann actually led to this idea shortly after his publication of *Mathematical Foundations of Quantum Mechanics* as Redei, (1996; 2005) found in von Neumann's letters. It was the dawn of quantum logic, a categorical reconsideration of which finally led to categorical quantum mechanics; von Neumann is a precursor of Abramsky in many respects.

From another angle, categorical quantum mechanics may even allow us to shed new light on ordinary interpretation problems. According to Quine's criterion for ontological commitment, accepting a physical theory makes us committed to the existence of the abstract entities that the theory quantifies over, including mathematical as well as physical ones. According to the Quine-Putnam indispensability argument, we must be ontologically committed to those mathematical entities that are indispensable for our best physical theories. Putnam says, "The positive argument for realism is that it is the only philosophy that doesn't make the success of science a miracle" (Putnam, 1975). This is also called the no miracles argument. These arguments aim at compelling us to take abstract entities in science at face value, and thus to accept platonic realism about those abstract entities. The categorical reconstruction of mathematics and physics arguably allows us to reduce our ontological commitment. It tells us how to do mathematics and physics without presupposing any substance. Put another way, it shows that what is truly indispensable is structure rather than substance. That is to say, substantival realism is reduced, via category theory, to structural realism. Let us call this the principle of ontology-lite, which yields a number of insights into the philosophy of physics and mathematics. It leads us to lightweight metaphysics, which is nonetheless not too light, i.e., not so light as nominalism, the lightest metaphysics. Even if the existence of abstract entities sound too unrealistic and too demanding, the principle of ontology-lite can reduce it to the existence of structures. For example, the many worlds interpretation suffers from the heavy ontology of indefinitely many worlds, which can be reduced to the existence of many structures. The ontology-lite of the many worlds interpretation is the many structures interpretation, which is ontologically less demanding. The ontology-lite of Hilbert space quantum mechanics is categorical quantum mechanics qua process theory, which is only committed to the structure of (symbolic or graphical) processes (symbolic or graphical processes are ontologically weather than substantival processes if any such thing). The ontology of categorical quantum mechanics is arguably lighter than the ontology of Hilbert space quantum mechanics for the following reason: categorical quantum mechanics is finitely (and graphically) axiomatisable in a weak formal system (and even implemented on a computer) whereas Hilbert space quantum mechanics can only be formalised in a stronger system such as the Zermelo set theory (which is weather than ZF but sill quite strong as a foundational system); moreover, the consistency strength of categorical quantum mechanics is presumably weaker than the consistency strength of Hilbert space quantum mechanics, because the foundational system for the former would have a weaker consistency strength than the one for the latter (this appears true because the Zermelo set theory can accommodate the Robinson arithmetic and thus is subject to Gödel's theorems, but categorical quantum mechanics, or Quantomatic as its formalisation, seems too weak to encompass the Robinson arithmetic and is not really subject to Gödel's theorems, on the grounds of what the author discussed with one of the founders of Quantomatic). Categorical quantum mechanics thus allows us to reduce the ontological commitment of quantum mechanics; there is no Hilbert space required in the mathematical ontology of quantum mechanics, but the (symbolic or graphical) ontology of structural processes.

In general, categorical structural realism is not so heavy as platonism (which may be too idealistic), and at the same time, not so light as nominalism (which may be too superficial). Structural ontologies are also epistemologically more certain than platonic ontologies in terms of consistency strength; for example, the structural ontology of topological spaces, such as constructive locale theory or formal topology, is weaker in consistency strength than the platonic, set-theoretical ontology of topological spaces (since constructive type theory is weaker than classical set theory, even though intuitionistic set theory has the same strength as classical set theory). At the same time, categorical structuralism embodies the concept of structure such that relations are prior to relata (or function is prior to substance; cf. Cassirer's genetic view); it has been debated in ontic structural realism whether such a concept of structure is possible or not. It is indeed possible in categorical structural realism.

2.3.4 Contextuality Across the Sciences

Just as von Neumann left the Hilbert space formalism shortly after having published *Mathematical Foundations of Quantum Mechanics*, Abramsky left categorical quantum mechanics shortly after having laid down a foundation for it. It seems to be a general characteristic of his work that he leaves an approach once he has laid a solid foundation for it (and he never sticks to a single approach). His next target was at more mainstream quantum foundations, especially Bell-type theorems therein, which tell us about the essential features of quantum theory, such as nonlocality and contextuality (nonlocality may be seen as a special case of contextuality). The shift may be called the Contextual Turn. He utilised sheaf theory to develop a general theory of nonlocality and contextuality (Abramsky & Brandenburger, 2011), which was applied beyond physics, for example, in linguistics in order to capture contextuality in meaning, and even in database theory (see, for example, Abramsky & Sadrzadeh, 2014, joint work with Mehrnoosh Sadrzadeh, and Abramsky, 2013, respectively). His contextual sheaf semantics of everything, ranging from physical to computational and linguistic systems, builds upon a conceptual understanding of sheaves as expressing the possibility of local information yielding global information when combined together, or in other words, the consistency and inconsistency between local and global information. He has thus characterised contextuality as local consistency plus global inconsistency (Abramsky & Brandenburger, 2011; Abramsky & Hardy, 2012). Contextuality is a central issue in quantum foundations; it is also a central resource in quantum computation and information (the 'paradox as resource' view). Yet the notion of contextuality is very broad and crossdisciplinary beyond physics. The following is a bird's-eye view of contextuality across the sciences:

- Contextuality of truth in epistemology: truth is a function of contexts; a single proposition may be true in one context and at the same time false in another; it may have different truth values in different contexts; epistemic contextualism has an origin in Wittgenstein's later philosophy (Rysiew, 2016).

- Contextuality of being in ontology: agents exist within contexts; being is inseparable from contexts such as environments; this sort of contextualism in ontology has an origin in Heidegger's philosophy and is related to the issues of situated AI, embedded-embodied AI, and Heideggerian AI (Dreyfus, 2007).
- Contextuality of meaning in language: words get meaning within contexts; their meaning may be different in different contexts; the indispensability of contexts in the meaning determination process may lead to some weak form of Quinean semantic holism (no meaning without some wider context).
- Contextuality of reality in quantum physics: measurement values or their statistics exist within contexts; there may be no global assignment of values and probabilities; this sort of contextuality in physics has an origin in Bell's and Kochen-Specker's No-Go theorems refuting classical (non-local) realism.
- Contextuality of reason in cognitive science: cognitive behavior is a function of contexts; a single question may have different answers in different contexts; contextual effects such as coexisting information and environmental noise in the real world may affect and change results of cognitive decision making.

The contextuality-by-default theory by Dzhafarov et al., (2016) allows us to link physical contextuality with congnitive contextuality, and it is experimentally verified that Bell-type inequalities are indeed violated in certain cognitive experiments by Cervantes et al., (2023). There is however a significant difference between cognitive and physical contextuality (or what they mean). To see this, let us think of the experimental Laplace's demon. If the Laplace's demon fixes all hidden parameters involved in cognitive experiments, cognitive contextuality disappears (under the classical brain assumption, which is verified by Tegmark, 2000). Yet even the demon cannot erase physical contextuality by fixing parameters involved. Cognitive contextuality, therefore, is an extrinsic property of the mind in its collective state dynamics, whereas physical contextuality is an intrinsic property of reality in its single state dynamics.

Compositionality is a central idea in category theory, in logic and semantics, and in philosophy of language; there is even a journal, *Compositionality*, dedicated to it. It is generally said that the origins of compositionality and contextuality in semantics are in Frege's philosophy, and Frege endorsed both the principle of compositionality and the principle of contextuality:

- The principle of compositionality: the meaning of a whole (expression) is a function of, and completely determined by, the meaning of its parts (and the syntactical way they are combined together). This is basically atomism; the meaning of atomic expressions recursively generate the meaning of more complex expressions. Compositionality is considered to be a source of the productivity, systematicity, and learnability of language; thanks to compositionality, we can systematically create new expressions.
- The principle of contextuality: the meaning of a word (or more complex expression) is a function of, and can only be determined within, contexts; the meaning of parts depends upon larger wholes surrounding them. This is basically a version of holism about meaning. In atomism, parts are prior to wholes, which are sec-

ondary; in holism, parts only exist as parts of wholes, which are primary. Some historical analysis (Janssen, 2001) tells that Frege only endorsed the principle of contextuality alone, and not the principle of compositionality.

The principle of contextuality may be seen as being in conflict with the principle of compositionality (if so Frege cannot endorse both). If meaning is compositional, the meaning of a whole must be determined with reference to the meaning of its parts only, i.e., without reference to anything larger, such as contexts. In contrast, contextuality is a holistic principle. Holism says that a whole cannot be reduced to the mere combination of its parts, whereas the central tenet of compositionality is that this is indeed possible, i.e., the meaning of a whole is composed of that of its parts. Burge, (2005) is one of the earliest commentators who was clearly aware of the tension in the Fregean philosophy of language:

> It is worth noting that Frege's reasoning here is prima facie *incompatible* with the idea that the notion of the denotation of a term has no other content than that provided by an analysis of the contribution of the term in fixing the denotation (or truth value) of a sentence. The argument presupposes [...] that the notion of term-denotation is more familiar than that of sentence denotation [...]

Dummett, (1981) also says as follows:

> It was meant to epitomize the way I hoped to reconcile that principle, taken as one relating to sense, with the thesis that the sense of a sentence is built up out of the senses of the words. This is a difficulty which faces most readers of Frege [...] The thesis that a thought is compounded out of parts comes into apparent conflict, not only with the context principle, but also with the priority thesis [...]

Note that Dummett here regards the compositionality of thought as deriving from that of sense (cf. the language of thought hypothesis). The essence of the tension between compositionality and contextuality may be understood in the following manner. The point is whether wholes have to refer to parts or parts have to refer to wholes in oder to determine meaning. Suppose that the two principles are both indispensable in meaning determination. Then, in order to determine meaning, wholes refer to parts, and parts refer to wholes (and mutual reference continues ad infinitum). There is a vicious circle here, and this is essentially an analogue of what is called hermeneutic circularity in the continental tradition of philosophy. We could speculate that, in the analytic tradition, the two principles came to be understood separately in order to keep the theory of meaning immune to vicious circles, and yet in the continental tradition, both of them were taken at face value at once, thus leading to the idea of hermeneutic circularity.

Philosophically, contextuality is in conflict with compositionality. Yet it has often happened in history that philosophy is damaged by science, just as Kant's philosophy was impeded by later developments of mathematical science (even though Neo-Kantians had some counterarguments and mathematicians such as Hilbert, Brouwer, and actually Gödel, nevertheless relied upon Kantian philosophy in order to support and develop their foundations of mathematics Corry, 2004). Abramsky's work elucidates contextuality via the methods of compositional semantics across logic, computer science, and category theory, and thus it may be seen as resolving the conflict between compositionality and contextuality. Note that some of the discussion above is based upon or adapted from material in Maruyama (2019a), Maruyama (2019b), Maruyama (2019b), which elaborate upon contextuality studies from different angles in a much more detailed manner.

2.4 Concluding Remarks: Vienna, Stanford, and Oxford

The twenty-first century has seen a remarkable shift in the conception of category theory; category theory as foundations of mathematics, as we have argued above, has gradually grown into category theory as foundations of science in general. Samson Abramsky has played a prominent rôle in this fundamental transformation of intellectual scenes surrounding category theory. Science now looks very different; it is significantly more interconnected and united in the after-Abramsky intellectual landscape than it was before. In light of this, we could even speculate that singularity is possibly really near, not in the technological sense, but in an epistemological sense; all knowledge will be interconnected and united as a coherent whole in the epistemological singularity (hopefully before the technological singularity comes, which otherwise may cause intelligence explosion and so possibly trigger the epistemological singularity as well, presumably in a way very different from the way the human mind grasps the world). This would sound too speculative; however it would be fair to say, at least, that category theory can contribute not only to the unity of mathematics but also to the unity of science. Even if the unity of science is an unfeasible ideal after all, categorical science as has been developed by Abramsky, or the very way he does science via his unifying categorical methodology, to say the least, gives us substantial hope for approximating the unity of science in the ideal limit of our scientific endeavour.

In the present article we have shed light on various philosophical contexts in which Abramsky's work may be placed and understood. Amongst other things, it may be construed as showing the categorical unity of science, which is in stark contrast with the old-fashioned unity of science in the Vienna Circle, and yet arguably serves to reunite 'the now severed sciences' (Meyer, 1864; Nye, 1993) in a way coherent with scientific pluralism. Even if there is no global Truth with capital T but only local truths, there can be the global network of local truths that keeps their diversity and plurality, and is nevertheless interlaced as a coherent whole. In particular, it is a decentred network; there is no such thing as the monistic centre of science or knowledge that grounds the rest of it at once (cf. the decline of Cartesian foundationalism in both analytic and continental philosophy as exemplified by Rorty's pluralist thesis that there is no Archimedean vantage point). The decentred network and unity of knowledge are supported by the decentred network and unity of people as well. Indeed, Abramsky and his work have been surrounded by many fellow researchers and their work, and his intellectual circle is as diverse as his intelligence per se. In light of this, the circle centring around Oxford may be called the Oxford

School of Categorical Unified Science, which is meant to include not only geographically Oxford-connected scholars, but also conceptually Oxford-style scholars. At the moment there may be no such school in broadly recognised form, but, decades later, historians and philosophers of science may retrospectively come to think that there was a new unity-of-science movement around Oxford.

Although we have traced Abramskty's intellectual history, leading from duality, to intensionality, and to contextuality, that is, nevertheless, not the whole story. There are many on-going projects of his own in the first place. In his recent work, for example, he attempts to bridge between 'structure' and 'power', that is, 'semantics and compositionality' on the one hand, and 'expressiveness and complexity' on the other (incidentally, *Structure and Power* is the title of a best-selling 1983 book on contemporary continental philosophy by Akira Asada, marking the birth of what is called 'new academism' in Japan, the transdisciplinary movement that gave a significant impact on the society as a whole, as opposed to 'new criticism' in the US and the UK in the mid-twentieth century). Structure and Power, in Abramsky's sense, reflect "two very different views of what the fundamental features of computation are: one focussing on structure and compositionality, the other on expressiveness and efficiency" (Abramsky, 2020). There are many other works by Abramsky we were unable to touch upon here; they are yet to be explored elsewhere in future work. We were also unable to follow his award-winning history in detail, but it includes the Lovelace Medal (2013) and the Alonzo Church Award (2017) as well as the two LiCS Test-of-Time awards (2007 and 2018) mentioned above. Yet we believe that his achievements are far more than mere awards, medals, and the like, impacting the very way we think about the fundamental elements of the world; its far-reaching consequences are yet to be explicated and articulated further.

Let us finally remark that we could learn as much from his way of living as from his scholarly achievements. He, for example, tends to work on one theme for several years, then suddenly leave it, and move on to another. Perhaps it is the only way to make depth coexistent with diversity in one's scholarly work. It may be risky, and possibly scary, for ordinary researchers to change their research topics so many times and cover many different fields within their single lifetimes, which may appear to be not so long as to do so successfully, so that they may hesitate to work in such a way. Yet there seems to be no such fear in his academic life; in terms of Isaiah Berlin's well-known opposition between the hedgehog and the fox (Berman, 1981), he is such a person as remarkably makes the characteristics of the former compatible with those of the latter. There are also many inspiring stories about his personality and his way of thinking. He once said, at a party celebrating his sixteenth birthday at Wolfson College, Oxford (which was founded by Isaiah Berlin, who is also known for his realistic pluralism), that the idea of a genius was wrong, and would be an obstacle to liberal criticism (this is not an empty saying, but he does practise the liberty and always accepts any criticism whatsoever), when we talked about Wittgenstein as a typical genius, who was conceived of as an untouchable genius during his undergraduate days as a philosophy student in Cambridge. So we presumably should not call him a genius, even if we call him a one-man-embodiment of the categorical unity of science.

Acknowledgements I would like to express my deepest gratitude to Samson Abramsky for his robust continual support and encouragements, without which my life might have been different from what it is currently. At the same time, I would like to dedicate this article to the memory of our wonderful friend and colleague Klaus Keimel, who is one of the founders of domain theory as a theory in pure mathematics rather than computer science, and according to what Bob Coecke told me, wrote a strong letter of recommendation for me, thus having allowed me to do a PhD with Samson and Bob in Oxford, without which this article would have never come into existence in the first place. I can however tell for sure that Klaus would nevertheless dislike this article, especially in light of his nature as a hardcore mathematician, and yet despite many such disagreements, we were actually good friends of each other. The same would apply to Samson, who would disagree with some aspects of this work, and we are nonetheless good friends of each other. As you may have noticed already, I am going to argue that these are indeed cases of the unity (of humans rather than science) that is coherent with plurality and diversity, which has been the subject matter of the present article. I would like to note that this work largely stemmed from the Symposium on the Categorical Unity of the Sciences (Kyoto, 22–23 March 2019), which I held in the precious occasion of Samson's visit in Kyoto, where I worked then as Hakubi Assistant Professor (since my DPhil study in Oxford), but soon after which I have got an opportunity to move to Australia as a Lecturer at ANU with the great help of Samson, Bob, and Jamie Vicary, to all of whom I am truly grateful for their gentle assistance. I hereby acknowledge that the present work was financially supported by JST PRESTO (grant code: JPMJPR17G9) and JSPS KAKENHI (grant code: 17K14231).

References

Abramsky, S. (1987). *Domain theory and the logic of observable properties*, Ph.D. thesis, London: Queen Mary University of London.

Abramsky, S. (1987). Domain theory in logical form. In *Proceedings of the 2nd annual IEEE symposium on logic in computer science* (pp. 47–53).

Abramsky, S. (2008). Information, processes and games. In *Handbook of the philosophy of science* (vol. 8, pp. 483–549). Philosophy of Information.

Abramsky, S. (2008). Temperley-Lieb algebra: From knot theory to logic and computation via quantum mechanics. In *Mathematics of quantum computing and quantum technology* (pp. 415–458). Oxon: Taylor & Francis.

Abramsky, S. (2015). Arrow's theorem by arrow theory. In *Logic without borders: Essays on set theory, model theory, philosophical logic and philosophy of mathematics* (pp. 15–30). Berlin: de Gruyter.

Abramsky, S., & Brandenburger, A. (2011). The sheaf-theoretic structure of non-locality and contextuality. *New Journal of Physics,13*, 113036.

Abramsky, S., & Coecke, B. (2004). A categorical semantics of quantum protocols. In *Proceedings of the 19th annual IEEE symposium on logic in computer science* (pp. 415–425).

Abramsky, S., & Hardy, L. (2012). Logical Bell inequalities. *Physical Review A,85*, 062114.

Abramsky, S., Gottlob, G., & Kolaitis, P. (2013). Robust constraint satisfaction and local hidden variables in quantum mechanics. In *Proceedings of the twenty-third international joint conference on artificial intelligence* (pp. 440–446).

Abramsky, S., Honda, K., & McCusker, G. (1998). A fully abstract game semantics for general references. In *Proceedings of the 13th annual IEEE symposium on logic in computer science* (pp. 334–344).

Abramsky, S. (2010). Logic and categories as tools for building theories. *Journal of Indian Council of Philosophical Research, 27,* 277–304.

Abramsky, S. (2013). Relational databases and Bell's theorem. *Lecture Notes in Computer Science, 8000,* 13–35.

Abramsky, S. (2014). Intensionality. *Definability and Computation, Outstanding Contributions to Logic, 5,* 121–142.
Abramsky, S. (2020). Whither semantics? *Theoretical Computer Science, 807,* 3–14.
Abramsky, S., & Jagadeesan, R. (1994). Games and full completeness for multiplicative linear logic. *The Journal of Symbolic Logic, 59,* 543–574.
Abramsky, S., & Sadrzadeh, M. (2014). Semantic unification: A sheaf theoretic approach to natural language. *Lecture Notes in Computer Science, 8222,* 1–13.
Abramsky, S., & Shah, N. (2018). Relating structure and power: Comonadic semantics for computational resources. *Lecture Notes in Computer Science, 11202,* 1–5.
Abramsky, S., & Winschel, V. (2017). Coalgebraic analysis of subgame-perfect equilibria in infinite games without discounting. *Mathematical Structures in Computer Science, 27,* 751–761.
Berlin, I. (1953). *The hedgehog and the fox: An essay on tolstoy's view of history*. London: Weidenfeld & Nicolson.
Berman, M. (1981). *The reenchantment of the world*. Ithaca: Cornell University Press.
Bourbaki, N. (1950). The architecture of mathematics. *The American Mathematical Monthly, 4,* 221–232.
Burge, T. (2005). *Truth, thought, reason: Essays on frege*. Oxford: Clarendon Press.
Car, E. H. (1961). *What is history?* Cambridge: Cambridge University Press.
Cassirer, E. (1923). *Substance and function, and Einstein's theory of relativity*. Chicago: Open Court.
Cassirer, E. (1955). *The philosophy of symbolic forms*. New Haven: Yale University Press.
Cassirer, E. (1945). Structuralism in modern linguistics. *Word, 1,* 99–120.
Cervantes, V., & Dzhafarov, E. (2023) Snow queen is evil and beautiful: Experimental evidence for probabilistic contextuality in human choices. In *Decision* (vol. 5, pp. 193–204).
Chalmers, D. (1996). *The conscious mind: In search of a fundamental theory*. Oxford: Oxford University Press.
Chiribella, G., D'Ariano, G. M., & Perinotti, P. (2023) Informational derivation of quantum theory. *Physical Review A,84,* 012311.
Coecke, B. (2007). Automated quantum reasoning: non-logic - semi-logic - hyper-logic. In *Proceedings of AAAI spring symposium: quantum interaction* (pp. 31–38).
Coecke, B., Sadrzadeh, M., & Clark, S. (2010). Mathematical foundations for a compositional distributional model of meaning. *Linguistic Analysis, 36,* 345–384.
Coquand, T. (2009). Space of valuations. *Annals of Pure and Applied Logic, 157,* 97–109.
Corry, L. (2004). *David Hilbert and the axiomatization of physics (1898–1918)*. Netherlands: Springer.
Daston, L. (1999). Die akademien und die einheit der wissenschaften: die disziplinierung der disziplinen. In *Die königlich preußische akademie der wissenschaften zu berlin im kaiserreich* (pp. 61–84). Berlin: Akademie Verlag.
Daston, L., & Galison, P. (2007). *Objectivity*. NewYork: Zone Books.
Dilthey, W. (1991). *Wilhelm Dilthey: selected works volume I: Introduction to the human sciences*. Princeton: Princeton University Press.
Dreyfus, H. (2007). Why Heideggerian AI failed and how fixing it would require making it more Heideggerian. *Philosophical Psychology, 20,* 247–268.
Dubois-Reymond, E. (1886). Über die Wissenschaftlichen zustände der Gegenwart. In *Reden* (vol. 2). Leipzig: Veit & Co.
Dummett, M. (1981). *The interpretation of Frege's philosophy*. London: Duckworth.
Dzhafarov, E., et al. (2016). On contextuality in behavioural data. *Philosophical Transactions A: Mathematical, Physical and Engineering Sciences, 374,* 20150234.
Friedman, M. (2000). *A parting of the ways: Carnap, Cassirer, and Heidegger*. Chicago: Open Court Press.
Galison, P. (1997). *Image and logic: A material culture of microphysics*. Chicago: University of Chicago Press.

Galison, P. (1999). Trading zone: Coordinating action and belief. In *Science studies reader* (pp. 137–160). NewYork: Routledge.

Gödel, K. (1995). *Collected works* (Vol. III). New York: Oxford University Press.

Gray, J. (2008). *Plato's ghost: The modernist transformation of mathematics*. Princeton: Princeton University Press.

Griffin, D. (2001). *Reenchantment without supernaturalism*. Ithaca: Cornell University Press.

Griffin, D. et al. (1992). *Founders of constructive postmodern philosophy: Peirce, James, Bergson, Whitehead, and Hartshorne*. New York: SUNY Press.

Hardy, L. (2001). *Quantum theory from five reasonable axioms*. arXiv:quant-ph/0101012v4.

Hilbert, D. (1983). On the infinite. In: *Philosophy of mathematics: Selected readings*. Cambridge: Cambridge University Press.

Hilbert, D. (1902). Mathematical problems. *Bulletin of the American Mathematical Society, 8,* 437–479.

Janssen, T. (2001). Frege, contextuality and compositionality. *Journal of Logic, Language, and Information, 10,* 87–114.

Jozsa, R. (2003). *Illustrating the concept of quantum information*. arXiv:quant-ph/0305114.

Kant, I. (2007). *Critique of judgement*. Oxford: Oxford University Press.

Kripke, S. (1982). *Wittgenstein on rules and private language*. Massachusetts: Harvard University Press.

Kunen, K. (2009). *Foundations of mathematics*. London: College Publications.

Lofts, S. (2000). *Cassirer: A "Repetition" of modernity*. Albany: SUNY Press.

Maruyama, Y. (2013). *From operational chu duality to coalgebraic quantum symmetry* (Vol. 8089, pp. 220–235). Springer LNCS.

Maruyama, Y. (2013). *Full Lambek hyperdoctrine: Categorical semantics for first-order substructural logics* (Vol. 8071, pp. 211–225). *Springer LNCS*.

Maruyama, Y. (2017). *Meaning and duality: From categorical logic to quantum physics*, D.Phil. Thesis, Department of Computer Science, Mathematical, Physical, and Life Sciences Division, University of Oxford.

Maruyama, Y. (2018). The frame problem, Gödelian incompleteness, and the Lucas-Penrose argument. In *Springer SAPERE* (Vol. 44, pp. 194–206).

Maruyama, Y. (2019). *Compositionality and contextuality: The symbolic and statistical theories of meaning* (Vol. 11939, pp. 161–174). Springer LNCS.

Maruyama, Y. (2019). *Contextuality across the Sciences: Bell-type Theorems in physics and cognitive science* (Vol. 11939, pp. 147–160) Springer LNCS.

Maruyama, Y. (2019). Foundations of mathematics: From Hilbert and Wittgenstein to the categorical unity of science. In *Wittgensteinian* (pp. 245–274). Springer Frontiers Collection.

Maruyama, Y. (2019). Quantum physics and cognitive science from a Wittgensteinian perspective. In *Wittgensteinian* (pp. 375–407). Springer Frontiers Collection.

Maruyama, Y. (2020). *Higher-order categorical substructural logic: Expanding the horizon of tripos theory* (Vol. 12062, pp. 187–203). Springer LNCS.

Maruyama, Y. (2010). Fundamental results for pointfree convex geometry. *Annals of Pure and Applied Logic, 161,* 1486–1501.

Maruyama, Y. (2012). Natural duality, modality, and coalgebra. *Journal of Pure and Applied Algebra, 216,* 565–580.

Maruyama, Y. (2013). Categorical duality theory: With applications to domains. *Convexity, and the Distribution Monad, Leibniz International Proceedings in Informatics, 23,* 500–520.

Maruyama, Y. (2016). Prior's Tonk, notions of logic, and levels of inconsistency: Vindicating the pluralistic unity of science in the light of categorical logical positivism. *Synthese, 193,* 3483–3495.

Maruyama, Y. (2016). Categorical harmony and paradoxes in proof-theoretic semantics. *Advances in Proof-Theoretic Semantics, Trends in Logic, 43,* 95–114.

Maruyama, Y. (2017). The dynamics of duality: A fresh look at the philosophy of duality. *RIMS Kokyuroku (Proceedings of RIMS, Kyoto Univesity), 2050,* 77–99.

Maruyama, Y. (2020). Topological duality via maximal spectrum functor. *Communications in Algebra, 48,* 2616–2623.
Meyer, J. L. (1864). Die modernen Theorien der Chemie.
Nishitani, K. (1990). *The self-overcoming of nihilism*. New York: State Univ. New York Press.
Nye, M. J. (1993). *From chemical philosophy to theoretical chemistry: dynamics of matter and dynamics of disciplines 1800–1950*. Berkeley: University of California Press.
Parsons, C. (2014). *Philosophy of mathematics in the twentieth century*. Harvard University Press.
Peirce, C. S. (1974). *Collected papers of Charles Sanders Peirce*. Massachusetts: Harvard University Press.
Putnam, H. (1975). *Mathematics, matter and method*. Cambridge: Cambridge University Press.
Putnam, H. (1983). *Realism and reason, philosophical papers* (Vol. 3). Cambridge: Cambridge University Press.
Rédei, M. (ed.) (2005). *John von Neumann: Selected letters*. American Mathematical Society/London Mathematical Society.
Rédei, M. (1996). Why John von Neumann did not Like the Hilbert space formalism of quantum mechanics (and what he liked instead). *Studies in History and Philosophy of Modern Physics, 27,* 493–510.
Rorty, R. (1970). *Philosophy and the mirror of nature*. Princeton: Princeton University Press.
Rorty, R. (Ed.). (1967). *The Linguistic turn: Recent essays in philosophical method*. Chicago: The University of Chicago Press.
Rorty, R. (1962). Realism, categories, and the "Linguistic Turn.". *International Philosophical Quarterly, 2,* 307–322.
Rysiew, P. (2016). Epistemic contextualism. *Stanford Encyclopedia of Philosophy*. Stanford: Stanford University.
Searle, J. (2002). The problem of consciousness, *Consciousness and language* (pp. 7–17). Cambridge: Cambridge University Press.
Tegmark, M. (2000). Importance of quantum decoherence in brain processes. *Physical Review E, 61,* 4194.
Voevodsky, V. et al. (2013). *Homotopy type theory: Univalent foundations of mathematics*. IAS: Princeton.
von Hemholtz, H. (1896). *Vorträge und Reden* (Vol. 1, 4th edn.). Braunschweig: Vieweg und Sohn.
Weingart, P. (2010). A short history of knowledge formations. In: *The Oxford handbook of interdisciplinarity* (pp. 3–14). Oxford: Oxford University Press.
Wheeler, J. (1989). Information, physics, quantum: the search for links. In *Proceedings of the Third International Symposium on Foundations of Quantum Mechanics* (pp. 354–358).
Wittgenstein, L. (1970). *Zettel*. California: University of California Press.
Wittgenstein, L. (1974). *Philosophical grammar*. Oxford: Basil Blackwell.
Wittgenstein, L. (1978). *Remarks on the foundations of mathematics*. Oxford: Basil Blackwell.

Chapter 3
Minimisation in Logical Form

Nick Bezhanishvili, Marcello M. Bonsangue, Helle Hvid Hansen, Dexter Kozen, Clemens Kupke, Prakash Panangaden, and Alexandra Silva

Abstract Recently, two apparently quite different duality-based approaches to automata minimisation have appeared. One is based on ideas that originated from the controllability-observability duality from systems theory, and the other is based on ideas derived from Stone-type dualities specifically linking coalgebras with algebraic structures derived from modal logics. In the present paper, we develop a more abstract view and unify the two approaches. We show that dualities, or more generally dual adjunctions, between categories can be lifted to dual adjunctions between categories of coalgebras and algebras, and from there to automata with initial as well as final states. As in the Stone-duality approach, algebras are essentially logics for reasoning about the automata. By exploiting the ability to pass between these categories, we show that one can minimize the corresponding automata. We give an abstract minimisation algorithm that has several instances, including the celebrated

N. Bezhanishvili
University of Amsterdam, Amsterdam, The Netherlands
e-mail: N.Bezhanishvili@uva.nl

M. M. Bonsangue
Leiden University, Leiden, The Netherlands
e-mail: m.m.bonsangue@liacs.leidenuniv.nl

H. H. Hansen (✉)
University of Groningen, Groningen, The Netherlands
e-mail: h.h.hansen@rug.nl

D. Kozen · A. Silva
Cornell University, Ithaca, NY, USA
e-mail: kozen@cornell.edu

A. Silva
e-mail: alexandra.silva@cornell.edu

C. Kupke
University of Strathclyde, Glasgow, Scotland
e-mail: clemens.kupke@strath.ac.uk

P. Panangaden
McGill University, Quebec, Canada
e-mail: prakash@cs.mcgill.ca

A. Palmigiano and M. Sadrzadeh (eds.), *Samson Abramsky on Logic and Structure in Computer Science and Beyond*, Outstanding Contributions to Logic 25,
https://doi.org/10.1007/978-3-031-24117-8_3

Brzozowski minimisation algorithm. We further develop three examples that have been treated in previous works: deterministic Kripke frames based on a Stone-type duality, weighted automata based on the self-duality of semimodules, and topological automata based on Gelfand duality. As a new example, we develop alternating automata based on the discrete duality between sets and complete atomic Boolean algebras.

Keywords Automata · Minimisation · Duality · Coalgebra · Algebra · Modal logic

3.1 Introduction

Category theory, algebra, and logic are deepening our understanding of program semantics and algorithms, and Samson has been a pioneer and leader in developing this field. His seminal paper *Domain Theory in Logical Form* Abramsky (1991) studies the connection between program logic and domain theory via Stone duality. This is an example of a fundamental duality in Computer Science between (operational or denotational) semantics and program logic which can be viewed as following the algebraic structure of the syntax.

Building on Stone's celebrated representation theorems for Boolean algebras (Stone, 1936) and distributive lattices (Stone, 1937), categorical dualities linking algebra and topology (Johnstone, 1982) have been widely used in logic and theoretical computer science (Chagrov & Zakharyaschev, 1997; Blackburn et al., 2001; Gierz et al., 2003). With algebras corresponding to the syntactic, deductive side of logical systems, and topological spaces to their semantics, Stone-type dualities provide a powerful mathematical framework for studying various properties of logical systems. More recently, it has also been fruitfully explored in more algorithmic applications: notably in understanding minimisation of various types of automata (Adámek et al., 2012; Bezhanishvili et al., 2012; Bonchi et al., 2012, 2014; Klin & Rot, 2016; Rot, 2016; Ganty et al., 2019). Among these, Bezhanishvili et al. (2012) and Bonchi et al. (2014) were published around the same time, and they had some similarities, but also some key differences. It was not clear whether the differences could be reconciled in a principled way. The main aim of this paper is to find a unifying perspective on the minimisation constructions in Bezhanishvili et al. (2012) and Bonchi et al. (2014) which we briefly recall here.

In Bezhanishvili et al. (2012), (generalised) Moore automata (without initial state) are modelled as coalgebras on base categories of algebras or topological spaces. The main observation used in Bezhanishvili et al. (2012) is that for many types of such coalgebras, one can define a category of algebras that is dually equivalent to the category of coalgebras. This dual equivalence generalises the Jónsson-Tarski duality known from modal logic, which in turn arises from Stone duality. The algebras in Bezhanishvili et al. (2012) are therefore understood as modal algebras, i.e., they consist of an algebra (that describes a propositional logic, e.g., Boolean logic) expanded with the modal operators. From this coalgebra-algebra duality it follows that maximal quotients of coalgebras correspond to minimal subobjects of algebras. Moreover, it is shown that for a given coalgebra, the minimal subalgebra of its dual modal algebra consists of the algebra of definable subsets. A maximal quotient can therefore be constructed by computing definable subsets and dualising. The minimisation-via-duality approach of Bezhanishvili et al. (2012) was shown to apply to partially observable DFAs (using duality of finite sets and finite Boolean algebras), linear weighted automata (using the self-duality of vector spaces), and belief automata viewed as coalgebras on compact Hausdorff spaces (using Gelfand duality). Moreover, for each of these examples it is shown that the definable subsets are determined by the subsets definable in the trace logic fragment consisting of formulas of the shape $[a_0] \cdots [a_n]p$.

In Bonchi et al. (2014), Brzozowski's double-reversal minimisation algorithm (Brzozowski, 1962) for deterministic finite automata (with both initial and final states) was described categorically and its correctness explained via the duality between reachability and observability known from control theory (cf. Kalman, 1959, Arbib & Zeiger, 1969, Arbib & Manes, 1974). This duality arises from a dual adjunction between algebras and coalgebras, not a full duality, but this is sufficient to formalise Brzozowski's algorithm in terms of a dual adjunction between categories of automata. This categorical formulation was then used to formulate Brzozowski-style minimisation algorithms for Moore automata (over Set) and weighted automata over commutative semirings, which include nondeterministic and linear weighted automata as instances. To be more precise, a weighted automaton is first determinised into a generalised Moore automaton with a semimodule statespace to which the double-reversal algorithm is applied, yielding in the end a minimal Moore automaton.

The perspective taken in Bonchi et al. (2014) is language-based; no link is made to modal logic. Conversely, the perspective taken in Bezhanishvili et al. (2012) is logic-based; no link is made to reachability, and language acceptance is only implicitly present via trace logic. Duality will play a central role in our unification of these approaches. Our work is very much inspired by Samson's perspective and we hope that he will regard it as being in the spirit of his own approach to formalising theories.

The contributions of the present paper are as follows.

1. A categorical framework within which minimisation algorithms can be understood and different approaches unified (Sect. 3.3). We start by illustrating the difference in the approaches from Bezhanishvili et al. (2012) and Bonchi et al. (2014) on classic deterministic automata (Sect. 3.3.1), and then proceed to a general setup for different automata types based on algebra and coalgebra (Sect. 3.3.2). Section 3.3.3 includes the categorical picture that unifies the work in Bezhanishvili et al. (2012) and Bonchi et al. (2014): in a nutshell, it is a stack of three interconnected adjunctions. It starts with a base dual adjunction that is subsequently lifted to a dual adjunction between coalgebras and algebras, and finally to a dual adjunction between automata. Section 3.3.4 extends this categorical picture to include trace logic. Section 3.3.5 presents an abstract understanding of

reachability and observability, and finally everything is summarised and abstract minimisation algorithms are stated in Sect. 3.3.6.

2. A thorough illustration of the general framework instantiated to concrete examples. In Sect. 3.4, we revisit a range of examples stemming from previous approaches: deterministic Kripke frames, weighted automata, and topological automata (belief automata). In Sect. 3.5, we include an extensive new example on alternating automata, which uses the duality of complete atomic Boolean algebras and sets. For weighted automata, we use our framework to extend a well-known result for weighted automata over a field (Schützenberger, 1961) to weighted automata over a principal ideal domain: the minimal weighted automaton over a principal ideal domain always exists, and, as expected, it has a state space smaller or equal than that of the original automaton.

We conclude the paper with a review of related work (Sect. 3.6).

3.2 Preliminaries

In this section, we fix notation and recall basic definitions of coalgebras and algebras. For a more detailed introduction to coalgebra, we refer to Rutten (2000). For general categorical notions, see e.g. Adámek et al. (2009).

Categories are denoted by $\mathcal{C}, \mathcal{D}, \ldots$, objects of categories by $X, Y, Z, \ldots$, and arrows or morphisms of categories by $f, g, h, \ldots$. We denote by Set the category of sets and functions. Let X_1, X_2 be in $\mathcal{C}$. The product of X_1 and X_2 (if it exists) is denoted by $X_1 \times X_2$ with projection maps $\pi_i \colon X_1 \times X_2 \to X_i$, $i = 1, 2$. Similarly, their coproduct (if it exists) is written $X_1 + X_2$ with coprojection (inclusion) maps $\mathsf{in}_i \colon X_i \to X_1 + X_2$. In Set, $X_1 \times X_2$ and $X_1 + X_2$ are the usual constructions of cartesian product and disjoint union. Let X be an object in $\mathcal{C}$ and A be a set. Assuming $\mathcal{C}$ has products, then $X^A := \prod_A X$ denotes the A-fold product of X with itself. Similarly, if $\mathcal{C}$ has coproducts, then $A \cdot X := \coprod_A X$ denotes the A-fold coproduct of X with itself.

The covariant powerset functor $\mathcal{P}\colon \mathsf{Set} \to \mathsf{Set}$ sends a set X to its powerset $\mathcal{P}(X)$ and a function $f\colon X \to Y$ to the direct-image map $\mathcal{P}(f)\colon \mathcal{P}(X) \to \mathcal{P}(Y)$. The contravariant powerset functor $\mathcal{Q}\colon \mathsf{Set} \to \mathsf{Set}^{\mathsf{op}}$ also sends a set X to its powerset, now denoted $\mathcal{Q}(X) = 2^X$, and a function $f\colon X \to Y$ to its inverse-image map $\mathcal{Q}(f)\colon \mathcal{Q}(Y) \to \mathcal{Q}(X)$.

3.2.1 Coalgebras, Algebras and Monads

Given an endofunctor $F\colon \mathcal{C} \to \mathcal{C}$, an *$F$-coalgebra* is a pair $(X, \gamma\colon X \to FX)$, where X is a $\mathcal{C}$-object and $\gamma\colon X \to FX$ is a $\mathcal{C}$-arrow. The functor F specifies the type of the coalgebra (which may be thought of as the type of observations and transitions), and the structure map γ specifies the dynamics. An *F-coalgebra morphism* from an F-coalgebra (X, γ) to an F-coalgebra (Y, δ) is a $\mathcal{C}$-arrow $h\colon X \to Y$ that preserves the coalgebra structure, *i.e.*, $\delta \circ h = Fh \circ \gamma$. F-coalgebras and F-coalgebra morphisms form a category denoted by $\mathsf{Coalg}_{\mathcal{C}}(F)$. A *final F-coalgebra* is a final object in $\mathsf{Coalg}_{\mathcal{C}}(F)$, i.e., an F-coalgebra (Z, ζ) is final if for all T-coalgebras (X, γ) there is a unique F-coalgebra morphism $h\colon (X, c) \to (Z, \zeta)$.

An *F-algebra* is a pair (X, α), where X is a $\mathcal{C}$-object and $\alpha\colon FX \to X$ is a $\mathcal{C}$-arrow. Now, the functor F can be seen to specify the type of operations of the algebra. An *F-algebra morphism* from an F-algebra (X, α) to an F-algebra (Y, β) is a $\mathcal{C}$-arrow $h\colon X \to Y$ that preserves the algebra structure, *i.e.*, $h \circ \alpha = \beta \circ Fh$. F-algebras and F-algebra morphisms form a category denoted by $\mathsf{Alg}_{\mathcal{C}}(F)$. An *initial F-algebra* is an initial object (A, α) in $\mathsf{Alg}_{\mathcal{C}}(F)$, i.e., for all F-algebras (X, β) there is a unique F-algebra morphism $h\colon (A, \alpha) \to (X, \beta)$.

A *monad* (on $\mathcal{C}$) is a triple (T, η, μ) consisting of a functor $T\colon \mathcal{C} \to \mathcal{C}$ and two natural transformations $\eta\colon Id \to T$ (the unit) and $\mu\colon TT \to T$ (the multiplication) satisfying $\mu \circ \eta T = id_T = \mu \circ T\eta$ and $\mu \circ T\mu = \mu \circ \mu T$. For brevity, we will sometimes refer to a monad simply by its functor part, leaving the unit and multiplication implicit. An *Eilenberg-Moore T-algebra* is a T-algebra (A, α) such that $\alpha \circ \eta_A = id_A$ and $\alpha \circ \mu_A = \alpha \circ T\alpha$. Eilenberg-Moore T-algebras and T-algebra morphisms form a category denoted by $\mathsf{EM}(T)$. In particular, for every X in $\mathcal{C}$, (TX, μ_X) is the free Eilenberg-Moore T-algebra on X, i.e., for every (A, α) in $\mathsf{EM}(T)$ and every $\mathcal{C}$-arrow $f\colon X \to A$ there is a unique T-algebra morphism (called the *free extension of f*) $f^\sharp\colon (TX, \mu_X) \to (A, \alpha)$ such that $f^\sharp \circ \eta_X = f$. Notice also that we have $f^\sharp = \alpha \circ Tf$.

3.2.2 Determinisation

Let (T, η, μ) be a monad on Set and $F\colon \mathsf{Set} \to \mathsf{Set}$ a functor given by $FX = B \times X^\Sigma$ where Σ is a set and B is the carrier of an Eilenberg-Moore T-algebra (B, β). Then FT-coalgebras can be seen as automata with input alphabet Σ, output in B and branching structure given by T. For example, nondeterministic automata are $F\mathcal{P}$-coalgebras where $FX = 2 \times X^\Sigma$ and $\beta = \vee\colon \mathcal{P}2 \to 2$ is the join (or max). Such FT-coalgebras can be "determinised" using a generalisation of the classic powerset construction (Silva et al., 2010), and the result can be seen as an F-coalgebra in the category $\mathsf{EM}(T)$. We follow Bartels (2004); Jacobs (2006) in explaining this general construction. As shown in Jacobs (2006), there is a so-called distributive law $\lambda\colon TF \Rightarrow FT$ of the monad (T, η, μ) over the functor F given by

$$\lambda_X\colon T(B \times X^\Sigma) \xrightarrow{\langle T\pi_1, T\pi_2\rangle} TB \times T(X^\Sigma) \xrightarrow{\beta \times \mathsf{st}} B \times (TX)^\Sigma \tag{3.1}$$

where $\mathsf{st}\colon T \circ (-)^\Sigma \Rightarrow (-)^\Sigma \circ T$ is the strength natural transformation that exists for all monads on Set. Such a distributive law λ corresponds to a lifting of

$F\colon \mathsf{Set} \to \mathsf{Set}$ to a functor $F_\lambda\colon \mathsf{EM}(T) \to \mathsf{EM}(T)$ (Johnstone, 1975), and it induces a functor $(-)^\sharp\colon \mathsf{Coalg}_{\mathsf{Set}}(FT) \to \mathsf{Coalg}_{\mathsf{EM}(T)}(F_\lambda)$ which sends an FT-coalgebra $\gamma = \langle o, t\rangle\colon X \to B \times (TX)^\Sigma$ to its determinisation $\gamma^\sharp = F\mu_X \circ \lambda_{TX} \circ T\gamma$, that is,

$$\gamma^\sharp = \; TX \xrightarrow{T\gamma} T(B \times (TX)^\Sigma) \xrightarrow{\lambda_{TX}} B \times (TTX)^\Sigma \xrightarrow{B\times(\mu_X)^\Sigma} B \times (TX)^\Sigma \tag{3.2}$$

Another perspective is that λ induces an Eilenberg-Moore T-algebra structure α on FTX, and $\gamma^\sharp\colon (TX, \mu_S) \to (FTX, \alpha)$ is the free extension of γ induced by α. This also justifies our use of the notation $(-)^\sharp$. The determinisation $\gamma^\sharp$ can be seen as a Moore automaton in $\mathsf{EM}(T)$. We will use the determinisation construction in order to place alternating automata and weighted automata in our general minimisation framework.

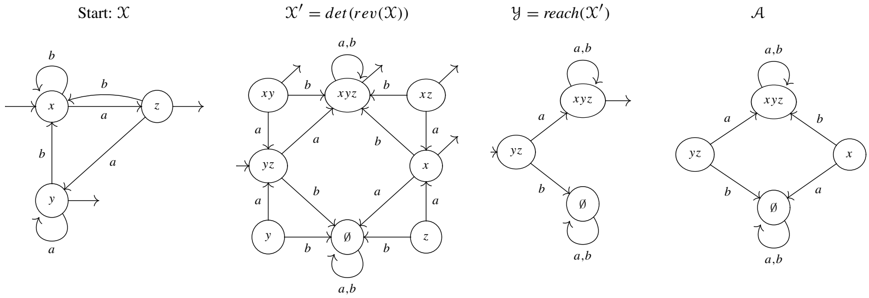
3.3 Minimisation via Dual Adjunctions

In this section, we will present a categorical picture that unifies the approaches of Bezhanishvili et al. (2012) and Bonchi et al. (2014). In particular, our picture formalises the role of trace logic in the minimisation algorithms. Some of the technical details of this part are known from Rot (2016); Kerstan et al. (2014); Hermida & Jacobs (1998); Bonchi et al. (2014)—precise connections are detailed throughout the subsections below and in Sect. 3.6.

3.3.1 An Illustrative Example

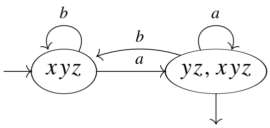
We illustrate the differences between the approaches of Bezhanishvili et al. (2012) and Bonchi et al. (2014) on classic deterministic finite automata (DFAs). A DFA can be minimised via Brzozowski's algorithm as well as via the approach in Bezhanishvili et al. (2012) using the duality between finite sets and finite Boolean algebras, and observing that a DFA is a PODFA with a single observation (or atomic proposition letter) p which is true precisely at the accepting states.

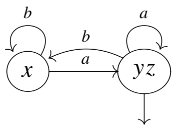
We will apply the two minimisation algorithms to the DFA $\mathcal{X}$ below left which is also found in (11) in Bonchi et al. (2014). The DFA $\mathcal{X}$ accepts the language $(a + b)^*a$. The result $\mathcal{X}'$ after the first reverse-determinise step in Brzozowski's algorithm is shown to the right of $\mathcal{X}$. Disregarding final states, $\mathcal{X}'$ is also the modal algebra obtained from $\mathcal{X}$. The reachable part of $\mathcal{X}'$ is the automaton $\mathcal{Y}$, and the algebra $\mathcal{A}$ is the subalgebra of definable subsets in the modal language with the single proposition letter p, and a modality for each letter of the alphabet.



After doing again reverse-determinise-reachability on $\mathcal{Y}$ to complete the Brzozowski algorithm, we get the automaton below on the left. Taking the dual automaton of atoms of $\mathcal{A}$, we get the coalgebra below on the right.

Result of Brzozowski's algorithm: Result of minimisation-via-duality





Accepts $(a + b)^*a$ State x accepts $(a + b)^*a$

The two deterministic automata (modulo initial state) are clearly isomorphic, but not identical.

3.3.2 Automata, Algebras and Coalgebras

Throughout this paper, we let Σ be a finite set. We will consider different types of automata, but they will all have input alphabet Σ.

A deterministic finite automaton (DFA), on alphabet Σ, consists of a set X (the state space), a transition map $t\colon X \to X^{\Sigma}$ (or equivalently $t\colon \Sigma \times X \to X$), an acceptance map $f\colon X \to 2$, and an initial state $i\colon 1 \to X$. We generalise this basic definition to arbitrary categories as follows.

Definition 3.1 Let $\mathcal{C}$ be a category, and let I and B be objects in $\mathcal{C}$. A $\mathcal{C}$-*automaton* (*with initialisation in I and output in B*) is a quadruple $\mathcal{X} = (X, t, i, f)$ consisting of a state space object (or carrier) X in $\mathcal{C}$, a Σ-indexed set of transition morphisms $\{t_a\colon X \to X \mid a \in \Sigma\}$, an initialisation morphism $i\colon I \to X$, and an output morphism $f\colon X \to B$. A $\mathcal{C}$- *automaton morphism* from $\mathcal{X}_1 = (X_1, t_1, i_1, f_1)$ to $\mathcal{X}_2 = (X_2, t_2, i_2, f_2)$ is a $\mathcal{C}$-morphism $h\colon X_1 \to X_2$ such that for all $a \in \Sigma$, $h \circ t_{1,a} = t_{2,a} \circ h$, $f_1 = f_2 \circ h$, and $h \circ i_1 = i_2$. Together, $\mathcal{C}$-automata (with initialisation in I and output in B) and their morphisms form a category which we denote by $\mathsf{Aut}_{\mathcal{C}}^{I,B}$.

A deterministic automaton is a Set-automaton with output in 2 and initialisation in 1.

A central observation in Bonchi et al. (2014) is that automata can be seen as coalgebras with initialisation, or dually, as algebras with output, as we briefly recall now. Assuming that $\mathcal{C}$ has products and coproducts, the transition morphisms $\{t_a : X \to X \mid a \in \Sigma\}$ correspond uniquely to morphisms of the following type:

$$\frac{\langle t_a \rangle_{a\in\Sigma} : X \to X^{\Sigma}}{[t_a]_{a\in\Sigma} : \Sigma \cdot X \to X} \tag{3.3}$$

Letting F and G be endofunctors on $\mathcal{C}$ given by $FX = B \times X^{\Sigma}$ and $GX = I + \Sigma \cdot X$, we see that a $\mathcal{C}$-automaton is an F-coalgebra $\langle f, \langle t_a \rangle_{a\in\Sigma} \rangle : X \to B \times X^{\Sigma}$ together with an intialisation morphism $i : I \to X$. Or equivalently, a G-algebra $[i, [t_a]_{a\in\Sigma}] : GX \to X$ together with an output morphism $f : X \to B$.

3.3.3 Dual Adjunctions of Coalgebras, Algebras and Automata

The categorical picture that unifies the work in Bezhanishvili et al. (2012) and Bonchi et al. (2014) is sketched in the diagram (3.4) below. This picture starts with a base dual adjunction that is lifted to a dual adjunction between coalgebras and algebras. This adjunction captures the construction in (Bezhanishvili et al., 2012) for obtaining observable coalgebras via duality. The coalgebra-algebra adjunction is then lifted to a dual adjunction between automata which captures the formalisation of the Brzozowski algorithm from Bonchi et al. (2014), which uses automata with initial states. In the remainder of the section, we will explain the details of how this picture comes about.

$$\begin{tikzcd}
\left(\mathsf{Aut}_{\mathcal{C}}^{I,S(O)}\right)^{\mathsf{op}} \arrow[d] \arrow[r, bend left, "\overline{P}'"] \arrow[r, phantom, "\top"] & \mathsf{Aut}_{\mathcal{D}}^{O,P(I)} \arrow[d] \arrow[l, bend left, "\overline{S}'"] \\
\mathsf{Coalg}_{\mathcal{C}}(F_{\mathcal{C}})^{\mathsf{op}} \arrow[d] \arrow[r, bend left, "\overline{P}"] \arrow[r, phantom, "\top"] & \mathsf{Alg}_{\mathcal{D}}(G_{\mathcal{D}}) \arrow[d] \arrow[l, bend left, "\overline{S}"] \\
\mathcal{C}^{\mathsf{op}} \arrow[loop left, "F_{\mathcal{C}}^{\mathsf{op}}"] \arrow[r, bend left, "P"] \arrow[r, phantom, "\top"] & \mathcal{D} \arrow[loop right, "G_{\mathcal{D}}"] \arrow[l, bend left, "S"]
\end{tikzcd} \tag{3.4}$$

$$F_{\mathcal{C}} = S(O) \times (-)^{\Sigma}, \qquad G_{\mathcal{D}} = O + \Sigma \cdot (-)$$

3.3.3.1 Base Dual Adjunction

Our starting point is a dual adjunction $S \dashv P$ between categories $\mathcal{C}$ and $\mathcal{D}$ as in the above picture. We will generally try to avoid the use of superscript op, and treat P and S as contravariant functors. The units of the dual adjunction will be denoted $\eta\colon Id \Rightarrow PS$ and $\varepsilon\colon Id \Rightarrow SP$. The natural isomorphism of Hom-sets $\theta_{X,Y}\colon \mathcal{C}(X, SY) \to \mathcal{D}(Y, PX)$, will sometimes be written in both directions simply as $f \mapsto f^\flat$. For $f\colon X \to SY$, its adjoint is $f^\flat = Pf \circ \eta_Y$, and for $g\colon Y \to PX$, its adjoint is $g^\flat = Sg \circ \varepsilon_X$.

In all our examples, $\mathcal{C}$ and $\mathcal{D}$ are concrete categories, and the dual adjunction arises from homming into a dualising object Δ (cf. Porst & Tholen, 1991), i.e., $P = \mathcal{C}(-, \Delta)$ and $S = \mathcal{D}(-, \Delta)$, and we will often denote both of them by $\Delta^{(-)}$. This means that adjoints are obtained simply by swapping arguments. E.g., for $f\colon Y \to \Delta^X$ we have $f^\flat(x)(y) = f(y)(x)$. Moreover, the units are given by evaluation. E.g. $\eta_X\colon X \to \Delta^{\Delta^X}$ is defined by $\eta_X(x)(f) = f(x)$.

Example 3.1 Consider the self-dual adjunction of Set given by the *contravariant powerset functor* $\mathcal{Q} = \mathsf{Set}(-, 2)$ which maps a set X to its powerset 2^X and a function $f: X \to Y$ to its inverse image map $f^{-1}\colon 2^Y \to 2^X$. The functor $\mathcal{Q}$ is dually self-adjoint with $\mathcal{Q}^{\mathsf{op}} \dashv \mathcal{Q}$, and the isomorphism of Hom-sets is given by taking exponential transposes, i.e., for $f\colon X \to 2^Y$ we have $f^\flat\colon Y \to 2^X$.

Dual adjunctions are also called *logical connections* as they form the basis of semantics for coalgebraic modal logics (Bonsangue & Kurz, 2005; Klin, 2007; Jacobs & Sokolova, 2010). In this logic perspective, $\mathcal{C}$ is a category of state spaces, $\mathcal{D}$ is a category of algebras (e.g. Boolean algebras) encoding a propositional logic, and the functor $G_\mathcal{D}$ encodes a modal logic. Intuitively, the adjoint P maps a state space C to the predicates over C, and S maps an algebra A to the theories of A. The logic given by $G_\mathcal{D}$ can be interpreted over $F_\mathcal{C}$-coalgebras by providing a so-called one-step modal semantics in the form of a natural transformation $\varrho\colon G_\mathcal{D} P \Rightarrow PF_\mathcal{C}$, or equivalently via its mate $\xi\colon F_\mathcal{C} S \Rightarrow SG_\mathcal{D}$. The pair $(G_\mathcal{D}, \varrho)$ is referred to as a logic. By assuming that the initial $G_\mathcal{D}$-algebra (A_0, α_0) exists, and viewing its elements as formulas, the semantics of formulas in an $F_\mathcal{C}$-coalgebra (C, γ) is obtained from the initial map $s^{G_\mathcal{D}} : (A_0, \alpha_0) \to P(\gamma) \circ \varrho_C$. As an underlying $\mathcal{D}$-map, it has type $s^{G_\mathcal{D}} : A_0 \to P(C)$, hence it maps formulas to predicates. Alternatively, the semantics can be specified by the theory map $th^{G_\mathcal{D}} : C \to S(A_0)$ which is defined as the adjoint of $s^{G_\mathcal{D}}$. We refer to Bonsangue & Kurz (2005); Klin (2007); Jacobs & Sokolova (2010) for a more detailed introduction to coalgebraic modal logic via dual ajdunctions.

3.3.3.2 Dual Adjunction Between Coalgebras and Algebras

The base dual adjunction in (3.4) lifts to one between coalgebras and algebras due to the shape of the functors $F_\mathcal{C}$ and $G_\mathcal{D}$. This follows from some basic results in Hermida & Jacobs (1998); Kerstan et al. (2014) as we explain now.

So assume that $\mathcal{C}$ has products, $\mathcal{D}$ has coproducts, and that we have a base dual adjunction $S \dashv P$ and functors $F_{\mathcal{C}}$ and $G_{\mathcal{D}}$ as in (3.4), in particular,

$$F_{\mathcal{C}}(C) = S(O) \times C^{\Sigma} \quad \text{and} \quad G_{\mathcal{D}}(D) = O + \Sigma \cdot D$$

We know from Hermida & Jacobs (1998, Cor. 2.15) (see also Kerstan et al., 2014, Theorem 2.5), that the dual adjunction $S \dashv P$ lifts to a dual adjunction $\overline{S} \dashv \overline{P}$ between $\mathsf{Coalg}_{\mathcal{C}}(F_{\mathcal{C}}) = \mathsf{Alg}_{\mathcal{C}^{\mathrm{op}}}(F_{\mathcal{C}}^{\mathrm{op}})$ and $\mathsf{Alg}_{\mathcal{D}}(G_{\mathcal{D}})$ if there is a natural isomorphism $\xi\colon F_{\mathcal{C}}S \overset{\sim}{\Rightarrow} SG_{\mathcal{D}}$. For our choice of $F_{\mathcal{D}}$ and $G_{\mathcal{C}}$, we have such a natural isomorphism, since for all $D \in \mathcal{D}$,

$$F_{\mathcal{C}}S(D) = S(O) \times S(D)^{\Sigma} \cong S(O + \Sigma \cdot D) = SG_{\mathcal{D}}(D) \tag{3.5}$$

since S (as a dual adjoint functor) turns colimits into limits. Let $\varrho\colon G_{\mathcal{D}}P \Rightarrow PF_{\mathcal{C}}$ be the mate of ξ, i.e., the adjoint of $\xi_P \circ F_{\mathcal{C}}\varepsilon$:

$$\varrho = PF_{\mathcal{C}}\varepsilon \circ P\xi P \circ \eta G_{\mathcal{D}}P \tag{3.6}$$

The lifted adjoint functors are defined for all $F_{\mathcal{C}}$-coalgebras $\gamma\colon C \to F_{\mathcal{C}}(C)$, all $F_{\mathcal{C}}$-coalgebra morphisms f, all $G_{\mathcal{D}}$-algebras $\alpha\colon G_{\mathcal{D}}(D) \to D$, and all $G_{\mathcal{D}}$-algebra morphisms g by:

$$\begin{array}{ll} \overline{P}(\gamma) = P\gamma \circ \varrho_C\colon G_{\mathcal{D}}PC \to PC, & \overline{P}(f) = P(f) \\ \overline{S}(\alpha) = \xi_D \circ S\alpha\colon SD \to F_{\mathcal{C}}SD, & \overline{S}(g) = S(g) \end{array} \tag{3.7}$$

Remark 1 If $F'_{\mathcal{C}}\colon \mathcal{C} \to \mathcal{C}$ is $F'_{\mathcal{C}}(C) = B \times C^{\Sigma}$ with $B \cong S(O)$, then $F'_{\mathcal{C}} \overset{\sim}{\Rightarrow} F_{\mathcal{C}}$, and hence $\mathsf{Coalg}_{\mathcal{C}}(F'_{\mathcal{C}}) \cong \mathsf{Coalg}_{\mathcal{C}}(F_{\mathcal{C}})$, so we can think of $F'_{\mathcal{C}}$-coalgebras as $F_{\mathcal{C}}$-coalgebras.

The isomorphism $\overline{\theta}$ of Hom-sets for $\overline{S} \dashv \overline{P}$ is simply the restriction of the isomorphism θ of Hom-sets for $S \dashv P$ to the relevant morphisms.

The natural transformation $\varrho\colon G_{\mathcal{D}}P \Rightarrow PF_{\mathcal{C}}$ provides the one-step semantics for a modal logic for $F_{\mathcal{C}}$-coalgebras as described at the end of Sect. 3.3.3.1. This makes most sense when the dual adjunction arises from a dualising object Δ in which case Δ is a domain of truth-values, i.e., the logic is Δ-valued, and when $\mathcal{D}$ is a category of algebras with operations given by a signature *Sgn*. Letting $\Phi_{\mathcal{D}}(X)$ denote the free algebra in $\mathcal{D}$ over a set X, an algebra functor $G_{\mathcal{D}} = \Phi_{\mathcal{D}}(\Omega) + \Sigma \cdot (-)$ then corresponds to a modal language $\mathbb{L}(G_{\mathcal{D}})$ that has atomic propositions from a finite set Ω, labelled modalities $[a]$, $a \in \Sigma$, and the propositional connectives are the operations from *Sgn*. That is, formulas in $\mathbb{L}(G_{\mathcal{D}})$ are generated by the following grammar:

$$\varphi ::= q \in \Omega \mid [a]\varphi, a \in \Sigma \mid \sigma(\varphi_1, \ldots, \varphi_n), \sigma \in Sgn \tag{3.8}$$

where n is the arity of the operation σ.

For our specific choice of functors $F_{\mathcal{C}}$ and $G_{\mathcal{D}}$, and when the adjunction arises from a dualising object Δ (i.e., $S(\Phi_{\mathcal{D}}(\Omega)) = \Delta^{\Phi_{\mathcal{D}}(\Omega)}$), we can compute the concrete definition of ϱ from (3.6), and we get the following Δ-valued modal semantics of the language $\mathbb{L}(G_{\mathcal{D}})$:

$$
\begin{aligned}
[\![q]\!](x) &= j(q), && \text{where } \gamma(x) = \langle j\colon \Delta^{\Phi_{\mathcal{D}}(\Omega)}, d\colon X^{\Sigma}\rangle \\
[\![[a]\varphi]\!](x) &= [\![\varphi]\!](d(a)), && \text{where } \gamma(x) = \langle j\colon \Delta^{\Phi_{\mathcal{D}}(\Omega)}, d\colon X^{\Sigma}\rangle \\
[\![\sigma(\psi_1,\ldots,\psi_n)]\!](x) &= \sigma([\![\psi_1]\!](x),\ldots,[\![\psi_n]\!](x))
\end{aligned}
\tag{3.9}
$$

This shows that ϱ gives the expected modal semantics for $F_{\mathcal{C}}$-coalgebras viewed as deterministic Σ-labelled Kripke frames with observations from Ω. In particular, the modalities are "deterministic" Kripke box/diamond-modalities.

Example 3.2 We consider the case of DFAs. Here $\mathcal{C} = \mathcal{D} = \mathsf{Set}$, $F_{\mathsf{Set}} = 2 \times (-)^{\Sigma}$ and $G_{\mathsf{Set}} = 1 + \Sigma \cdot (-)$, and the self-dual adjunction of Set is given by the contravariant powerset functor $\mathcal{Q} = \mathsf{Set}(-, 2)$ (Example 3.1). The formulas of $\mathbb{L}(G_{\mathsf{Set}})$ are built from a single atomic proposition q, and a modality $[a]$ for each $a \in \Sigma$, since $\mathcal{D} = \mathsf{Set}$ means that there are no propositional connectives. The initial G_{Set}-algebra is Σ^*, the set of finite words over Σ, which is easily seen to be in bijection with the set of formulas. The logic we obtain is *trace logic* (Klin, 2007), but here interpreted over DFAs rather than labelled transition systems as in Klin (2007). The natural transformation ϱ has type $\varrho_X\colon 1 + \Sigma \cdot 2^X \to 2^{2\times X^{\Sigma}}$, given concretely below together with the induced semantics, where we write $x \Vdash \varphi$ iff $[\![\varphi]\!](x) = 1$:

$$
\begin{aligned}
\varrho_X(q) &= \{(b,d) \in 2 \times X^{\Sigma} \mid b = 1\} && x \Vdash q \iff x \text{ is accepting} \\
\varrho_X(a,U) &= \{(b,d) \in 2 \times X^{\Sigma} \mid d(a) \in U\} && x \Vdash [a]\varphi \iff x \xrightarrow{a} y \text{ and } y \Vdash \varphi
\end{aligned}
$$

3.3.3.3 Dual Adjunction Between Automata

In order to obtain the upper adjunction in (3.4) (which formalises the Brzozowski algorithm), we will use algebra and coalgebra structure on both sides, hence we assume that $\mathcal{C}$ and $\mathcal{D}$ both have products and coproducts. The lifting is a small extension of $\overline{S} \dashv \overline{P}$ obtained by defining how an initialisation map $I \to C$ for an $F_{\mathcal{C}}$-coalgebra γ is turned into an observation map $PC \to PI$ for the $G_{\mathcal{D}}$-algebra $\overline{P}(\gamma)$, and vice versa for $\overline{S}$.

Theorem 3.1 *Assume that $\mathcal{C}$ and $\mathcal{D}$ both have products and coproducts, and that we have the dual adjunctions and functors $F_{\mathcal{C}}$ and $G_{\mathcal{D}}$ as specified in the two lower parts of (3.4). The dual adjunction $\overline{S} \dashv \overline{P}$ between $\mathsf{Coalg}_{\mathcal{C}}(F_{\mathcal{C}})$ and $\mathsf{Alg}_{\mathcal{D}}(G_{\mathcal{D}})$ lifts to a dual adjunction $\overline{S}' \dashv \overline{P}'$ between $\mathsf{Aut}_{\mathcal{C}}^{I,SO}$ and $\mathsf{Aut}_{\mathcal{D}}^{O,PI}$ by defining $\overline{P}'$ and $\overline{R}'$ as follows for all $\gamma\colon C \to F_{\mathcal{C}}C$ and $\alpha\colon G_{\mathcal{D}}D \to D$:*

$$
\begin{aligned}
\overline{P}'(\gamma, i\colon I \to C) &= (\overline{P}(\gamma)\colon G_{\mathcal{D}}PC \to PC,\ P(i)\colon PC \to PI), & \overline{P}'(f) &= P(f) \\
\overline{S}'(\alpha, j\colon D \to PI) &= (\overline{R}(\alpha)\colon SD \to F_{\mathcal{C}}SD,\ j^{\flat}\colon I \to SD), & \overline{S}'(g) &= S(g)
\end{aligned}
$$

Proof This is a minor generalisation of Proposition 9.1 in Bonchi et al. (2014). It suffices to show that for all $\mathcal{C}$-arrows $i\colon I \to C$, and all $\mathcal{D}$-arrows $g\colon D \to PI$ and $h\colon D \to PX$: $g = Pi \circ h$ iff $g^\flat = h^\flat \circ i$. First, if $g = Pi \circ h$, then $g^\flat = Sg \circ \varepsilon_I = Sh \circ SPi \circ \varepsilon_I = Sh \circ \varepsilon_X \circ i = h^\flat \circ i$, where the third equality follows from naturality of ε. Conversely, if $g^\flat = h^\flat \circ i$, then $g = Pg^\flat \circ \eta_D = Pi \circ Ph^\flat \circ \eta_D = Pi \circ h$. □

The final $F_\mathcal{C}$-coalgebra exists and has carrier $S(O)^{\Sigma^*}$. The final morphism $!\colon C \to S(O)^{\Sigma^*}$ assigns to each state in C an $S(O)$-weighted language. For $\mathcal{X} = \langle \gamma, i\rangle \in \mathsf{Aut}_\mathcal{C}^{I,S(O)}$, we define its language semantics as the composition $I \xrightarrow{i} C \xrightarrow{!_\gamma} S(O)^{\Sigma^*}$. This $\mathcal{C}$-morphism can be seen as a Σ^*-indexed family of $\mathcal{C}$-morphisms $\langle\!|\mathcal{X}|\!\rangle_w\colon I \to SO$ defined for all $w = a_1 \cdots a_k \in \Sigma^*$ by

$$\langle\!|\mathcal{X}|\!\rangle_w \quad = \quad I \xrightarrow{i} X \xrightarrow{t_{a_1}} \cdots \xrightarrow{t_{a_k}} X \xrightarrow{f} S(O)$$

Computing the adjoint transpose $\langle\!|\mathcal{X}|\!\rangle_w^\flat = P\langle\!|\mathcal{X}|\!\rangle \circ \eta_O$, we get the $\mathcal{D}$-morphism:

$$\langle\!|\mathcal{X}|\!\rangle_w^\flat \quad = \quad P(I) \xleftarrow{P(i)} P(X) \xleftarrow{P(t_{a_1})} \cdots \xleftarrow{P(t_{a_k})} P(X) \xleftarrow{f^\flat} O$$

Hence $\langle\!|\mathcal{X}|\!\rangle_w^\flat = \langle\!|\overline{P}'(\mathcal{X})|\!\rangle_{w^R}$ where $w^R = a_k \cdots a_1$ is the reversal of w. Similarly, we find that for all $\mathcal{Y} \in \mathsf{Aut}_\mathcal{D}^{O,P(I)}$, $\langle\!|\mathcal{Y}|\!\rangle_w^\flat = \langle\!|\overline{S}'(\mathcal{Y})|\!\rangle_{w^R}$. In the case of DFAs from Example 3.2 where $I = O = 1$ and $S(O) = P(I) \cong 2$, the above says that the adjoint functors reverse the language accepted by the automaton.

3.3.4 Language Semantics and Trace Logic

In this section, we give a general condition on the output sets that ensures that we can link trace logic with the full modal logic via an adjunction. This places trace logic in the general picture. In Bezhanishvili et al. (2012), it was shown in each of the concrete examples that trace logic is equally expressive as the full modal logic. The results of this section give a general explanation of this fact.

Assume that the category $\mathcal{D}$ is monadic over Set with adjunction $\Phi_\mathcal{D} : \mathcal{D} \rightleftarrows \mathsf{Set} : U_\mathcal{D}$. This adjoint situation allows us to relate the Set-based language semantics to the final $F_\mathcal{C}$-coalgebra semantics as we will show now.

Consider the functor $G\colon \mathsf{Set} \to \mathsf{Set}$ defined as $G(X) = \Omega + \Sigma \cdot X = \Omega + \Sigma \times X$ where Ω is a finite set of observations. Then the set $\Sigma^*\Omega$ is an initial G-algebra with algebra structure $\Omega + \Sigma \times (\Sigma^*\Omega) \to \Sigma^*\Omega$ given by prefixing $\mathrm{o} \in \Omega$ with the empty word $\mathrm{o} \mapsto \varepsilon\mathrm{o}$ and concatenation $(a, w) \mapsto aw$. Then we can compose the adjunction $\Phi_\mathcal{D} \dashv U_\mathcal{D}$ with the dual adjunction $S \dashv P$ to obtain a dual adjunction between $\mathcal{C}$ and Set as follows:

$$
\begin{tikzcd}
\mathcal{C}^{op} \arrow[loop left, "F_{\mathcal{C}}^{op}"] \arrow[r, bend left, "P"] \arrow[r, phantom, "\top"] & \mathcal{D} \arrow[l, bend left, "S"] \arrow[loop below, "G_{\mathcal{D}}"] \arrow[r, bend left, "U_{\mathcal{D}}"] \arrow[r, phantom, "\top"] & \mathsf{Set} \arrow[l, bend left, "\Phi_{\mathcal{D}}"] \arrow[loop right, "G"]
\end{tikzcd}
\tag{3.10}
$$

Lemma 3.1 *Assume we have the situation in (3.10), and that $F_{\mathcal{C}}$, $G_{\mathcal{D}}$, G are defined by:*

$$
F_{\mathcal{C}}(C) = S\Phi_{\mathcal{D}}(\Omega) \times C^{\Sigma}, \qquad G_{\mathcal{D}}(D) = \Phi_{\mathcal{D}}(\Omega) + \Sigma \cdot D, \qquad G(X) = \Omega + \Sigma \cdot X.
$$

Then (3.10) lifts to

$$
\begin{tikzcd}
\mathsf{Coalg}_{\mathcal{C}}(F_{\mathcal{C}})^{op} \arrow[r, bend left, "\overline{P}"] \arrow[r, phantom, "\top"] & \mathsf{Alg}_{\mathcal{D}}(G_{\mathcal{D}}) \arrow[l, bend left, "\overline{S}"] \arrow[r, bend left, "\overline{U_{\mathcal{D}}}"] \arrow[r, phantom, "\top"] & \mathsf{Alg}_{\mathsf{Set}}(G) \arrow[l, bend left, "\overline{\Phi_{\mathcal{D}}}"]
\end{tikzcd}
\tag{3.11}
$$

Proof The dual adjunction on the left lifts because of a special case of (3.5). For similar reasons, the adjunction on the right lifts, because there is a natural isomorphism $\kappa : \Phi_{\mathcal{D}} G \overset{\sim}{\Rightarrow} G_{\mathcal{D}} \Phi_{\mathcal{D}}$ that can be obtained as follows

$$
\kappa\colon \Phi_{\mathcal{D}} G X = \Phi_{\mathcal{D}}(\Omega + \Sigma \cdot X) \cong \Phi_{\mathcal{D}}(\Omega) + \Sigma \cdot \Phi_{\mathcal{D}}(X) = G_{\mathcal{D}} \Phi_{\mathcal{D}}(X), \tag{3.12}
$$

since $\Phi_{\mathcal{D}}$ (being a left adjoint) preserves colimits. By Hermida & Jacobs (1998), Theorem 2.14, $\Phi_{\mathcal{D}} \dashv U_{\mathcal{D}}$ lifts to an adjunction $\overline{\Phi_{\mathcal{D}}} \dashv \overline{U_{\mathcal{D}}}$ between $\mathsf{Alg}_{\mathcal{D}}(G_{\mathcal{D}})$ and $\mathsf{Alg}_{\mathsf{Set}}(G)$ where the functor $\overline{\Phi_{\mathcal{D}}}$ maps a G-algebra (X, α) to the $G_{\mathcal{D}}$-algebra $(\Phi_{\mathcal{D}}(X), \Phi_{\mathcal{D}}\alpha \circ \kappa^{-1})$.

By composition of adjunctions, also $S\Phi_{\mathcal{D}} \dashv U_{\mathcal{D}} P$ lifts. This could also be verified by noticing that for all sets X, there is natural isomorphism

$$
\xi^{\mathsf{trc}} := S\kappa \circ \xi\Phi_{\mathcal{D}} \colon F_{\mathcal{C}} S\Phi_{\mathcal{D}} \overset{\sim}{\Rightarrow} S\Phi_{\mathcal{D}} G \tag{3.13}
$$

where $\xi\colon F_{\mathcal{C}} S \overset{\sim}{\Rightarrow} SG_{\mathcal{D}}$ from (3.5) is the mate of the modal logic $(G_{\mathcal{D}}, \varrho)$. Hence by Hermida & Jacobs (1998, Theorem 2.14, Corollary 2.15) (see also Kerstan et al., 2014, Theorem 2.5), the adjunction $S\Phi_{\mathcal{D}} \dashv U_{\mathcal{D}} P$ lifts to one between $\mathsf{Coalg}_{\mathcal{C}}(F_{\mathcal{C}})^{op}$ and $\mathsf{Alg}_{\mathsf{Set}}(G)$.

Letting $\varrho^{\mathsf{trc}}\colon GU_{\mathcal{D}}P \Rightarrow U_{\mathcal{D}}PF_{\mathcal{C}}$ be the mate of ξ^{trc} from (3.13), then $(G, \varrho^{\mathsf{trc}})$ is a modal logic for $F_{\mathcal{C}}$-coalgebras. Since its formulas are the elements of the intial G-algebra of traces, we refer to $(G, \varrho^{\mathsf{trc}})$ as a trace logic.

Lemma 3.2 *The theory maps th^G and $th^{G_{\mathcal{D}}}$ of the logics $(G, \varrho^{\mathsf{trc}})$ and $(G_{\mathcal{D}}, \varrho)$ coincide.*

Proof Due to the adjunctions in (3.10), the intial G-algebra $\Sigma^*\Omega$ of traces is mapped by $\overline{\Phi_{\mathcal{D}}}$ to an initial $G_{\mathcal{D}}$-algebra, which in turn is mapped by $\overline{S}$ to a final $F_{\mathcal{D}}$-coalgebra. The coincidence of the theory maps follows from them being adjoints of the initial maps.

Since the mates ξ and ξ^{trc} are both natural isomorphisms, it follows from Klin (2007); Jacobs & Sokolova (2010) (and $\mathcal{C}$ having a suitable factorisation system, cf. Theorem 3.2) that the full modal logic $(G_{\mathcal{D}}, \varrho)$ and trace logic $(G, \varrho^{\mathsf{trc}})$ are both expressive for $F_{\mathcal{C}}$-coalgebras. In other words, the propositional connectives from $\mathcal{D}$-structure in the logic language $\mathbb{L}(G_{\mathcal{D}})$ do not add any expressive power to $\mathbb{L}(G) = \Sigma^*\Omega$. In summary, we arrive at the following proposition.

Proposition 3.1 *With the above assumptions, the trace logic* $(G, \varrho^{\mathsf{trc}})$ *and the full logic* $(G_{\mathcal{D}}, \varrho)$ *are equally expressive over* $F_{\mathcal{C}}$*-coalgebras, meaning that for all* $F_{\mathcal{C}}$*-coalgebras* $\gamma\colon C \to F_{\mathcal{C}}(C)$*, and all states* c_1, c_2 *in* C *(recall that* $\mathcal{C}$ *is a concrete category),* c_1 *and* c_2 *are logically equivalent for* $(G, \varrho^{\mathsf{trc}})$ *iff they are logically equivalent for* $(G_{\mathcal{D}}, \varrho)$*.*

By the uniqueness of final coalgebras up to isomorphism, it follows that there is an isomorphism $\sigma\colon S\Phi_{\mathcal{D}}(\Omega)^{\Sigma^*} \xrightarrow{\sim} S\,\Phi_{\mathcal{D}}(\Sigma^*\Omega)$ which links the language semantics in the automata/coalgebraic sense with trace logic semantics given by initiality.

Proposition 3.2 *For all* $F_{\mathcal{C}}$*-coalgebras* γ*, its language semantics defined as the unique morphism into the final* $F_{\mathcal{C}}$*-coalgebra* $S\Phi_{\mathcal{D}}(\Omega)^{\Sigma^*}$ *corresponds to the trace theory map* th^G *into the final* $F_{\mathcal{C}}$*-coalgebra* $\overline{S}\,\overline{\Phi_{\mathcal{D}}}(\Sigma^*\Omega)$*, (and with the theory map* $th^{G_{\mathcal{D}}}$*) via the isomorphism* σ*.*

We remark that it is straightforward to extend $\overline{\Phi_{\mathcal{D}}} \dashv \overline{U_{\mathcal{D}}}$ to an adjunction of automata by taking adjoints of additional output maps to the algebras. We omit the details.

Finally, we show that trace logic expressiveness can be extended to coalgebras for what we can think of as subfunctors of $F_{\mathcal{C}}$. This will be needed for the topological automata in Sect. 3.4.3.

Remark 2 Let $F'_{\mathcal{C}}$ be a functor on $\mathcal{C}$ which preserves monos and such that there is a natural transformation $\tau\colon F'_{\mathcal{C}} \Rightarrow F_{\mathcal{C}}$ which is abstract mono, i.e., all components are mono. Assume that $\mathcal{C}$ has factorisation system (E, M) with $E \subseteq Epi$ and $M \subseteq Mono$. Defining $\xi' = \xi^{\mathsf{trc}} \circ \tau_S$, then $\xi'\colon F'_{\mathcal{C}} S\Phi_{\mathcal{D}} \Rightarrow S\Phi_{\mathcal{D}}G$ defines semantics of trace formulas over $F'_{\mathcal{C}}$-coalgebras which is essentially the same as the semantics over $F_{\mathcal{C}}$-coalgebras. Since τ is abstract mono and ξ^{trc} is a natural iso, it follows that ξ' is abstract mono, and hence the associated logic is expressive (Klin, 2007; Jacobs & Sokolova, 2010).

3.3.5 Reachability and Observability

Recall that a classic DFA is *reachable* if all states are reachable by reading some word from the initial state; it is *observable* if no two states accept the same language; and it is *minimal* if it is both reachable and observable.

A main point emphasised in Bonchi et al. (2014) is that reachability is an algebraic concept, and observability is a coalgebraic concept. We will call an algebra *reachable* if it has no proper subalgebras, and a coalgebra is *observable* if it has no proper quotients. Both concepts apply to $\mathcal{C}$-automata as they are both coalgebras and algebras (cf. Sect. 3.3.2), and a $\mathcal{C}$-automaton is then minimal if its algebraic part is reachable, and its coalgebraic part is observable.

Both Bezhanishvili et al. (2012) and Bonchi et al. (2014) show that a reachable algebra dualises to an observable coalgebra, but the conditions and arguments differ. Note that in Bezhanishvili et al. (2012), observable coalgebras are referred to as minimal automata. In Bonchi et al. (2014), automata were generally considered as automata over Set, and the reachable part of an automaton was defined as the image of the initial G-algebra inside the automaton (using its G-algebra structure, after possibly forgetting $\mathcal{D}$-structure). In Bezhanishvili et al. (2012), an initial $G_{\mathcal{D}}$-algebra generally did not exist. Instead, a reachable algebra was obtained by taking a least subalgebra, the existence of which was ensured by assuming that $\mathcal{D}$ is wellpowered and having an epi-mono factorisation system. It is straighforward to show that when conditions for both are satisfied, the two reachability notions coincide, i.e., if an initial $G_{\mathcal{D}}$-algebra exists, and $\mathcal{D}$ is wellpowered with epi-mono factorisation system, then the least subalgebra is obtained by factorisation of the initial morphism.

The assumptions in Lemma 3.1 are most closely related to the setup of Bonchi et al. (2014), as we have an initial $G_{\mathcal{D}}$-algebra. The connection to the logical perspective of (Bezhanishvili et al., 2012) comes from viewing the initial $G_{\mathcal{D}}$-algebra as a generalisation of the Lindenbaum algebra. For an $F_{\mathcal{C}}$-coalgebra (C, γ), the factorisation of the initial morphism to $\overline{P}(\gamma)$ then yields the subalgebra of $\mathbb{L}(G_{\mathcal{D}})$-definable subsets of C (or more abstractly, $\mathbb{L}(G_{\mathcal{D}})$-definable Δ-valued predicates on C). By Lemma 3.1, $\Phi_{\mathcal{D}}(\Sigma^*\Omega)$ is also initial, and hence the reachable part of $\overline{P}(\gamma)$ is equivalently characterised as the factorisation of the unique morphism from $\Phi_{\mathcal{D}}(\Sigma^*\Omega)$, and this factorisation is easily seen to be the subalgebra generated by the trace logic definable subsets.

Finally, by Lemma 3.1, a quotient of an initial $G_{\mathcal{D}}$-algebra is mapped by $\overline{S}$ to a subobject of a final $F_{\mathcal{C}}$-coalgebra since the dual adjoint functors turn colimits into limits. A subcoalgebra of a final coalgebra is necessarily observable. The following proposition summarises our discussion.

Proposition 3.3 *Under the assumptions of Lemma 3.1, and assuming further that $\mathcal{D}$ has a factorisation system (E, M) such that $E \subseteq Epi$ and $M \subseteq Mono$, we then have:*

For all $(D, \delta) \in \mathsf{Alg}_{\mathcal{D}}(G_{\mathcal{D}})$, let reach$(D, \delta)$ be the reachable part of (D, δ) obtained by (E, M)-factorisation of the initial morphism:

$$\overline{\Phi_{\mathcal{D}}}(\Sigma^*\Omega, \alpha) \overset{e}{\twoheadrightarrow} \mathit{reach}(D, \delta) \overset{m}{\hookrightarrow} (D, \delta).$$

The epimorphism $e\colon \overline{\Phi_{\mathcal{D}}}(\Sigma^*\Omega, \alpha) \twoheadrightarrow \mathit{reach}(D, \delta)$ *is mapped by* $\overline{S}$ *to a monomorphism*

$$\overline{S}(e)\colon \overline{S}(\mathit{reach}(D, \delta)) \hookrightarrow \overline{S\Phi_{\mathcal{D}}}(\Sigma^*\Omega, \alpha).$$

As a subcoalgebra of a final coalgebra, $\overline{S}(\mathit{reach}(D, \delta))$ *is an observable* $F_{\mathcal{C}}$*-coalgebra.*

Note that $\overline{S}$ maps epis to monos, but monos are not necessarily mapped to epis, unless we have a full duality. In particular, with only a dual adjunction we cannot argue that a least subalgebra of $\overline{P}(\gamma)$ is mapped by $\overline{S}$ to a largest quotient of γ, as in Bezhanishvili et al. (2012).

Extending the notion of reachable part to $\mathcal{D}$-automata is done simply by taking the reachable part of their $G_{\mathcal{D}}$-algebraic part and restricting the output map. Proposition 3.3 thus also tells us how to obtain an observable $\mathcal{C}$-automaton by taking the reachable part of the dual $\mathcal{D}$-automaton.

Brzozowski's algorithm produces a minimal $\mathcal{C}$-automaton by also taking the reachable part of the resulting observable $\mathcal{C}$-automaton, that is, with respect to the algebraic structure of $\mathcal{C}$-automata given by $G_{\mathcal{C}} = I + \Sigma \cdot (-)$. In order to do so, we need to assume that also $\mathcal{C}$ has a suitable factorisation system.

3.3.6 Abstract Minimisation Algorithms

We now put everything together into one diagram with which we can describe both approaches from Bezhanishvili et al. (2012) and Bonchi et al. (2014) including the role of trace logic.

$$\begin{tikzcd}
\left(\mathsf{Aut}_{\mathcal{C}}^{I, S\Phi_{\mathcal{D}}(\Omega)}\right)^{\mathsf{op}} \arrow[r, bend left, "\overline{P}'"] \arrow[r, phantom, "\top"] \arrow[d] & \mathsf{Aut}_{\mathcal{D}}^{\Phi_{\mathcal{D}}(\Omega), P(I)} \arrow[l, bend left, "\overline{S}'"] \arrow[r, bend left, "\overline{U_{\mathcal{D}}}'"] \arrow[r, phantom, "\top"] \arrow[d] & \mathsf{Aut}_{\mathsf{Set}}^{\Omega, U_{\mathcal{D}} P(I)} \arrow[l, bend left, "\overline{\Phi_{\mathcal{D}}}'"] \arrow[d] \\
\mathsf{Coalg}_{\mathcal{C}}(F_{\mathcal{C}})^{\mathsf{op}} \arrow[r, bend left, "\overline{P}"] \arrow[r, phantom, "\top"] \arrow[d] & \mathsf{Alg}_{\mathcal{D}}(G_{\mathcal{D}}) \arrow[l, bend left, "\overline{S}"] \arrow[r, bend left, "\overline{U_{\mathcal{D}}}"] \arrow[r, phantom, "\top"] \arrow[d] & \mathsf{Alg}_{\mathsf{Set}}(G) \arrow[l, bend left, "\overline{\Phi_{\mathcal{D}}}"] \arrow[d] \\
\mathcal{C}^{\mathsf{op}} \arrow[loop left, "F_{\mathcal{C}}^{\mathsf{op}}"] \arrow[r, bend left, "P"] \arrow[r, phantom, "\top"] & \mathcal{D} \arrow[l, bend left, "S"] \arrow[loop below, "G_{\mathcal{D}}"] \arrow[r, bend left, "U_{\mathcal{D}}"] \arrow[r, phantom, "\top"] & \mathsf{Set} \arrow[l, bend left, "\Phi_{\mathcal{D}}"] \arrow[loop right, "G"]
\end{tikzcd}$$

$$F_{\mathcal{C}}(C) = S\Phi_{\mathcal{D}}(\Omega) \times C^{\Sigma}, \quad G_{\mathcal{D}}(D) = \Phi_{\mathcal{D}}(\Omega) + \Sigma \cdot D, \quad G(X) = \Omega + \Sigma \cdot X. \tag{3.14}$$

Theorem 3.2 *Let $\mathcal{C}$, $\mathcal{D}$ be concrete categories, both having products and coproducts, and both having factorisation systems (E, M) such that $E \subseteq Epi$ and $M \subseteq Mono$. Let Ω be a finite set (of observations), and I an (initialisation) object in $\mathcal{C}$, and assume that we have the adjoint situation between $\mathcal{C}$, $\mathcal{D}$, Set and functors as described at the bottom level of (3.14). Then the lower adjunctions lift to the upper two levels in (3.14) as shown in Sects. 3.3.3.2, 3.3.3.3 and 3.3.4, and we have the following abstract algorithms:*

Algo1 *Given an $F_{\mathcal{C}}$-coalgebra γ, compute $\overline{S}(reach(\overline{P}(\gamma)))$ which will be an observable $F_{\mathcal{C}}$-coalgebra.*

Algo2 *Given a $\mathcal{C}$-automaton (γ, i), compute $reach(\overline{S}'(reach(\overline{P}'(\gamma, i))))$, which will be a reachable and observable (i.e., minimal) $\mathcal{C}$-automaton.*

Of course, the abstract algorithms only become actual algorithms, when all structures involved have finite representations.

Concerning **Algo1**, we note that, in general, $\overline{S}(reach(\overline{P}(\gamma)))$ can be much larger than γ as the application of both $\overline{S}$ and $\overline{P}$ might yield some kind of completion of γ. However, if $\overline{P} \vdash \overline{S}$ is a dual equivalence, then $\overline{S}(reach(\overline{P}(\gamma)))$ is a maximal quotient of γ. All instances of **Algo1** contained in Bezhanishvili et al. (2012) and in this paper are of this form. In the general case, following Rot (2016), one can obtain the maximal quotient of γ by factoring a morphism from γ to the result $\overline{S}(reach(\overline{P}(\gamma)))$ of **Algo1**.

Also, when $\overline{P} \vdash \overline{S}$ is a dual equivalence (as in Bezhanishviliet al., 2012) the initial state is easily found back in the observable coalgebra resulting from **Algo1** as its language equivalence class, so the extension to **Algo2** seems almost trivial. In case $\overline{P} \vdash \overline{S}$ is not a full duality, the transformation of the initial state goes via the dual adjunction, and factorisation on the dual side. This is formalised in Theorem 3.1, and illustrated by the example in Sect. 3.3.1.

Brzozowski's algorithm and its generalisation to weighted automata in Sect. 3.4.2 are instances of **Algo2** as they use initial states. The classic Brzozowski algorithm is the case $\mathcal{C} = \mathcal{D} = \mathsf{Set}$, $G_{\mathcal{D}} = G$, and $\Omega = I = 1$. The set-based algorithm for weighted automata in Bonchi et al. (2014) is neither of the above algorithms, but it can be described as constructing $reach(\overline{U_{\mathcal{D}}}'\,\overline{P}'(\gamma, i))$, that is, reachability is computed over Set, and then dualise back (without going through $\mathcal{D} = \mathbb{S}\mathsf{Mod}$) to get a Set-based Moore automaton. As shown in Bonchi et al. (2014), this may result in the reachable part of the reversed automaton being infinite (cf. Example 8.3 of Bonchi et al. (2014)), whereas it might be finitely generated as a coalgebra/automaton over $\mathcal{D}$.

Remark 3 We end this section by observing that the requirements regarding products, coproducts and factorisation systems hold in all our examples, since $\mathcal{C}$ and $\mathcal{D}$ are monadic over Set meaning that they are equivalent to an Eilenberg-Moore category $\mathsf{EM}(T)$ for a Set-monad T. For such a category $\mathsf{EM}(T)$, we know that it is complete, cocomplete and exact (Borceux, 1994), Theorem 4.3.5. W.r.t factorisation systems, $(Epi, Mono)$ is generally not a factorisation system for $\mathsf{EM}(T)$, rather

$(RegEpi, Mono)$ is. Using the fact that regular epis are the surjective morphisms, and monos are the injective morphisms, one can prove that in $\mathsf{Coalg}_{\mathcal{C}}(F_{\mathcal{C}})$ and $\mathsf{Alg}_{\mathcal{D}}(G_{\mathcal{D}})$ the surjective and injective morphisms form a factorisation system.

3.4 Revisiting Examples

3.4.1 Deterministic Kripke Models

A central example from Bezhanishvili et al. (2012) are deterministic Kripke models (in *loc.cit* referred to as PODFAs, i.e., partially observable DFAs). We will first recall the definitions of deterministic Kripke models and their dual Boolean algebras with operators corresponding to a modal logic of tests. After that we will see how this duality can be seen as a special case of our general duality picture, which has as immediate corollary a minimisation algorithm for the case of finite models. In addition, results from Sect. 3.3.4 entail that the modal test language without propositional operators is sufficiently expressive to specify deterministic Kripke models up to bisimulation and to compute their observable quotient.

For the remainder of the section we fix an arbitrary finite set Σ of *actions* and an arbitrary finite set Ω of *observations*.

Definition 3.2 A *deterministic Kripke model* is a quintuple $\mathcal{S} = (S, \Sigma, \Omega, t : S \to S^{\Sigma}, f : S \to 2^{\Omega})$ where S is a set of *states*, t is a *transition function* and f is an *observation function*. A function $h : S_1 \to S_2$ is a morphism between Kripke models $(S_1, \Sigma, \Omega, t_1, f_1)$ and $(S_2, \Sigma, \Omega, t_2, f_2)$ if for all $s \in S_1$ and all $a \in \Sigma$ we have $h(t_1(s)(a)) = t_2(h(s))(a)$ and $f_1(s) = f_2(s)$. We write DKM for the category of deterministic Kripke models.

In other words, deterministic Kripke models are Kripke models where for each action $a \in \Sigma$ the corresponding relation is the graph of a (total) function. It is well-known that there is a duality between DKM and a suitable category BAO of Boolean algebras. We will now recall the definition of BAO and some known facts concerning this duality.

Definition 3.3 The category BAO of (deterministic) *Boolean algebras with operators* (BAOs) has as objects Boolean algebras B with the usual operations $\wedge$ and $\neg$ with a greatest element $\top$ and least element $\perp$ together with unary operators $(a) : B \to B$, for each action $a \in \Sigma$, such that (a) is a Boolean homomorphism. For each observation $\mathrm{o} \in \Omega$, we also have constants $\underline{\mathrm{o}}$. We denote an object of BAO by

$$\mathcal{B} = (B, \{(a) | a \in \Sigma\}, \{\underline{\mathrm{o}} | \mathrm{o} \in \Omega\}, \top, \wedge, \neg).$$

The BAO morphisms are the usual Boolean homomorphisms preserving, in addition, the constants and commuting with the unary operators. Finally, we denote by FBAO the category of *finite* Boolean algebras with operators.

The following fact is well-known (cf. e.g. Blackburn et al., 2001; Givant & Halmos, 2009).

Fact There is a dual adjunction between Set and BA as depicted below given by the contravariant functor $\mathbb{P}$ that maps a set to its Boolean algebra of subsets and the functor $\mathit{Uf} := \mathrm{Hom}(-, \mathbf{2})$, i.e., the contravariant functor the maps a Boolean algebra to its collection of ultrafilters. This adjunction restricts to a dual equivalence between the category FSet of finite sets and the category FBA of finite Boolean algebras.

$$
\begin{tikzcd}
\mathsf{Set}^{\mathsf{op}} \arrow[loop above, "F^{\mathsf{op}}"] \arrow[rr, bend left, "\mathbb{P}"] \arrow[rr, phantom, "\top"] & & \mathsf{BA} \arrow[loop above, "G_{\mathsf{BA}}"] \arrow[ll, bend left, "\mathit{Uf}"]
\end{tikzcd}
\qquad
\begin{array}{l}
F(X) = 2^{\Omega} \times X^{\Sigma} \\
G_{\mathsf{BA}}(X) = \Phi_{\mathsf{BA}}(\Omega) + \Sigma \cdot X \\
\mathit{Uf}(\Phi_{\mathsf{BA}}(\Omega)) \cong 2^{\Omega}
\end{array}
$$

We are now going to show how this example fits into our general framework. As a corollary we obtain a minimisation procedure for finite deterministic Kripke models.

Proposition 3.4 *We have the following equivalences:*

1. $\mathsf{DKM} \cong \mathsf{Coalg}_{\mathsf{Set}}(F)$ *for* $F = 2^{\Omega} \times X^{\Sigma}$
2. $\mathsf{BAO} \cong \mathsf{Alg}_{\mathsf{BA}}(G_{\mathsf{BA}})$ *for* $G_{\mathsf{BA}} = \Phi_{\mathsf{BA}}(\Omega) + \Sigma \cdot X$

Both equivalences are an immediate consequence of the definitions. In the sequel, we will make no distinction between F-coalgebras and deterministic Kripke models and, likewise, between G_{BA}-algebras and BAOs. As a consequence of the proposition we obtain the following duality results by applying our general framework.

Proposition 3.5 *The dual adjunction* $\mathit{Uf} \dashv \mathbb{P}$ *lifts to a dual adjunction between* DKM *and* BAO *and to an adjunction between* $\mathsf{Aut}_{\mathsf{Set}}^{1,2^{\Omega}}$ *and* $\mathsf{Aut}_{\mathsf{BA}}^{2,\Phi_{\mathsf{BA}}(\Omega)}$. *If we start with the dual equivalence* $\mathsf{FSet} \cong \mathsf{FBA}$, *both liftings are dual equivalences as well.*

Proof For the dual adjunction between DKM and BAO recall from Proposition 3.4 that both categories are equivalent to categories of F-coalgebras and G_{BA}-algebras for certain functors F and G_{BA}, respectively. Furthermore, we have $\mathit{Uf}(\Phi_{\mathsf{BA}}(\Omega)) \cong 2^{\Omega}$, which follows from the well-known fact that the set of homomorphisms of type $\Phi_{\mathsf{BA}}(\Omega) \to \mathbf{2}$ (i.e., ultrafilters) is in one-one correspondence with the set of functions of type $\Omega \to 2$. Therefore the functors F and G_{BA} have the shape required by our general lifting result from Sect. 3.3.3.2 and we obtain functors $\overline{\mathbb{P}} : \mathsf{Coalg}(F)^{\mathsf{op}} \to \mathsf{Alg}(G_{\mathsf{BA}})$ and $\overline{\mathit{Uf}} : \mathsf{Alg}(G_{\mathsf{BA}}) \to \mathsf{Coalg}(F)^{\mathsf{op}}$ with $\overline{\mathit{Uf}} \dashv \overline{\mathbb{P}}$.

To extend the adjunction $\overline{\mathit{Uf}} \dashv \overline{\mathbb{P}}$ between $\mathsf{Coalg}(F)$ and $\mathsf{Alg}(G_{\mathsf{BA}})$ further to a dual adjunction $\overline{\mathit{Uf}}' \dashv \overline{\mathbb{P}}'$ between $\mathsf{Aut}_{\mathsf{Set}}^{1,2^{\Omega}}$ and $\mathsf{Aut}_{\mathsf{BA}}^{\Phi_{\mathsf{BA}}(\Omega),2}$—the latter is a slight extension of the former by adding a initial state to deterministic Kripke models and adding an acceptance predicate to BAOs—it suffices to note that $\mathbb{P}1 \cong 2$ such that the result follows from the general theorem in Sect. 3.3.3.3.

The fact that the obtained adjunctions restrict to equivalences when we replace the base categories Set and BA with FSet and FBA, respectively, is a matter of routine checking.

This shows, in particular, that we get a duality between finite deterministic Kripke models and FBAO's. Proposition 3.5 is the key for obtaining a minimal realization via logical theories. Towards obtaining observable coalgebras via logical theories, we define a modal logic for DKMs.

Definition 3.4 Consider (cf. (3.8)) the language $\mathbb{L}(G_{\mathsf{BA}})$:

$$\varphi ::= \top \mid \hat{\mathrm{o}}, \mathrm{o} \in \Omega \mid [a]\varphi, a \in \Sigma \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi.$$

with 2-valued semantics defined as in (3.9) which corresponds to the usual semantics by identifying the predicate $[\![\varphi]\!] : S \to 2$ with the set $\{s \in S \mid [\![\varphi]\!](s) = 1\}$. For a given deterministic Kripke model $\mathcal{S} = (S, t, f)$ we say that a subset U of S is *definable by* $\mathbb{L}(G_{\mathsf{BA}})$ if $U = [\![\varphi]\!]$ for some $\varphi \in \mathbb{L}(G_{\mathsf{BA}})$. We let $\mathrm{Def}(\mathcal{S}) = (\mathrm{Def}(S), \{(a)_{\mathcal{S}}\}_{a\in\Sigma}, \{[\![\hat{\mathrm{o}}]\!]\}_{\mathrm{o}\in\Omega})$ be the BAO-subalgebra of $\overline{P}(\mathcal{S})$ based on the definable subsets of $\mathcal{S}$, where $(a)_{\mathcal{S}}([\![\varphi]\!]) = [\![[a]\varphi]\!]$.

In other words, the modal logic has (deterministic) Σ-indexed modalities and atomic propositions from Ω. It might seem strange that we introduced negation and indeed one does not need it, because one can get exactly the same Boolean algebra without having negation explicitly in the logic. This reflects the fact that for deterministic systems, unlike for non-deterministic ones, bisimulation and language equivalence coincide.

For a given DKM $\mathcal{S}$, the algebra $\mathrm{Def}(\mathcal{S})$ of definable subsets is a least BAO-subalgebra (=zero generated subalgebra) of the dual BAO $\overline{P}(\mathcal{S})$. By our definition in Sect. 3.3.5, $\mathrm{Def}(\mathcal{S})$ is clearly reachable, since it has no proper BAO-subalgebras. We instantiate the discussion in Sect. 3.3.5 further for DKMs. As is well-known, the Lindenbaum algebra for $\mathbb{L}(G_{\mathsf{BA}})$ is an initial BAO, and the subalgebra $\mathrm{Def}(\mathcal{S})$ is also the image of the Lindenbaum algebra in $\overline{P}(\mathcal{S})$ under the initial BAO-morphism. This image is obtained from the factorisation of the initial morphism in the factorisation system consisting of surjective and injective BAO-homomorphisms (cf. Remark 3).

Central to Bezhanishvili et al. (2012) was the result that the fragment of trace logic formulas is as expressive as the full modal logic.

Definition 3.5 A *trace logic formula* is a formula φ of the form $[a_1]\dots[a_n]\hat{\mathrm{o}}$ for some $\hat{o} \in \Omega$, $n \in \mathbb{N}$ and $a_i \in \Sigma$ for $i \in \{1, \dots, n\}$. For a DKM $\mathcal{S}$, we denote by $\mathrm{Def}^*(\mathcal{S})$ the Boolean subalgebra of $\overline{\mathbb{P}}(\mathcal{S})$ generated by subsets that are definable by a trace formula.

The expressiveness of trace logic was proven in the general setting in Proposition 3.1, and it is equivalent to the following statement:

$$\mathrm{Def}(\mathcal{S}) = \mathrm{Def}^*(\mathcal{S}) \quad \text{for all DKMs } \mathcal{S}. \tag{3.15}$$

Note that Eq. (3.15) can be seen as a normal form theorem for modal logics over DKMs. This result can also be obtained by observing that $\mathrm{Def}^*(\mathcal{S})$ is the image of $\overline{\Phi}_{\mathsf{BA}}(\Sigma^*\Omega)$, which is an initial BAO by Lemma 3.1. So both $\mathrm{Def}(\mathcal{S})$ and $\mathrm{Def}^*(\mathcal{S})$

are images of an initial object inside $\overline{P}(\mathcal{S})$, hence they must be equal, since initial objects are unique up to isomorphism.

We finish with a key observation from Bezhanishvili et al. (2012) that allows to compute quotients of finite DKMs via duality.

Corollary 3.1 *Given a finite DKM* $\mathcal{S}$*, the quotient of* $\mathcal{S}$ *modulo bisimulation can be effectively computed as* $\overline{Uf}(\mathrm{Def}^*(\mathcal{S}))$.

Proof By **Algo1** in Theorem 3.2 we have that $\overline{Uf}(\mathit{reach}(\overline{\mathbb{P}}(\mathcal{S}))$ and thus $\overline{Uf}(\mathrm{Def}^*(\mathcal{S}))$ are observable. As $\mathrm{Def}^*(\mathcal{S})$ is a BAO-subalgebra of $\overline{\mathbb{P}}(\mathcal{S})$ and as FSet and FBA are dually *equivalent*, we get that $\overline{Uf}(\mathrm{Def}^*(\mathcal{S}))$ is a quotient of $\overline{Uf}(\overline{\mathbb{P}}(\mathcal{S})) \cong \mathcal{S}$. Therefore, as $\overline{Uf}(\mathrm{Def}^*(\mathcal{S}))$ is an observable quotient of $\mathcal{S}$, it is the quotient modulo bisimulation of $\mathcal{S}$.

Remark 4 We note that the full duality between finite DKMs and finite BAOs, which was the basis of the minimisation-via-duality in Bezhanishvili et al. (2012), is not an instance of Theorem 3.2, since the category of finite Boolean algebras is not monadic over Set. **Algo1**, of course, applies to finite DKMs as they are just DKMs, and it will produce the same result as minimisation-via-duality from Bezhanishvili et al. (2012), since the full duality between finite sets and finite Boolean algebras is a restriction of the dual adjunction between Set and BA.

3.4.2 Weighted Automata

We need some basic definitions on semirings and semimodules to present the example of weighted automata. Recall that a *semiring* is a tuple $(\mathbb{S}, +, \cdot, 0, 1)$ where $(\mathbb{S}, +, 0)$ and $(\mathbb{S}, \cdot, 1)$ are monoids, the former of which is commutative, and multiplication distributes over finite sums:

$$r \cdot 0 = 0 = 0 \cdot r \qquad r \cdot (s + t) = r \cdot s + r \cdot t \qquad (r + s) \cdot t = r \cdot t + s \cdot t$$

We just write $\mathbb{S}$ to denote a semiring. Examples of semirings are: every field, the Boolean semiring 2, the semiring $(\mathbb{N}, +, \cdot, 0, 1)$ of natural numbers, and the tropical semiring $(\mathbb{N} \cup \{\infty\}, \min, +, \infty, 0)$. All these semirings are examples of *commutative* semirings, as the $\cdot$ operation is also commutative.

For a semiring $\mathbb{S}$, an $\mathbb{S}$-*semimodule* is a commutative monoid $(M, +, 0)$ with a left-action $\mathbb{S} \times M \to M$ denoted by juxtaposition rm for $r \in \mathbb{S}$ and $m \in M$, such that for every $r, s \in \mathbb{S}$ and every $m, n \in M$ the following laws hold:

$$\begin{aligned} (r + s)m &= rm + sm & \qquad r(m + n) &= rm + rn \\ 0m &= 0 & r0 &= 0 \\ 1m &= m & r(sm) &= (r \cdot s)m \end{aligned}$$

Every semiring $\mathbb{S}$ is an $\mathbb{S}$-semimodule, where the action is taken to be just the semiring multiplication. Semilattices are another example of semimodules (for the Boolean semiring $\mathbb{S}$).

An $\mathbb{S}$-semimodule homomorphism is a monoid homomorphism $h\colon M_1 \to M_2$ such that $h(rm) = rh(m)$ for each $r \in \mathbb{S}$ and $m \in M_1$. $\mathbb{S}$-semimodule homomorphisms are also called $\mathbb{S}$-*linear maps* or simply *linear maps*. The set of all linear maps from an $\mathbb{S}$-semimodule M_1 to M_2 is denoted by $\mathbb{S}\mathsf{Mod}(M_1, M_2)$.

Free $\mathbb{S}$-*semimodules* over a set X exist and can be built using the functor $\mathcal{V}_{\mathbb{S}}\colon \mathsf{Set} \to \mathsf{Set}$ defined on sets X and maps $h\colon X \to Y$ as follows:

$$\mathcal{V}_{\mathbb{S}}(X) \quad = \{\, \varphi\colon X \to \mathbb{S} \mid \varphi \text{ has finite support} \,\},$$

$$\mathcal{V}_{\mathbb{S}}(h)(\varphi) = \big(y \mapsto \textstyle\sum_{x \in h^{-1}(y)} \varphi(x)\big),$$

where a function $\varphi\colon X \to \mathbb{S}$ is said to have finite support if $\varphi(x) \neq 0$ holds only for finitely many elements $x \in X$. $\mathcal{V}_{\mathbb{S}}(X)$ is the free $\mathbb{S}$-semimodule on X when equipped with the following pointwise $\mathbb{S}$-semimodule structure:

$$(\varphi_1 + \varphi_1)(x) = \varphi_1(x) + \varphi_2(x) \qquad (s\varphi_1)(x) = s \cdot \varphi_1(x)\,.$$

We sometimes write the elements of $\mathcal{V}_{\mathbb{S}}(X)$ as formal sums $s_1x_1 + \cdots + s_nx_n$ with $s_i \in \mathbb{S}$ and $x_i \in X$. $\mathcal{V}_{\mathbb{S}}(X)$ is a monad and the category of Eilenberg-Moore algebras is $\mathbb{S}\mathsf{Mod}$, the category of $\mathbb{S}$-semimodules and $\mathbb{S}$-linear maps. As usual, free $\mathbb{S}$-semimodules enjoy the following universal property: for every function $h\colon X \to M$ from a set X to a semimodule M, there exists a unique linear map $h^\sharp\colon \mathcal{V}_{\mathbb{S}}(X) \to M$ that is called the *linear extension* of h.

We can define for an $\mathbb{S}$-semimodule M its *dual space* $M^\star$ to be the set $\mathbb{S}\mathsf{Mod}(M, \mathbb{S})$ of all linear maps between M and $\mathbb{S}$, endowed with the $\mathbb{S}$-semimodule structure obtained by taking pointwise addition and monoidal action: $(g + h)(m) = g(m) + h(m)$, and $(sh)(m) = s \cdot h(m)$. Note that $\mathbb{S} \cong \mathcal{V}_{\mathbb{S}}(1)$ and that $\mathbb{S}^\star = \mathbb{S}\mathsf{Mod}(\mathbb{S}, \mathbb{S}) \cong \mathbb{S}$.

3.4.2.1 Weighted Automata and Weighted Languages

A *weighted automaton* with finite input alphabet Σ and weights over a semiring $\mathbb{S}$ is given by a set of states X, a function $t\colon X \to \mathcal{V}_{\mathbb{S}}(X)^\Sigma$ (encoding the transition relation in the following way: the state $x \in X$ can make a transition to $y \in X$ with input $a \in \Sigma$ and weight $s \in \mathbb{S}$ if and only if $t(x)(a)(y) = s$), a final state function $f\colon X \to \mathbb{S}$ associating an output weight with every state, and an initial state function $i\colon 1 \to \mathcal{V}_{\mathbb{S}}(X)$. A diagrammatic representation is given in Fig. 3.1(a).

We see that a weighted automaton is an $F\mathcal{V}_{\mathbb{S}}$-coalgebra $\langle f, t\rangle\colon X \to \mathbb{S} \times (\mathcal{V}_{\mathbb{S}}X)^\Sigma$, where $F\colon \mathsf{Set} \to \mathsf{Set}$ is given by $F(X) = \mathbb{S} \times X^\Sigma$, together with an initialisation map $i\colon 1 \to X$.

The function $t\colon X \to \mathcal{V}_{\mathbb{S}}(X)^\Sigma$ can be inductively extended to words $w \in \Sigma^*$:

$$
\begin{tikzcd}
1 \arrow[dr, "i"'] & & \mathbb{S} \\
& X \arrow[ur, "f"'] \arrow[d, "t"] & \\
& \mathcal{V}_{\mathbb{S}}(X)^{\Sigma} &
\end{tikzcd}
$$

(a)

$$
\begin{tikzcd}
1 \arrow[dr, "i"'] & & \mathbb{S} \arrow[rr, dashed] & & \mathbb{S} \\
& X \arrow[ur, "f"] \arrow[r, "\eta"] \arrow[d, "t"] & V(X) \arrow[u, "f^{\sharp}"] \arrow[dl, "t^{\sharp}"] \arrow[r, dashed] & \mathbb{S}^{\Sigma^*} \arrow[ur, "o"] \arrow[d, "d"] & \\
& V(X)^{\Sigma} \arrow[rr, dashed] & & (\mathbb{S}^{\Sigma^*})^{\Sigma} &
\end{tikzcd}
$$

(b)

Fig. 3.1 (a) Weighted automata as Set-automata, and (b) their determinisation as $\mathbb{S}$Mod-automata

$$
\begin{aligned}
t(x)(\varepsilon) &= 1.x \\
t(x)(aw) &= v_1 t(x_1)(w) + \cdots + v_n t(x_n)(w), \text{ where } t(x)(a) = v_1 x_1 + \cdots + v_n x_n
\end{aligned}
$$

Weighted automata recognise functions in $\mathbb{S}^{\Sigma^*}$, or *formal power series over* $\mathbb{S}$. More precisely, the formal power series recognised by a weighted automaton $\mathcal{X} = (X, t, i, f)$ is the function $\mathcal{L}(\mathcal{X})\colon \Sigma^* \to \mathbb{S}$ that maps $w \in \Sigma^*$ to $f(t(i)(w)) \in \mathbb{S}$. More concretely, the value $\mathcal{L}(\mathcal{X})(w)$, for $w = a_1 a_2 \cdots a_n$, is the sum of all $v_1 \cdot \ldots \cdot v_n \cdot f(x_{n+1})$ over all paths $p_w = x_1 \xrightarrow{a_1, v_1} \ldots \xrightarrow{a_n, v_n} x_{n+1}$ labelled by w.

Observe that $\mathbb{S}$ is (isomorphic to) the carrier of the free Eilenberg-Moore $\mathcal{V}_{\mathbb{S}}$-algebra on one generator $\mathcal{V}_{\mathbb{S}}(1)$. Hence, as described in Sect. 3.2.2, we can determinise a weighted automaton $\mathcal{X}$ into a Moore automaton $\mathcal{X}^{\sharp}$ in $\mathbb{S}$Mod. More precisely, letting $\mathcal{X} = (X, t, i, f)$ be a weighted automaton, we determinise its coalgebraic part $\langle f, t\rangle\colon X \to \mathbb{S} \times \mathcal{V}_{\mathbb{S}}(X)^{\Sigma}$ into $\langle f^{\sharp}, t^{\sharp}\rangle\colon \mathcal{V}_{\mathbb{S}}(X) \to \mathbb{S} \times \mathcal{V}_{\mathbb{S}}(X)^{\Sigma}$ and take as initialisation morphism $\mathcal{V}_{\mathbb{S}}(i)\colon \mathcal{V}_{\mathbb{S}}(1) \to \mathcal{V}_{\mathbb{S}}(X)$. The result is an $\mathbb{S}$Mod-automaton $\mathcal{X}^{\sharp} = (\mathcal{V}_{\mathbb{S}}(X), t^{\sharp}, \mathcal{V}_{\mathbb{S}}(i), f^{\sharp})$ with initialisation in $\mathbb{S} \cong \mathcal{V}_{\mathbb{S}}(1)$ and output in $\mathbb{S}$. We view such automata as Moore automata over $\mathbb{S}$Mod. The construction is illustrated in Fig. 3.1b.

The unique map from the determinised Moore automaton into the final Moore automaton of weighted languages gives the language semantics of weighted automata described concretely above. In $\mathbb{S}$Mod, the value of $\mathcal{L}(\mathcal{X})(w)$ can be computed using the usual matrix representation of linear maps: the initialisation morphism $\mathcal{V}_{\mathbb{S}}(i)$ corresponds to a column vector $\eta \circ i\colon \mathcal{V}_{\mathbb{S}}(X)$, the output morphism $f^{\sharp}$ is a row vector, and the transition morphism $t^{\sharp}$ can be represented as a Σ-indexed collection of $X \times X$-matrices t_a where $t_a(y, x) = t(x)(a)(y)$ for all $x, y \in X$. $\mathcal{L}(\mathcal{X})(w)$ is then obtained by the following matrix multiplication $f \times t_{a_n} \times \cdots \times t_{a_0} \times i$.

3.4.2.2 Brozowski's Algorithm for Weighted Automata

There is self-dual adjunction of $\mathbb{S}$Mod obtained by taking dual space: $(-)^{\star} = \mathbb{S}\mathsf{Mod}(-, \mathbb{S})$. A special case is the self-dual adjunction of vector spaces in case $\mathbb{S}$ is a field, which restricts to a duality between finite-dimensional vector

$$
\begin{tikzcd}
\left(\mathsf{Aut}^{\mathbb{S},\mathbb{S}}_{\mathbb{S}\mathsf{Mod}}\right)^{\mathsf{op}} \arrow[r, bend left, "\overline{(-)^{\star}}'"] \arrow[r, phantom, "\top"] \arrow[d] & \mathsf{Aut}^{\mathbb{S},\mathbb{S}}_{\mathbb{S}\mathsf{Mod}} \arrow[l, bend left, "\overline{(-)^{\star}}'"] \arrow[d] \\
\mathbb{S}\mathsf{Mod}^{\mathsf{op}} \arrow[r, bend left, "(-)^{\star}"] \arrow[r, phantom, "\top"] \arrow[loop below, "F_{\mathbb{S}\mathsf{Mod}}"] & \mathbb{S}\mathsf{Mod} \arrow[l, bend left, "(-)^{\star}"] \arrow[loop below, "G^{\mathsf{op}}_{\mathbb{S}\mathsf{Mod}}"]
\end{tikzcd}
$$

(a)

$$
\begin{tikzcd}
\mathsf{WAut} \text{ over } \mathsf{Set} \arrow[d, "(-)^{\sharp}"] \arrow[dr, bend left] & \\
\mathsf{Aut}^{\mathbb{S},\mathbb{S}}_{\mathbb{S}\mathsf{Mod}} \arrow[r, "\overline{(-)^{\star}}'"] & \left(\mathsf{Aut}^{\mathbb{S},\mathbb{S}}_{\mathbb{S}\mathsf{Mod}}\right)^{\mathsf{op}} \arrow[d, "reach^{\mathsf{op}}"] \\
\mathsf{Aut}^{\mathbb{S},\mathbb{S}}_{\mathbb{S}\mathsf{Mod}} \arrow[d, "reach"'] & \left(\mathsf{Aut}^{\mathbb{S},\mathbb{S}}_{\mathbb{S}\mathsf{Mod}}\right)^{\mathsf{op}} \arrow[l, "\overline{(-)^{\star}}'"'] \\
\mathsf{Aut}^{\mathbb{S},\mathbb{S}}_{\mathbb{S}\mathsf{Mod}} &
\end{tikzcd}
$$

(b)

Fig. 3.2 **a** The dual adjunction for weighted automata, and **b** Brzozowski's algorithm for weighted automata

spaces. This duality was used in Bezhanishvili et al. (2012) to obtain observable Moore automata over vector spaces.

We lift the base adjunction to one between Moore automata in $\mathbb{S}\mathsf{Mod}$ using Theorem 3.1. Let $\mathcal{C} = \mathcal{D} = \mathbb{S}\mathsf{Mod} = \mathsf{EM}(\mathcal{V}_{\mathbb{S}})$ and $F_{\mathbb{S}\mathsf{Mod}}(X) = \mathbb{S} \times X^{\Sigma}$ and $G_{\mathbb{S}\mathsf{Mod}}(X) = \mathbb{S} + \Sigma \cdot X$. Since $\mathbb{S}^{\star} \cong \mathbb{S}$, the conditions for Theorem 3.1 hold, and the adjunction lifts, as illustrated in Fig. 3.2a.

We can now give the Brzozowski algorithm for weighted automata by instantiating **Algo2** of Theorem 3.2 for the determinised automaton. Start with a weighted automaton in Set, determinise it into a Moore automaton in $\mathsf{Aut}^{\mathbb{S},\mathbb{S}}_{\mathbb{S}\mathsf{Mod}}$ (to have a canonical representative of the accepted language), reverse and determinise, take the reachable part (w.r.t $G_{\mathbb{S}\mathsf{Mod}}$-structure over $\mathbb{S}\mathsf{Mod}$), reverse and determinise, take the reachable part again. Diagramatically, **Algo2** is (putting $^{\mathsf{op}}$ on the right-hand side to start and end in $\mathsf{Aut}^{\mathbb{S},\mathbb{S}}_{\mathbb{S}\mathsf{Mod}}$) shown in Fig. 3.2b.

At this point we have built a minimal Moore automaton $min(\mathcal{X}^{\sharp})$ over $\mathbb{S}\mathsf{Mod}$ accepting the same language as the weighted automaton $\mathcal{X}$ we started with and, moreover, the state space is a subsemimodule of the semimodule generated by the original state space.

The last step missing is to recover a weighted automaton over Set with a state space Y such that $\mathcal{V}_{\mathbb{S}}(Y)$ is the state space of $min(\mathcal{X}^{\sharp})$. Unfortunately, subsemimodules of free, finitely generated semimodules are not necessarily free and finitely generated. Therefore our construction does not guarantee, in general, that the resulting automaton $min(\mathcal{X}^{\sharp})$ corresponds to a weighted automaton in Set. Fortunately, we know from a result of Tan (2016) that for a commutative semiring $\mathbb{S}$, every nonzero subsemimodule N of a finitely generated free $\mathbb{S}$-semimodule M is free if and only if $\mathbb{S}$ is a principal ideal domain (Tan, 2016), Theorem 4.3. Furthermore, because N is free, it follows that it is also finitely generated and of rank smaller than that of M (Tan, 2016), Theorem 4.3. In other words, the minimal weighted automaton over a principal ideal domain exists and has a state space smaller or equal than that of the original automaton if the latter is finite.

Recall that a principal ideal domain is an integral domain in which every ideal is principal, i.e., can be generated by a single element. Examples include any Euclidean domain, thus any field, the ring of integers, the ring of polynomials in one variable with coefficients in a field, and the ring of formal power series over a field and one variable. The ring of polynomials in two or more variables and the ring of polynomials with integer coefficients are not principal ideal domains.

3.4.2.3 Logic for Weighted Automata

The $\mathbb{S}$-valued logic (cf. (3.8)) corresponding to the functor $G_{\mathbb{S}\mathsf{Mod}}(X) = \mathcal{V}_{\mathbb{S}}(1) + \Sigma \cdot X$ (recall that $\mathbb{S} \cong \mathcal{V}_{\mathbb{S}}(1)$, i.e., $\Omega = 1$), has formulas generated by the following grammar:

$$\varphi \ ::= \ \downarrow \mid [a]\varphi,\ a \in \Sigma \mid 0 \mid s \cdot \varphi,\ s \in \mathbb{S} \mid \varphi + \varphi$$

where $\downarrow$ is a single atomic proposition (denoting termination) and the linear propositional connectives are interpreted via semimodule structure. Trace logic is the fragment that is built only from $\downarrow$ and modalities. The results from Sect. 3.3.4 tell us that trace logic is already expressive for $\mathbb{S}\mathsf{Mod}$-automata.

3.4.3 Topological Automata via Gelfand Duality

A very popular model heavily used in reforcement learning is the *partially observable Markov decision process* (POMDP). The idea is that one can only see the observations and not exactly which state the system is in. Many algorithms in machine learning deal with this situation by constructing a new automaton called the *belief automaton*. The state space of this automaton is the set of probability distributions over the original states. When seeking to minimize this using duality (Bezhanishvili et al., 2012), the original idea was to exploit the fact that the state space of the belief automaton is a compact Hausdorff space and use Gelfand duality. However, we have since felt that convex duality is a better match for this situation. Nevertheless, the notion of a topological automaton is interesting in its own right and may be the basis for later extensions and examples. This section, therefore develops Gelfand duality and its application to topological automata.

Given a finite set X, we write $\mathcal{D}_{\leq 1}(X)$ for the set of discrete *sub*distributions on X endowed with the relative topology when viewed as a subset of $[0, 1]^X$. This is a compact Hausdorff space. As in other sections, we fix a finite set Σ of actions or inputs, and a finite set Ω of observations.

Definition 3.6 A **compact Hausdorff automaton** is a 5-tuple

$$\mathcal{H} = (S, t : S \to S^{\Sigma}, f : S \to \mathcal{D}_{\leq 1}(\Omega), i : S)$$

where S is a compact Hausdorff space, t is a *continuous* transition function (to the product space S^{Σ}), f is a *continuous* observation function, and $i : S$ is an initial state.

A compact Hausdorff automaton is easily seen to be a coalgebra for the functor $F: \mathsf{KHaus} \to \mathsf{KHaus}$ given by $F(X) = \mathcal{D}_{\leq 1}(\Omega) \times X^{\Sigma}$ together with an initialisation morphism $i: 1 \to S$ where 1 is the discrete, one-element space. Since Gelfand duality is a full duality, the initial state plays only a minor role as mentioned in Sect. 3.3.6. In Bezhanishvili et al. (2012), compact Hausdorff automata were defined without initial state.

We recall a few basic facts about C^*-algebras, and refer to Arveson (1976); Johnstone (1982) for further information. Usually C^*-algebras are considered as algebras over the complex field. Here, we are concerned with probabilistic computation, and therefore we consider C^*-algebras over the field $\mathbb{R}$ of reals.

A *(real-valued) Banach algebra* A is Banach space (complete normed real vector space) equipped with an associative multiplication such that $\|xy\| \leq \|x\|\|y\|$ for all x, y. This requirement makes multiplication continuous in the norm topology. A *(real)* C^**-algebra* is a Banach algebra together with an *involution* $(-)^*$ which is a linear, norm-preserving map on A such that $(xy)^* = y^*x^*$ and $(x^*)^*$, and which in addition satisfies the C^*-axiom: $\|x^*x\| = \|x\|^2$ for all $x \in A$. A C^*-algebra A is *unital* if it has a multiplicative unit 1 whose norm is $1 \in \mathbb{R}$, and A is *commutative* if the multiplication is commutative.

A *homomorphism of* C^**-algebras* is a bounded, linear map that preserves the multiplication and the involution. A homomorphism of unital C^*-algebras is additionally required to preserve the unit. We denote by $\mathsf{CUC^*Alg}$ the category of unital, commutative, real-valued C^*-algebras and their homomorphisms.

In Negrepontis (1971) it was shown[1] that the unit interval functor $U: \mathsf{CUC^*Alg} \to \mathsf{Set}$, which sends an $A \in \mathsf{CUC^*Alg}$ to $\{a \in A \mid 0 \leq a \leq 1\}$, has a left adjoint $M: \mathsf{Set} \to \mathsf{CUC^*Alg}$ given by

$$M(X) = C([0,1]^X) = \{f: [0,1]^X \to \mathbb{R} \mid f \text{ continuous}\} \quad (3.16)$$

$$M(g: X \to Y) = f(v \circ g) \quad \text{where } f \in C([0,1]^X), v \in [0,1]^Y. \quad (3.17)$$

where $[0, 1]^X$ is equipped with the product topology.

[1] Strictly speaking, she showed it for complex-valued C^*-algebras, but the result also holds for real-valued ones.

We denote by KHaus the category of compact Hausdorff spaces and continuous maps. Given a compact Hausdorff space X, the hom-set $C(X) = \mathrm{Hom}_{\mathsf{KHaus}}(X, \mathbb{R})$ becomes a commutative, unital, real-valued C^*-algebra by defining operations pointwise. In particular, the unit is the constantly 1 map, and for $f \in C(X)$, the norm is $\|f\| = \sup\{|f(x)| \mid x \in X\}$; recall that for a compact space and a continuous function the supremum is attained. For a morphism $g \colon X \to Y$ in KHaus, defining $C(g)(h) = h \circ g$ makes $C(-)$ a functor from KHaus to $\mathsf{CUC^*Alg}^{\mathsf{op}}$.

Conversely, for $A \in \mathsf{CUC^*Alg}$, the set $\hat{A} = \mathrm{Hom}_{\mathsf{CUC^*Alg}}(A, \mathbb{R})$ becomes a compact Hausdorff space (called the *spectrum of* A) by equipping it with the weak *-topology τ which is generated by the sets $O_x = \{\Phi \in \hat{A} \mid \Phi(x) \neq 0\}$ for all $x \in A$. We define $\mathit{Spec}(A) = (\hat{A}, \tau)$. For a morphism $h \colon A \to B$ in $\mathsf{CUC^*Alg}$, defining $\mathit{Spec}(h)(\Phi) = \Phi \circ h$ makes Spec a functor from $\mathsf{CUC^*Alg}^{\mathsf{op}}$ to KHaus.

The functors C and Spec establish a dual equivalence between KHaus and $\mathsf{CUC^*Alg}$ known as Gelfand duality

$$\begin{tikzcd} \mathsf{KHaus}^{\mathsf{op}} \arrow[r, bend left, "C"] \arrow[r, phantom, "\cong"] & \mathsf{CUC^*Alg} \arrow[l, bend left, "\mathit{Spec}"] \end{tikzcd} \tag{3.18}$$

For the purposes of this paper, we only need a dual adjunction. We will take C to be the right adjoint. As this dual adjunction is in fact a dual equivalence, the unit and the counit of this adjunction are natural isomorphisms. The unit $\eta_A \colon A \to C(\mathit{Spec}(A))$ is known as the *Gelfand transform*, and is given by $\eta_A(x)(\Phi) = \Phi(x)$. For all $A \in \mathsf{CUC^*Alg}$, η_A is an isometric isomorphism in $\mathsf{CUC^*Alg}$.

We first lift the base dual adjunction between KHaus and $\mathsf{CUC^*Alg}$ to an a dual ajdunction between the category of F-coalgebras and G-algebras for the functors

$$\begin{array}{ll} F \colon \mathsf{KHaus} \to \mathsf{KHaus}, & F(X) = \mathcal{D}_{\leq 1}(\Omega) \times X^{\Sigma} \\ G \colon \mathsf{CUC^*Alg} \to \mathsf{CUC^*Alg}, & G(A) = M(\Omega)/J + \Sigma \cdot A \end{array} \tag{3.19}$$

Recall from (3.16) that M is the left adjoint of the unit interval functor U. Finally, $J \subseteq M(\Omega)$ is an ideal of the $\mathsf{CUC^*Alg}$-algebra $M(\Omega)$ which we describe in a moment. Note that $\mathsf{CUC^*Alg}$ has coproducts. This follows from the fact that KHaus has products and using Gelfand duality.

In order to lift the base dual adjunction $\mathit{Spec} \dashv C$ to a dual adjunction between $\mathsf{Coalg}(F)$ and $\mathsf{Alg}(G)$ as in Sect. 3.3.3.2, we need to show that $\mathit{Spec}(M(\Omega)/J) \cong \mathcal{D}_{\leq 1}(\Omega)$. First, we define the ideal J. Fix a finite set Y and consider the C^*-algebra $M(Y) \in \mathsf{CUC^*Alg}$ defined by $[0, 1]^Y$. For each $y \in Y$, we have a projection map $\pi_y \in M(Y) \to \mathbb{R}$ given by $\pi_y(v) = v(y)$. Let $\pi = \sum_{y \in Y} \pi_y$. Then $\pi \colon [0, 1]^Y \to \mathbb{R}$ is linear and $\pi \in M(Y)$. We will take J to be the ideal corresponding to the congruence generated by the equality obtained by rewriting $\pi \preceq 1$ as an equality as follows:

$$\pi \preceq 1 \iff \pi \vee 1 = 1 \iff \tfrac{1}{2}(\pi + 1) + \tfrac{1}{2}|\pi + 1| = 1 \iff |1 - \pi| = 1 - \pi$$

Definition 3.7 We define the ideal J of $M(Y)$ as the principal ideal generated by the element $(|\pi^-| - \pi^-)$ where $\pi^- := 1 - \pi$. That is,

$$J = \{m \in M(Y) \mid \exists k \in M(Y) : m = k(|\pi^-| - \pi^-)\}.$$

The congruence relation $\equiv_J$ on $M(Y)$ arising from the ideal J is then defined standardly as follows: For $m, n \in M(Y)$, $m \equiv_J n$ if $m - n \in J$. We write $M(Y)/J$ for the quotient of $M(Y)$ with respect to $\equiv_J$.

Due to space limitations, we omit the rather technical proof.

Lemma 3.3 *For any set Y, $\mathcal{D}_{\leq 1}(Y) \cong Spec(M(Y)/J)$ in* KHaus.

From Lemma 3.3 and Sect. 3.3.3.2, it follows that the base dual adjunction lifts to one between F-coalgebras and G-algebras. This adjunction in turn can be easily lifted to one between automata using Theorem 3.1 from Sect. 3.3.3.3. The categories of automata are here the category $\mathsf{CHA} = \mathsf{Aut}^{1,\mathcal{D}_{\leq 1}(\Omega)}_{\mathsf{KHaus}}$ of compact Hausdorff automata and the category $\mathsf{CAO} = \mathsf{Aut}^{M(Obs)/J,\mathbb{R}}_{\mathsf{CUC^*Alg}}$ of $\mathsf{CUC^*Alg}$-automata with initialisation in $M(Obs)/J$ and output in $\mathbb{R} \cong C(1)$.

$$\begin{tikzcd}
\mathsf{CHA}^{\mathsf{op}} \arrow[r, bend left, "\overline{C}'"] \arrow[r, phantom, "\top"] \arrow[d] & \mathsf{CAO} \arrow[l, bend left, "\overline{Spec}'"] \arrow[d] \\
\mathsf{KHaus}^{\mathsf{op}} \arrow[r, bend left, "C"] \arrow[r, phantom, "\top"] & \mathsf{CUC^*Alg} \arrow[l, bend left, "Spec"]
\end{tikzcd}$$

The abstract algorithms **Algo1** and **Algo2** apply since KHaus and $\mathsf{CUC^*Alg}$ are monadic over Set (cf. Sect. 3.3.6). In particular, KHaus is the Eilenberg-Moore category of the ultrafilter monad (Manes, 1969). In order to show that the associated trace logic is expressive we need an extra argument, since the functor F defined in (3.19) does not have the shape required by Lemma 3.1 and Theorem 3.2. However, we can apply Remark 2 after observing the following. Let $F' := Spec(M(\Omega)) \times (-)^{\Sigma}$. Then the associated natural isomorphism $\xi^{\mathsf{trc}'} \colon F' Spec M \Rightarrow Spec M G$ specifies semantics of trace logic over F'-coalgebras. To obtain a suitable $\tau \colon F \Rightarrow F'$ note that quotienting with J in $\mathsf{CUC^*Alg}$ yields an epi $e : M(O) \twoheadrightarrow M(O)/J$ from which we get a mono $Spec(e) : Spec(M(O)/J) \rightarrowtail Spec(M(O))$ in KHaus. Pre-composing $Spec(e)$ with the isomorphism $h : \mathcal{D}_{\leq 1}(O) \xrightarrow{\sim} Spec(M(O)/J)$ given by Lemma 3.3 and defining $\tau := (Spec(e) \circ h) \times id$, it follows that $\tau \colon F \Rightarrow F'$ has all components mono in KHaus. It now follows that trace logic is also expressive for F-coalgebras, i.e., for compact Hausdorff automata.

Remark 5 In order to view Gelfand duality (3.18) as a concrete duality obtained from a dualising object, we need to expand the setting a bit, since $\mathbb{R}$ is not a compact Hausdorff space. This can be done by considering the dual adjunction between locally compact Hausdorff spaces and not-necessarily unital commutative C^*-algebras. Gelfand duality is a restriction of this dual adjunction.

3.5 Alternating Automata

Alternating finite automata (aka *Boolean automata* or *parallel automata*) were first studied in Chandra & Stockmeyer (1976); Kozen (1976); Chandra et al. (1981); Leiss (1981); Brzozowski & Leiss (1980) as a finite-state analog of alternating Turing machines (Chandra et al., 1981). Let Σ be a fixed finite *input alphabet*. An *alternating finite automaton* (AFA) over Σ is a tuple $\mathcal{A} = (X, i, t, F)$, where

- X is a finite set of *states*,
- $F \subseteq X$ are the *final states*,
- $t : \Sigma \to X \to 2^X \to 2$ is the *transition function*, and
- $i : 2^X \to 2$ is the *acceptance condition*.

Intuitively, the machine $\mathcal{A}$ operates as follows. Let $k = |X|$. Initially k processes are started, each assigned to a different state, reading the first symbol of the input word $w \in \Sigma^*$. In each step, a process at state s reads the next input symbol a and spawns k child processes, each of which moves to a different state and continues in the same fashion, while the parent process at s waits for the child processes to report back a Boolean value. In this way a k-branching computation tree is generated. When the end of the input word is reached, a process at state s reports 1 back to its parent if $s \in F$, 0 otherwise. A non-leaf process waiting at state s, having read input symbol a, collects the k-tuple $b \in 2^X$ of Boolean values reported by its children, computes $tasb$, and reports that Boolean value back to its parent. When the initial processes have all received values, say $c \in 2^X$, the machine accepts if $ic = 1$, otherwise it rejects.

Alternating automata accept all and only regular sets. It was shown in Kozen (2006) by combinatorial means that a language $L \subseteq \Sigma^*$ is accepted by a k-state AFA iff its reverse $\{w^R \mid w \in L\}$ is accepted by a 2^k state deterministic finite automaton (DFA).

Our purpose in this section is to recast this result in the framework of our general duality principle. The duality involves the category CABA of *complete atomic Boolean algebras* and the category Set of *discrete spaces*, which underlie *powerset Boolean algebras*.

3.5.1 **CABA, EM(N), *and* Set$^{\mathsf{op}}$**

3.5.1.1 CABA

A *complete Boolean algebra* (CBA) is a structure $(B, \neg, \bigvee, \bigwedge, 0, 1, \leq)$, where B is a set, $\neg$ is a unary operation on B, $\bigvee$ and $\bigwedge$ are infinitary operations on the powerset of B, 0 and 1 are constants, and $\leq$ is a partial order on B, such that

- $(B, \neg, \vee, \wedge, 0, 1, \leq)$ is an ordinary Boolean algebra (BA), where $\vee$ and $\wedge$ are the restrictions of $\bigvee$ and $\bigwedge$, respectively, to two-element sets; and
- $\bigvee A$ and $\bigwedge A$ give the supremum and infimum of A, respectively, with respect to $\leq$.

The CBA-morphisms are BA-homomorphisms that preserve $\bigvee$ and $\bigwedge$.

An *atom* of a BA is a $\leq$-minimal nonzero element. A BA is *atomic* if every nonzero element has an atom $\leq$-below it. A *complete atomic Boolean algebra* (CABA) is an atomic CBA. A CABA-morphism is just a CBA-morphism. Together, CABAs and their morphisms form the category CABA.

It is known that every CABA is isomorphic to the powerset Boolean algebra on its atoms, thus every element is the supremum of the atoms below it. CBAs and CABAs satisfy infinitary de Morgan and distributive laws:

$$\neg \bigvee a = \bigwedge \{\neg x \mid x \in a\} \qquad (\bigvee a) \wedge x = \bigvee \{y \wedge x \mid y \in a\}$$
$$\neg \bigwedge a = \bigvee \{\neg x \mid x \in a\} \qquad (\bigwedge a) \vee x = \bigwedge \{y \vee b \mid y \in a\}$$

as well as other useful infinitary properties such as commutativity, associativity, and idempotence of $\bigvee$ and $\bigwedge$. The free CABA on generators X is the powerset CABA $(2^{2^X}, \bigcup, \bigcap, \sim, \emptyset, 2^X)$. See, e.g., Givant & Halmos (2009) for further information on the theory of CBAs and CABAs.

3.5.1.2 EM(N)

The self-dual adjunction $\mathcal{Q}^{\mathsf{op}} \dashv \mathcal{Q}$ of the contravariant powerset functor (Example 3.1) gives rise to a Set-monad $N = \mathcal{Q} \circ \mathcal{Q}^{\mathsf{op}}$, where for X a set and $f : X \to Y$ a set function,

$$NX = \mathcal{Q}\mathcal{Q}^{\mathsf{op}} X = 2^{2^X} \qquad Nf = (f^{-1})^{-1} : 2^{2^X} \to 2^{2^Y}$$

The unit and multiplication are

$$\eta_X(x) = \{a \mid x \in a\}, \qquad \mu_X(H) = \{a \mid \eta_{\mathcal{Q}X}(a) \in H\} = \eta^{-1}_{\mathcal{Q}X}(H).$$

This is called the *double powerset* or *neighbourhood monad*. The category of Eilenberg-Moore algebras of N is denoted $\mathsf{EM}(N)$.

3.5.1.3 Equivalence of CABA, Set$^{\mathsf{op}}$, and EM(N)

It is known that the Eilenberg-Moore algebras of the double powerset monad N are exactly the CABAs. These two categories are also dually equivalent to Set, that is, equivalent to Set$^{\mathsf{op}}$, as observed in Taylor (2002).

The equivalence of the three categories can be shown via the composition of three faithful functors that are injective on objects:

$$\mathsf{Set}^{\mathsf{op}} \xrightarrow{J} \mathsf{EM}(N) \xrightarrow{D} \mathsf{CABA} \xrightarrow{At} \mathsf{Set}^{\mathsf{op}}. \tag{3.20}$$

Here J is the Eilenberg-Moore comparison functor (Adámek et al., 2009; Mac Lane, 1971). Concretely, J sends a set X to the CABA 2^X and a function $f : X \to Y$ to its inverse image map. That is, $J = \mathsf{Set}(-, 2)$ with Boolean structure. The functor At takes a CABA to its set of atoms and a CABA-morphism $f : A \to B$ to $At\, f : At\, B \to At\, A$, where $At\, f\,(b)$ is the unique atom a of A such that $\uparrow a = f^{-1}(\uparrow b)$ and $\uparrow a$ and $\uparrow b$ are the principal ultrafilters on atoms a and b, respectively. In a CABA, there is a bijection between principal ultrafilters and atoms, and we have that $At \cong \mathsf{CABA}(-, 2)$. In other words, the equivalence given by J and At is a concrete duality with dualising object 2.

Although the equivalence between EM(N) and CABA is fairly well known, the details are rarely provided. We therefore describe the functor D that produces a CABA from an EM(N)-algebra (X, α). Let TX be the term monad for CABA terms over indeterminates X.[2] Let $D(X, \alpha) = (X, D\alpha)$, where

$$D\alpha : TX \to X \qquad\qquad D\alpha = \alpha \circ (\tau N \circ T\eta)_X, \tag{3.21}$$

where $T\eta_X : TX \to TNX$ substitutes $\eta_X(x)$ for $x \in X$ in a term and $\tau_{NX} : TNX \to NX$ is the evaluation map of the powerset CABA $(2^{2^X}, \bigcup, \bigcap, \sim, \emptyset, 2^X)$. In particular (and in more conventional notation), this gives the following definitions of the Boolean operations:

$$\begin{gathered}
\textstyle\bigvee_n x_n = \alpha(\bigcup_n \eta_X(x_n)) \qquad \bigwedge_n x_n = \alpha(\bigcap_n \eta_X(x_n)) \\
\neg x = \alpha(\sim\eta_X(x)) \qquad 0 = \alpha(\emptyset) \qquad 1 = \alpha(2^X).
\end{gathered} \tag{3.22}$$

The action of D on morphisms is the identity.

The natural transformation $\tau N \circ T\eta : T \to N$ in (3.21) relating CABA terms and double powerset is invertible up to CABA equivalence. Consider the natural transformation

[2] TX consists of CBA terms with the arity of the infinitary operations bounded by $2^{2^{|X|}}$. There can be no such bound for CBA in general, as there are CBAs of arbitrarily large cardinality generated by X; thus there is no term monad for CBA. However, CABAs generated by X are of cardinality at most double exponential in $|X|$, and we can bound arities accordingly.

$$\upsilon : N \to T \qquad \upsilon_X(A) = \bigvee_{a \in A} \left(\bigwedge_{x \in a} x \wedge \bigwedge_{x \notin a} \neg x \right), \quad A \in 2^{2^X}.$$

It can be shown that

$$\tau N \circ T\eta \circ \upsilon = id_N \qquad \upsilon \circ \tau N \circ T\eta \equiv id_T.$$

By the latter we mean that for any term $\theta \in TX$, $\upsilon_X(\tau_{NX}(T\eta_X(\theta))) \equiv \theta$ modulo the axioms of CABA. This essentially says that there is a disjunctive normal form for CABA terms.

3.5.2 Language Acceptance of Alternating Automata

Let $\mathcal{A} = (X, t, f, i)$ be an AFA with states X and components

$$i : 1 \to 2^{2^X} \qquad t_a : X \to 2^{2^X}, \; a \in \Sigma \qquad f : X \to 2$$

where i is the (transposed) acceptance condition, t_a are the transitions, and $f : X \to 2$ is the characteristic function for the subset F of accepting states.

The language accepted by $\mathcal{A}$ is $\mathcal{L}(\mathcal{A}) \triangleq \{w \in \Sigma^* \mid i(t'_w(F)) = 1\}$, where

$$t'_w : 2^X \to 2^X \qquad t'_\varepsilon(A) = A \qquad t'_{aw}(A)(s) = t_a(s)(t'_w(A)).$$

As constructed in Kozen (2006), the associated DFA for the reverse language is $\mathcal{A}'$ with states 2^X and components

$$f^\flat : 1 \to 2^X \qquad t^\flat_a : 2^X \to 2^X, \; a \in \Sigma \qquad i^\flat : 2^X \to 2.$$

This is a deterministic automaton, that is, a coalgebra for the functor $F = 2 \times (-)^\Sigma$ with start state $f^\flat$, transitions $t^\flat_a$, and accept states $i^\flat$. The language accepted by $\mathcal{A}'$ is $\mathcal{L}(\mathcal{A}') \triangleq \{w \in \Sigma^* \mid i^\flat(t^\flat_w(f^\flat)) = 1\}$, where

$$t^\flat_\varepsilon = id_{2^X} \qquad t^\flat_{wa} = t^\flat_a \circ t^\flat_w.$$

The combinatorial construction of Kozen (2006) amounts to recurrying the components of the automata. Denoting the reverse of a string w by w^R and using the fact that $t^\flat_a = t'_a$, it can be shown inductively that $t^\flat_w = t'_{w^R}$, therefore the language accepted by $\mathcal{A}'$ is the reverse of the language accepted by $\mathcal{A}$:

$$\mathcal{L}(\mathcal{A}') = \{w \mid i^\flat(t^\flat_w(f^\flat)) = 1\} = \{w \mid i(t'_{w^R}(f)) = 1\} = \{w \mid w^R \in \mathcal{L}(\mathcal{A})\}.$$

3.5.3 Alternating Automata as $\mathsf{EM}(N)$-Automata

We now show how the relationship between $\mathcal{A}$ and $\mathcal{A}'$ comes about as an instance of a dual adjunction of automata as described in Sect. 3.3.3, in particular Sect. 3.3.3.3. We use the base equivalence between $\mathsf{EM}(N)$ and $\mathsf{Set}^{\mathsf{op}}$ described in Sect. 3.5.1. For the sake of uniformity with the general setup in Sect. 3.3.3, we take R as the right adjoint (hence we put the op on $\mathsf{EM}(N)$), and consider R and J as contravariant functors.

$$
\begin{tikzcd}
\left(\mathsf{Aut}^{N(1),2}_{\mathsf{EM}(N)}\right)^{\mathsf{op}} \arrow[r, bend left, "\overline{R}'"] \arrow[r, phantom, "\top"] \arrow[d] & \mathsf{Aut}^{1,2}_{\mathsf{Set}} \arrow[l, bend left, "\overline{J}'"] \arrow[d] \\
\mathsf{EM}(N)^{\mathsf{op}} \arrow[r, bend left, "R"] \arrow[r, phantom, "\cong"] & \mathsf{Set} \arrow[l, bend left, "J"]
\end{tikzcd}
\tag{3.23}
$$

More precisely, we show that $\mathcal{A}' = \overline{R}'(\mathcal{A}^\sharp)$, where $\mathcal{A}^\sharp$ is the deterministic automaton over $\mathsf{EM}(N)$ obtained by applying the determinisation construction from Sect. 3.2.2 for N to $\mathcal{A}$. The functor R is the composition $R = At \circ D$ (see (3.20)).

Recall from Sect. 3.2.2 that determinisation for N takes free extensions of the transition function and output function. That is, given an alternating automaton $\mathcal{A}$ with states X and components

$$
i : 1 \to 2^{2^X} \qquad t_a : X \to 2^{2^X},\ a \in \Sigma \qquad f : X \to 2
$$

over Set, we have a deterministic automaton $\mathcal{A}^\sharp$ with

$$
i^\sharp : 2^{2^1} \to 2^{2^X} \qquad t_a^\sharp : 2^{2^X} \to 2^{2^X},\ a \in \Sigma \qquad f^\sharp : 2^{2^X} \to 2
$$

over $\mathsf{EM}(N)$, using the CABA structure on 2. In $\mathcal{A}^\sharp$, we leave algebraic structure on 2^{2^X} and 2 implicit. Formally, they are the powerset CABAs on 2^{2^X} and 2, respectively; these are isomorphic to the free $\mathsf{EM}(N)$-algebras (NX, μ_X) and $(N\emptyset, \mu_\emptyset)$ on generators X and $\emptyset$, respectively.

We easily see that $\langle 2^{2^X}, f^\sharp, t^\sharp, i^\sharp \rangle$ instantiates the definition from Sect. 3.3.2 of an $\mathsf{EM}(N)$-automaton with initialisation in 2^{2^1} and output in 2, i.e., $\mathcal{A}^\sharp$ is in $\mathsf{Aut}^{N(1),2}_{\mathsf{EM}(N)}$. For ease of notation, we will sometimes write the initialisation morphism $i^\sharp$ as its corresponding Set-function i.

A dual automaton in $\mathsf{Aut}^{1,2}_{\mathsf{Set}}$ (with states X) is a coalgebra for $F = 2 \times (-)^\Sigma$ together with an initial state $j : 1 \to X$, or equivalently an algebra for $G = 1 + \Sigma \times (-)$ with output $f : X \to 2$. It is easy to check that the conditions for Theorem 3.1 hold. First note that $I = 2^{2^1}$ and $O = 1$. We then easily verify that $F_{\mathsf{EM}(N)} \cong J(1) \times (-)^\Sigma$ by noting that $J(1) = 2^1 \cong 2$. Similarly, to see that $G \cong R(2^{2^1}) + \Sigma \cdot (-)$, we note that $R(2^{2^1}) = At(2^{2^1}) = 2^1 \cong 2$. Hence the base dual adjunction $J \dashv R$ lifts to $\overline{J}' \dashv \overline{R}'$ between automata categories, and the lifted adjoints are given by (3.7) and Theorem 3.1. We describe the reversal functor $\overline{R}'$ a bit more concretely as a contravariant functor from $\mathsf{Aut}^{N(1),2}_{\mathsf{EM}(N)}$ to $\mathsf{Aut}^{1,2}_{\mathsf{Set}}$. The base adjunction of (3.23) gives us a bijection of homsets:

$$\theta : \mathsf{EM}(N)((A, \alpha), JX) \to \mathsf{Set}(X, R(A, \alpha))$$

natural in (A, α) and X. Given an automaton in $\mathsf{Aut}^{N(1),2}_{\mathsf{EM}(N)}$

$$i : 1 \to (A, \alpha) \qquad t_a : (A, \alpha) \to (A, \alpha) \qquad f : (A, \alpha) \to 2$$

(again, we leave the algebraic structure on 2 implicit), $\overline{R}'$ produces the deterministic automaton over Set

$$\theta f : 1 \to R(A, \alpha) \quad R(t_a) : R(A, \alpha) \to R(A, \alpha) \quad Ri^\sharp : R(A, \alpha) \to 2. \quad (3.24)$$

Applying $\overline{R}'$ to $\mathcal{A}^\sharp$, which is

$$i : 1 \to 2^{2^X} \qquad t_a^\sharp : 2^{2^X} \to 2^{2^X} \qquad f^\sharp : 2^{2^X} \to 2$$

we get the reversed, deterministic automaton $\overline{R}'(\mathcal{A}^\sharp)$ (over Set):

$$\theta f^\sharp : 1 \to 2^X \qquad \bar{R}t_a^\sharp : 2^X \to 2^X \qquad Ri^\sharp : 2^X \to 2.$$

Theorem 3.3 *For any alternating automaton*

$$\mathcal{A} = (X, \{t_a : X \to 2^{2^X} \mid a \in \Sigma\}, i : 1 \to 2^{2^X}, f : X \to 2),$$

we have that $\mathcal{A}' \cong \overline{R}'(\mathcal{A}^\sharp)$.

Proof The state space of $\mathcal{A}'$ is 2^X and the state space of $\overline{R}'(\mathcal{A}^\sharp)$ is the set of atoms of the CABA $D(2^{2^X}, \mu_X)$ which is the set $\{\{a\} \mid a \subseteq X\}$. To see that $\theta(f^\sharp) = f^\flat$, observe that $\theta^{-1}(f^\flat) = f^{\flat^{-1}}$, and expand the definitions to show that $f^\sharp = f^{\flat^{-1}}$. To see that $R(t_a{}^\sharp) = t_a^\flat$, we show that for all $d : Y \to 2^{2^X}$, and all $a \subseteq X$, $R(d^\sharp)(\{a\}) = \{d^\flat(a)\}$. Thus $R(d^\sharp) = d^\flat$ up to bijections relating atoms $\{a\}$ and their singleton elements a. The atoms of $(2^{2^X}, \mu_X)$ are of the form $\{a\}$ for $a \subseteq X$, hence for $A \in 2^{2^Y}$,

$$\begin{aligned}
R(d^\sharp)(\{a\}) &= \bigwedge\{A \in 2^{2^Y} \mid \{a\} \leq d^\sharp(A)\} \\
&= \bigcap\{A \in 2^{2^Y} \mid a \in d^\sharp(A)\} && \text{since } \bigwedge \text{ is } \bigcap \text{ in}(2^{2^X}, \mu_X) \\
&= \bigcap\{A \in 2^{2^Y} \mid d^\flat(a) \in A\} && \text{since } d^\sharp = d^{\flat^{-1}} \\
&= \{d^\flat(a)\} && \text{since} d^\flat(a) \in 2^Y.
\end{aligned}$$

where $d^\sharp = d^{\flat^{-1}}$ can be shown by expanding the definitions. The argument for $R(i^\sharp) = i^\flat$ is similar.

The relationship between an AFA and its determinised version can be understood as follows. In an AFA, when reading an input word, we generate a computation tree downwards, and once we reach the end of the word, we evaluate the outputs going back up using Boolean functions, and at the top all outputs are aggregated into a single Boolean value with the acceptance condition. In the determinised AFA, we propagate the acceptance condition forwards as a Boolean function (encapsulated in the state) and once we reach the end of the input word, we use the Boolean function to evaluate immediately instead of propagating back up.

The modal logic of alternating automata (cf. (3.8)) has a single termination predicate, labelled modalities, and no propositional connectives, since $\mathcal{D} = \mathsf{Set}$. Hence formulas correspond to words in Σ^*. The dual DFA of an AFA represents its logical semantics, or predicate transformer semantics, where the observations at the end of the word are propagated backwards to the initial state. Since predicate transformers move backwards, the language of an AFA is the reversed language of the dual DFA.

Finally, we note that all conditions for Theorem 3.2 hold (with $\mathcal{D} = \mathsf{Set}$ and $\Phi_{\mathcal{D}} = U_{\mathcal{D}} = Id$). Hence we also get a Brzozowski style minimisation algorithm for alternating automata by instantiating **Algo2** of Sect. 3.3.6. Reachability in $\mathsf{Aut}^{1,2}_{\mathsf{Set}}$ is just the standard automata-theoretic notion, whereas now the more abstract algebraic notion from Sect. 3.3.5 is relevant "on the left" in the category $\mathsf{Aut}^{N(1),2}_{\mathsf{EM}(N)}$. As with weighted automata (cf. Sect. 3.4.2), we are not guaranteed that the result of the minimisation algorithm is again an alternating automaton (understood as an FN-coalgebra over Set), since a subalgebra of a free CABA need not be free.

3.6 Conclusion and Related Work

In this paper, we presented a unifying categorical perspective on the minimisation constructions presented in Bezhanishvili et al. (2012) and Bonchi et al. (2014), revisited some examples from these two papers in light of the general framework, and presented a new example of alternating automata. We also filled in some details regarding topological automata (belief automata) that were missing from Bezhanishvili et al. (2012).

Our starting points are Brzozowski's algorithm (Brzozowski, 1962) for the minimisation of deterministic automata and the use of Stone-type duality between computational processes and their logical characterisation (Abramsky, 1991). The connection between these two seemingly unrelated points is given by the duality principle between reachability and observability originally introduced in systems theory (Kalman, 1959) and then extended to automata theory in Arbib & Zeiger (1969); Arbib & Manes (1975, 1980).

The duality between reachability and observability has been studied, e.g. in Bidoit et al. (2001), to relate coalgebraic and algebraic specifications in terms of observations and constructors. In this context most notable is the use of Stone-type dualities between automata and varieties of formal languages (Gehrke, 2009; Gehrke et al., 2008) which recently culminated into a general algebraic and coalgebraic understanding of equations, coequations, Birkhoff's and Eilenberg-type correspondences (Ballester-Bolinches et al., 2015; Salamanca et al., 2015, 2016; Adamek et al., 2015; Adámek et al., 2018).

Our unifying categorical perspective is based on a dual adjunction between base categories lifted to a dual adjunction between coalgebras and algebras, as introduced in (Bonsangue & Kurz, 2006; Klin, 2007; Kerstan et al., 2014) in the context of coalgebraic modal logic, and in Bezhanishvili et al. (2012); Klin & Rot (2016) to capture the observable behaviour of a coalgebra. Our novelty is to lift the coalgebra-algebra adjunction to a dual adjunction between automata which generalises the formalisation of Brzozowski's algorithm from Bonchi et al. (2014), and formalising the relationship of trace logic to the full modal logic and language semantics.

Our paper focuses on comparing and unifying our earlier approaches from Bezhanishvili et al. (2012) and Bonchi et al. (2014) under a common umbrella, but we hasten to remark that the concept of minimisation via logic presented in Sect. 3.3.3 is already in Rot (2016). At its core, Rot (2016) uses a dual adjunction that is lifted to a dual adjunction between coalgebras and algebras. A logic is then used to provide a construction for obtaining observable coalgebras. This is esssentially what we call **Algo1**. The setting of Rot (2016) is more general as no assumptions are made on the specific shape of the algebra and coalgebra functors involved. Instead the necessary functor requirements are axiomatised. One achievement of Rot (2016) is to generalise the setup in Bezhanishvili et al. (2012) from dual equivalences to dual adjunctions. The central contribution in Rot (2016) is to combine the duality-based framework with coalgebraic partition-refinement (Adámek et al., 2012) such that a logic-based treatment of Brzozowski and partition refinement is obtained. Compared to Rot (2016), our framework is more restricted, as we confine ourselves to functors of certain shapes, but we believe this strikes a good balance between generality and a categorical setting for studying many different types of automata. Furthermore, our categorical framework incorporates a formalisation of the full Brzozowski algorithm via the small extension of the coalgebra-algebra adjunction to the adjunction of automata, i.e., structures that have both initial and final states.

Other categorical approaches to automata minimisation have been proposed in the literature; we mention here just a few. In Colcombet & Petrisan (2017) languages and their acceptors are regarded as functors which provides a different perspective on minimisation in which Brzozowski's algorithm can also be formulated. In Adámek et al. (2012) the authors study coalgebras in categories equipped with factorisation structures in order to devise a generic partition refinement algorithm. From the language-theoretic point of view, the relation between the automata constructions resulting from the automata-based congruences, together with the duality between right and left congruences, allows to relate determinisation and minimisation operations (Ganty et al., 2019).

Acknowledgements We thank Jan Rutten and Filippo Bonchi for their contributions to the earlier stages this work, and Jurriaan Rot for fruitful discussions.

References

Abramsky, S. (1991). Domain theory in logical form. *Annals of Pure and Applied Logic, 51*(1), 1–77.

Adámek, J., Bonchi, F., Hülsbusch, M., König, B., Milius, S., & Silva, A. A coalgebraic perspective on minimization and determinization. In Lars Birkedal, editor, *Foundations of Software Science and Computational Structures - 15th International Conference, (FOSSACS 2012)*, volume 7213 of *Lecture Notes in Computer Science*, pages 58–73. Springer, 2012.

Adámek, J., Herrlich, H., & Strecker, G. E. (2009). *Abstract and Concrete Categories - The Joy of Cats*. Dover Publications.

Adamek, J., Myers, R. S. R., Urbat, H., & Milius, S. (2015). Varieties of languages in a category. In *Proceedings of the 2015 30th annual ACM/IEEE symposium on Logic in Computer Science (LICS)*, LICS '15 (pp. 414–425). IEEE Computer Society.

Adámek, J., Milius, S., Myers, R. S. R., & Urbat, H. (2018). Generalized Eilenberg theorem: Varieties of languages in a category. *ACM Transaction of Computational Logic*, *20*(1).

Arbib, M. A., & Manes, E. G. (1974). Machines in a category: An expository introduction. *SIAM Review, 16,* 163–192.

Arbib, M. A., & Zeiger, H. P. (1969). On the relevance of abstract algebra to control theory. *Automatica, 5,* 589–606.

Arbib, M. A., & Manes, E. G. (1975). Adjoint machines, state-behavior machines, and duality. *J. of Pure and Applied Algebra, 6*(3), 313–344.

Arbib, M. A., & Manes, E. G. (1980). Foundations of system theory: The Hankel matrix. *Journal of Computer and System Sciences, 20,* 330–378.

Arveson, W. (1976). *An invitation to C*-algebras*, *Graduate Texts in Mathematics* (Vol. 39). Springer.

Ballester-Bolinches, A., Cosme-Llópez, E., & Rutten, J. (2015). The dual equivalence of equations and coequations for automata. *Information and Computation, 244*(C), 49–75.

Bartels, F. (2004). *On generalised coinduction and probabilistic specification formats*. Ph.D. thesis, Vrije Universiteit Amsterdam.

Bezhanishvili, N., Kupke, C., Panangaden, P. (2012). Minimization via duality. In Ong, L., & de Queiroz, R. (Eds.), *Proceedings of WoLLIC' 12*, *LNCS* (Vol. 7456, pp. 191–205). Springer.

Bidoit, M., Hennicker, R., & Kurz, A. (2001). On the duality between observability and reachability. In Honsell, F., & Miculan, M. (Eds.), *FoSSaCS*, *Lecture Notes in Computer Science* (Vol. 2030, pp. 72–87). Springer.

Blackburn, P., de Rijke, M., & Venema, Y. (2001). *Modal logic*. Cambridge: Cambridge University Press.

Bonchi, F., Bonsangue, M., Hansen, H. H., Panangaden, P., Rutten, J., & Silva, A. (2014). Algebra-coalgebra duality in Brzozowski's minimization algorithm. *ACM Transactions on Computational Logic, 15*(1).

Bonchi, F., Bonsangue, M. M., Boreale, M., Rutten, J. J. M. M., & Silva, A. (2012). A coalgebraic perspective on linear weighted automata. *Information and Computation, 211,* 77–105.

Bonsangue, M. M., & Kurz, A. (2005). Duality for logics of transition systems. In *FoSSaCS' 05*.

Bonsangue, M. M., & Kurz, A. (2006). Presenting functors by operations and equations. In Aceto, L., Ingólfsdóttir, A. (Eds.) *Foundations of Software Science and Computation Structures, 9th international conference, FOSSACS 2006*, *Lecture Notes in Computer Science* (Vol. 3921, pp. 172–186). Springer.

Borceux, F. (1994). *Handbook of Categorical Algebra 2: Categories and structure*. Cambridge University Press.

Brzozowski, J. A., & Leiss, E. (1980). On equations for regular languages, finite automata, and sequential networks. *Theoretical Computer Science, 10,* 19–35.

Brzozowski, J. A. (1962). Canonical regular expressions and minimal state graphs for definite events. In *Mathematical theory of automata*, *MRI symposia series* (Vol. 12, pp. 529–561). Polytechnic Institute of Brooklyn, Polytechnic Press.

Chagrov, A., & Zakharyaschev, M. (1997). *Modal logic*, *Oxford logic guides* (Vol. 35). New York: The Clarendon Press.

Chandra, A., Kozen, D., & Stockmeyer, L. (1981). Alternation. *Journal of the Association for Computing Machinery, 28*(1), 114–133.

Chandra, A. K., & Stockmeyer, L. J. (1976). Alternation. In *Proceedings of 17th Symposium on Foundations of Computer Science (FOCS)* (pp. 98–108). IEEE.

Colcombet, T., & Petrisan, D. (2017). Automata minimization: a functorial approach. In Bonchi, F., & König, B. (Eds.), *7th Conference on Algebra and Coalgebra in Computer Science, (CALCO 2017)*, *LIPIcs* (Vol. 72, pp. 8:1–8:16). Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Ganty, P., Gutiérrez, E., & Valero, P. (2019). A congruence-based perspective on automata minimization algorithms. In Rossmanith, P., Heggernes, P., & Katoen, J-P. (Eds.), *44th Int. Symposium on Mathematical Foundations of Computer Science, (MFCS 2019)*, *LIPIcs* (Vol. 138, pp. 77:1–77:14). Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Gehrke, M. (2009). Stone duality and the recognisable languages over an algebra. In Kurz, A., Lenisa, M., & Tarlecki, A. (Eds.), *CALCO*, *Lecture Notes in Computer Science* (Vol. 5728, pp. 236–250). Springer.

Gehrke, M., Grigorieff, S., & Pin, J-E. (2008). Duality and equational theory of regular languages. In Aceto, L., Damgård, I., Ann Goldberg, L., Halldórsson, M. M., Ingólfsdóttir, A., & Walukiewicz, I. (Eds.), *ICALP (2)*, *Lecture notes in computer science* (Vol. 5126, pp. 246–257). Springer.

Gierz, G., Hofmann, K. H., Keimel, K., Lawson, J. D., Mislove, M., & Scott, D. S. (2003). *Continuous lattices and domains*, *Encyclopedia of mathematics and its applications* (Vol. 93). Cambridge: Cambridge University Press.

Givant, S., & Halmos, P. (2009). *Introduction to Boolean algebras*. Springer.

Hermida, C., & Jacobs, B. (1998). Structural induction and coinduction in a fibrational setting. *Information and Computation, 145,* 107–152.

Huntington, E. V. (1904). Sets of independent postulates for the algebra of logic. *Transactions of the American Mathematical Society, 5*(3), 288–309.

Jacobs, B. (2006). A bialgebraic review of deterministic automata, regular expressions and languages. In Futatsugi, K., Jouannaud, J.-P., & Meseguer, J. (Eds.), *Algebra, Meaning and Computation: Essays dedicated to Joseph A. Goguen on the Occasion of his 65th Birthday*, *LNCS* (Vol. 4060, pp. 375–404). Springer.

Jacobs, B., & Sokolova, A. (2010). Exemplaric expressivity of modal logics. *Journal of Logic and Computation, 20*(5), 1041–1068.

Johnstone, P. T. (1982). *Stone spaces*. Cambridge: Cambridge University Press.

Johnstone, P. T. (1975). Adjoint lifting theorems for categories of algebras. *Bulletin London Mathematical Society, 7,* 294–297.

Kalman, R. (1959). On the general theory of control systems. *IRE Transactions on Automatic Control, 4*(3), 110–110.

Kerstan, H., König, B., & Westerbaan, B. (2014). Lifting adjunctions to coalgebras to (re)discover automata constructions. In Bonsangue, M. M. (Ed.) *Coalgebraic Methods in Computer Science* (pp. 168–188). Berlin, Heidelberg: Springer.

Klin, B. (2007). Coalgebraic modal logic beyond sets. In Fiore, M. (Ed.), *Proceedings of the 23rd conference on the Mathematical Foundations of Programming Semantics, (MFPS 2007)*, *Electronic Notes in Theoretical Computer Science* (Vol. 173, pp. 177–201). Elsevier.

Klin, B., & Rot, J. (2016). Coalgebraic trace semantics via forgetful logics. *Logical Methods in Computer Science, 12*(4).

Kozen, D. (1976). On parallelism in Turing machines. In *Proceedings of 17th Symposium on Foundations of Computer Science* (pp. 89–97). IEEE.

Kozen, D. (2006). *Theory of Computation*. New York: Springer.

Mac Lane, S. (1971). *Categories for the Working Mathematician*. New York: Springer.

Leiss, E. (1981). Succinct representation of regular languages by Boolean automata. *Theoretical Computer Science, 13,* 323–330.

Manes, E. (1969). A triple-theoretic construction of compact algebras. In Eckman, B. (Ed.) Seminar on triples and categorical homology theory, *Lecture Notes in Mathematics* (Vol. 80, pp. 91–118). Springer.

Negrepontis, J. W. (1971). Duality in analysis from the point of view of triples. *Journal of Algebra, 19,* 228–253.

Porst, H. E., & Tholen, W. (1991). Concrete dualities. In Herrlich, H., & Porst, H-E. (Eds.), *Category theory at work*. Heldermann Verlag.

Rot, J. (2016). Coalgebraic minimization of automata by initiality and finality. In Birkedal, L. (Ed.) *The thirty-second conference on the mathematical foundations of programming semantics, (MFPS 2016)*, *Electronic notes in theoretical computer science* (Vol. 325, pp. 253–276). Elsevier.

Rutten, J. J. M. M. (2000). Universal coalgebra: A theory of systems. *Theoretical Computer Science, 249*(1), 3–80.

Salamanca, J., Ballester-Bolinches, A., Bonsangue, M. M., Cosme-Llópez, E., & Rutten, J. J. M. M. (2015). Regular varieties of automata and coequations. In Hinze, R., & Voigtländer, J. (Eds.), *Mathematics of Program Construction - 12th international conference, (MPC 2015), proceedings*, *Lecture Notes in Computer Science* (Vol. 9129, pp. 224–237). Springer.

Salamanca, J., Bonsangue, M. M., & Rot, J. (2016). Duality of equations and coequations via contravariant adjunctions. In Hasuo, I. (Ed.), *Coalgebraic Methods in Computer Science - 13th IFIP WG 1.3 international workshop, (CMCS 2016)*, *Lecture Notes in Computer Science* (Vol. 9628, pp. 73–93). Springer.

Schützenberger, M. P. (1961). On the definition of a family of automata. *Information and Control, 4*(2–3), 245–270.

Silva, A., Bonchi, F., Bonsangue, M. M., & Rutten, J. J. M. M. (2010). Generalizing the powerset construction, coalgebraically. In Lodaya, K., & Mahajan, M. (Eds.), *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2010, December 15-18, 2010, Chennai, India*, *LIPIcs* (Vol. 8, pp. 272–283). Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Stone, M. H. (1936). The theory of representations for Boolean algebras. *Transactions of the American Mathematical Society, 40*(1), 37–111.

Stone, M. H. (1937). Topological representation of distributive lattices and Brouwerian logics. *Časopis Pešt. Mat. Fys., 67,* 1–25.

Street, R. (1972). The formal theory of monads. *Journal of Pure and Applied Algebra, 2,* 149–168.

Tan, Y.-J. (2016). Free sets and free subsemimodules in a semimodule. *Linear Algebra and its Applications, 496,* 527–548.

Taylor, P. (2002). Subspaces in abstract Stone duality. *Theory and Applications of Categories, 10*(13), 300–366.

Chapter 4
A Cook's Tour of Duality in Logic: From Quantifiers, Through Vietoris, to Measures

Mai Gehrke, Tomáš Jakl, and Luca Reggio

Abstract We identify and highlight certain landmark results in Samson Abramsky's work which we believe are fundamental to current developments and future trends. In particular, we focus on the use of

- topological duality methods to solve problems in logic and computer science;
- category theory and, more particularly, free (and co-free) constructions;
- these tools to unify the 'power' and 'structure' strands in computer science.

Keywords Duality theory · Topological methods in logic · Vietoris space · Quantifiers and measures · Structural limits · Lindenbaum-Tarski algebras · Free constructions

This project has been funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No.670624). Tomáš Jakl has received partial support from the EPSRC grant EP/T007257/1. Luca Reggio has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No.837724.

M. Gehrke · T. Jakl
CNRS and Université Côte d'Azur, Nice, France
e-mail: mai.gehrke@unice.fr

T. Jakl
e-mail: tomas.jakl@cl.cam.ac.uk

T. Jakl
Department of Computer Science and Technology, University of Cambridge, Cambridge, UK

L. Reggio (✉)
Department of Computer Science, University of Oxford, Oxford, UK
e-mail: l.reggio@ucl.ac.uk

A. Palmigiano and M. Sadrzadeh (eds.), *Samson Abramsky on Logic and Structure in Computer Science and Beyond*, Outstanding Contributions to Logic 25,
https://doi.org/10.1007/978-3-031-24117-8_4

4.1 Algebras from Logic

Boole wanted to view propositional logic as arithmetic. This idea, of seeing logic as a kind of algebra, reached a broader and more foundational level with the work of Tarski and the Polish school of algebraic logicians. The basic concept is embodied in what is now known as the *Lindenbaum-Tarski algebra* of a logic. In the classical cases, this algebra is obtained by quotienting the set of all formulas $\mathcal{F}$ by logical equivalence, that is,

$$\mathcal{L} = \mathcal{F}/_{\approx} \quad \text{where} \quad \varphi \approx \psi \text{ if, and only if, } \varphi \text{ and } \psi \text{ are logically equivalent.}$$

When the equivalence relation $\approx$ is a congruence for the connectives of the logic, $\mathcal{L}$ may be seen as an algebra in the signature given by the connectives. This is the case for many propositional logics as well as for first-order logic. There is, however, a fundamental difference in how well this works at these two levels of logic.

For example, for Classical Propositional Logic (CPL), Intuitionistic Propositional Calculus (IPC) and modal logics, the Lindenbaum-Tarski algebra is the *free algebra* over the set of primitive propositions of the appropriate variety. In the above mentioned cases, these are Boolean algebras, Heyting algebras, and modal algebras of the appropriate signature, respectively. Further, for algebras in these varieties, congruences are given by the equivalence classes of the top elements which, logically speaking, are the theories of the corresponding logics. Consequently, we have that the Lindenbaum-Tarski algebras of theories, in which one quotients out by logical equivalence modulo the theory, account for the full varieties of Boolean algebras, Heyting algebras and modal algebras.

The picture is not always quite this simple, even at the propositional level. E.g. the Lindenbaum-Tarski algebra of positive propositional logic (i.e. the fragment of CPL without negation, which we will denote PPL) is indeed the free bounded distributive lattice over the set of primitive propositions. However, since there are lattices with multiple congruences giving the same filter, we do not have the same natural correspondence between the full variety of distributive lattices and the theories of PPL. This sort of problem can be dealt with and this is the subject of the far-reaching theory of Abstract Algebraic Logic, see Font & Verdú (1991) for the example of PPL.

Let us now consider (classical) first-order logic. Here also, logical equivalence is a congruence for the logical connectives. We have the Boolean connectives, and unary connectives $\exists x$ and $\forall x$, a pair for each individual variable x of the logical language.[1] The latter give rise to pairs of unary operations that are inter-definable by conjugation with negation. Thus, in the Boolean setting, it is enough to consider the $\exists x$ operations. These are (unary) *modal operators*.

In its most basic form, modal propositional logic corresponds to the variety of modal algebras (MAs), which are Boolean algebras augmented by a unary operation that preserves finite joins. The algebraic approach is a powerful tool in the study of modal logics, see e.g. Rautenberg et al. (2006) for a survey. In particular, the

[1] Typically one also considers some named constants, which we are not mentioning here.

Lindenbaum-Tarski algebra for this logic is the free modal algebra over the propositional variables, the normal modal logic extensions correspond to the subvarieties of the variety of MAs, and theories in these logics correspond to the individual algebras in the corresponding varieties.

The Lindenbaum-Tarski algebra of first-order formulas modulo logical equivalence is a multimodal algebra, with modalities $\Diamond_x$, one for each variable x in the first-order language. These modalities satisfy some equational properties such as[2]

$$\varphi \leq \Diamond_x \varphi \qquad \Diamond_x(\varphi \wedge \Diamond_x \psi) = \Diamond_x \varphi \wedge \Diamond_x \psi \qquad \Diamond_x \Diamond_y \varphi = \Diamond_y \Diamond_x \varphi.$$

A fundamental problem, as compared with the propositional examples given above, is that these Lindenbaum-Tarski algebras *are not free* in any reasonable setting. Tarski and his students introduced the variety of cylindric algebras of which these are examples, see Monk (1986) for an overview. However, not all cylindric algebras occur as Lindenbaum-Tarski algebras for first-order theories. For one, when we have an infinite set of variables, and thus of modalities, for every element φ in the algebra there is a finite set V_φ of variables such that $\Diamond_x \varphi = \varphi$ for all $x \notin V_\varphi$.

Even though cylindric algebras have been extensively studied, little is known specifically about the ones arising as Lindenbaum-Tarski algebras of first-order theories. A notable exception is the paper of Myers (1976) characterising the algebras for first-order logic over empty theories. Another important insight, due to Rasiowa and Sikorski, is the fact that the completeness theorem for first-order logic may be obtained using the Lindenbaum-Tarski construction (Rasiowa & Sikorski, 1950). Their proof uses the famous Rasiowa-Sikorski Lemma. This lemma, which may be seen as a consequence of the Baire Category Theorem in topology, states that, given a specified countable collection of subsets with suprema in a Boolean algebra, one can separate the elements of the Boolean algebra with ultrafilters that are inaccessible by these suprema.

The lack of freeness of the Lindenbaum-Tarski algebras of first-order logic is overcome by moving from lattices with operators to categories and categorical logic. In the equational setting, algebraic theories can equivalently be described as Lawvere theories, i.e. categories with finite products and a distinguished object X such that every object is a finite power of X.[3] Similarly, theories in a given fragment of first-order logic correspond to a certain class of categories.

For instance, theories in the positive existential fragment of first-order logic, also called coherent theories, correspond to coherent categories. Every coherent theory T yields a coherent category, the *syntactic category* of T, which may be seen as a generalisation of the Lindenbaum-Tarski construction, and which is free in an appropriate sense. Central to this construction is the fundamental insight, of Lawvere, that

[2] Throughout, if no confusion arises, we write φ for the corresponding element of the Lindenbaum-Tarski algebra, i.e. the logical equivalence class $[\varphi]_\approx$ of the formula φ.

[3] For a variety of algebras $\mathscr{V}$, the associated Lawvere theory is the *dual* of the category of finitely generated free $\mathscr{V}$-algebras with homomorphisms; the distinguished object is the free algebra on one generator.

quantifiers are adjoints to substitution maps. Thus, existential quantifiers are encoded in coherent categories as lower adjoints to certain homomorphisms between lattices of subobjects. Further, there is some sense in which the correspondence between theories and quotients is regained (at the level of so-called classifying toposes of the theories). See Makkai & Reyes (1977). Other fragments of first-order logic can be dealt with in a similar fashion, e.g. intuitionistic first-order theories correspond to Heyting categories, and classical first-order theories to Boolean coherent categories. See Johnstone (2002) for a thorough exposition.

To make the relation between syntactic categories and Lindenbaum-Tarski algebras more explicit, we recall the notion of Boolean hyperdoctrines, tightly related to Boolean coherent categories. Consider the category **Con** of contexts and substitutions. A context is a finite list of variables $\overline{x}$, and a substitution from $\overline{x}$ to a context $\overline{y} = y_1, \ldots, y_n$ is a tuple $\langle t_1, \ldots, t_n \rangle$ of terms with free variables in $\overline{x}$. Given a first-order theory T, let $P(\overline{x})$ be the Lindenbaum-Tarski algebra of first-order formulas with free variables in $\overline{x}$, up to logical equivalence modulo T. A substitution $\langle t_1, \ldots, t_n \rangle : \overline{x} \to \overline{y}$ induces a Boolean algebra homomorphism $P(\overline{y}) \to P(\overline{x})$ sending a formula $\varphi(\overline{y})$ to $\varphi(\langle t_1, \ldots, t_n \rangle / \overline{y})$.[4] This yields a functor

$$P : \mathbf{Con}^{\mathrm{op}} \to \mathbf{BA}.$$

The product projection $\pi_y : \overline{x}, y \to \overline{x}$ in **Con** induces the Boolean algebra embedding $P(\pi_y) : P(\overline{x}) \hookrightarrow P(\overline{x}, y)$, which admits both lower and upper adjoints:

$$\begin{aligned} \exists_y \dashv P(\pi_y), \quad & \exists_y(\varphi(\overline{x}, y)) = \exists y.\varphi(\overline{x}, y), \\ P(\pi_y) \dashv \forall_y, \quad & \forall_y(\varphi(\overline{x}, y)) = \forall y.\varphi(\overline{x}, y). \end{aligned}$$

This accounts for the *Boolean hyperdoctrine* structure of P. The syntactic category of the theory T can be obtained from P by means of a 2-adjunction between Boolean hyperdoctrines and Boolean categories, cf. Pitts (1983) or Coumans (2012, Chap. 5).

While the categorical perspective solves a number of problems, it is not easily amenable to the inductive point of view that we want to highlight here. We will get back to this in Sect. 4.2.2.

4.2 Topological Methods in Logic

Topological methods in logic have their origin in the work of M. H. Stone. The paper Stone (1936) established what is nowadays presented as a dual equivalence between the category **BA** of Boolean algebras with homomorphisms and a full subcategory **BStone** of the category of topological spaces with continuous maps. The objects of **BStone** are the so-called *Boolean (Stone) spaces*, i.e. compact Hausdorff spaces

[4] More precisely, the morphisms in **Con** are defined as equivalence classes of substitutions, by identifying two tuples $\langle s_1, \ldots, s_n \rangle$ and $\langle t_1, \ldots, t_n \rangle$ if they give rise to the same homomorphism.

whose collection of *clopen* (simultaneously closed and open) subsets forms a basis for the topology. Usually referred to as *Stone duality for Boolean algebras*, this is the prototypical example of a dual equivalence induced by a dualizing object, i.e. an object sitting at the same time in two categories. In fact, the quasi-inverse functors providing the equivalence between $\mathbf{BA}^{\mathrm{op}}$ and **BStone** are given by enriching the set of homomorphisms into the appropriate structure on the two-element set $\mathbf{2} = \{0, 1\}$, which can be seen either as the two-element Boolean algebra or as the two-element Boolean space when equipped with the discrete topology.

Given a Boolean algebra B, the space X_B obtained by equipping the set of homomorphisms

$$\hom_{\mathbf{BA}}(B, \mathbf{2})$$

with the subspace topology induced by the product topology on $\mathbf{2}^B$ is a Boolean space, the *(Stone) dual space* of B. Under the correspondence sending a Boolean algebra homomorphism $h\colon B \to \mathbf{2}$ to the subset $h^{-1}(1) \subseteq B$, the points of X_B can be identified with the *ultrafilters* on B. In logical terms, these are the complete consistent theories over B. Conversely, given a Boolean space X, the set of continuous maps

$$\hom_{\mathbf{BStone}}(X, \mathbf{2})$$

forms a Boolean subalgebra B_X of the product algebra $\mathbf{2}^X$, where $\mathbf{2}$ is now viewed as a Boolean algebra. When equipped with the induced Boolean operations, B_X is called the *dual algebra* of X. Upon identifying a continuous function $f\colon X \to \mathbf{2}$ with the clopen subset $f^{-1}(1) \subseteq X$, the Boolean algebra B_X can be described as the field of clopen subsets of X with the set-theoretic Boolean operations. Stone duality states that these object assignments extend to functors, and there are isomorphisms $B \cong B_{X_B}$ and $X \cong X_{B_X}$ (natural in B and X, respectively). Throughout, the element of B_{X_B} corresponding to $a \in B$ will be denoted by $\widehat{a}$.

Shortly after his seminal work in 1936, Stone generalised the duality to bounded distributive lattices (Stone, 1938); there, the relevant category of spaces consists of spectral spaces with perfect maps. A different formulation of the duality for distributive lattices, induced by the dualizing object $\mathbf{2}$ regarded either as a lattice or as a discrete *ordered* space where $0 < 1$, was later introduced in Priestley (1970).

When combined with the algebraic semantics, as outlined in the previous section, Stone duality yields a powerful framework for developing and applying topological methods in logic. The potential advantages of applying duality are of two types. For one, duality theory often connects syntax and semantics. To wit, in the case of CPL, the Lindenbaum-Tarski algebra is the free Boolean algebra on the set V of propositional variables, and its dual space is the Cantor space $\mathbf{2}^V$ of all valuations over V. The second type of advantage is that it *often is easier*, technically, to solve a problem on the dual side.

The use of duality is not restricted to the Boolean setting. Indeed, generalisations and extensions of Stone duality have been exploited to study fragments and extensions of CPL. Many other special cases have since been developed based on Stone's and Priestley's dualities for bounded distributive lattices (corresponding to PPL). Here we just mention the duality for Heyting algebras, the algebraic semantics of IPC, mainly developed by Leo Esakia (1974; 2019). Stone duality was also extended by Jónsson and Tarski to Boolean algebras with operators by introducing the powerful framework of canonical extensions (Jónsson & Tarski, 1951, 1952). This was a crucial step for many applications, e.g. in modal logic.

In theoretical computer science, the link between syntax and semantics provided by Stone-type dualities is particularly central as the two sides correspond to specification languages and to spaces of computational states, respectively. The ability to translate faithfully between these two worlds has often proved itself to be a powerful theoretical tool as well as a handle for solving problems. A prime example is Abramsky's seminal work (Abramsky, 1987, 1991) linking program logic and domain theory via Stone duality for bounded distributive lattices, which was awarded the IEEE LICS "Test of Time" Award in 2007. Other examples include large parts of modal and intuitionistic logics, where Jónsson-Tarski duality yields Kripke semantics (Blackburn et al., 2001). For a particular example, see Ghilardi's work in modal and intuitionistic logic on unification (Ghilardi, 2004) and normal forms (Ghilardi, 1995).

By contrast, Stone duality has not played a significant role, at least overtly, in more algorithmic areas of theoretical computer science until recently. In the theory of regular languages, finite and profinite monoids are an important tool, in particular for proving decidability, ever since their introduction in the 1960s and 1980s, respectively, see Pin (2009) for a survey. While it was observed as early as 1937 by Birkhoff that profinite topological algebras are based on Boolean spaces (Birkhoff, 1937), the connection with Stone duality was not used in automata theory until much more recently. It was exploited first in an isolated case by Pippenger (1997), and then more structurally by Gehrke et al. (2008). Further, realising that these methods are instances of Stone duality provides an opportunity to generalise them to the setting of computational complexity and the search for lower bounds (Gehrke & Krebs, 2017). This line of work connects tools from semantics, such as Stone duality, with problems and methods on the algorithmic side of computer science, such as decidability and Eilenberg-Reiterman theory. Similarly, recent work of Samson Abramsky and co-workers connects categorical tools from semantics, such as comonads, with concepts from finite model theory, such as tree-width and tree-depth (Abramsky et al., 2017; Abramsky & Shah, 2018).

Finite model theory, computational complexity theory and the theory of regular languages all belong to the branch of computer science where the use of resources in computing is the main focus, whereas category theory and Stone duality have long been central tools in semantics of programming languages. While the trend of making connections and seeking unifying results that bridge the gap between semantics and algorithmic issues has long been on the way (e.g. in the form of semantic work on resource sensitive logics), making this overt and placing it front and center stage is a recent phenomenon in which Samson Abramsky has played a central role. In particular, one may mention the 2017 semester-long program at The Simons Institute for the Theory of Computing on Logical Structures in Computation of which he was a co-organiser, and the ensuing work and ongoing project with Anuj

Dawar focussing on bridging what they aptly call the *Structure versus Power* gap in theoretical computer science. The 2014 ERC project Duality in Formal Languages and Logic—a unifying approach to complexity and semantics (DuaLL), in which our recent work has taken place, shares these goals.

In Sect. 4.2.1, we highlight some of the ideas and concepts from Samson Abramsky's work in semantics that are playing an important role in our recent work on the DuaLL project, which we will describe in Sect. 4.3. In Sect. 4.2.2, we briefly review two settings from logic pertinent to our work, and give a duality-centric description of the treatment of the function space construction in Abramsky's Domain Theory in Logical Form. This allows us to make a connection to the profinite methods in automata theory.

4.2.1 Modal Logic and the Vietoris Functor

An important contribution of Samson Abramsky's is to use the duality between syntax and semantics, *combined with a step-wise description of connectives* in logic applications. This phenomenon is the driving force behind his sweeping and elegant general solution to domain equations in the paper Domain Theory in Logical Form (DTLF), (Abramsky, 1991). We will get back to this with a few more details in Sect. 4.2.2. In Abramsky (2005), which is the published version of various talks given during the genesis of DTLF, Abramsky gives a simpler example of this general idea. The setting is non-well-founded sets, and the object he considers is the free modal algebra (over the empty set). Other early uses of similar methods are due to Ghilardi (1992; 1995). Subsequently, the treatment of the free modal algebra given in Abramsky's talks, in particular his talk at the 1988 British Colloquium on Theoretical Computer Science in Edinburgh, has been identified as an important contribution to modal logic in its own right, see e.g. Rutten & Turi (1993); Kupke et al. (2004); Venema & Vosmaer (2014), and it is also very pertinent to the duality theoretic treatment of quantifiers which we will discuss in Sect. 4.3.

The step-wise description of an algebra from a set of generators is what is often called *Noetherian induction* in algebra and *induction on the complexity of a formula* in logic: The algebra is generated layer by layer, starting with the generators—which are said to be of rank 0—by adding consecutive layers of the operations to obtain higher rank elements. Also, instead of doing this with all the operations, we may do it relative to a fragment. In the case of modal algebras, for example, we may consider as rank 0 all Boolean combinations of generators, rank less than or equal to 1 any element which may be expressed as a Boolean combination of rank 0 and diamonds of rank 0 elements, and so on. This is a fine tool for the purpose of induction, but it is not a good tool for constructing algebras in general. However, if the operation is freely added modulo some equations which are of pure rank 1, then it is in fact a powerful method of *construction*. This is exactly the situation for free modal algebras, which are Boolean algebras with an additional operation satisfying the equations

$$\Diamond 0 \approx 0 \quad \text{and} \quad \Diamond(x \vee y) \approx \Diamond x \vee \Diamond y.$$

These equations are both of pure rank 1. That is, in each equation, all occurrences of each variable are in the scope of exactly *one* layer of modal operators.

From a categorical point of view, one may see algebras in a variety as Eilenberg-Moore algebras for a finitary monad, but having a pure rank 1 axiomatisation means that these are also presentable as the *algebras for an endofunctor*, see Kurz & Rosický (2012) where this is studied in greater generality. In the case of MAs, define the endofunctor $\mathbb{M}$ on Boolean algebras which takes a Boolean algebra B to the Boolean algebra freely generated by elements $\Diamond a$, for every $a \in B$, subject to the equations for modal algebras viewed as relations on these generators:

$$\Diamond 0 \approx 0 \quad \text{and} \quad \Diamond(a \vee b) \approx \Diamond a \vee \Diamond b \quad (\forall a, b \in B).$$

Then B, equipped with a unary operation $f\colon B \to B$, is a modal algebra if and only if the map $\Diamond a \mapsto f(a)$ extends to a Boolean algebra homomorphism $h\colon \mathbb{M}(B) \to B$. It also follows that the free modal algebra over a Boolean algebra B may be *constructed inductively*, as the colimit of the sequence

$$B_0 \xhookrightarrow{i_0} B_1 \xhookrightarrow{i_1} B_2 \xhookrightarrow{i_2} \ldots$$

where $B_0 = B$, B_{n+1} is the coproduct $B \oplus \mathbb{M}(B_n)$, the map i_0 is the embedding of B in the coproduct, and $i_{n+1} = \mathrm{id}_B \oplus \mathbb{M}(i_n)$. Note that, if B is finite, then so are all the algebras in the sequence. Moreover, if we start with the free Boolean algebra on a set V, then the colimit of the sequence is the free modal algebra over V, and B_n is the Boolean subalgebra consisting of all formulas of rank at most n.

Further, we may of course dualize $\mathbb{M}$ to get a functor on **BStone** and a co-inductive description of the dual of free modal algebras. This dual endofunctor is the Vietoris functor. Recall that, given a Boolean space X, the *Vietoris hyperspace of* X is the collection $\mathcal{V}(X)$ of closed subsets of X equipped with the topology generated by the sets of the form

$$\Diamond U = \{C \in \mathcal{V}(X) \mid C \cap U \neq \emptyset\} \quad \text{and} \quad (\Diamond U)^c$$

for U a clopen subset of X. With respect to this topology, $\mathcal{V}(X)$ is again a Boolean space. See Vietoris (1922); Michael (1951). Furthermore, for every continuous map $f\colon X \to Y$, the forward-image map $f(-)\colon \mathcal{V}(X) \to \mathcal{V}(Y)$ is continuous. Hence, we obtain a functor

$$\mathcal{V}\colon \mathbf{BStone} \to \mathbf{BStone}.$$

Abramsky showed that the dual Stone space of the free modal algebra on no generators coincides with the final coalgebra for the functor $\mathcal{V}$. In general, the dual of the sequence of embeddings given above is

$$X \xleftarrow{\pi_X} X \times \mathcal{V}(X) = X_1 \xleftarrow{\mathrm{id}_X \times \mathcal{V}(\pi_X)} X \times \mathcal{V}(X_1) = X_2 \longleftarrow \dots$$

This result provides also a coalgebraic perspective on the duality between modal algebras and descriptive general Kripke frames. As such, it has had a strong influence on the very active coalgebraic approach to modal logic. The Vietoris hyperspace construction also appeared earlier in modal logic in the work (published in Russian) of Leo Esakia, cf. Esakia (1974). See also Esakia (2019) for the recent English translation of Esakia's 1985 book.

4.2.2 *Three Examples of Dual Spaces in Logic*

In this section we discuss duality methods in logic in three settings: classical first-order logic, Büchi's logic on words, and Domain Theory in Logical Form.

First-order logic and spaces of types. For classical first-order logic, the dual space of the Lindenbaum-Tarski algebra of formulas is fairly easy to describe. Fix a countably infinite set of first-order variables $v_1, v_2, \dots$ and a first-order signature σ, i.e. σ may contain relation symbols as well as function symbols and constants. Denote by FO_ω the set of all first-order formulas in the signature σ over the set of variables. Given a theory T, that is, any set of first-order sentences in the signature σ, consider the collection

$$\mathrm{Mod}_\omega(T) = \{(A, \alpha\colon \omega \to A) \mid A \text{ is a } \sigma - \text{structure and } A \models T\}$$

of models of T equipped with an assignment of the variables. The satisfaction relation $\models \subseteq \mathrm{Mod}_\omega \times \mathrm{FO}_\omega$ induces the equivalence relations of elementary equivalence and logical equivalence on these sets, respectively:

$$(A, \alpha) \equiv (A', \alpha') \quad \text{iff} \quad \forall \varphi \in \mathrm{FO}_\omega \;\; A, \alpha \models \varphi \iff A', \alpha' \models \varphi$$

and

$$\varphi \approx \psi \quad \text{iff} \quad \forall (A, \alpha) \in \mathrm{Mod}_\omega(T) \;\; A, \alpha \models \varphi \iff A, \alpha \models \psi.$$

The quotient $\mathrm{FO}_\omega(T) = \mathrm{FO}_\omega/{\approx}$, i.e. the Lindenbaum–Tarski algebra of T, carries a natural Boolean algebra structure. On the other hand, $\mathrm{Typ}_\omega(T) = \mathrm{Mod}_\omega/{\equiv}$ is naturally equipped with a topology, generated by the sets

$$[\![\varphi]\!] = \{[(A, \alpha)] \mid A, \alpha \models \varphi\}$$

for $\varphi \in \mathrm{FO}_\omega$, and is known as the *space of types* of T. Gödel's completeness theorem may now be stated as follows:

the space $\mathrm{Typ}_\omega(T)$ is the Stone dual of $\mathrm{FO}_\omega(T)$.

For every $n \in \mathbb{N}$, we can consider the Boolean subalgebra $\mathrm{FO}_n(T)$ of $\mathrm{FO}_\omega(T)$ consisting of the equivalence classes of formulas with free variables in $v_1, \ldots, v_n$. The dual space of $\mathrm{FO}_n(T)$ is then the space of n-types of T. In particular, for $n = 0$, we see that the dual space of the Lindenbaum-Tarski algebra of sentences $\mathrm{FO}_0(T)$ is the space of elementary equivalence classes of models of T.

Methods based on spaces of types play a central role in model theory. Their use can be traced back to Tarski's work, but the functorial nature of the construction was brought out and exploited nearly thirty years later by Morley in (Morley, 1974). In fact, it has been suggested that the notion of type space may be more fundamental than the notion of model (Macintyre, 2003). This point of view is related to the categorical approach, as the type space functor of a theory T can be essentially identified with the (pointwise) dual of the hyperdoctrine associated with T.

This approach relies on the presentation of the algebra $\mathrm{FO}_\omega(T)$ as the colimit of the following diagram of Boolean algebra embeddings:

$$\mathrm{FO}_0(T) \hookrightarrow \mathrm{FO}_1(T) \hookrightarrow \mathrm{FO}_2(T) \hookrightarrow \ldots$$

Interestingly, this presentation does not fit with the inductive treatment of modal logic in Sect. 4.2.1, as the sentences, which is what we want to understand, belong to all the algebras in the chain. If we want to construct the Lindenbaum-Tarski algebra $\mathrm{FO}_\omega(T)$ inductively, by adding a layer of quantifier $\exists$ at each step, we should start from the Boolean subalgebra $\mathrm{FO}^0(T)$ of $\mathrm{FO}_\omega(T)$ consisting of the *quantifier-free* formulas. The algebra $\mathrm{FO}^0(T)$ sits inside the algebra $\mathrm{FO}^1(T)$ of formulas with quantifier rank at most 1, and so forth. The colimit of the diagram

$$\mathrm{FO}^0(T) \hookrightarrow \mathrm{FO}^1(T) \hookrightarrow \mathrm{FO}^2(T) \hookrightarrow \ldots$$

is again the algebra $\mathrm{FO}_\omega(T)$. In Sect. 4.3, we will illustrate how the inductive methods used in Büchi's logic apply in the general first-order setting (and beyond) using the ideas set forth in Sect. 4.2.1.

Büchi's logic on words and profinite monoids. The connection between logic and automata goes back to the work of Büchi, Elgot, Rabin and others in the 1960s. In particular, Büchi's logic on words provides a powerful tool for the study of formal languages. The basic idea consists in regarding words on a finite alphabet A, i.e. elements of the free monoid A^*, as finite models for so-called *logic on words*. That is, a word $w \in A^*$ is seen as a relational structure on the initial segment of the natural numbers

$$\{1, \ldots, |w|\},$$

where $|w|$ is the length of w, equipped with a unary relation P_a for each $a \in A$ which singles out the positions in w where the letter a appears. Büchi's theorem states that the Lindenbaum-Tarski algebra of monadic second-order sentences for logic on words with the successor relation (interpreted over finite words) is isomorphic to the Boolean subalgebra of $\mathcal{P}(A^*)$ consisting of the regular languages (Büchi, 1966).

Since we are beyond first-order logic, and we have restricted to the finite models, the dual of the Lindenbaum-Tarski algebra is *not* A^*, i.e. the collection of (elementary equivalence classes of) finite models. For the FO fragment of logic on words we can identify the dual with a space of models provided we allow for pseudofinite words. See e.g. van Gool & Steinberg (2017). However, this is not the case for monadic second-order logic and duality guides the right choice for the space of generalised models as the dual of the Lindenbaum-Tarski algebra. The latter coincides with (the underlying space of) the profinite completion $\widehat{A^*}$ of the monoid A^*, or equivalently, the free profinite monoid on the set A.

The observation that the space underlying the free profinite monoid is the dual of the Boolean algebra of languages recognised by finite monoids essentially goes back to Birkhoff (1937), and was rediscovered by Almeida in the setting of automata theory (Almeida, 1989). Further, the fact that the monoid multiplication of $\widehat{A^*}$ also arises from duality for Boolean algebras with operators as the dual of certain quotienting operations on regular languages was shown in Gehrke et al. (2008).

This type space tells us what generalised models for these logics should be, namely the points of the free profinite monoids. The realisation that these are an important tool in automata theory came in the 1980s (Reiterman, 1982; Almeida, 1994). However, it was introduced, not via logic and duality, but rather via the connection between automata and finite semigroups, where the multiplication available on the profinite monoid also plays a fundamental role.

An essential insight in the proof of Büchi's theorem is the fact that every monadic second-order formula is equivalent on words to an existential monadic second-order formula, and thus the iterative approach is not relevant as the hierarchy collapses. See Ghilardi & van Gool (2016) for a duality and type-theoretic approach via model companions. However, for the first-order fragment the iterative approach is very powerful. The first, and still prototypical application, is Schützenberger's theorem which applies an iterative method, similar to the one of Sect. 4.2.1, to characterise the first-order fragment via duality. To be more precise, Schützenberger (1965) shows that the star-free languages are precisely those recognised by (finite) aperiodic monoids. To prove this, Schützenberger identified a semidirect product construction which captures dually the application of concatenation product on languages. The fact that star-free languages are precisely those given by first-order sentences of Büchi's logic was subsequently shown in McNaughton & Papert (1971), though some passages in the introduction of Schützenberger (1965) suggest that Schützenberger was aware of this connection when he proved his result.

Domain Theory in Logical Form. In denotational semantics one seeks mathematical models of programs, which should be assigned in a compositional way. The compositionality means that program constructors should correspond to type constructors, and solutions to domain equations should correspond to program specifications. Scott's original solution to the domain equation

$$X \cong [X, X],$$

seeking a domain X which is isomorphic to the domain of its endomorphisms, was obtained by constructing a profinite poset, that is, a spectral space. Much further work confirmed that categorical methods, topology and in particular duality are central to the theory, cf. Scott & Strachey (1971), Plotkin (1976), Smyth & Plotkin (1982), Smyth (1983), Larsen & Winskel (1991). Rather than seeing Stone duality and its variants as useful technical tools for denotational semantics, Abramsky put Stone duality front and center stage: A *program logic* is given in which denotational types correspond to theories and the ensuing Lindenbaum-Tarski algebras of the theories are bounded distributive lattices, whose dual spaces yield the domains as types. The constructors involved in the domain equations thus have duals under Stone duality, and solutions are obtained as duals of the solutions of the corresponding equation on the lattice side. In Abramsky (1987) Stone duality is restricted to the so-called Scott domains. That is, algebraic domains that are consistently complete. These are fairly simple and are closed under many constructors, including function space. In Abramsky (1991) the larger category of bifinite domains, which, in addition, is closed under powerdomain constructions, is used. We will say a bit more about bifinite domains later, but for now, we illustrate with a simple example at the level of spectral spaces.

The Smyth powerdomain, $\mathbb{S}(X)$, is the space whose points are the compact and saturated[5] subsets of X equipped with the upper Vietoris topology (Smyth, 1983). That is, the topology is generated by the subbasis given by the sets

$$\Box U = \{K \in \mathbb{S}(X) \mid K \subseteq U\}, \quad \text{for } U \subseteq X \text{ open.}$$

At first sight, this may seem like quite an exotic object to pull out of a hat to study non-determinism. However, in Abramsky's duality with program logic, this construct is the Stone dual of adding a layer of (demonic) non-determinism. Indeed, if X is a spectral space, then so is $\mathbb{S}(X)$, and if L is the dual of X, then $\mathbb{S}(X)$ is the dual of

$$F_\Box(L) = \mathbb{F}_{DL}(\Box L)/_\approx.$$

Here, $\mathbb{F}_{DL}(\Box L)$ denotes the free distributive lattice[6] on the set of formal generators $\Box L = \{\Box a \mid a \in L\}$, and $\approx$ is the congruence given by the following scheme of relations between the generators:

$$\Box\left(\bigwedge G\right) \approx \bigwedge \Box G \quad \text{for } G \subseteq L \text{ finite.}$$

Note that the Smyth powerdomain generalises the Vietoris hyperspace construction for Boolean spaces and, indeed, when $L = B$ is a Boolean algebra, the Booleanization of the lattice $F_\Box(B)$ coincides with the Boolean algebra $\mathbb{M}(B)$ from Sect. 4.2.1.

[5] A subset $K \subseteq X$ is saturated provided it is an intersection of opens, or equivalently, it is an up-set in the specialisation order of the space X.

[6] All distributive lattices are assumed to be bounded, and lattice homomorphisms preserve these bounds.

Now the domain equation $X = \mathbb{S}(X)$ is solved by the final coalgebra for $\mathbb{S}$. However, a priori, there is no guarantee that it exists. On the other hand, the dual equation $L = F_\Box(L)$ is solved by the initial algebra, i.e. the free $\Box$-algebra over the empty set. As explained in Sect. 4.2.1, the latter algebra is guaranteed to exist since algebraic varieties are closed under filtered colimits.

Even though the duality theoretic paradigm supplied by the program logic makes it clearer why $\mathbb{S}(X)$ is the right object, one may still wonder how difficult it is to discover that $F_\Box(L)$ and $\mathbb{S}(X)$ are dual to each other. But this also is made quite algorithmic by duality: The dual of a free distributive lattice, such as $\mathbb{F}_{DL}(\Box L)$, is simply the Sierpinski cube $\mathbf{2}^{\Box L}$.[7] Indeed, a subset $S \subseteq \Box L$ corresponds to the unique homomorphism $h_S \colon \mathbb{F}_{DL}(\Box L) \to \mathbf{2}$ extending the characteristic map $\chi_S \colon \Box L \to \mathbf{2}$. Viewed as a theory (or prime filter) it is $F_S = \{\varphi \mid \exists S' \subseteq S \text{ finite with } \bigwedge S' \leq \varphi\}$. Also, a quotient of $\mathbb{F}_{DL}(\Box L)$ such as $F_\Box(L)$ is dual to a subspace of $\mathbf{2}^{\Box L}$, namely the one consisting of all those $S \subseteq \Box L$ such that

$$\Box(\bigwedge G) \in F_S \iff \bigwedge \Box G \in F_S, \quad \text{for } G \subseteq L \text{ finite.}$$

By the definition of F_S, this is equivalent to

$$\Box(\bigwedge G) \in S \iff \Box G \subseteq S, \quad \text{for } G \subseteq L \text{ finite.}$$

Note that $\mathbf{2}^{\Box L}$ is homeomorphic to $\mathcal{P}(L)$ with the topology generated by the sets $\widetilde{a} = \{S \in \mathcal{P}(L) \mid a \in S\}$ for $a \in L$. Viewed as subsets of L, the elements that belong to the dual of $F_\Box(L)$ are precisely the filters of L. That is, $\mathbb{S}(X)$ is homeomorphic to the space $\mathrm{Filt}(L)$ equipped with the topology generated by the sets $\widetilde{a}$ for $a \in L$. This algorithmic method, using duality for quotients of free algebras and then inductively adding layers of a connective, has been applied widely in the setting of propositional logics, see e.g. Ghilardi (1992); Gehrke & Bezhanishvili (2011); Ghilardi (2010); Coumans & van Gool (2012).

In Abramsky (1991) a large number of constructors such as $\mathbb{S}$ are treated, including the function space which, given two spaces X and Y, yields the space $[X, Y]$ of all continuous functions $X \to Y$ in the compact-open topology. This case is more subtle, but it is closely related to the one above, and to the duality between lattices with residuation and Stone topological algebras, which is at the heart of the duality theory of profinite methods in automata theory. For these reasons, we go in a bit more detail. The following are extracts of a book in preparation (Gehrke & van Gool, 2023).

Consider the duality as above but for the operator type of implication. That is, given distributive lattices (DLs) L and M, define

$$F_\to(L \times M) = \mathbb{F}_{DL}(\to (L \times M))/_\approx,$$

[7] In this section, the dualizing object $\mathbf{2}$ is regarded as either a distributive lattice, or a spectral space by equipping the two-element set with the Sierpinski topology.

where $\to(L \times M) = \{a \to b \mid a \in L,\ b \in M\}$ are the formal generators and $\approx$ is the congruence given by the following two schemes of relations between the generators:

(i) $a \to \bigwedge G = \bigwedge\{a \to b \mid b \in G\}$ for $a \in L$ and $G \subseteq M$ finite;
(ii) $\bigvee F \to b = \bigwedge\{a \to b \mid a \in F\}$ for $F \subseteq L$ finite and $b \in M$.

Going through the same exercise as outlined above to identify the elements of $\mathbf{2}^{L\times M}$ which are compatible with the schemes (i) and (ii), one obtains the following result.

Theorem 4.2.1 *Let L and M be DLs, and let X and Y be their respective dual spaces. The dual of $F_\to(L \times M)$ is the space $[X, \mathbb{S}(Y)]$ of continuous functions from X to the Smyth powerspace of Y, in the compact-open topology.*

This provides a dual description of $[X, \mathbb{S}(Y)]$, but we are interested in $[X, Y]$ which is a subspace of $[X, \mathbb{S}(Y)]$. However, it is not in general a closed subspace in the patch topology, reflecting the fact that $[X, Y]$ is not in general a spectral space. One would need to move to frames, sober spaces and geometric theories to describe $[X, Y]$ as the dual of a quotient. However, we have the following approximation.

Proposition 4.2.2 *Let L and M be DLs, and X, Y their respective dual spaces. The dual of the quotient of $F_\to(L \times M)$ by a congruence θ is a subspace of $[X, Y]$ if and only if for all $x \in X$, $a \in F_x$, and finite subset $G \subseteq M$, there is $a' \in F_x$ such that*

$$[a \to (\bigvee G)]_\theta \ \leq\ [\bigvee\{a' \to b \mid b \in G\}]_\theta.$$

Here, F_x denotes the prime filter of L corresponding to the point $x \in X$.

The above property may be thought of as saying that the operations $x \to (-)$, for $x \in X$, preserve finite joins. For this reason, it has been called 'preserving joins at primes'. Cf. Sect. 3.2 of Gehrke (2016), where it is used to characterise the lattices with residuation that are dual to topological algebras based on Boolean spaces.

There is a special case in which we can get our hands on the property of preserving joins at primes with a finitary scheme of relations between generators. This is the case where the lattice L has enough join prime elements, i.e. every $a \in L$ is a finite join of join prime elements of L. This is for example true in free distributive lattices (where the meets of finite sets of generators are join prime), and it is intimately related to the interaction of domain theory and Stone duality as we have the following theorem.

Theorem 4.2.3 (Abramsky (1991), Theorem 2.4.5) *A lattice has enough join primes if, and only if, its dual space endowed with the Scott topology is a domain.*

Let L be a lattice with enough join primes, and X its dual space. If $P = J(L)$ is the subposet of join prime elements of L, the free distributive lattice on the *poset* P is isomorphic to L. Further, $X \cong \mathrm{Idl}(P^{\mathrm{op}})$, the free directed join completion of P^{op} in the Scott topology, while $P^{\mathrm{op}} \cong \mathrm{Comp}(X)$, the set of compact elements of X. In particular, X is an algebraic domain. Accordingly, we see that everything, i.e. L,

X, and the compact elements of X, is determined by P. The posets P that occur in this way were described already in Plotkin (1976), where the profinite domains were characterised as those algebraic domains for which the set of compact elements form a 'MUB-complete poset' in the nomenclature of Abramsky & Jung (1995). We now have a corollary of Proposition 4.2.2.

Corollary 4.2.4 *Let L and M be DLs with dual spaces X and Y, respectively. Suppose L has enough join primes and let $P = J(L)$. Then the quotient of $F_{\to}(L \times M)$ by the congruence θ given by the following scheme is dual to the function space $[X, Y]$:*

$$p \to \bigvee G \approx \bigvee \{p \to b \mid b \in G\} \text{ for } p \in P \text{ and } G \subseteq M \text{ finite}.$$

In the above, we have just talked about spectral spaces and domains, but in order to have a class of spectral domains not only closed under function spaces and products, but also under the various versions of powerdomain, one must restrict oneself to the so-called bifinite domains. These were introduced (in the setting of domains with a least element) in Plotkin (1976) as generated by special MUB-complete posets P now known as Plotkin orders (Abramsky & Jung, 1995, Definition 4.2.1). These also have a beautiful very self-dual description relative to Stone duality.

The following definition applies to categories concrete over the category **Pos** of posets and monotone maps, such as the category of DLs or that of spectral spaces and spectral maps (w.r.t. the specialization order) with the obvious forgetful functors.

Definition 4.2.5 Let $\mathcal{C}$ be a category equipped with a faithful functor $U : \mathcal{C} \to$ **Pos**. A pair of morphisms $C \xrightarrow{f} D \xrightarrow{g} C$ in $\mathcal{C}$ is an *embedding-retraction-pair (e-r-p)* provided $(U(f), U(g))$ is an adjoint pair, and $U(f)$ is injective.[8] Further, such an e-r-p is said to be *finite* if $U(C)$ is finite.

We have the following easy duality result.

Proposition 4.2.6 *In Stone duality, the dual of a (finite) embedding-retraction-pair on either side of the duality is a (finite) embedding-retraction-pair on the other side.*

We may then define bifiniteness in the setting of spectral spaces, rather than in the setting of domains as it is customarily done.

Definition 4.2.7 Let X be a spectral space, and L its dual lattice. We say that X and L are *bifinite* provided the following two equivalent conditions are satisfied:

1. X is the cofiltered limit of the retractions of its finite e-r-p's;
2. L is the filtered colimit of the embeddings of its finite e-r-p's.

The following proposition, which clearly implies that a bifinite lattice must have enough join primes, allows us to conclude that bifinite spectral spaces are bifinite domains. Thus, the above definition is no more general than the standard one.

[8] It follows from these two conditions that $U(f)$ is an embedding with left inverse $U(g)$.

Proposition 4.2.8 *Let L be a distributive lattice and $K \subseteq L$ a finite sublattice. Then the following conditions are equivalent:*

1. *There is a lattice homomorphism $h\colon L \to K$ making (i, h) an embedding-retraction-pair, where $i\colon K \to L$ is the inclusion;*
2. *(i) For all $b \in L$, ${\downarrow}b \cap K$ is a principal downset;*
   (ii) $J(K) \subseteq J(L)$.

4.3 Quantifiers, Free Constructions and Duality

In the categorical logic approach, cf. Sects. 4.1 and 4.2.2, the stratification of the algebra of formulas (up to logical equivalence modulo T) provided by the hyperdoctrine $P\colon \mathbf{Con}^{\mathrm{op}} \to \mathbf{BA}$ is in a sense impredicative. Indeed, it starts from the algebra of sentences $P(\emptyset)$, which is what we ultimately want to understand, to build all formulas on a countably infinite set of variables. This contrasts with the step-wise construction of algebras of formulas outlined in Sect. 4.2.1.

We want to understand quantification as a step-by-step construction. To this end, in this section we analyse from a duality theoretic viewpoint the inductive process of applying a layer of quantifiers in three settings. First, we focus on existential quantification in first-order logic over arbitrary structures. Then, on semiring and probabilistic quantifiers in first-order logic over finite structures.

As explained in Sect. 4.1, Lindenbaum-Tarski algebras of predicate logics typically fail to be free algebras. The challenge then consists, in a sense, in building free objects which approximate the Lindenbaum-Tarski algebra we are interested in. We illustrate this idea in the following examples.

4.3.1 Existential Quantification and Vietoris

For existential quantification in first-order logic, the framework can be loosely described as follows. Assume we are given a Boolean algebra of formulas B, and we build a new Boolean algebra $B_{\exists x}$ by adding a layer of the quantifier $\exists x$ to the formulas in B. We then have a quotient map

$$\mathbb{M}(B) \twoheadrightarrow B_{\exists x}$$

sending $\Diamond\varphi$ to $\exists x.\varphi$, where $\mathbb{M}(B)$ is the Boolean algebra obtained by freely adding one layer of modality as described in Sect. 4.2.1. Dually, we get a continuous embedding

$$\mathcal{V}(X) \hookleftarrow X_{\exists x}$$

where X and $X_{\exists x}$ are the dual spaces of B and $B_{\exists x}$, respectively. We have approximated the space $B_{\exists x}$ by means of the Vietoris space $\mathcal{V}(X)$, whose dual is a *free* object (namely, the free modal algebra on B). The problem then consists in characterising $X_{\exists x}$ as a subspace of $\mathcal{V}(X)$. This is addressed by observing that $X_{\exists x}$ is the image of a continuous map into $\mathcal{V}(X)$ constructed in a canonical way. In the remaining of this section we provide the necessary details.

Recall from Sect. 4.2.2 that a first-order formula $\varphi \in \mathrm{FO}_\omega(T)$ can be identified with the set $[\![\varphi]\!] \subseteq \mathrm{Mod}_\omega/\!\equiv$ consisting of the (equivalence classes of) models with assignments satisfying φ. If the free variables of φ are contained in $v_1, \ldots, v_n$, we can restrict the variable assignments accordingly. Write

$$\mathrm{Mod}_n = \{[(A, \alpha\colon \{v_1, \ldots, v_n\} \to A)] \mid A \text{ is a } \sigma\text{-structure and } A \models T\},$$

where $[(A, \alpha)] = [(A', \alpha')]$ if and only if $A, \alpha \models \varphi \Leftrightarrow A', \alpha' \models \varphi$ for every $\varphi \in \mathrm{FO}_n(T)$. Henceforth, we abuse notation and denote an arbitrary element of Mod_n by (A, α) instead of $[(A, \alpha)]$. Then, $\mathrm{FO}_n(T)$ embeds into $\mathcal{P}(\mathrm{Mod}_n)$ via the map

$$\mathrm{FO}_n(T) \hookrightarrow \mathcal{P}(\mathrm{Mod}_n), \quad [\varphi] \mapsto [\![\varphi]\!]_n = \{(A, \alpha) \in \mathrm{Mod}_n \mid A, \alpha \models \varphi\}.$$

The projection map

$$\pi_i\colon \mathrm{Mod}_n \twoheadrightarrow \mathrm{Mod}_{n\setminus i}$$

which forgets the value of the assignments on the variable v_i induces a Boolean algebra embedding

$$\pi_i^{-1}\colon \mathcal{P}(\mathrm{Mod}_{n\setminus i}) \hookrightarrow \mathcal{P}(\mathrm{Mod}_n)$$

by applying the contravariant power-set functor. As in the hyperdoctrine approach, the homomorphism π_i^{-1} has a lower adjoint and it is given by taking direct images under π_i.

$$\begin{tikzcd} \mathcal{P}(\mathrm{Mod}_{n\setminus i}) \arrow[r, bend left, "\pi_i^{-1}"] \arrow[r, phantom, "\top"] & \mathcal{P}(\mathrm{Mod}_n) \arrow[l, bend left, "\pi_i(-)"] \end{tikzcd}$$

This lower adjoint map can be thought of as the quantifier $\exists v_i$. Indeed, it is readily seen that $\pi_i([\![\varphi]\!]_n) = [\![\exists v_i.\varphi]\!]_{n\setminus i}$. More generally, abstracting away from the Boolean subalgebra $\mathrm{FO}_n(T) \hookrightarrow \mathcal{P}(\mathrm{Mod}_n)$, we can consider any Boolean algebra embedding

$$j\colon B \hookrightarrow \mathcal{P}(\mathrm{Mod}_n)$$

and regard it as a 'semantically given logic'. The Boolean algebra obtained by adding a layer of the quantifier $\exists v_i$ to B can be identified with the Boolean subalgebra $B_\exists^i$ of $\mathcal{P}(\mathrm{Mod}_{n\setminus i})$ generated by the set of direct images

$$\{\pi_i(j(\varphi)) \mid \varphi \in B\}.$$

We now focus on the dual of the transformation $B \rightsquigarrow B^i_\exists$. Let $f\colon \beta(\mathrm{Mod}_n) \twoheadrightarrow X$ be the continuous map dual to $j\colon B \hookrightarrow \mathcal{P}(\mathrm{Mod}_n)$. Here, $\beta(\mathrm{Mod}_n)$ denotes the Čech-Stone compactification of Mod_n regarded as a discrete space, and is the dual Stone space of $\mathcal{P}(\mathrm{Mod}_n)$. We obtain a continuous map

$$R\colon \beta(\mathrm{Mod}_{n\setminus i}) \xrightarrow{\beta(\pi_i)^{-1}} \mathcal{V}(\beta(\mathrm{Mod}_n)) \xrightarrow{\mathcal{V}(f)} \mathcal{V}(X).$$

The first component of R is the preimage map $x \mapsto \beta(\pi_i)^{-1}(x)$, where the function $\beta(\pi_i)\colon \beta(\mathrm{Mod}_n) \to \beta(\mathrm{Mod}_{n\setminus i})$ is the Stone dual of $\pi_i^{-1}\colon \mathcal{P}(\mathrm{Mod}_{n\setminus i}) \to \mathcal{P}(\mathrm{Mod}_n)$. The map $\beta(\pi_i)^{-1}$ is continuous because π_i^{-1} has a lower adjoint. Indeed, the join-semilattice homomorphism $\pi_i(-)\colon \mathcal{P}(\mathrm{Mod}_n) \to \mathcal{P}(\mathrm{Mod}_{n\setminus i})$ induces a Boolean algebra homomorphism $\mathbb{M}(\mathcal{P}(\mathrm{Mod}_n)) \to \mathcal{P}(\mathrm{Mod}_{n\setminus i})$, whose dual map is precisely $\beta(\pi_i)^{-1}$.

We then have the following result.

Proposition 4.3.1 *The image of the continuous map* $R\colon \beta(\mathrm{Mod}_{n\setminus i}) \to \mathcal{V}(X)$ *is the dual space of* $B^i_\exists$.

Proof It is not difficult to verify that $R^{-1}(\Diamond\widehat{\varphi}) = \widehat{\pi_i(j(\varphi))}$ for every $\varphi \in B$, see e.g. Corollary 3.2 of Borlido & Gehrke (2019). Consequently, the Boolean algebra dual to the image of R can be identified with the subalgebra of $\mathcal{P}(\mathrm{Mod}_{n\setminus i})$ generated by the elements of the form $\pi_i(j(\varphi))$ for $\varphi \in B$, which is precisely $B^i_\exists$. $\square$

To sum up, the transformation $B \rightsquigarrow B^i_\exists$ which adds one layer of quantifier $\exists v_i$ dually corresponds to taking the image of the continuous map $R\colon \beta(\mathrm{Mod}_{n\setminus i}) \to \mathcal{V}(X)$, canonically constructed from the continuous function $f\colon \beta(\mathrm{Mod}_n) \twoheadrightarrow X$. For a step-by-step treatment of quantifiers, we now want to add to $B^i_\exists$ the formulas which were already in B. Hence, we take the Boolean subalgebra of $\mathcal{P}(\mathrm{Mod}_n)$ generated by the union $B \cup B^i_\exists$, which coincides with the image of the obvious Boolean algebra homomorphism $B + B^i_\exists \to \mathcal{P}(\mathrm{Mod}_n)$. This corresponds, dually, to taking the image of the continuous product map

$$\beta(\mathrm{Mod}_n) \xrightarrow{(R\circ\beta(\pi_i))\times f} \mathcal{V}(X) \times X.$$

An essential obstacle to a two-sided duality theory for quantifiers is the lack of a characterisation of the continuous maps $\beta(\mathrm{Mod}_n) \to \mathcal{V}(X) \times X$ arising this way. We will return to this point in Sect. 4.4.

4.3.2 Semiring Quantifiers and Measures

The existential quantifier $\exists$ captures the existence, or non-existence, of an element satisfying a property. As such, it is a two-valued query. Semiring quantifiers, as studied for instance in logic on words, generalise $\exists$ by allowing us to count the number of witnesses in a given semiring.[9] Recall that a *semiring* is a tuple $(S, +, \cdot, 0, 1)$ where $(S, +, 0)$ is a commutative monoid, $(S, \cdot, 1)$ is a monoid, the operation $\cdot$ distributes over $+$, and $0 \cdot s = 0 = s \cdot 0$ for all $s \in S$. If S is a fixed finite semiring, every element $k \in S$ determines a quantifier $\exists_k$. Given a first-order formula φ with one free variable v and a finite structure A, the semantics of the sentence $\exists_k v.\varphi(v)$ is given as follows:

$$A \models \exists_k v.\varphi(v) \quad \text{iff} \quad 1 + \cdots + 1 (\text{ repeated } m\text{-times}) \text{ is equal to } k \text{ in } S$$
$$\text{where } m \text{ is the number of elements } a \in A \text{ such that } A \models \varphi(a).$$

Notice that A must be finite, for otherwise the set $\{a \in A \mid A \models \varphi(a)\}$ may be infinite and the sum $1 + \cdots + 1$ undefined. This problem could be overcome by requiring that S be complete in an appropriate sense. The existential quantifier $\exists$ is recovered by letting $S = \mathbf{2}$ be the two-element Boolean ring and $k = 1$.

Let Fin_n be the subset of Mod_n consisting of the finite models with assignments. Given a Boolean algebra embedding $j \colon B \hookrightarrow \mathcal{P}(\mathrm{Fin}_n)$ we can construct, akin to the case of $\exists$, a Boolean algebra $B^i_{\exists_S}$ obtained by adding a layer of semiring quantifiers $\exists_k v_i$ for $k \in S$. For every $\varphi \in B$ and $(A, \alpha) \in \mathrm{Fin}_{n\setminus i}$, write $m_{\varphi,(A,\alpha)}$ for the number of elements a in A such that $(A, \alpha \cup \{v_i \mapsto a\})$ belongs to $j(\varphi)$. Then, $B^i_{\exists_S}$ can be defined as the Boolean subalgebra of $\mathcal{P}(\mathrm{Fin}_{n\setminus i})$ generated by the sets

$$\{(A, \alpha) \in \mathrm{Fin}_{n\setminus i} \mid 1 + \cdots + 1 (m_{\varphi,(A,\alpha)}\text{-times}) \text{ is equal to } k\}, \ \text{ for } \varphi \in B \text{ and } k \in S.$$

In order to describe the dual of the transformation $B \rightsquigarrow B^i_{\exists_S}$, we need to understand which construction plays the role of the Vietoris hyperspace in the case of semiring quantifiers. For this purpose, notice that the Vietoris space $\mathcal{V}(X)$ can be identified with a space of two-valued finitely additive measures on X, whenever X is a Boolean space.[10] Regard X as a measurable space where the measurable subsets are precisely the clopens, i.e. the elements of the Boolean algebra B dual to X. A finitely additive $\mathbf{2}$-valued measure on X is then a function $\mu \colon B \to \mathbf{2}$ satisfying

$$\mu(0) = 0 \quad \text{and} \quad \mu(a \vee b) \vee \mu(a \wedge b) = \mu(a) \vee \mu(b) \ \ \forall a, b \in B.$$

[9] A particular class of semiring quantifiers is given by the modular quantifiers, which count in a finite cyclic ring $\mathbb{Z}/q\mathbb{Z}$. These were introduced in logic on words in Straubing et al. (1995).

[10] Perhaps more natural would be to first identify $\mathcal{V}(X)$ with the space of filters on the dual Boolean algebra of X, as explained towards the end of Sect. 4.2.2 in the case of the Smyth powerspace, and then observe that filters can be seen as two-valued finitely additive measures.

Denote by $\mathcal{M}(X, \mathbf{2})$ the collection of all finitely additive $\mathbf{2}$-valued measures on X, and equip it with the subspace topology induced by the product topology on $\mathbf{2}^B$.

Proposition 4.3.2 *For every Boolean space X, the Vietoris hyperspace $\mathcal{V}(X)$ is homeomorphic to $\mathcal{M}(X, \mathbf{2})$ via the map*

$$\mathcal{V}(X) \to \mathcal{M}(X, \mathbf{2}), \quad C \mapsto \mu_C, \quad \textit{where} \;\; \mu_C(a) = \begin{cases} 1 & \textit{if } \widehat{a} \cap C \neq \emptyset, \\ 0 & \textit{otherwise.} \end{cases}$$

Proof It is straightforward to verify that the map in the statement is a continuous bijection, with inverse $\mathcal{M}(X, \mathbf{2}) \to \mathcal{V}(X), \mu \mapsto \bigcap \{\widehat{a} \subseteq X \mid \mu(\neg a) = 0\}$. Every continuous bijection between compact Hausdorff spaces is a homeomorphism, hence the statement follows. □

For semiring quantifiers, the hyperspace $\mathcal{V}(X)$ will thus be replaced by $\mathcal{M}(X, S)$, the space of finitely additive S-valued measures on X. An element of $\mathcal{M}(X, S)$ is a function $\mu\colon B \to S$ satisfying

$$\mu(0) = 0 \quad \text{and} \quad \mu(a \vee b) + \mu(a \wedge b) = \mu(a) + \mu(b) \;\; \forall a, b \in B, \tag{4.1}$$

and the set $\mathcal{M}(X, S)$ is equipped with the subspace topology induced by the product topology on S^B. The equations in (4.1), encoding finite additivity, translate into equaliser diagrams in the category of Boolean spaces. Hence, the resulting space $\mathcal{M}(X, S)$ is again Boolean. Explicitly, the topology of $\mathcal{M}(X, S)$ is generated by the (clopen) subsets of the form

$$[a, k] = \{\mu \in \mathcal{M}(X, S) \mid \mu(a) = k\}, \quad \text{for } a \in B \text{ and } k \in S.$$

In order to describe the dual of the construction $B \rightsquigarrow B^i_{\exists_S}$, we perform two steps. First, given a finite model with assignment $(A, \alpha) \in \mathrm{Fin}_{n\setminus i}$, let

$$\delta_{(A,\alpha)}\colon \mathrm{Fin}_n \to S \tag{4.2}$$

be the 'S-valued characteristic function' of $\pi_i^{-1}(A, \alpha)$, where $\pi_i\colon \mathrm{Fin}_n \to \mathrm{Fin}_{n\setminus i}$ is the map which forgets the assignment of the ith variable. That is, $\delta^i_{(A,\alpha)}(A', \alpha')$ is 1 if $A = A'$ and α agrees with α' on the variables $v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n$, and 0 otherwise. Since A is finite, $\delta^i_{(A,\alpha)}$ belongs to the set $\mathbf{S}(\mathrm{Fin}_n)$ of finitely supported S-valued functions on Fin_n. In the second step, in order to construct a measure, we extend the function $\delta^i_{(A,\alpha)}$ to subsets of Fin_n by adding up all the non-zero values in a given subset. More generally, if T is a set and $g\colon T \to S$ is a finitely supported function, the map

$$\int g\colon \mathcal{P}(T) \to S, \quad P \mapsto \int_P g \quad \text{computed as} \quad \sum_{x \in P} g(x)$$

is a finitely additive S-valued measure on $\beta(T)$. We obtain an integration map[11]

$$\int : \mathbf{S}(T) \to \mathcal{M}(\beta(T), S).$$

Now, let $f: \beta(\mathrm{Fin}_n) \to X$ be the dual of the embedding $j: B \hookrightarrow \mathcal{P}(\mathrm{Fin}_n)$. Consider the composite

$$\mathrm{Fin}_{n\setminus i} \xrightarrow{\delta^i_{(-)}} \mathbf{S}(\mathrm{Fin}_n) \xrightarrow{\int} \mathcal{M}(\beta(\mathrm{Fin}_n), S) \xrightarrow{f_*} \mathcal{M}(X, S) \tag{4.3}$$

where f_* sends a measure to its pushforward along f, i.e. $f_*(\mu)(a) = \mu(f^{-1}(\widehat{a}))$ for every $\mu \in \mathcal{M}(\beta(\mathrm{Fin}_n), S)$ and $a \in B$. The space $\mathcal{M}(X, S)$ is compact and Hausdorff, whence the above composition extends to a (unique) continuous function

$$R: \beta(\mathrm{Fin}_{n\setminus i}) \to \mathcal{M}(X, S). \tag{4.4}$$

The following result generalises Proposition 4.3.1 and can be proved in a similar manner (we omit the details here).

Theorem 4.3.3 *The image of the continuous map* $R: \beta(\mathrm{Mod}_{n\setminus i}) \to \mathcal{M}(X, S)$ *is the dual space of* $B^i_{\exists_S}$.

The connection between semiring quantifiers and spaces of finitely additive measures was first explored, in the context of logic on words, in Gehrke et al. (2017). The treatment in this section could be adapted to deal with any profinite semiring, such as the *tropical semiring* $(\mathbb{N} \cup \{\infty\}, \min, +, \infty, 0)$, and not just the finite ones. See Reggio (2020).

4.3.3 Probabilistic Quantifiers and Structural Limits

Topological methods are also employed in the study of structural limits in finite model theory. A systematic investigation of limits of finite structures has been developed by Nešetřil and Ossona de Mendez and is based on an embedding, called the *Stone pairing*, of the collection of finite structures into a space of probability measures (Nešetřil & Ossona de Mendez, 2012, 2020). The latter space is complete, thus it provides the limit objects for those sequences of finite structures which embed as Cauchy sequences. Although this space of measures and the Stone pairing embedding did not originate from duality, in recent work we showed that a closely related version

[11] In fact, the construction $X \mapsto \mathcal{M}(X, S)$ yields a monad on **BStone** and the integration map can be upgraded to a monad morphism $\int: \mathbf{S} \circ U \to U \circ \mathcal{M}(-, S)$, where $\mathbf{S}$ is the semiring monad on **Set** and U : **BStone** $\to$ **Set** is the forgetful functor. Cf. Gehrke et al. (2017). While, for the purpose of this section, we may assume S is any pointed monoid, the monadic treatment requires the full semiring structure.

of the Stone pairing can be understood—via duality—as the embedding of finite structures into a space of types. Namely, the space of 0-types of an extension of first-order logic obtained by adding a layer of certain probabilistic quantifiers (Gehrke et al., 2020). In the following, we highlight the similarities between the Stone pairing embedding and the space-of-measures construction introduced above in the context of existential and semiring quantification.

For every first-order formula φ with free variables contained in $v_1, \dots, v_n$, and finite structure A, the *Stone pairing* of φ and A is defined as

$$\langle \varphi, A \rangle \;=\; \frac{|\{\overline{a} \in A^n \mid A \models \varphi(\overline{a})\}|}{|A|^n}.$$

In other words, $\langle \varphi, A \rangle$ is the probability that a random assignment of the variables $v_1, \dots, v_n$ in A satisfies the formula φ. Upon fixing the second coordinate, the map $\langle -, A \rangle$ is a finitely additive measure on the dual space of the Lindenbaum-Tarski algebra of all first-order formulas FO_ω, with values in the unit interval $[0, 1]$. I.e.,

$$\langle \bot, A \rangle = 0 \quad \text{and} \quad \langle \varphi \vee \psi, A \rangle + \langle \varphi \wedge \psi, A \rangle = \langle \varphi, A \rangle + \langle \psi, A \rangle \;\; \forall \varphi, \psi \in \mathrm{FO}_\omega.$$

Since the Boolean algebra FO_ω is dual to the space of models and valuations Mod_ω, we obtain an embedding

$$\langle -, - \rangle \colon \mathrm{Fin} \longrightarrow \mathcal{M}(\mathrm{Mod}_\omega, [0, 1]), \quad A \mapsto \langle -, A \rangle$$

where Fin is the collection of finite structures, up to isomorphism (with the notation of Sect. 4.3.2, $\mathrm{Fin} = \mathrm{Fin}_0$). This is the *Stone pairing* embedding introduced by Nešetřil and Ossona de Mendez.

By restricting $\langle -, A \rangle$ to suitable fragments of first-order logic, Nešetřil and Ossona de Mendez obtained a unifying framework that captures various notions of convergence of finite structures, such as Lovasz–Szegedy convergence, Benjamini–Schramm convergence, elementary convergence, etc.[12] Their insight was that each of these notions of convergence corresponds to a fragment of first-order logic. Further, since the ensuing spaces of finitely additive measures are complete, they admit a limit for every sequence of finite structures which embeds as a Cauchy sequence.

In Sect. 4.3.2, we defined a map from a set of finite structures with evaluations into a space of finitely additive measures, see Eq. (4.3), and showed that it dually captures the adding of a layer of semiring quantifiers. By analogy, we may ask if the Stone pairing also corresponds to applying a layer of quantifiers. One immediate obstacle is that the spaces $[0, 1]$ and $\mathcal{M}(\mathrm{Mod}_\omega, [0, 1])$ are not Boolean, whence not amenable to the methods of Stone duality for Boolean algebras.

We can overcome this problem by replacing $[0, 1]$ with a profinite version of the unit interval obtained from a codirected system of finitary approximations of real

[12] Note that the restriction of the Stone pairing embedding to a fragment of FO may fail to be injective.

numbers in $[0, 1]$. This profinite space Γ is naturally equipped with a Priestley space structure and can therefore be studied using Stone-Priestley duality for distributive lattices. To define Γ, we divide the unit interval into n segments of equal length, i.e.

$$\Gamma_n = \{0 < \tfrac{1}{n} < \tfrac{2}{n} < \cdots < 1\}.$$

The chain Γ_n provides a finite approximation of $[0, 1]$. The higher the value of $n \in \mathbb{N}$, the better the approximation is. Whenever $n \mid m$, we consider the flooring function $\Gamma_m \to \Gamma_n$ sending $\frac{a}{m}$ to the largest $\frac{b}{n} \in \Gamma_n$ such that $\frac{b}{n} \leq \frac{a}{m}$. Note that the finite chains Γ_n with flooring functions between them form a codirected diagram in the category $\mathbf{Pos}_f$ of finite posets with monotone maps. The limit of this diagram is an object Γ of the pro-completion of $\mathbf{Pos}_f$, which is the category of Priestley spaces with continuous monotone maps.[13] See e.g. Corollary VI.3.3 in Johnstone (1986).

Concretely, the elements of Γ are the sequences of approximations $(x_n)_n \in \prod_{n\in\mathbb{N}} \Gamma_n$ which are compatible with the flooring functions. Every $q \in (0, 1]$ determines an element $q^- \in \Gamma$, namely the sequence

$$q^- = (q_1^-, q_2^-, q_3^-, \ldots) \quad \text{where} \quad q_n^- = \max\{\tfrac{a}{n} \in \Gamma_n \mid \tfrac{a}{n} < q\}$$

which approximates q from below while never reaching it. Further, if q is rational, we also get a lower approximating sequence $q^\circ \in \Gamma$ which eventually stabilises at q:

$$q^\circ = (q_1^\circ, q_2^\circ, q_3^\circ, \ldots) \quad \text{where} \quad q_n^\circ = \max\{\tfrac{a}{n} \in \Gamma_n \mid \tfrac{a}{n} \leq q\}.$$

In fact, any point of Γ is of one of these two types. We can thus think of Γ as a copy of the unit interval where all the non-zero rationals are doubled (in the picture, q is rational while r is irrational):

$\mathbf{\Gamma}$ =


Equivalently, Γ is a copy of the Cantor space with an extra top element which is topologically isolated (corresponding to 1°). The natural order of Γ, illustrated in the previous picture, is the total order defined by the two conditions

- $r^\circ < s^-$ if and only if $r < s$ in $[0, 1]$, and
- $q^- < q^\circ$ for every $q \in (0, 1]$,

and its topology is the interval topology. Note that Γ retracts onto $[0, 1]$. Indeed, the continuous surjection

$$\gamma\colon \Gamma \to [0, 1], \quad q^-, q^\circ \mapsto q$$

[13] A *Priestley space* is a pair $(X, \leq)$ where X is a compact space and $\leq$ is a partial order such that, whenever $x \not\leq y$, there is a clopen subset $C \subseteq X$ which is upward closed and satisfies $x \in C$ and $y \notin C$.

has a (lower semicontinuous) section

$$\iota\colon [0,1] \to \Gamma, \quad \iota(q) = \begin{cases} q^\circ & \text{if } q \text{ is rational} \\ q^- & \text{otherwise.} \end{cases}$$

The additive structure of [0, 1] lifts to Γ (as can be derived by duality for additional operators) so that it makes sense to consider the set $\mathcal{M}(X,\Gamma)$ of finitely additive probability measures on a Boolean space X with values in Γ. This construction can be generalised to any Priestley space X, and it turns out that the assignment $X \mapsto \mathcal{M}(X,\Gamma)$ is an endofunctor on the category of Priestley spaces. In particular, a continuous monotone map of Priestley spaces $f\colon X \to Y$ is sent to the map

$$f_*\colon \mathcal{M}(X,\Gamma) \to \mathcal{M}(Y,\Gamma)$$

taking a measure to its pushforward along f. Furthermore, the retraction-section pair $\gamma\colon \Gamma \leftrightarrows [0,1] :\iota$ lifts to a retraction-section pair

$$\gamma^\#\colon \mathcal{M}(X,\Gamma) \leftrightarrows \mathcal{M}(X,[0,1]) :\iota^\#, \quad \text{where} \quad \gamma^\#(\mu) = \gamma \circ \mu \quad \text{and} \quad \iota^\#(\mu) = \iota \circ \mu.$$

Now we define a Γ-valued variant of the Stone pairing by following the strategy set out in Sect. 4.3.2 in the case of semiring quantifiers. Fix $n \in \mathbb{N}$, and let $\mathcal{F}(\mathrm{Fin}_n,\Gamma)$ be the set of finitely supported functions $\mathrm{Fin}_n \to \Gamma$ with total value 1°. We get a map $\delta_{(-)}\colon \mathrm{Fin} \to \mathcal{F}(\mathrm{Fin}_n,\Gamma)$ sending a finite structure A to

$$\delta_A\colon \mathrm{Fin}_n \to \Gamma, \quad \text{where} \quad \delta_A(A',\alpha') = \begin{cases} \left(\frac{1}{|A|^n}\right)^\circ & \text{if } A' = A \\ 0^\circ & \text{otherwise.} \end{cases}$$

The map $\delta_{(-)}$ is the (normalized) Γ-valued version of the function introduced in (4.2) for semiring quantifiers. In a similar way, to move from finitely supported functions to measures, for every set T we consider the integration map

$$\int\colon \mathcal{F}(T,\Gamma) \to \mathcal{M}(\beta(T),\Gamma), \quad f \mapsto \int f.$$

Lastly, define the following composition

$$R_n\colon \mathrm{Fin} \xrightarrow{\delta_{(-)}} \mathcal{F}(\mathrm{Fin}_n,\Gamma) \xrightarrow{\int} \mathcal{M}(\beta(\mathrm{Fin}_n),\Gamma) \xrightarrow{f_*} \mathcal{M}(\mathrm{Mod}_n,\Gamma)$$

where $f\colon \beta(\mathrm{Fin}_n) \to \mathrm{Mod}_n$ is the dual map of the Boolean algebra homomorphism

$$\mathrm{FO}_n \to \mathcal{P}(\mathrm{Fin}_n), \quad \varphi \mapsto [\![\varphi]\!] \cap \mathrm{Fin}_n.$$

The map R_n can be extended to a continuous function $\widetilde{R}_n : \beta(\mathrm{Fin}) \to \mathcal{M}(\mathrm{Mod}_n, \Gamma)$, corresponding to the map in (4.4). Using the fact that the space Mod_ω is the codirected limit of the Mod_n's for $n \in \mathbb{N}$, and the functor $\mathcal{M}(-, \Gamma)$ preserves codirected limits, we can 'glue' the maps $\widetilde{R}_n$ to get a continuous function $\widetilde{R} : \beta(\mathrm{Fin}) \to \mathcal{M}(\mathrm{Mod}_\omega, \Gamma)$. The restriction $R : \mathrm{Fin} \to \mathcal{M}(\mathrm{Mod}_\omega, \Gamma)$ of $\widetilde{R}$ is an equivalent Γ-valued version of the Stone pairing, as expressed by the commutativity of the following diagram.

$$
\begin{tikzcd}
 & \mathcal{M}(\mathrm{Mod}_\omega, \Gamma) \arrow[dd, bend left, "\gamma^\#"] \\
\mathrm{Fin} \arrow[ur, "R"] \arrow[dr, "<-,->"'] & \\
 & \mathcal{M}(\mathrm{Mod}_\omega, [0,1]) \arrow[uu, bend left, "\iota^\#"]
\end{tikzcd}
$$

The map R, and more precisely the way it is constructed, provides an interesting link between the theory of structural limits and the inductive study of semiring quantifiers. Further, the duality approach allows us to see (the Γ-valued version of) the Stone pairing as an embedding of the finite structures into a space of types. This is the content of the following theorem, which is a special case of more general results in Gehrke et al. (2020).

Theorem 4.3.4 *The Boolean space* $\mathcal{M}(\mathrm{Mod}_\omega, \Gamma)$ *is dual to the Lindenbaum-Tarski algebra of the propositional logic having as atoms* $\mathbf{p}_{\geq q}\,\varphi$ *and* $\mathbf{p}_{<q}\,\varphi$*, for each* $\varphi \in \mathrm{FO}_\omega$ *and* $q \in [0,1] \cap \mathbb{Q}$*, and the following inference rules (along with the usual ones for the Boolean connectives):*

$$
\frac{\mathbf{p}_{\geq q}\,\varphi}{\mathbf{p}_{\geq p}\,\varphi}\ (\text{if } p \leq q) \qquad \frac{\mathbf{p}_{\geq q}\,\varphi}{\mathbf{p}_{\geq q}\,\psi}\ (\text{if } \varphi \vdash \psi) \qquad \frac{}{\mathbf{p}_{\geq 0}\,\bot} \qquad \frac{}{\mathbf{p}_{<q}\,\bot}\ (\text{if } q > 0) \qquad \frac{}{\mathbf{p}_{\geq q}\,\top} \qquad \frac{\mathbf{p}_{\geq q}\,\varphi}{\neg\mathbf{p}_{<q}\,\varphi}
$$

$$
\frac{\mathbf{p}_{\geq p}\,\varphi \wedge \mathbf{p}_{\geq q}\,\psi}{\mathbf{p}_{\geq p+q-r}\,(\varphi \vee \psi) \vee \mathbf{p}_{\geq r}\,(\varphi \wedge \psi)} \qquad \frac{\mathbf{p}_{\geq p+q-r}\,(\varphi \vee \psi) \wedge \mathbf{p}_{\geq r}\,(\varphi \wedge \psi)}{\mathbf{p}_{\geq p}\,\varphi \vee \mathbf{p}_{\geq q}\,\psi}\ (\text{if } 0 \leq p+q-r \leq 1)
$$

The intended models for this extension of FO are the measures $\mu \in \mathcal{M}(\mathrm{Mod}_\omega, \Gamma)$, and the probabilistic quantifiers $\mathbf{p}_{\geq q}$ and $\mathbf{p}_{<q}$ are interpreted as follows:

$$
\mu \models \mathbf{p}_{\geq q}\,\varphi \ \Leftrightarrow\ \mu(\varphi) \geq q^\circ \quad \text{and} \quad \mu \models \mathbf{p}_{<q}\,\varphi \ \Leftrightarrow\ \mu(\varphi) < q^\circ.
$$

In particular, if A is a finite structure, $< -, A > \models \mathbf{p}_{\geq q}\,\varphi$ if and only if φ is satisfied in A with probability at least q. Similarly for $\mathbf{p}_{<q}\,\varphi$. Note that these probabilistic quantifiers bind all free variables in a formula. Thus, once applied a layer of quantifiers to FO_ω, we obtain an algebra of *sentences*. These sentences are seen as propositional atoms for a new logic and, by the previous theorem, the Stone pairing can be seen as embedding the collection of finite structures (up to isomorphism) into the space of 0-types for this logic.

Therefore, we see that Nešetřil and Ossona de Mendez's Stone pairing dually corresponds to adding a layer of probabilistic quantifiers. As such, it can be regarded as an instance of the inductive approach described in Sect. 4.2.1.

4.4 Outlook

We saw in Sect. 4.3.1 that adding a layer of existential quantifier $\exists$ to a Boolean algebra B of first-order formulas (with free variables in $v_1, \ldots, v_n$) dually corresponds to taking the image of a continuous map $\beta(\mathrm{Mod}_n) \to \mathcal{V}(X) \times X$, where X is the dual Stone space of B. A similar statement holds for semiring quantifiers, cf. Sect. 4.3.2. This continuous map is defined in a canonical way, and ensures the *soundness* of the construction. But we do not know, so far, how to characterise the continuous maps $\beta(\mathrm{Mod}_n) \to \mathcal{V}(X) \times X$ arising in this manner, which would establish the *completeness* of the construction. This is a notable obstacle to a full duality theoretic understanding of step-by-step quantification in predicate logics. On the other hand, such a completeness result is available for semiring quantifiers in logic on words, and makes use of the richer structure of the spaces of models (in the form of monoid actions). See Proposition VI.7 and Theorem VI.8 of Gehrke et al. (2017), where this is called a 'Reutenauer-type theorem'. A question arises, whose answer would significantly further the use of topological methods in logic: *Is there a Reutenauer-type result for first-order logic over arbitrary structures?*

In this paper we have discussed several examples of topological methods in logic and computer science, highlighting their duality theoretic nature. However, there are topological methods in logic which have been successfully developed and applied, but for which no duality theoretic explanation is available so far. An appealing example is the theory of limits of schema mappings as developed in database theory by Kolaitis and his collaborators Kolaitis et al. (2018). Understanding these tools and results from a duality theoretic perspective may yield new useful insights and is an exciting venue for future investigations. Another example are 0–1 laws in finite model theory, illustrating the limits of the expressive power of first-order logic over finite structures, see e.g. Fagin (1976). These are only some of the many opportunities for further development of the duality approach, which would contribute to unify the 'structure' and 'power' strands in theoretical computer science.

One of the main themes of our present contribution has been the analysis of step-by-step constructions in logic, which yield *free* objects on the algebra side and *co-free* objects on the space side. Note that, even though the step-wise process of adding a layer of connectives yields a *monad* in the (co)limit, the one-step functor is typically a *comonad*. For instance, the functor on Boolean algebras which adds one layer of modality $\Diamond$ is a comonad, whose dual is the Vietoris monad on Boolean spaces.

The recent work of Samson Abramsky and his coauthors on comonads for model-theoretic games (Abramsky et al., 2017; Abramsky & Shah, 2018) is tightly related to this viewpoint. The connection between the comonadic approach and the duality one remains to be explored, and is an interesting avenue of research. In this direction, one may point out that the Ehrenfeucht-Fraïssé comonad introduced by Abramsky and Shah arises as the density comonad for a certain (contravariant) realization functor from a category of primitive positive sentences into the category of structures.

Besides the inductive treatment of quantifiers, another important theme of this paper has been the lack of freeness of Lindenbaum-Tarski algebras of first-order theories. Indeed, we pointed out that this is one of the main obstacles to a satisfactory algebraic and duality theoretic approach to predicate logics.

Another place where the lack of freeness plays an important role is quantum information and computation, to which Samson Abramsky has greatly contributed. There, as recently observed by Abramsky, the lack of freeness (of certain Boolean subalgebras of partial Boolean algebras) can be regarded as an obstruction to classicality. In fact, in the presence of freeness, the Kochen-Specker theorem does not apply. See Abramsky & Barbosa (2021). Interestingly, in this context, this obstruction represents a (quantum) advantage.

We conclude with a question concerning a wider issue, which is instrumental in addressing the divide between structure and power, one of the main focuses of Samson Abramsky's recent research. A difference between general model theory and finite model theory which is often emphasised is the fact that the major structure theorems such as compactness, Löwenheim-Skolem, etc. do not carry over to the finite setting. Rossman's Finite Homomorphism Preservation Theorem is a major advance because it provides such a theorem which does persist in the finite setting. Another take on this would be to conjecture that *topological variants of all the classical structure theorems* hold in the finite setting. A first result in this direction is Reiterman's theorem for finite algebras, which shows that Birkhoff's variety theorem has a finite variant once we topologize. *In weaker logics of resources, as studied for example in finite model theory, is there a topological component missing at the level of the associated Lawvere theories/categorical semantics?*

References

Abramsky, S. (1987). Domain theory in logical form. In *Proceedings of the second annual IEEE symposium on logic in computer science (LICS 1987)* (pp. 47–53). IEEE Computer Society Press.

Abramsky, S. (1991). Domain theory in logical form. *Annals of Pure and Applied Logic, 51*, 1–77.

Abramsky, S. (2005). A Cook's tour of the finitary non-well-founded sets. In S. Artemov, H. Barringer, A. d'Avila Garcez, L. Lamb, & J. Woods (Eds.), *We will show them! Essays in honour of Dov Gabbay* (Vol. 1, pp. 1–18). College Publications.

Abramsky, S., & Barbosa, R. S. (2021). The logic of contextuality. In 29th EACSL annual conference on computer science logic, CSL (pp. 5:1–5:18).

Abramsky, S., Dawar, A., & Wang, P. (2017). The pebbling comonad in finite model theory. In *32nd annual ACM/IEEE symposium on logic in computer science, LICS* (pp. 1–12).

Abramsky, S., & Jung, A. (1995). *Domain Theory* (pp. 1–168). USA: Oxford University Press, Inc..

Abramsky, S., & Shah, N. (2018). Relating Structure and Power: Comonadic semantics for computational resources. In *27th EACSL annual conference on computer science logic, CSL* (pp. 2:1–2:17).

Almeida, J. (1989). Residually finite congruences and quasi-regular subsets in uniform algebras. *Portuguese Mathematical Society, 46*(3), 313–328.

Almeida, J. (1994). *Finite semigroups and universal algebra*. River Edge, NJ: World Scientific Publishing Co. Inc. Translated from the 1992 Portuguese original and revised by the author.

Birkhoff, G. (1937). Moore-Smith convergence in general topology. *Annals of Mathematics, 38*(1), 39–56.

Blackburn, P., de Rijke, M., & Venema, Y. (2001). *Modal logic*, *Cambridge Tracts in Theoretical Computer Science* (Vol. 53). Cambridge University Press.

Borlido, C., & Gehrke, M. (2019). A note on powers of Boolean spaces with internal semigroups. Preprint available at arXiv:1811.12339.

Büchi, J. R. (1966). On a decision method in restricted second order arithmetic. In E. Nagel, P. Suppes, & A. Tarski (Eds.), *Logic, methodology and philosophy of science, Studies in Logic and the Foundations of Mathematic* (Vol. 44, pp. 1–11). Elsevier.

Coumans, D. C. S. (2012). *Canonical extensions in logic*. Ph.D. thesis, Radboud Universiteit Nijmegen.

Coumans, D. C. S., & van Gool, S. J. (2012). On generalizing free algebras for a functor. *Journal of Logic and Computation, 23*(3), 645–672.

Esakia, L. L. (1974). Topological Kripke models. *Soviet Mathematics Doklady, 15*, 147–151.

Esakia, L. L. (2019). *Heyting algebras*, *Trends in logic—studia logica library* (Vol. 50). Cham: Springer. Original edition (in Russian) 1985.

Fagin, R. (1976). Probabilities on finite models. *Journal of Symbolic Logic, 41*(1), 50–58.

Font, J. M., & Verdú, V. (1991). Algebraic logic for classical conjunction and disjunction. *Studia Logica, 50*(3), 391–419.

Gehrke, M. (2016). Stone duality, topological algebra, and recognition. *Journal of Pure and Applied Algebra, 220*(7), 2711–2747.

Gehrke, M., & Bezhanishvili, N. (2011). Finitely generated free Heyting algebras via Birkhoff duality and coalgebra. *Logical Methods in Computer Science, 7*(2:9), 1–24.

Gehrke, M., Grigorieff, S., & Pin, J. -É. (2008). Duality and equational theory of regular languages. *Automata, languages and programming II*, *LNCS* (Vol. 5126, pp. 246–257). Berlin: Springer.

Gehrke, M., Jakl, T., & Reggio, L. (2020). A duality theoretic view on limits of finite structures. In Goubault-Larrecq, J., & König, B. (Eds.), *FOSSACS - Proceedings of the 23rd international conference*, *Lecture notes in computer science* (Vol. 12077, pp. 299–318). Springer.

Gehrke, M., & Krebs, A. (2017). Stone duality for languages and complexity. *ACM SIGLOG News, 4*(2), 29–53.

Gehrke, M., Petrişan, D., & Reggio, L. (2017). Quantifiers on languages and codensity monads. In *32nd annual ACM/IEEE symposium on logic in computer science, LICS* (pp. 1–12).

Gehrke, M. & van Gool, S. J. (2023). Topological duality for distributive lattices: theory and applications. Draft of text book on Stone duality to be published by Cambridge University Press.

Ghilardi, S. (1992). Free Heyting algebras as bi-Heyting algebras. *Comptes Rendus Mathematique Reports Academy Science Canada, 14*(6), 240–244.

Ghilardi, S. (1995). An algebraic theory of normal forms. *Annals of Pure and Applied Logic, 71*, 189–245.

Ghilardi, S. (2004). Unification, finite duality and projectivity in varieties of Heyting algebras. *Annals of Pure and Applied Logic, 127*, 99–115.

Ghilardi, S. (2010). Continuity, freeness, and filtrations. *Journal of Applied Non-Classical Logics, 20*(3), 193–217.

Ghilardi, S., & van Gool, S. (2016). Monadic second order logic as the model companion of temporal logic. In *Proceedings of the 31st annual ACM/IEEE symposium on logic in computer science*, *LICS '16* (pp. 417–426), New York, NY, USA: Association for Computing Machinery.

van Gool, S. J., & Steinberg, B. (2017). Pro-aperiodic monoids via saturated models. In *34th symposium on theoretical aspects of computer science, STACS* (pp. 39:1–39:14).

Johnstone, P. T. (1986). *Stone spaces*, *Cambridge Studies in Advanced Mathematics* (Vol. 3). Cambridge University Press. Reprint of the 1982 edition.

Johnstone, P. T. (2002). *Sketches of an elephant: a topos theory compendium. Vol. 2*, *Oxford Logic Guides* (Vol. 44). Oxford: The Clarendon Press, Oxford University Press.

Jónsson, B., & Tarski, A. (1951). Boolean algebras with operators I. *American Journal of Mathematics, 73*, 891–939.

Jónsson, B., & Tarski, A. (1952). Boolean algebras with operators II. *American Journal of Mathematics, 74*, 127–162.

Kolaitis, P. G., Pichler, R., Sallinger, E., & Savenkov, V. (2018). Limits of schema mappings. *Theory of Computing Systems, 62*(4), 899–940.

Kupke, C., Kurz, A., & Venema, Y. (2004). Stone coalgebras. *Theoretical Computer Science, 327*(1), 109–134. Selected Papers of CMCS '03.

Kurz, A., & Rosický, J. (2012). Strongly complete logics for coalgebras. *Logical Methods in Computer Science, 8*(3:14), 1–32.

Larsen, K. G., & Winskel, G. (1991). Using information systems to solve recursive domain equations. *Information and Computation, 91*(2), 232–258.

Macintyre, A. (2003). Model theory: Geometrical and set-theoretic aspects and prospects. *Bulletin of Symbolic Logic, 9*(2), 197–212.

Makkai, M., & Reyes, G. E. (1977). *First order categorical logic*, *Lecture notes in mathematics* (Vol. 611). Berlin-New York: Springer.

McNaughton, R., & Papert, S. (1971). *Counter-free automata*. Cambridge, Mass.-London: The M.I.T. Press. With an appendix by William Henneman, M.I.T. Research Monograph, No. 65.

Michael, E. (1951). Topologies on spaces of subsets. *Transactions of the American Mathematical Society, 71*, 152–182.

Monk, J. D. (1986). The contributions of Alfred Tarski to algebraic logic. *The Journal of Symbolic Logic, 51*(4), 899–906.

Morley, M. (1974). Applications of topology to $L_{\omega_1,\omega}$. In *Proceedings of the Tarski symposium (Proceedings of symposium pure mathematics, Vol. XXV, Univ. California, Berkeley, Calif., 1971)* (pp. 233–240).

Myers, D. (1976). Cylindric algebras of first-order languages. *Transactions of the American Mathematical Society, 216*, 189–202.

Nešetřil, J., & Ossona de Mendez, P. (2012). A model theory approach to structural limits. *Commentationes Mathematicae Universitatis Carolinae, 53*(4), 581–603.

Nešetřil, J., & Ossona de Mendez, P. (2020). A unified approach to structural limits and limits of graphs with bounded tree-depth. *Memoirs of the American Mathematical Society, 263*(1272).

Pin, J.-É. (2009). Profinite methods in automata theory. In Albers, S., & Marion, J.-Y., (Eds.), *26th International symposium on theoretical aspects of computer science STACS 2009* (pp. 31–50).

Pippenger, N. (1997). Regular languages and Stone duality. *Theory of Computing Systems, 30*(2), 121–134.

Pitts, A. M. (1983). An application of open maps to categorical logic. *Journal of Pure and Applied Algebra, 29*, 313–326.

Plotkin, G. D. (1976). A powerdomain construction. *SIAM Journal on Computing, 5*(3), 452–487.

Priestley, H. A. (1970). Representation of distributive lattices by means of ordered Stone spaces. *Bulletin of the London Mathematical Society, 2*, 186–190.

Rasiowa, H., & Sikorski, R. (1950). A proof of the completeness theorem of Gödel. *Fundamenta Mathematicae, 37*, 193–200.

Rautenberg, W., Walter, F., & Zakharyaschev, M. (2006). Willem Blok and modal logic. *Studia Logica, 83*(1), 15–30.

Reggio, L. (2020). Codensity, profiniteness and algebras of semiring-valued measures. *Journal of Pure and Applied Algebra, 224*(1), 181–205.

Reiterman, J. (1982). The Birkhoff theorem for finite algebras. *Algebra Universalis, 14*(1), 1–10.

Rutten, J. J. M. M., & Turi, D. (1993). On the foundations of final semantics: Non-standard sets, metric spaces, partial orders. In J. W. de Bakker, W. P. de Roever, & G. Rozenberg (Eds.), *Semantics: Foundations and applications* (pp. 477–530). Berlin Heidelberg: Springer.

Schützenberger, M.-P. (1965). On finite monoids having only trivial subgroups. *Information and Control, 8*(2), 190–194.

Scott, D., & Strachey, C. (1971). Towards a mathematical semantics for computer languages. In *Proceedings of the symposium on computers and automata*, *Microwave Research Institute symposia series* (pp. 19–46). Polytechnic Institute of Brooklyn Press. Also Tech. Mon. PRG-6, Oxford U. Computing Lab.

Smyth, M. B. (1983). Power domains and predicate transformers: A topological view. In J. Diaz (Ed.), *Automata, languages and programming* (pp. 662–675). Berlin Heidelberg: Springer.

Smyth, M. B., & Plotkin, G. D. (1982). The category-theoretic solution of recursive domain equations. *SIAM Journal on Computing, 11*(4), 761–783.

Stone, M. H. (1936). The theory of representations for Boolean algebras. *Transactions of the American Mathematical Society, 40*(1), 37–111.

Stone, M. H. (1938). Topological representations of distributive lattices and Brouwerian logics. *Časopis pro pěsiování matematiky a fysiky, 67*, 1–25.

Straubing, H., Therien, D., & Thomas, W. (1995). Regular languages defined with generalized quantifiers. *Information and Computation, 118*(2), 289–301.

Venema, Y., & Vosmaer, J. (2014). Modal logic and the Vietoris functor. In *Leo Esakia on duality in modal and intuitionistic logics*, *Outst. Contrib. Log.* (Vol. 4, pp. 119–153). Dordrecht: Springer.

Vietoris, L. (1922). Bereiche zweiter Ordnung. *Monatshefte Mathematical Physics, 32*(1), 258–280.

Chapter 5
Stone Duality for Relations

Alexander Kurz, Andrew Moshier, and Achim Jung

Abstract We show how Stone duality can be extended from maps to relations. This is achieved by working order-enriched and defining a relation from A to B as both an order-preserving function $A^{\mathrm{op}} \times B \to \mathbb{2}$ and as a subobject of $A \times B$. We show that dual adjunctions and equivalences between regular categories, taken in a suitably order-enriched sense, extend to (framed bi)categories of relations.

5.1 Introduction

In this article we will extend Stone-type dualities from maps to relations. We view relations $A \looparrowright B$ as generalising functions, not as generalising subsets. Accordingly, composition of relations $A \looparrowright B$ and $B \looparrowright C$ and the functorial embedding from maps $A \to B$ into relations $A \looparrowright B$ will play a major role. On the other hand, relations $R \subseteq A_1 \times \ldots A_n$ as generalised subsets are outside of the scope of this paper.

Motivation stems, independently, from domain theory and from duality theory, as we will explain in more detail now.

Domain Theory. Starting from Scott (1970), domain theory is, at least in part, concerned with describing infinite data as well as continuous functions via finite approximants. This leads to Scott's algebraic domains and approximable maps, the latter being relations between the finite approximants of two domains that capture continuity of functions between the domains themselves. Smyth (1992) continued the development of this idea by supposing the finite approximants play the role of propo-

A. Kurz (✉) · A. Moshier
Fowler School of Engineering, Chapman University, Orange, CA, USA
e-mail: akurz@chapman.edu

A. Moshier
e-mail: moshier@chapman.edu

A. Jung
School of Computer Science, University of Birmingham, Birmingham, England
e-mail: a.jung@cs.bham.ac.uk

A. Palmigiano and M. Sadrzadeh (eds.), *Samson Abramsky on Logic and Structure in Computer Science and Beyond*, Outstanding Contributions to Logic 25,
https://doi.org/10.1007/978-3-031-24117-8_5

sitions in a logic of properties of the domain elements. Abramsky (1991) investigated a similar idea in the context of SFP domains, providing analysis of a wide variety of domain constructions in terms of relations on the corresponding distributive lattices. Jung & Sünderhauf (1996) extended the techniques to general stably compact spaces and proximity lattices (distributive lattices equipped with a suitable "way below" relation). Jung et al. (1999) then extended the Jung-Sünderhauf duality to relations on the stably compact spaces. This permitted many constructions (products, coproducts, lifting, etc.) on stably compact spaces to be dealt with by Abramsky's logical form methods. Following up on Kegelmann et al., in a more purely topological setting, Moshier (2004) establishes a duality for compact Hausdorff spaces and proximity lattices that satisfy a simple strong form of distributivity.

This duality for compact Hausdorff spaces can be derived from the duality of Boolean algebras and Stone spaces by a sequence of purely category theoretic constructions. In order to do this, one needs to work order-enriched and so the construction starts out from the duality of bounded distributive lattices and Priestley spaces (= ordered Stone spaces) and proceeds as follows.

- Extend the duality of distributive lattices and Priestley spaces from functions to relations.
- Complete these relational categories by the (ordered) Karoubi envelope (= ordered splitting of idempotents), obtaining a duality for weakening relations of continuous spaces.
- Restrict this relational duality to maps.

Each step in this construction is purely categorical and, therefore, preserves dual adjunctions. In fact, starting from the dual equivalence of distributive lattices and Priestley spaces we arrive at the dual equivalence of proximity lattices and Nachbin spaces (=ordered compact Hausdorff spaces).[1]

In this paper we concentrate on the first step, which consists of extending a duality of maps to a duality of relations.

Duality Theory. For the applications we have in mind, we need that a relation $A \looparrowright B$ is both on the algebraic side and on the topological side a subobject of $A \times B$, or, in the ordered setting, an upward closed subobject of $A^{\mathrm{op}} \times B$. But since the dual of a subobject of the product is a quotient of the coproduct and not itself again a subobject of a product, this endeavor seems to be doomed to fail. One of the main points of this article is to show that in the order-enriched setting, for so-called weakening closed relations, it is possible to circumvent these problems by exploiting a duality of certain spans and cospans.

Indeed, at the heart of the construction is the observation that in the order-enriched setting relations can be both tabulated as spans and co-tabulated as cospans. This will allow us to define the Stone dual of a relation R as the cospan obtained from dualising

[1] Splitting of idempotents in relations and then restricting to maps gives the exact completion, see Carboni & Vitale (1998), Sect.3 for ordinary categories and the introduction of Lack (1999) for further references. Bourke (2010), Garner & Lack (2012); Bourke & Garner (2013) develop enriched generalisations of regularity and exactness.

the span tabulating R. The main result of this paper shows that this construction extends a given duality of maps to a duality for relations.

In order to formulate this result precisely we first review order-enriched category theory (Sect. 5.2) followed by a study of order-enriched spans and cospans (Sect. 5.3). We then show how the extension from maps to relations works in the category of posets (Sect. 5.4). Building on this, we will be in a position to extend the duality of bounded distributive lattices and Priestley spaces to relations (Sect. 5.5). As it turns out, this result can be generalized to order-regular categories (Sect. 5.6), which do support a general duality theory of relations (Sect. 5.7).

In a nutshell, the three technical observations at the heart of the paper are the following.

First, if we tabulate a relation $R \subseteq X \times Y$ in sets as a span $X \leftarrow R \rightarrow Y$ and then let $X \leftarrow R' \rightarrow Y$ be the pullback of the pushout of R, the two relations R and R' will in general not coincide. On the other hand, if we view $X \leftarrow R \rightarrow Y$ as a span of discrete posets and we let $X \leftarrow R' \rightarrow Y$ be the comma of the cocomma of R, then the two relations R and R' are equal. This will be reviewed in detail in Sect. 5.3.

Second, if we

- tabulate a relation in finite sets as a span $X \leftarrow R \rightarrow Y$,
- dualise it to a cospan of Boolean algebras $\mathbb{2}^X \rightarrow \mathbb{2}^R \leftarrow \mathbb{2}^Y$,
- tabulate it via pullback as a Boolean relation $\mathbb{2}^X \leftarrow R' \rightarrow \mathbb{2}^R$,
- dualise it to a cospan of sets $X \leftarrow \mathbb{2}^{R'} \rightarrow Y$,
- tabulate it via pullback as a relation $X \leftarrow R'' \rightarrow Y$,

then, in general, the double dual R'' will be different from R. On the other hand, if we view R as a relation of discrete posets and we repeat the same steps with the categories of posets and distributive lattices, replacing pullbacks by comma objects, the double dual R'' will coincide with the original relation R.

Third, spans work well on both sides of the duality in order to inherit algebraic and topological structure, allowing us to extend the duality from finite posets and finite distributive lattices to Priestley spaces and distributive lattices (and other similar dualities).

Examples of dual relations arise from different questions including the following.

- Given a topological space equipped with an equivalence relation, preorder or partial order, what is the algebraic structure dual to the quotient of the topological space by its equivalence relation (or by its preorder or by its partial order)?
- Given a non-deterministic computation formalised as a relation in a category of domains or topological spaces, what is its dual relation between preconditions and postconditions?
- Given algebraic structure extended with relations, what is its topological dual?
- In particular, given a sequent calculus formalised as a relation in a category of algebras, what is its dual semantics for which it is sound and complete?

Answers to some of these questions in concrete examples (Sects. 5.4.4, 5.5.3, 5.6.3, 5.7.3) are meant to be read before going into the details of the technical developments.

Contributions of the paper include:

- Formula (5.9) for computing the dual of a relation.
- Example 5.5.15 showing that, as a consequence of (5.9), the dual of the relation that quotients Cantor space to the unit interval is the way below relation on the algebra of clopens.
- Theorem 5.5.9 on the equivalence of Priestley and distributive lattice relations.
- Theorem 5.6.9 on extending functors between concretely order-regular categories from maps to relations
- Theorem 5.7.6 on extending equivalences of categories of maps to equivalences of categories of relations.
- Theorem 5.7.11 on extending adjunctions of categories of maps to adjunctions of framed bicategories of relations.

Related Work. We draw on a range of previous work. From the point of view of *domain theory* this paper is in the tradition of Abramsky's Domain Theory in Logical From, see Abramsky (1991) and Smyth (1992). Both emphasize domains as systems of data that can be described by finitary (logical) means. We bring this together with the tradition of domain theory as enriched category theory introduced by Smyth & Plotkin (1982) and continued by e.g. Wagner (1994), Rutten (1996), Rutten (1998), Bonsangue et al. (1998), Waszkiewicz (2002), Stubbe (2007), Gutierres & Hofmann (2013). We also rely on Kelly's monograph on *enriched category theory* (Kelly, 1982) and work by Guitart (1980) and Street (1974, 1980) who investigated relations in category-enriched categories whereas we specialise to poset-enriched categories. The *categorical theory of relation lifting* started with Barr (1970) who also showed that the relation lifting of a set-functor is functorial iff the functor preserves weak pullbacks (or exact squares). Work by Trnková (1980), Freyd & Scedrov (1990), Hermida & Jacobs (1998), Hermida (2011), and Moss (1999) has also been influential. Extending adjunctions (as opposed to equivalences) to categories of relations requires tools from *higher category theory* with work by Grandis & Paré (1999), Grandis & Paré (2004), Grandis (2020) and Shulman (2008) being particularly valuable. In the field of *ordered algebra*, work by Scott (1970, 1976), Goguen et al. (1977) and, in particular, Bloom & Wright (1983) and Kelly & Power (1993) was important, as well as our own continuation (Kurz & Velebil, 2017) which introduced order-regular categories. *Weakening relation algebras* are studied by Jipsen and Galatos in Jipsen (2017), Galatos & Jipsen (2020). Our paper is also part of *coalgebra*, in particular of the line of research extending set-based coalgebra to coalgebras over enriched categories initiated by Rutten (1998) and Worrell (2000). In particular, we take from Bilkova et al. (2012), Bilkova et al. (2013) the insight, ultimately going back to Street (1980), that, in the order-enriched setting, relations can be both tabulated and co-tabulated. Last but not least, from the field of *duality theory*, we rely on the classical results of Stone (1937) and Priestley (1970), summarised in the monographs of Johnstone (1982) and Davey & Priestley (2002).

5.2 Preliminaries on Ordered Category Theory

We review some known material on order-enriched categories. Most important for us is that weakening relations can be both tabulated via spans and co-tabulated via cospans. This observation is pivotal for our duality of relations.

5.2.1 *Ordered Categories and Weighted Limits*

An important aspect of ordered categories is that they offer a richer notion of limits. Of particular importance to us will be the ordered analogues of pullback, pushout and coequalizer, also known as comma object,[2] co-comma object and co-inserter. Comma objects tabulate (and cocomma objects co-tabulate) relations. Coinserters take quotients wrt theories of inequations.

Throughout this paper, Pos denotes the category of partially ordered sets (aka posets) and order-preserving (aka monotone) functions.

A **Pos-category** $\mathcal{C}$ is a category in which the homsets are posets and where composition is monotone in both arguments. In other words, a Pos-category is a category enriched over Pos. A **Pos-functor** is a functor that is **locally monotone**, that is, a functor that preserves the order on the homsets.

If $\mathcal{C}$ is a Pos-category, then $\mathcal{C}^{\mathrm{op}}$ denotes the Pos-category which turns around the arrows and $\mathcal{C}^{\mathrm{co}}$ denotes the Pos-category which turns around the order on the homsets.

Since Pos is cartesian closed and complete and cocomplete we are in the framework studied in Kelly's monograph (Kelly, 1982). If we want to emphasise this, we follow Kelly and prefix the notions with "Pos-", but, still following Kelly, we also may drop the prefix if it is clear from the context. If we want to emphasise non enriched categories, we speak of "ordinary" categories, "ordinary" functors, etc.

Pos is itself a Pos-category. We write

$$[A, B]$$

for the poset of maps $A \to B$ ordered pointwise.

Notions such as epi and mono carry over from ordinary category theory to Pos-enriched category theory unchanged. But they are not always the most useful notions. For example, more important to us than injection is **embedding**, that is, a map $m : A \to B$ in Pos that is order-reflecting. If $m : A \to B$ is an embedding then m is injective and A inherits the order from B.

Definition 5.2.1 Let $\mathcal{C}$ be a Pos-category. An arrow $m : A \to B$ is a **P-mono** if $\mathcal{C}(-, m) : \mathcal{C}(X, A) \to \mathcal{C}(X, B)$ is an embedding. An arrow is a **P-epi** if it is a P-mono in $\mathcal{C}^{\mathrm{op}}$.

---

[2] The name "comma object" stems from Lawvere's comma categories.

Remark 5.2.2 Explicitly, m is a P-mono iff $m \circ f \leq m \circ g \;\Rightarrow\; f \leq g$ and e is a P-epi iff $f \circ e \leq g \circ e \;\Rightarrow\; f \leq g$.

Whereas Set has epi/mono factorizations, Pos has P-epi/P-mono factorizations:

Example 5.2.3 In Pos the P-monos are precisely the embeddings and the P-epis are precisely the epis or surjections. They form the

$$(Onto, Emb)$$

factorization system that will play a major role later.

While pullbacks will continue to play a role in Pos, we also need what could be called order-pullbacks or P-pullbacks, but are more commonly known as quasi-pullbacks or comma objects.

Definition 5.2.4 (*Comma, P-kernel, cocomma*) Given a diagram (aka a cospan) $A \to C \leftarrow B$, the **comma object** (or just comma for short) of the cospan is a span $A \leftarrow W \to B$ such that in the diagram

$$\begin{tikzcd}
& W \arrow[dl] \arrow[dr] & \\
A \arrow[dr] & \leq & B \arrow[dl] \\
& C &
\end{tikzcd}$$

the left-hand composition is smaller than the right-hand composition and such that for any other span $A \leftarrow X \to B$ with this property there is a unique $X \to W$ such that the two triangles in

$$\begin{tikzcd}
& X \arrow[d, dotted] \arrow[ddl, bend right] \arrow[ddr, bend left] & \\
{=} & W \arrow[dl] \arrow[dr] & {=} \\
A \arrow[dr] & \leq & B \arrow[dl] \\
& C &
\end{tikzcd}$$

commute. Moreover, there is a 2-dimensional requirement: If there are two cones $A \xleftarrow{f_i} X \xrightarrow{g_i} B$ with $f_1 \leq f_2$ and $g_1 \leq g_2$, then also $h_1 \leq h_2$ for the unique arrows $h_i : X \to W$. In the special case where the two legs of the cospan are the same arrow

f, we speak of the order-kernel or **P-kernel** of f. A **cocomma** in $\mathcal{C}$ is a comma in $\mathcal{C}^{\mathrm{op}}$.

While we will encounter comma-objects in other categories than Pos, we will only need to compute it in Pos itself.

Example 5.2.5 In Pos, the comma of the cospan (j, k) is given by $W = \{(a, b) \mid j(a) \leq k(b)\}$ together with the two projections on the domain of j and k, respectively. The order on W is inherited from the order on A and B, that is, the induced $W \to A \times B$ is an embedding.

The next example highlights one of the reasons why we need to work order-enriched. In the order-enriched setting, the order on a cocomma object C in $A \to C \longleftarrow B$ can encode any weakening relation $R : A \looparrowright B$ (see the next subsection for more on weakening relations).

Example 5.2.6 In Pos, the cocomma of a span $A \xleftarrow{p} R \xrightarrow{q} B$ is the cospan $A \xrightarrow{j} C \xleftarrow{k} B$ where the carrier of C is the disjoint union of A and B and the order on C is inherited from A, B and R. In detail, $\leq_C$ is the smallest partial order satisfying $a \leq_C a' \Leftrightarrow a \leq_A a'$ and $a \leq_C b \Leftarrow aRb$ and $b \leq_C b' \Leftrightarrow b \leq_B b'$.

In universal algebra regular factorizations play a crucial role. The regular factorization of an arrow f is obtained by taking the coequalizer of its kernel. In the ordered setting, we factor f by taking the coinserter (or P-coequalizer) of its P-kernel. Intuitively, while coequalizers quotient by equations, coinserters quotient by inequations.

Definition 5.2.7 Given a pair of two parallel arrows (f, g) the **coinserter** e is the universal arrow wrt the property $e \circ f \leq e \circ g$. In detail, this means that if there are $k_1 \leq k_2$ such that $k_i \circ f \leq k_i \circ g$ then there are unique $h_1 \leq h_2$ such that $h_i \circ e = k_i$. An arrow that is a coinserter is also called a **P-regular epi**.

Example 5.2.8 The coinserters in Pos are precisely the surjections. In fact, in Pos the notions of surjection, epi, P-epi, and P-regular epi coincide. In the category of preorders, the coinserter of (p, q) with $p, q : X \to Y$ is simply given by $(Y, \sqsubseteq)$ where $\sqsubseteq$ is the smallest preorder containing the order of Y and $\{(p(x), q(x)) \mid x \in X\}$. So we see clearly how taking a coinserter corresponds to adding inequations. A coinserter in Pos is computed by first taking the coinserter in preorders and then quotienting by the equivalence $y \equiv y' \;\Leftrightarrow y \sqsubseteq y' \;\&\; y' \sqsubseteq y$.

Remark 5.2.9 (*Inserter*) An inserter in $\mathcal{C}$ is a coinserter in $\mathcal{C}^{\mathrm{op}}$. Inserters will only appear in minor remarks and examples in this paper. It is enough to know that in Pos, the inserter of (j, k) with $(j, k) : X \to Y$ is the subposet of X given by $\{x \mid j(x) \leq k(x)\}$. For a reader who wishes to see examples of how the duality of inserters and coinserters plays out in a setting similar to ours we refer to Dahlqvist & Kurz (2017).

Remark 5.2.10 (*On Terminology*) The point of view of enriched category theory and the one of universal algebra often suggest different terminology.

- Bloom & Wright (1983) noticed that many results in ordered universal algebra can be stated verbatim the same way as the corresponding results in ordinary universal algebra if one is careful about how to define the corresponding notions in the ordered setting. They mark these ordered notions by prefixing them with a "P-". Sometimes these notions agree with those from enriched category theory. For example, a P-category is a Pos-category, a P-functor is a Pos-functor, a P-monad is a Pos-monad, but the same is not true for P-monos, P-epis, P-faithful, P-kernel, P-coequalizer. One theme is that P-notions often add a requirement of order-reflection. Another is that P-notions work well with inequational theories instead of only with equational theories. As a rule, in a category with discrete homsets, the P-notions should coincide with the ordinary notions.
- On the other hand, the category theoretic notions have the advantage that they make sense in other enriched categories. For example, some results in ordered algebra arise as the poset-collapse of more general results from category-enriched categories, which have a well-developed theory (see e.g. Street 1974, 1980, 1982; Kelly & Power 1993; Bourke 2010; Bourke & Garner 2013) that can be exploited in the poset-enriched setting.
- Another advantage of category theoretic notions such as comma object and coinserter is that they include the 2-dimensional aspect of weighted limits, as opposed to Bloom and Wright's P-kernel or P-coequalizer. The 2-dimensional aspect is essential in abstract Pos-categories, but comes for free in Pos itself, as well as in other concrete Pos-categories, which explains why the difference does not matter for the purposes of this paper.
- We summarize our compromise terminology in Table 5.1. All of the P-notions are from Bloom & Wright (1983).

Table 5.1 Summary of terminology

Category theory	Universal algebra
Pos-category	P-category
Pos-functor	P-functor
–	P-faithful
Representably fully faithful	P-mono
–	P-epi
Comma object	–
–	P-kernel
Coinserter	P-coequaliser
(Coinserter)	P-regular epi

There are other weighted limits than comma objects and inserters. For our purposes, the easiest way to define the totality of all weighted limits is to use a theorem of Kelly, (1982, (3.68)) which states that if a complete and cocomplete category has the special weighted limits known as powers and the special weighted colimits known as tensors, then it has all weighted limits and all weighted colimits:

Definition 5.2.11 Let A be an object of a Pos-category $\mathcal{C}$ and $X \in \mathsf{Pos}$. Then the co-tensor or **power** $X \pitchfork B$ is defined as the unique up-to-iso solution of the equation

$$[X, \mathcal{C}(A, B)] \cong \mathcal{C}(A, X \pitchfork B)$$

and the dual notion of co-power or **tensor** $X \bullet A$ is determined by

$$[X, \mathcal{C}(A, B)] \cong \mathcal{C}(X \bullet A, B).$$

Example 5.2.12 In posets, the power $X \pitchfork B$ is the poset of monotone functions $X \to B$. In distributive lattices, with X a poset and B a distributive lattice, $X \pitchfork B$ is the distributive lattice of monotone functions $X \to B$.

We can now define completeness in the enriched sense.

Definition 5.2.13 A Pos-category is **(finitely) complete** if it has (finite) products, equalizers and powers, and it is **(finitely) cocomplete** if it has (finite) coproducts, coequalizers and tensors. In particular, a complete Pos-category has commas and inserters and a cocomplete Pos-category has cocommas and coinserters.

5.2.2 Weakening Relations

This section introduces the protagonists of this paper, namely monotone, or weakening-closed, relations. Let

$$\mathsf{Rel}(\mathsf{Pos}) \qquad \text{or} \qquad \overline{\mathsf{Pos}}$$

denote the Pos-category where objects are posets $A, B, \ldots$, arrows $A \looparrowright B$ are monotone maps $A^{\mathrm{op}} \times B \to \mathbb{2}$, and 2-cells are given pointwise (in other words, if we identify a relation with $\{(a, b) \mid R(a, b) = 1\}$, then relations are ordered by set-inclusion). Since $\mathbb{2} = \{0 < 1\}$ is a poset, all homsets $\overline{\mathsf{Pos}}(A, B)$ are posets. We let $B(a, b) = 1$ if $a \leq_B b$ and $B(a, b) = 0$ otherwise. The identity of A is the order of A and composition is ordinary relational composition. Composition of $R : A \looparrowright B$ and $S : B \looparrowright C$ is written as $R \,;\, S$ or $S \cdot R$.

We call these relations **monotone relations** or **weakening-closed relations** or **weakening relations** for short. They are also the Pos-enriched cousins of their category-enriched relatives known as profunctors, distributors, or bimodules. The term weakening-closure derives from the fact that the monotonicity of $A^{\mathrm{op}} \times B \to \mathbb{2}$ amounts to the rule

$$\frac{a' \leq a \, R \, b \leq b'}{a' \, R \, b'}$$

which is known as weakening in the case where R is a Gentzen-style $\vdash$ in a proof theoretic setting.

For every map $f : A \to B$ in Pos there is a relation (called companion in Grandis & Paré, 2004, Shulman, 2008)

$$f_* : A \nrightarrow B \tag{5.1}$$

given by $(a, b) \mapsto B(fa, b) : A^{\mathrm{op}} \times B \to \mathbb{2}$ and a relation (called adjoint or conjoint in Grandis & Paré, 2004, Shulman, 2008)

$$f^* : B \nrightarrow A \tag{5.2}$$

given by $(a, b) \mapsto B(b, fa) : A \times B^{\mathrm{op}} \to \mathbb{2}$. Recall that a relation $L : A \nrightarrow B$ is left-adjoint to $R : B \nrightarrow A$, written as

$$L \dashv R,$$

if we have (unit) $a \leq a' \Rightarrow \exists b \,.\, L(a, b) \wedge R(b, a')$ and (counit) $\exists a \,.\, R(b, a) \wedge L(a, b') \Rightarrow b \leq b'$. We have that

$$f_* \dashv f^*$$

in $\overline{\mathsf{Pos}}$. Moreover, the left-adjoints recover the maps among the relations: If we have $L \dashv R$, then there is a monotone function f in Pos such that $L = f_*$ and $R = f^*$.[3]

The functor $(-)_* : \mathsf{Pos} \to \overline{\mathsf{Pos}}$ is covariant on 1-cells and contravariant on 2-cells.

The functor $(-)^* : \mathsf{Pos} \to \overline{\mathsf{Pos}}$ is contravariant on 1-cells and covariant on 2-cells.

This notation can be used to explain how a span $(A \xleftarrow{p} W \xrightarrow{q} B)$ represents, or **tabulates**, the relation $Rel(p, q) = q_* \cdot p^*$ and how a cospan $(A \xrightarrow{j} C \xleftarrow{k} B)$ represents, or **cotabulates**, the relation $Rel(j, k) = k^* \cdot j_*$.

We conclude with a couple of useful observations.

Proposition 5.2.14 *The identity relation on A is the comma of $A \xrightarrow{\mathrm{id}} A \xleftarrow{\mathrm{id}} A$.*

[3] In the discrete setting, a function f and its relation f_* are the same set $\{(x, f(x))\}$ of pairs. In the ordered setting, f_* corresponds to the set of pairs $\{(x, y) \mid f(x) \leq y\}$. To recover f from an adjunction $L \dashv R$ we obtain from the unit that (i) every $a \in A$ gives rise to an upset $aL = L(a, -)$ and a downset $Ra = R(-, a)$ with non-empty intersection and we obtain from the counit that (ii) this intersection can contain at most one element. Thus $fa = aL \cap Ra$.

Proposition 5.2.15 *A monotone function m is an embedding in* Pos *if and only if* $m^* \cdot m_* = \mathrm{Id}$. *A monotone function e is a surjection in* Pos *if and only if* $\mathrm{Id} = e_* \cdot e^*$.[4]

5.2.3 Ordered Algebra

Stone duality for relations takes place in an order-enriched setting. To understand the algebraic side of the duality, we review some aspects of order-enriched algebra. For the purposes of this paper, ordered algebra is Pos-enriched algebra. In particular, all operations are order-preserving. This has the advantage that a relation $A \looparrowright B$ between two ordered algebras can be simply defined as a monotone function $A^{\mathrm{op}} \times B \to \mathbb{2}$ such that the legs of the corresponding span are algebra homomorphisms. We explain this now in more detail and conclude with examples of algebraic structure with order-reversing operations.

Our notion of an ordered (quasi)-variety is the one of Bloom & Wright (1983). As in the ordinary case, a **P-variety** can be defined in various equivalent ways. (Recall that a functor is locally monotone if it preserves the order on the homsets.) A P-variety $U : \mathcal{A} \to \mathsf{Pos}$ is, equivalently,

- a category of algebras with monotone operations for a finitary signature definable by a set of inequations.
- a category of algebras with monotone operations for a finitary signature closed under HSP. Here we need to take H as closure under quotients by inequations, or closure under coinserters, to use the terminology of Sect. 5.2.1. Similarly, closure under SP needs to be generalized to include all weighted limits. This can be done by adding closure under powers, or by generalizing closure under S from equalizers to inserters.
- a category of algebras for a locally monotone monad $T : \mathsf{Pos} \to \mathsf{Pos}$ that is *strongly finitary* in the sense that it is the Pos-enriched left-Kan extension of its restriction to finite discrete posets.

We settle for the last item as our official definition, since it is the most succinct one and liberates us from repeating the standard definitions of universal algebra such as signature, inequations, and closure under HSP. The equivalence of the last item with the previous two can be found in Theorem 6.9 of Kurz & Velebil (2017), which also contains a full explanation of the technical notions involved as well as further references.

Convenient properties that follow from this definition are that coinserters (= quotients by inequations) are surjections and that free constructions as well as the monad T preserve surjections.[5] This need not be the case for the more general

[4] One can replace "=" by "$\leq$" since the other direction is, respectively, the unit and counit of the adjunction and always holds.

[5] Kurz & Velebil (2017), Theorem 6.3 shows that strongly finitary functors preserve surjections. On the other hand, Kurz & Velebil (2017), Exle 6.4 shows that, conversely, being finitary and preserving surjections is not enough to imply strongly finitary.

notion of finitary monads on Pos, which are a special case of the notion of algebra theory studied by Kelly & Power (1993).

Since surjections coincide with P-regular epis, we can also say that $\mathcal{A}$ is the category of algebras for a strongly finitary P-regular monad and, since Pos is P-regular but not regular in the ordinary sense, we may, with a slight abuse of language, simply say that P-varieties are the categories of algebras for a strongly finitary, regular monad on Pos.[6]

Example 5.2.16 The category DL of bounded distributive lattices is a P-variety. Note that, as a P-variety, DL is different from the ordinary variety of distributive lattices as DL now has ordered homsets. The category BA of Boolean algebras is the full subcategory of DL consisting of Boolean algebras. Note that, because DL-morphisms between Boolean algebras preserve negation, $\mathsf{DL}(B, B')$ is discrete if B, B' are Boolean algebras.

Remark 5.2.17 It was shown in Dahlqvist & Kurz (2017), Thmeorem 12 that the inclusion $\mathsf{BA} \to \mathsf{DL}$ is the free completion of BA wrt a certain class of inserters. Informally, we may say that DL is the smallest category containing BA and closed under Pos-enriched subobjects.

Definition 5.2.18 A DL-relation $R : A \looparrowright B$ is a weakening relation $UA \looparrowright UB$ that is tabulated by a span in DL.

In other words, $R : A \looparrowright B$ is DL-relation if it is weakening closed and a subalgebra of $A \times B$. Spelling this out in detail this means that R is closed under the following rules.

$$\frac{a' \leq aRb \leq b'}{a'Rb'} \qquad \frac{}{0R0} \qquad \frac{}{1R1} \qquad \frac{aRb \quad a'Rb'}{(a \wedge a')R(b \wedge b')} \qquad \frac{aRb \quad a'Rb'}{(a \vee a')R(b \vee b')}$$

Example 5.2.19 The property of being a subalgebra interacts with weakening closure in a subtle way.

1. If $A = B = \mathbb{2} \in \mathsf{DL}$, then there are only two relations $A \looparrowright B$, namely the identity $\leq_{\mathbb{2}}$ and the total relation.
2. If A and B are the 3-chain distributive lattice, then the smallest weakening-closed DL-relation $A \looparrowright B$ is $\{(0, 0), (0, b), (0, 1), (a, 1), (1, 1)\}$, where the middle elements of the chains are called a and b, respectively. This relation is the weakening-closure of the initial span $(p : \mathbb{2} \to A, q : \mathbb{2} \to B)$.

As we will see in this paper, this interplay between weakening closure and the subalgebra property is crucial to extend Stone duality to relations. It has some, maybe at first sight unexpected, consequences for structures that have order-reversing operations. For example, if we equip Boolean algebras with their natural order then the only weakening-closed BA-relation is the total relation.

[6] The notion of a P-regular category was introduced in Kurz & Velebil (2017), Definition 3.18 where it was simply called regular.

Example 5.2.20 Let $R : A \looparrowright B$ be a Boolean relation between Boolean algebras equipped with their natural order. Then R is the total relation. Indeed, because of $(0, 0) \in R$, we have, by weakening closure, $(0, 1) \in R$ and then by closure under negation $(1, 0) \in R$, hence, again by weakening closure, $(a, b) \in R$ for all $a \in A$ and $b \in B$.

This example raises the question of what the order of Boolean algebras should be in the order-enriched setting. There are two possible answers. In the first remark below, the order on a Boolean algebra is discrete, in the second the order is the natural order, as inherited from distributive lattices.

Remark 5.2.21 (*The P-variety of Boolean algebras is discrete*) If we want Boolean algebras to form a P-variety, all operations need to be monotone. Since Boolean algebras have negation, the order on the homsets as well as the order on individual Boolean algebras (witnessed by the forgetful functor) must be discrete. This does not contradict closure under ordered quotients since ordered congruences in Boolean algebras are necessarily symmetric and hence equivalence relations (Dahlqvist & Kurz, 2017, Sect. 2.2). It also does not contradict closure under weighted limits, since the forgetful functor to Pos has a left-adjoint given by composing the connected component functor $\mathsf{Pos} \to \mathsf{Set}$ with the ordinary free construction of Boolean algebras. More technically, we can say that the category of discretely ordered Boolean algebras is exact in the ordered sense (Kurz & Velebil, 2017, Exle.3.22). It then follows from Kurz & Velebil (2017), Theorem 5.9 that the forgetful functor from discretely ordered Boolean algebras to Pos is a P-variety.

The previous remark is only of interest since it indicates that the theory of P-varieties specializes to the theory of ordinary varieties in the discrete case. But this discrete point of view fails to exhibit any new order related structure. Therefore, in the rest of the paper, we consider BA as a full subcategory of the P-variety DL equipped with the forgetful functor $\mathsf{BA} \to \mathsf{DL} \to \mathsf{Pos}$. From this point of view the homsets between Boolean algebras are still discrete, but their carriers are not.

Remark 5.2.22 (BA *is not a P-variety*) While $\mathsf{BA} \to \mathsf{DL} \to \mathsf{Pos}$ equips Boolean algebras with their natural order, $\mathsf{BA} \to \mathsf{Pos}$ is now not a P-variety, since BA is not closed under weighted limits. For example, the power $\mathbb{2} \pitchfork \mathbb{2}$, see Definition 5.2.11, is the three element distributive lattice. In fact, every DL is an inserter of Boolean algebras in a canonical way (Dahlqvist & Kurz, 2017, Proposition 10) and DL is a closure of BA under weighted limits.

Let us note that, mutatis mutandis, the last two remarks also apply to other ordered structures such as Heyting algebras. Our approach can deal with mixed variance only indirectly by embedding the mixed variant signatures into order-preserving signatures. The next example illustrates that this is related to our interest in heterogeneous relations, that is, relations $A \looparrowright B$ where $A \neq B$.

Remark 5.2.23 Let us illustrate why order-reversing operations present a problem for binary relations $A \looparrowright B$. For example, in the case of Boolean algebras or Heyting algebras, we might want to add, respectively, to Definition 5.2.18 of a DL-relation the clauses

$$\frac{a\ R\ b}{\neg b\ R\ \neg a} \qquad\qquad \frac{a_1\ R\ b_1 \quad a_2\ R\ b_2}{(b_1 \to a_2)\ R\ (a_1 \to b_2)}$$

as eg in Pigozzi, (2004, Definition 2.1). The as and bs are switching sides and this presents no problems if $A = B$. But in this paper we are mainly interested in "multi-lingual" relations (Jung et al. 1999) connecting objects $A \neq B$. Future work should take a cue from Greco et al. (2020) who account for rules such as the ones above with the help of opposite relations.

5.2.4 *Ordered Stone Duality*

As we have seen in Sect. 5.2.3 on Ordered Algebra, weakening relations are more interesting in the ordered category DL then in the discrete category of Boolean algebras. We therefore decided to treat BA as a full subcategory of DL and dualize Boolean relations inside the larger category of distributive lattices. This lines up nicely with the way that Johnstone (1982) introduces Stone duality where he first presents the duality of spectral spaces and distributive lattices and then obtains the duality of Stone spaces and Boolean algebras as the discrete restriction. We follow this approach in that we take the duality for distributive lattice as more fundamental, but find it convenient to rely on Priestley (1970) version of the duality as laid out for example in Davey & Priestley (2002).

Let us recall that the dual equivalence between the category DL of distributive lattices and the category Pri of Priestley spaces is mediated by two contravariant functors $\mathsf{DL}(-, \mathbb{2})$ and $\mathsf{Pri}(-, \mathbb{2})$ which we both abbreviate as $\mathbb{2}^-$. We only need to add to this that the two contravariant functors determined by homming into $\mathbb{2}$

$$\begin{tikzcd} \mathsf{Pri} \arrow[r, bend left, "\mathbb{2}^-"] & \mathsf{DL} \arrow[l, bend left, "\mathbb{2}^-"] \end{tikzcd}$$

are not only a dual equivalence of categories, but also a dual equivalence of Pos-categories, covariant on the order of the homsets. This means, for example, that a cocomma in DL can be computed as the dual of a comma in Pri.

5.3 The Duality of Spans and Cospans

Since duality sends the span tabulating a relation to a cospan, we need to understand the relationship of spans and cospans. We will see that restricting to weakening-closed spans yields a satisfactory duality. This material owes much to Street (1974), Street (1980) and Guitart (1980).

5.3.1 Spans and Cospans

Given a Pos-category $\mathcal{C}$ and objects A, B we define the Pos-categories

$$\mathsf{Span}(\mathcal{C}, A, B) \qquad \text{and} \qquad \mathsf{Cospan}(\mathcal{C}, A, B).$$

(We may drop the reference to $\mathcal{C}$ in the notation). Objects in $\mathsf{Span}(\mathcal{C}, A, B)$ are spans $(p : W \to A, q : W \to B)$. Arrows $f : (p : W \to A, q : W \to B) \to (p' : W' \to A, q' : W' \to B)$ are arrows $f \in \mathcal{C}$ such that $p' \circ f = p$ and $q' \circ f = q$. Cospan is defined dually.

Remark 5.3.1 Every span (p, q) and every cospan (j, k) give rise to relations

$$Rel(p, q) = q_* \cdot p^* \qquad \text{and} \qquad Rel(j, k) = k^* \cdot j_*$$

if $\mathcal{C}$ is Pos or just a concrete Pos-category.

In general, if $\mathcal{C}$ has comma and cocomma objects, there are Pos-functors

$$Cocomma : \mathsf{Span}(A, B) \to \mathsf{Cospan}(A, B)$$

and

$$Comma : \mathsf{Cospan}(A, B) \to \mathsf{Span}(A, B)$$

where *Comma* takes a cospan and maps it to its comma square and *Cocomma* takes a span and maps it to its co-comma square. Grandis and Pare, (2004, Sect. 5.3) describe this as a colax/lax adjunction between double categories of spans as cospans, but the following suffices for our purposes.

Proposition 5.3.2 *Cocomma $\dashv$ Comma for all Pos-categories $\mathcal{C}$ with comma and cocomma objects. The induced monad and comonad are idempotent. Restricting the functors Comma and Cocomma to a skeleton of $\mathsf{Span}(A, B)$ and $\mathsf{Cospan}(A, B)$, this means that Comma $\circ$ Cocomma is a closure operator and Cocomma $\circ$ Comma is an interior operator. Moreover, there is a bijection between fixed points of Comma $\circ$ Cocomma and fixed points of Cocomma $\circ$ Comma. Furthermore, if $\mathcal{C} = \mathsf{Pos}$, then these fixed points are in bijection with the weakening relations $A \looparrowright B$.*

Remark 5.3.3 In case that $\mathcal{C} = \mathsf{Pos}$, there is a canonical choice of skeleton of $\mathsf{Span}(A, B)$ given by the weakening closed subsets of $A \times B$. The monad *Comma* $\circ$ *Cocomma* then maps a span (p, q) to the graph of the relation $q_* \cdot p^*$. We write

$$Graph(R)$$

for the graph of a relation R. *Comma* maps a cospan (j, k) to the graph of the relation $k^* \cdot j_*$.

We can reformulate the definition of graph so that it generalizes to order-regular categories (Kurz & Velebil, 2017). Instead of fully-faithful we would then say representably fully-faithful (or P-mono) and instead of onto we would say P-regular epi. But in this paper we work concretely over Pos and we can say surjective and embedding instead.

Definition 5.3.4 (*Graph*) In the category Pos, we say that a span (p, q) is **embedding** if arrow between spans $(p, q) \to \mathit{Comma}(\mathit{Cocomma}(p, q))$ is fully faithful; is **weakening-closed** if $(p, q) \to \mathit{Comma}(\mathit{Cocomma}(p, q))$ is onto; is a **graph** (of a weakening relation) if $(p, q) \to \mathit{Comma}(\mathit{Cocomma}(p, q))$ is iso. A span (p, q) represents a relation R if $q_* \cdot p^* = R$. We say that (p, q) tabulates R if it is the graph of R.

It follows from the proposition that every relation has not only a unique tabulation as a graph, but also a unique cotabulation, which is known as the collage of a relation and was introduced by Street (1980) to characterize relations in the case of bicategories.

Definition 5.3.5 (*Collage*) In the category Pos, we say that a cospan $(j : A \to C, k : B \to C)$ is **bipartite** if $\mathit{Cocomma}(\mathit{Comma}(j, k)) \to (j, k)$ is fully faithful; is **onto** if $\mathit{Cocomma}(\mathit{Comma}(j, k)) \to (j, k)$ is onto; is a **collage** if $\mathit{Cocomma}(\mathit{Comma}(j, k)) \to (j, k)$ is iso. A cospan (j, k) represents the relation $k^* \cdot j_*$ and cotabulates it if (j, k) is bipartite and onto.

The terminology is summarised in Table 5.2.

Example 5.3.6 In the category Pos the collage of a relation $R : A \looparrowright B$, or, equivalently, $\mathit{Cocomma}(p, q)$ of a span tabulating R, is given by a poset C such that $C(a, a') = A(a, a')$, $C(a, b) = R(a, b)$ and $C(b, b') = B(b, b')$. We write

$$\mathit{Collage}(R)$$

for this particular cospan cotabulating R.

The next example shows that while the legs of a collage are order-reflecting in Pos, this need not be the case in DL. A similar example can be built in all non-trivial categories of algebras which have a constant. It follows that a general characterization of collages (or cocomma cospans) in algebraic categories needs special investigation.

Example 5.3.7 Let $(p : W \to A, q : W \to B)$ be the span where p is the identity on the free DL on one generator $\{a\}$, let B be the initial DL with elements $\{0 < 1\}$, and let q map a to 0. One verifies that (q, id_B) is the cocomma of (p, q). And we have $a \not\leq 0$ but $q(a) \leq q(0)$, so that q is not an embedding. The reason is that we have

Table 5.2 Duality of spans and cospans

Spans	Cospans
Weakening-closed	Bipartite
Embedding (full subobject of product)	Onto (quotient of coproduct)
Graph of a relation	Collage of a relation

$$a \le 0_B = 0_A$$

where the inequation comes from the span and the equation comes from the laws of DL.

5.3.2 *Exact Squares*

Given a diagram

$$\begin{tikzcd} & W \arrow[dl, "p"'] \arrow[dr, "q"] & \\ A \arrow[dr, "j"'] \arrow[rr, phantom, "\le"] & & B \arrow[dl, "k"] \\ & C & \end{tikzcd} \tag{5.3}$$

in Pos, we always have that $jp \le kq$ implies $Rel(p, q) \le Rel(j, k)$ ("going over is smaller or equal to going under"). A square $((p, q), (j, k))$ with $jp \le kq$ is called exact if $Rel(p, q) = Rel(j, k)$.[7] Without referring to relations this can be expressed equivalently as in

Definition 5.3.8 A square in Pos as in (5.3) satisfying $jp \le kq$ is **exact** if for all a, b such that $ja \le kb$ there is w such that $a \le pw$ and $qw \le b$.

In our context, one of the reasons why exact squares are important, is that an exact square says that the span and the cospan represent the same relation. Informally, exact squares represent relations without preference being given to either spans or cospans.

Let us repeat that (5.3) is exact iff

$$q_* \cdot p^* = k^* \cdot j_*, \tag{5.4}$$

which is sometimes called the Beck-Chevalley-Condition. It is also important to note that (5.4) gives the square (5.3) a direction, which we denote by

[7] Note that $Rel(p, q) = Rel(j, k)$ implies $jp \le kq$ (since $Rel(p, q) = Rel(j, k)$ is $k^* \cdot j_* = q_* \cdot p^*$ which implies $j_* \cdot p_* \ge k_* \cdot q_*$ which is equivalent to $j \cdot p \le k \cdot q$).

$$A \nrightarrow B.$$

In other words, the span and the cospan in (5.3) represent the same relation if read from A to B, but not necessarily the other way around (in fact, if the cospan is a collage in Pos, the relation represented by reading the cospan backwards is empty).

The following can be verified easily by direct computation.

Proposition 5.3.9 *Comma squares and cocomma squares in* Pos *are exact.*

We will see in Sect. 5.6, that most of the results about spans and cospans in Pos generalise to concretely order-regular categories. The exactness of cocommas is one of the exceptions: It either may fail to hold or require more work. On the other hand, the next two propositions do generalise.

Proposition 5.3.10 *A comma in an exact square is the comma of the cospan.*

Proof Let (p', q') be a comma and (j, k) a cospan so that the square (p', q', j, k) is exact. Let (j', k') be the cospan of which (p', q') is the comma. Now suppose that (p, q) is a cone over (j, k), that is, $jp \le kq$. We show first that (p, q) is also a cone over (j', k'). We know $jp(w) \le kq(w)$ for all w. It follows from the exactness of (p', q', j, k) that there is w' such that $p(w) \le p'(w')$ and $q'(w') \le q(w)$, which implies

$$j'p(w) \le j'p'(w') \le k'q'(w') \le k'q(w).$$

We have shown that $j'p \le k'q$, that is, that (p, q) is a cone over (j', k'). Since (p', q') is the comma of (j', k'), there is a unique arrow $(p, q) \to (p', q')$. It follows that (p', q') is the comma of (j, k). $\square$

We also have the dual property for cocommas:

Proposition 5.3.11 *A cocomma in an exact square is the cocomma of the span.*

5.3.3 Identity and Composition of Spans and Cospans

Since we have a correspondence between relations and (co)spans and we know how to compose relations, an obvious question is how to describe composition directly on (co)spans. But let us first quickly look at identities.

The span (id, id) represents the identity relation, but it is not weakening closed in general. The graph of the identity relation is given by the comma object of the cospan (id, id). On the other hand, the collage of the identity relation is simply the cospan (id, id).

Composition of relations can be done directly on representing spans by taking comma objects (or any exact square, for that matter). If in the diagram

$$
\begin{tikzcd}
 & & {} \arrow[dl, "r"'] \arrow[dr, "s"] & & \\
 & {} \arrow[dl, "p"'] \arrow[dr, "q"] & & {} \arrow[dl, "p'"'] \arrow[dr, "q'"] & \\
 {} & & {} & & {}
\end{tikzcd}
\tag{5.5}
$$

(r, s) is the comma span of (q, p') then $(pr, q's)$ represents $Rel(p', q') \cdot Rel(p, q)$, which is immediate if we have exactness of comma squares.

It is important to note that this composition does not preserve graphs. For example, if $p = q' : 2 \to 1$ and $q = p' = id_2$ then $Rel(pr, q's)$ is the identity on 1 but $(pr, q's)$ not an embedding span.

But composition of spans does preserve weakening closure:

Proposition 5.3.12 *If in* (5.5) *we have that* (p, q) *and* (p', q') *are weakening-closed and* (r, s) *is the comma span of* (q, p')*, then* $(pr, q's)$ *is weakening-closed.*

Composition of cospans is done by cocomma squares, dualising (5.5), and relying on exactness of cocomma squares.

$$
\begin{tikzcd}
 {} \arrow[dr, "j"'] & & {} \arrow[dl, "k"] \arrow[dr, "j'"'] & & {} \arrow[dl, "k'"] \\
 & {} \arrow[dr, "i"'] & & {} \arrow[dl, "l"] & \\
 & & {} & &
\end{tikzcd}
\tag{5.6}
$$

Composition by cospans does not preserve collages. Indeed, similarly to the previous example, if we take (j, k) and (j', k') to be the collages of $\{(0, 0), (0, 1)\}$ and $\{(0, 0), (1, 0)\}$ respectively, then (ij, lk') is not a collage (because it is not onto, ie, there are elements neither in the image of ij nor in the image of lk'.

But composition by cospans does preserve being bipartite:

Proposition 5.3.13 *If in* (5.6) *we have that* (j, k) *and* (j', k') *are bipartite and* (i, l) *is the cocomma span of* (k, j')*, then* (ij, lk') *is bipartite.*

5.4 Dual Relations in Posets

The purpose of this section is to extend to relations the well-known dualising functor

$$f : X \to Y \quad \mapsto \quad \mathbb{2}^f : \mathbb{2}^Y \to \mathbb{2}^X$$

taking a monotone function to its inverse image. As suggested by the previous section, this can be done by applying the functor $\mathbb{2}^-$ to either the legs of a tabulating span or to the legs of a co-tabulating cospan. We show that these two procedures agree and that $\mathbb{2}^-$ extends to a functor on $\overline{\mathsf{Pos}} = \mathsf{Rel}(\mathsf{Pos})$.

The contravariance of $\mathbb{2}^-$ means that the extension is contravariant on the order of the homsets (2-cells): If $r \subseteq r'$ are two relations, then tabulating them as $(p, q) \to (p', q')$ and applying a contravariant functor F gives cospans

$$(Fp', Fq') \to (Fp, Fq). \tag{5.7}$$

As explained in the next remark, it follows that the extension must be covariant on relations (1-cells). (This fits well with relations as "objects of the arrows-object" of a double category (Grandis & Paré, 2004; Shulman, 2008), a point of view that will play a role in Sect. 5.7.)

Remark 5.4.1 (*Covariance on relations*) Let F, G be Pos-functors that are contravariant on 1-cells and covariant on 2-cells. Assume that we have a construction $\mathcal{C} \mapsto \mathsf{Rel}(\mathcal{C})$ with functors $(-)_* : \mathcal{C} \to \mathsf{Rel}(\mathcal{C})^{\mathrm{co}}$ and $(-)^* : \mathcal{C} \to \mathsf{Rel}(\mathcal{C})^{\mathrm{op}}$ (or, equivalently, $(-)^* : \mathcal{C}^{\mathrm{op}} \to \mathsf{Rel}(\mathcal{C})$). Further assume that there are functors $\overline{F}$ and $\overline{G}$ that are contravariant on 2-cells. Then to complete the diagram

$$\begin{tikzcd}
\mathsf{Rel}(\mathcal{X}) \arrow[rr, bend left, "\overline{F}"] & & \mathsf{Rel}(\mathcal{A}) \arrow[ll, bend left, "\overline{G}"] \\
\\
\mathcal{X} \arrow[uu, dotted, "(-)"] \arrow[rr, bend left, "F"] & & \mathcal{A}^{\mathrm{op}} \arrow[uu, dotted, "(-)"'] \arrow[ll, bend left, "G"]
\end{tikzcd}$$

we are forced to set things up in such a way that the extensions $\overline{F}, \overline{G}$ are covariant on relations. Indeed, after fixing one of the embeddings, say we use $(-)_*$ on the $\mathcal{X}$-side, we need to use the other one $(-)^*$ on the $\mathcal{A}$-side, since this is the only way to accommodate that $\overline{F}$ and $\overline{G}$ are contravariant on 2-cells. This in turn forces the extensions $\overline{F}, \overline{G}$ to be covariant on 1-cells.

The reason for later choosing $(-)_*$ on the space-side and $(-)^*$ on the algebra-side is explained at the beginning of Sect. 5.5.2. Here we only need the functors, defined in Sect. 5.2.2 , $(-)_* : \mathsf{Pos} \to \overline{\mathsf{Pos}}^{\mathrm{co}}$ and $(-)^* : \mathsf{Pos}^{\mathrm{op}} \to \overline{\mathsf{Pos}}$, where we continue to abbreviate $\overline{\mathsf{Pos}} = \mathsf{Rel}(\mathsf{Pos})$.

5.4.1 Extending to Relations via Spans

We derive condition (5.9), which allows us to calculate the dual of a relation in specific examples. The formula arises from applying $\mathbb{2}^-$ to a graph and then converting the resulting cospan to a relation. Recall that a cospan (j, k) represents the relation $k^* \cdot j_*$ given by $(x, y) \in k^* \cdot j_* \;\Leftrightarrow\; j(x) \leq k(y)$.

Proposition 5.4.2 ($\overline{\mathbb{2}^-}$ via spans) *Given a weakening relation $r : X \looparrowright Y$ in* Pos*, define $\overline{\mathbb{2}}(r)$ via first converting r into its graph $X \xleftarrow{p} R \xrightarrow{q} Y$ and then applying $\mathbb{2}^-$ to the legs of the span, yielding a cospan*

$$\mathbb{2}^X \xrightarrow{\mathbb{2}^p} \mathbb{2}^R \xleftarrow{\mathbb{2}^q} \mathbb{2}^Y$$

which in turn gives rise to a relation

$$\overline{\mathbb{2}}(r) = (\mathbb{2}^q)^* \cdot (\mathbb{2}^p)_* : \mathbb{2}^X \looparrowright \mathbb{2}^Y. \tag{5.8}$$

Then

$$(A, B) \in \overline{\mathbb{2}}(r) \;\Leftrightarrow\; R[A] \subseteq B \tag{5.9}$$

where $R[A] = \{b \mid \exists a \in A \,.\, aRb\}$.

Proof We have

$$\begin{aligned}(A, B) \in \overline{\mathbb{2}}(r) \;&\Leftrightarrow\; \mathbb{2}^p(A) \subseteq \mathbb{2}^q(B)\\ &\Leftrightarrow\; \forall x \in X.\forall y \in Y.\, x \in A \;\&\; xRy \;\Rightarrow\; y \in B\end{aligned}$$

or, in one picture,

$$\begin{tikzcd} & R \arrow[dl] \arrow[dr] & \\ X \arrow[dr, "A"'] \arrow[rr, phantom, "\leq"] & & Y \arrow[dl, "B"] \\ & \mathbb{2} & \end{tikzcd} \tag{5.10}$$

iff $(A, B) \in \overline{\mathbb{2}}(r)$. $\square$

Remark 5.4.3 For logics with many-valued valuations in a poset D we have

$$(A, B) \in \overline{D}(r) \iff \forall x \in X.\forall y \in Y.\, xRy \;\Rightarrow\; A(x) \leq B(y).$$

5.4.2 Extending to Relations via Cospans

In this section, we see that extending $\mathbb{2}^-$ via cospans gives the same dual relations as the extension via spans from the previous section, see (5.9) and (5.12). The formula (5.12) arises from applying $\mathbb{2}^-$ to a cotabulating cospan and then turning the resulting span into a relation. Recall that a span (p, q) represents the relation $q_* \cdot p^*$ given by $(x, y) \in q_* \cdot p^* \;\Leftrightarrow\; \exists w \,.\, x \le p(w) \;\&\; q(w) \le y$.

Proposition 5.4.4 ($\overline{\mathbb{2}^-}$ via cospans) *Given a weakening relation $r : X \nrightarrow Y$, first convert r into a cospan $X \xrightarrow{j} R \xleftarrow{k} Y$ and then apply $\mathbb{2}^-$, yielding a span*

$$\mathbb{2}^X \xleftarrow{\mathbb{2}^j} \mathbb{2}^R \xrightarrow{\mathbb{2}^k} \mathbb{2}^Y,$$

and hence a relation

$$\overline{\mathbb{2}}(r) = (\mathbb{2}^k)_* \cdot (\mathbb{2}^j)^* : \mathbb{2}^X \nrightarrow \mathbb{2}^Y. \tag{5.11}$$

Then we have $(A, B) \in \overline{\mathbb{2}}(r)$ if and only if

$$\forall b \in Y. \forall a \in X. a \in A \;\&\; aRb \;\Rightarrow\; b \in B. \tag{5.12}$$

Proof We have by definition of $\overline{\mathbb{2}}$ that

$$(A, B) \in \overline{\mathbb{2}}(r) \;\Leftrightarrow\; \exists C \in \mathbb{2}^R \,.\, A \subseteq \mathbb{2}^j(C) \;\&\; \mathbb{2}^k(C) \subseteq B. \tag{5.13}$$

For "only if", assume $x \in A$ and xRy. From $A \subseteq \mathbb{2}^j(C)$ we know $jx \in C$ and from xRy that $jx \le ky$. Since $C : R \to \mathbb{2}$ is monotone we have $ky \in C$ and it follows from $\mathbb{2}^k(C) \subseteq B$ that $y \in B$. For "if" define C to be the upper closure of $\{j(x) \mid x \in A\}$. $\square$

Remark 5.4.5 Recalling the definition of a collage from Example 5.3.6, it is clear that for the equivalence (5.12), it is crucial that $\mathbb{2}^R$ consists of upward closed sets. This is also highlighted by the diagram

$$\begin{tikzcd}
 & Graph(r) \arrow[dl] \arrow[dr] & \\
X \arrow[dr, "j"] \arrow[ddr, bend right, "A"'] & \le & Y \arrow[dl, "k"'] \arrow[ddl, bend left, "B"] \\
\le & R \arrow[d, dotted, "C"] & \le \\
 & \mathbb{2} &
\end{tikzcd} \tag{5.14}$$

which can be used to express Proposition 5.4.4 more categorically by saying that $(A, B) \in \overline{2}(r)$ iff (A, B) is a cocone for the span $Graph(r)$. As an aside, since (j, k) is a cocomma, we can always find a C for which the "$\leq$" in the two triangles can be replaced by "=". This shows that the span $2^X \leftarrow 2^R \rightarrow 2^Y$ is weakening closed and that the "$\subseteq$" in (5.13) can be replaced by "=". Finally, comparing (5.10) and (5.14) explains why we obtain the same result whether dualising the relation r with R being the graph in (5.10) or with R being the collage in (5.14).

5.4.3 Functoriality and Universality of the Extension

So far in this section, we have seen how to extend the contravariant functor $2^- :$ Pos $\rightarrow$ Pos to relations. In order to know that this extension is functorial and does not depend on a choice of span (or choice of cospan), we need to know that the functor preserves factorizations and exact squares.

In more detail, we will employ the general results about extending functors to weakening relations known from Bilkova et al. (2012), Theorem 4.1 for the extension via spans and Bilkova et al. (2013), Theorem 5.10 for the extension via cospans. These results have also been presented in the survey Kurz & Velebil (2016) as Theorems 3.8 and 3.10, which may be the most convenient reference for our purposes. We will later need a generalization of Bilkova et al. (2012), Theorem 4.1 from posets to concrete categories over posets. The reader may therefore also refer to Theorem 5.6.9 (and the dual Theorem 5.6.10) of this paper and instantiate the categories $\mathcal{X}$ and $\mathcal{A}$ with Pos.

Remark 5.4.6 To conclude that the extension $\overline{2}$ is functorial, we will use the extension-via-spans theorem (see Bilkova et al., 2012, Theorem 4.1 or Kurz and Velebil, 2016, Theorem 3.8 or Theorem 5.6.9) which guarantees that the extension $\overline{F}$ of F in

$$\begin{tikzcd} \overline{\mathsf{Pos}}^{\mathrm{co}} \arrow[r, "\overline{F}"] & \mathcal{K}^{\mathrm{co}} \\ \mathsf{Pos} \arrow[u, "(-)_*"] \arrow[ur, "F"'] & \end{tikzcd}$$

is universal and functorial, if F satisfies the following properties.

1. F preserves maps, that is, every Ff has a right adjoint $(Ff)^r$ in $\mathcal{K}$ (which is a left-adjoint in $\mathcal{K}^{\mathrm{co}}$).
2. F preserves exact squares, that is, $Fq \cdot (Fp)^r = (Fk)^r \cdot Fj$ for every exact square (5.3).
3. $Fe \cdot (Fe)^r = \mathrm{Id}$ for all surjections e in Pos.

For the extension via cospans, we have the same theorem with Property 3 being replaced by

3. $(Fj)^r \cdot (Fj) = \mathrm{Id}$ for all embeddings j in Pos.

To show that $\overline{\mathbb{2}}$ is a functor we verify that $(-)^* \circ \mathbb{2}$ satisfies properties 1–3 above. Since the extension is universal and therefore unique, it also follows that the span and the cospan extension agree, giving a different argument for what we have seen by direct calculation in Propositions 5.4.2 and 5.4.4.

We first recall the well-known fact that $\mathbb{2}$ preserves Onto-Embedding factorizations.

Lemma 5.4.7 *The contravariant functor $\mathbb{2}^-: \mathsf{Pos} \to \mathsf{Pos}$ maps surjections to embeddings and embeddings to surjections.*

Proof Let $f : X \to Y$, hence $\mathbb{2}^f : [Y, \mathbb{2}] \to [X, \mathbb{2}]$. If f is onto and $\mathbb{2}^f(p) \leq \mathbb{2}^f(q)$, then $p \circ f \leq q \circ f$ and $p \leq q$, proving that $\mathbb{2}^f$ is an embedding. If f is an embedding and $p : X \to \mathbb{2}$, then there is $q : Y \to \mathbb{2}$ such that $q \circ f = p$ (eg, one can take q as the left or right Kan-Extension of p along f). □

Of central importance is that $\mathbb{2}^-$ preserves exact squares:

Lemma 5.4.8 *Let*

$$\begin{tikzcd} & W \arrow[dl, "p"'] \arrow[dr, "q"] & \\ X \arrow[dr, "f"'] \arrow[rr, phantom, "\leq"] & & Y \arrow[dl, "g"] \\ & Z & \end{tikzcd} \tag{5.15}$$

be an exact square, that is, $f \circ p \leq g \circ q$ and $\forall x, y\,.\,(fx \leq gy \Rightarrow \exists w\,.\,x \leq pw\ \&\ qw \leq y)$ or, equivalently,

$$q_* \cdot p^* = g^* \cdot f_*.$$

Then

$$\begin{tikzcd} & \mathbb{2}^W & \\ \mathbb{2}^X \arrow[ur, "\mathbb{2}^p"] \arrow[rr, phantom, "\leq"] & & \mathbb{2}^Y \arrow[ul, "\mathbb{2}^q"'] \\ & \mathbb{2}^Z \arrow[ul, "\mathbb{2}^f"] \arrow[ur, "\mathbb{2}^g"'] & \end{tikzcd} \tag{5.16}$$

is exact, that is,

$$(\mathbb{2}^q)^* \cdot (\mathbb{2}^p)_* = (\mathbb{2}^g)_* \cdot (\mathbb{2}^f)^* \,.$$

Proof Assume $ap \leq bq$. We have to show that there is c such that $a \leq cf$ and $cg \leq b$.

$$\begin{tikzcd}
& W \arrow[dl, "p"'] \arrow[dr, "q"] & \\
X \arrow[dr, "f"] \arrow[ddr, bend right, "a"'] \arrow[rr, phantom, "\leq"] \arrow[d, phantom, "\leq"] & & Y \arrow[dl, "g"'] \arrow[ddl, bend left, "b"] \arrow[d, phantom, "\leq"] \\
{} & Z \arrow[d, dotted, "c"] & {} \\
& \mathbb{2} &
\end{tikzcd} \tag{5.17}$$

Let $c = \{z \in Z \mid \exists x \in a \,.\, f(x) \leq z\}$. Then $a \leq cf$. It remains to show $cg \leq b$, which follows from $ap \leq bq$. Indeed, if $gy \in c$ then there is $x \in a$ such that $fx \leq gy$. From exactness, we get a w such that $x \leq pw$ and $qw \leq y$, which, together with our assumption, implies $a(x) \leq b(y)$, that is, due to $x \in a$, the required $y \in b$. $\square$

Remark 5.4.9 The proof does not depend on the span (p, q) being weakening closed. This can be used to simplify the computation of dual relations by choosing smaller generating spans.

Relying on terminology from Sects. 5.2.2 and 5.3.3, we are now ready to prove

Theorem 5.4.10 *The extensions* $\overline{\mathbb{2}} : \mathsf{Rel}(\mathsf{Pos})^{\mathrm{co}} \to \mathsf{Rel}(\mathsf{Pos})$ *of* $\mathbb{2}$ *defined by applying* $\mathbb{2}$ *to a tabulating span as in Proposition 5.4.2 or to a co-tabulating cospan as in Proposition 5.4.4 agree and are functorial. They are also universal wrt the properties 1–3 on page 181. Moreover,* $\overline{\mathbb{2}}_{X,Y} : \mathsf{Rel}(\mathsf{Pos})(X, Y)^{\mathrm{op}} \to \mathsf{Rel}(\mathsf{Pos})(\mathbb{2}^X, \mathbb{2}^Y)$, $R \mapsto \{(a, b) \mid R[a] \subseteq b\}$ *is a complete meet-semilattice homomorphism.*

Proof Let $F = (-)^* \cdot \mathbb{2}$ in Remark 5.4.6. To verify Property 1 of Remark 5.4.6, we note that F takes a map $f : X \to Y$ and sends it to the relation $Ff = (\mathbb{2}^f)^*$, see Sect. 5.2.2. This verifies that $(\mathbb{2}^f)^r = (\mathbb{2}^f)_*$ is the left-adjoint of Ff in $\mathsf{Pos} = \mathcal{K}^{\mathrm{co}}$ and the right-adjoint of Ff in $\mathsf{Pos}^{\mathrm{co}} = \mathcal{K}$.

For Property 2, we use that $\mathbb{2}$ preserves exact squares by Lemma 5.4.8. That $(-)^*$, and $(-)_*$, preserves exact squares is immediate from writing out the definitions.

Both the span and the cospan version of Property 3 follow from Proposition 5.2.15 and Lemma 5.4.7.

Finally, we need to verify that $\overline{F}$ agrees with $\overline{\mathbb{2}}$ as defined in Propositions 5.4.2 or 5.4.4. To this end, thanks to the universality and uniqueness of $\overline{F}$, it suffices to show that $\overline{\mathbb{2}} \circ (-)_* = Ff$. In case of the extension by spans, on the left-hand side, a map $f : X \to Y$ is sent by $(-)_*$ to the span (p, q) of the cospan (f, id), which in turn is then dualised by $\overline{\mathbb{2}}$ to $(\mathbb{2}^q)^* \cdot (\mathbb{2}^p)_*$, see (5.8). Since (p, q) and (id, f) both form exact squares with the cospan (f, id) and since $\mathbb{2}^-$ preserves exact squares, we have $(\mathbb{2}^q)^* \cdot (\mathbb{2}^p)_* = (\mathbb{2}^f)^* \cdot \mathrm{id}_* = Ff$.

In case of the extension by cospans, on the left-hand side, a map $f : X \to Y$ is sent by $(-)_*$ to the cospan (f, id), which in turn is then dualised by $\overline{\mathbb{2}}$ to $\mathrm{id}_* \cdot (\mathbb{2}^f)^*$, see (5.11), which equals Ff. $\square$

Remark 5.4.11 To extend functors to relations via spans, it is in fact sufficient to require that the functor preserves exact squares with weakening-closed spans, since spans are composed by commas (see Sect. 5.3.2) and commas are weakening closed.

Remark 5.4.12 (*Independence of choice of span*) Under the conditions of the extension theorem, it is the case that the relation lifting on a relation R can be computed by applying the functor to any representing *weakening-closed* span. But if, as it is the case in our situation, a category has cocomma objects and cocomma objects are exact, then the relation lifting can be computed on any span, including those that are not weakening closed. This follows from the facts that (i) two spans represent the same relation iff they have isomorphic cocommas (Proposition 5.3.2), that (ii) cocommas are exact (Proposition 5.3.9) and that (iii) $\mathbb{2}^-$ preserves exact squares (Lemma 5.4.8 and Remark 5.4.9). A dual argument shows that if two cospans represent the same relation, then applying $\mathbb{2}^-$ to both cospans gives the same relation.

Theorem 5.7.11 will show that we can extend not only the functor $\mathbb{2}^-$, but also to the adjunction $\mathbb{2}^- \dashv (\mathbb{2}^-)^{\mathrm{op}} : \mathsf{Pos}^{\mathrm{op}} \to \mathsf{Pos}$.

5.4.4 Examples

We illustrate different interpretations of the dual $\overline{\mathbb{2}}(R)$ of a relation R

$$(A, B) \in \overline{\mathbb{2}}(R) \;\Leftrightarrow\; R[A] \subseteq B.$$

through four different applications to Hoare logic, duality theory, domain theory and coalgebraic logic.

Hoare Logic. First, an example from program verificiation and the relational theory of computation.

Example 5.4.13 If $R \subseteq X \times X$ is the relation representing a non-deterministic computation, then $(A, B) \in \overline{\mathbb{2}}(R)$ iff inputs satisfy A then outputs satisify B. In other words, $(A, B) \in \overline{\mathbb{2}}(R)$ iff (A, B) are a pair of pre- and post-conditions of the computation R, or,

$$\{A\}R\{B\}$$

in a notation common in program verification and Hoare logic. Note that this is indeed a weakening relation as we have as one of the rules of Hoare logic

$$\frac{A' \leq A \quad \{A\}\ R\ \{B\} \quad B \leq B'}{\{A'\}\ R\ \{B'\}}.$$

Moreover, the meet preserving function $\overline{\mathbb{2}}_{X,Y}$ maps a relation to its theory of precondition and postcondition pairs while its left-adjoint

$$\begin{tikzcd} \mathsf{Rel}(\mathsf{Pos})(X, Y)^{\mathrm{op}} \arrow[r, bend right, "\text{Theory}"'] \arrow[r, phantom, "\perp"] & \mathsf{Rel}(\mathsf{Pos})(\mathbb{2}^X, \mathbb{2}^Y) \arrow[l, bend right, "\text{Implementation}"'] \end{tikzcd}$$

takes a 'specification' $S \in \mathsf{Rel}(\mathsf{Pos})(\mathbb{2}^X, \mathbb{2}^Y)$ to its largest relation 'implementation' $\bigcup\{R \mid S \subseteq \overline{\mathbb{2}}(R)\}$. $\square$

Duality Theory. We describe quotienting by an equivalence relation or preorder in terms of the dual relation. We emphasise that even to describe the dual of equivalence relations on a discrete set one is led to consider weakening relations with respect to a non-discrete order, namely the inclusion order between subsets. From a technical point of view, this stems, on the one hand, from the fact that we work with a dualising object $\mathbb{2}$ that is equipped with an order and, on the other hand, from the fact that the relationship between spans and cospans is not mediated via pullback/pushout but via comma/cocomma, see Definition 5.2.4 and Sect. 5.3. Working with a discrete $\mathbb{2}$ and with discrete spans/cospans, we would not obtain a dual equivalence between, say, relations on finite sets and relations on finite Boolean algebras.

Example 5.4.14 Let R be a relation on a set X.

1. If R is reflexive then $\overline{\mathbb{2}}(R) \subseteq \mathrm{Id}_{\mathbb{2}^X}$.
2. If R is reflexive and transitive then $\overline{\mathbb{2}}(R) \subseteq \mathrm{Id}_{\mathbb{2}^X}$ and $\overline{\mathbb{2}}(R); \overline{\mathbb{2}}(R) \supseteq \overline{\mathbb{2}}(R)$. Such a relation $\overline{\mathbb{2}}(R)$ is called *interpolative*.
3. If R is an equivalence relation, then $\mathbb{2}^{X/R}$ is bijective to the set $\{(A, A) \mid R[A] \subseteq A\}$ of reflexive elements of $\overline{\mathbb{2}}(R)$.

Reflexive and transitive relations are idempotent relations above identity and interpolative relations are idempotent relations below identity. So item 2 becomes the obvious statement that duality maps idempotent relations above identity to idempotent relations below identity. Since reflexive and transitive relations are monads, we can also view item 2 as the duality of monads and comonads. $\square$

The example generalises to posets X. The first two items transfer verbatim, noting that Id_A now refers to the order of A.

Example 5.4.15 Let R be a weakening relation on a poset X. If R is a preorder, then the set $\mathbb{2}^{X/R}$ of upper sets of X/R is bijective to $\{(A, A) \mid R[A] \subseteq A\}$. Here X/R is the partial order quotient of X wrt R.

These observations will lead to new duality results for categories where objects are endo-relations, see Sect. 5.6.3.

Bitopological spaces. We present an example from the theory of bitopological spaces. A bitopological space (X, τ_-, τ_+) is a set X with two topologies. While certain complete lattices, known as frames can be considered as algebraic duals of topological spaces (see Johnstone 1982 for details), pairs of lattices (L_-, L_+) dualise bitopological spaces. This setting is of interest because adding two weakening relations

$$con : L_- \looparrowright L_+^\partial \qquad tot : L_+^\partial \looparrowright L_-$$

to the pair of lattices one can characterise a large class of well-known topological spaces by a finitary structure (Jung & Moshier, 2006). The functor from bitopological spaces to so-called d-frames is easily explained. It takes a space (X, τ_-, τ_+) to the frames $L_- = \tau_-$ and $L_+ = \tau_+$ with con defined as the set of pairs $(a_-, a_+) \in \tau_- \times \tau_+$ such that $a_- \cap a_+ = \emptyset$ and tot as the set of pairs such that $a_- \cup a_+ = X$. (The names con and tot should remind us of 'consistent' and 'total'.) The functor from d-frames to bitopological spaces takes a structure (L_-, L_+, con, tot) to the bitopological space (X, τ_-, τ_+) where X is the set of pairs (p_-, p_+) of frame morphisms $p_- : L_- \to \mathbb{2}$ and $p_+ : L_+ \to \mathbb{2}$ such that

$$\forall (a_-, a_+) \in con \,.\, p_-(a_-) = 0 \text{ or } \; p_+(a_+) = 0 \tag{5.18}$$

$$\forall (a_+, a_-) \in tot \,.\, p_-(a_-) = 1 \text{ or } \; p_+(a_+) = 1 \tag{5.19}$$

and the topologies τ_- and τ_+ are generated by basic opens $\{p \mid p(a) = 1\}$ where a ranges over L_- and L_+, respectively. Using formula (5.9) to compute the dual of a relation, one can verify

Example 5.4.16 (*Duals of d-frames*) The carrier of the dual of a d-frame (L_-, L_+, con, tot) is the intersection $\overline{\mathbb{2}}(con) \cap \overline{\mathbb{2}}(tot)$ of the dual of con and the dual of tot. For the proof, one verifies that (p_-, p_+) satisfies (5.18) iff $(p_-, p_+) \in \overline{\mathbb{2}}(con)$ and that (p_-, p_+) satisfies (5.19) iff $(p_-, p_+) \in \overline{\mathbb{2}}(tot)$.

Modal Logic. In modal and coalgebraic logic the notion of R-coherent pairs arises from the study of bisimulations for so-called neighbourhood frames (Hansen et al., 2009).

Example 5.4.17 A quick look at Definition 2.1 in Hansen et al. (2009) of R-coherent pairs shows that, given a relation $R \subseteq X_1 \times X_2$, a pair (U_1, U_2) with $U_i \subseteq X_i$ is R coherent if $(U_1, U_2) \in \overline{\mathbb{2}}(R)$ and $(U_2, U_1) \in \overline{\mathbb{2}}(R^{-1})$.

Due to the presence of the converse relation R^{-1} in the definition above, given $X_1 \leftarrow R \to X_2$, the relation of R-coherence is the pullback of $\mathbb{2}^{X_1} \to \mathbb{2}^R \leftarrow \mathbb{2}^{X_2}$. This observation opened the way to coalgebraic generalisations (Bakhtiari & Hansen, 2017; de Groot et al., 2020). It would be interesting to pursue these in the ordered setting.

5.5 Dual Relations in Priestley Spaces

We will use the results from the previous section on weakening relations to show that the dual equivalence of Priestley spaces and distributive lattices extends from maps to relations.

5.5.1 Priestley Spaces and Distributive Lattices

We start out by defining the category $\mathsf{Rel}(\mathsf{DL})$ of distributive lattice relations and the category $\mathsf{Rel}(\mathsf{Pri})$ of Priestley relations. We defined DL-relations in Definition 5.2.18.

Definition 5.5.1 ($\mathsf{Rel}(\mathsf{DL})$) The category $\mathsf{Rel}(\mathsf{DL})$, abbreviated to $\overline{\mathsf{DL}}$, has the same objects as DL and DL-relations as arrows. Homsets are ordered by inclusion.

Pri-relations can be defined in the same way. Recall that a Priestley space $(X, \leq, \tau)$ is a compact Hausdorff space (X, τ) with an order relation satisfying the Priestley separation axiom, that is, $x \not\leq y$ only if there is a clopen downset U_- and a clopen upset U_+ such that $U_- \cap U_+ = \emptyset$ and $x \in U_+$ and $y \in U_-$.

Definition 5.5.2 ($\mathsf{Rel}(\mathsf{Pri})$) A Pri-relation $A \looparrowright B$ is a topologically-closed and upward-closed subspace of $A^{\mathrm{op}} \times B$. The category $\mathsf{Rel}(\mathsf{Pri})$, or $\overline{\mathsf{Pri}}$ for short, has the same objects as Pri and Pri-relations as arrows. Homsets are ordered by inclusion.

For future reference we prove some properties that will be needed later. In particular, the properties below establish that DL and Pri are examples of concretely order-regular categories as defined in Sect. 5.6. (Note that if a functor $\mathcal{A} \to \mathcal{B}$ creates limits or lifts limits and $\mathcal{B}$ is complete, then the functor preserves limits.)

Proposition 5.5.3 *$U : \mathsf{DL} \to \mathsf{Pos}$ creates Pos-limits and $(Onto, Emb)$ factorisations. DL is order-regular. Comma squares in DL are exact. Identities and composition in $\mathsf{Rel}(\mathsf{DL})$ are inherited from $\mathsf{Rel}(\mathsf{Pos})$.*

Proof These properties of the first two sentences are true for all P-varieties (and P-quasi-varieties) (Kurz & Velebil, 2017). The others follow from this. □

Proposition 5.5.4 *The forgetful functor $V : \mathsf{Pri} \to \mathsf{Pos}$ lifts Pos-limits and factorisations uniquely. Comma squares in Pri are exact. Identities and composition in $\mathsf{Rel}(\mathsf{DL})$ are inherited from $\mathsf{Rel}(\mathsf{Pos})$.*

Proof (1) Ordinary limits in Pri are equalisers of products equipped with the subspace topology. Cotensors $I \pitchfork X$, with I a poset and X a Priestley space, are given by $\{(x_i)_{i \in I} \mid i \leq_I j \ \Rightarrow \ x_i \leq_X x_j\}$, which is a closed subspace of the $|I|$-fold power of X and hence a Priestley space. It follows from Kelly (1982), Theorem 3.73 that all weighted limits exist. V lifts these limits uniquely, since the property of being a limit prescribes that limits must be equipped with the subspace topology. 2) Pri has a factorisation system consisting of embeddings with the subspace topology and surjections. 3) The statements about comma squares, identities and composition follow from the above. $\square$

The next lemma contains the crucial technical observation.

Lemma 5.5.5 *The contravariant functors* $\mathbb{2}^-:\mathsf{DL}\to\mathsf{Pri}$ *and* $\mathbb{2}^-:\mathsf{Pri}\to\mathsf{DL}$ *preserve exact squares.*

Proof For the proof, we use the notation of Lemma 5.4.8. For $\mathbb{2}^-:\mathsf{DL}\to\mathsf{Pri}$, suppose we have the exact square (5.15) in DL and its image under $\mathbb{2}$ in Pri as in (5.16). We need to show that we can find an appropriate DL-morphism c in (5.17). The forward image of a via f is a filter basis, that is,

$$f[a_+] = \{z \in Z \mid \exists x\,.\,a(x) = 1 \;\&\; f(x) \le z\} \tag{5.20}$$

is a filter. Likewise,

$$g[b_-] = \{z \in Z \mid \exists y\,.\,b(y) = 0 \;\&\; z \le g(y)\} \tag{5.21}$$

is an ideal.

Assume $a \circ p \le b \circ q$, that is, $\mathbb{2}^p(a) \subseteq \mathbb{2}^q(b)$. Then $f[a_+]$ is disjoint from $g[b_-]$. For suppose not. Then for some x and y, $a(x) = 1$, $b(y) = 0$, and $f(x) \le g(y)$. By exactness, there is a w so that $x \le p(w)$ and $q(w) \le y$. But then our assumption tells us that $a(x) \le b(y)$, contradicting $a(x) = 1$ and $b(y) = 0$. Hence $f[a_+]$ is disjoint from $g[b_-]$.

Therefore, by the prime ideal theorem, $f[a_+]$ and $g[b_-]$ extend to some $c \in \mathbb{2}^Z$, so that $z \in f[a_+]$ implies $c(z) = 1$ and $z \in g[b_-]$ implies $c(z) = 0$. That is, $a \le \mathbb{2}^f(c)$ and $\mathbb{2}^g(c) \le b$, as required.

For $\mathbb{2}^-:\mathsf{Pri}\to\mathsf{DL}$, suppose (5.17) is in Pri. Define $f[a_+]$ and $g[b_-]$ as above. Evidently,

$$f[a_+] = \uparrow f(a^{-1}(\{1\})) \qquad \text{and} \qquad g[b_-] = \downarrow g(b^{-1}(\{0\})).$$

Adapting the argument for DL above, suppose $\mathbb{2}^p(a) \subseteq \mathbb{2}^q(b)$. Then $f[b_+] \cap g[b_-] = \emptyset$. Because $a^{-1}(\{1\})$ is closed, it is compact. So $f(a^{-1}(\{1\}))$ is compact, hence closed. The upper set determined by any closed set is closed. So $f[a_+]$ is an upper compact set. Likewise, $g[b_-]$ is a lower compact set.

Fix $z \in f[a_+]$. For each $z' \in g[b_-]$, $z \not\le z'$. So there is a clopen downset $U_{z'}$ containing z' and excluding z. These cover $g[b_-]$. So finitely many, say $U_{z'_0}$, ..., $U_{z'_{n-1}}$, suffice to cover. Thus the intersection of their complements is an upper clopen containing z and disjoint from $g[b_-]$. Call it V_z. The upper clopens V_z cover $f[a_+]$. And again finitely many, say V_{z_0}, ..., $V_{z_{m-1}}$, suffice to cover $f[a_+]$. The union of these is an upper clopen that covers $f[a_+]$ and is disjoint from $g[b_-]$.

Let c be the corresponding element of $\mathbb{2}^Z$. Then $a \le \mathbb{2}^f(c)$ and $\mathbb{2}^g(c) \le b$. $\square$

Corollary 5.5.6 *In* DL *and* Pri *cocomma squares are exact.*

Proof The homming-into-$\mathbb{2}$ functors mediating the dual equivalence between DL and Pri are locally monotone and hence Pos-enriched. Therefore cocommas in DL (or Pri) are commas in Pri (or DL), which are exact. And exactness is preserved by $\mathbb{2}^-$. □

Remark 5.5.7 Duality is helpful here. Recall from Example 5.3.6 that in Pos, the exactness of cocommas was immediately obvious from their explicit characterization of cocommas as collages. But we do not have such a characterization for DLs, see also Example 5.5.11.

We will see in Remark 5.7.8 that the relationship of Corollary 5.5.6 between exactness of cocommas and preservation of exact squares extends to other concretely order-regular categories.

Finally, we will need the following result, which is well-known and follows from the fact that the duality respects the factorisation systems of Priestley spaces and distributive lattices. We sketch a direct proof.

Lemma 5.5.8 *The contravariant functors* $\mathbb{2}^- : \mathsf{Pri} \to \mathsf{DL}$ *and* $\mathbb{2}^- : \mathsf{DL} \to \mathsf{Pri}$ *map surjections to embeddings and embeddings to surjections.*

Proof For $\mathbb{2}^- : \mathsf{Pri} \to \mathsf{DL}$, we let $f : X \to Y$ so that $\mathbb{2}^f : [Y, \mathbb{2}] \to [X, \mathbb{2}]$. If f is onto, then $\mathbb{2}^f$ is an embedding, for the same reason as in Pos. If f is an embedding and $p : X \to \mathbb{2}$ is a clopen upset, then by the Priestley separation axiom there is a clopen upset $q : Y \to \mathbb{2}$ containing $\{f(x) \mid x \in p\}$ and disjoint from $\{f(x) \mid x \notin p\}$. Therefore $q \circ f = p$, ie, $\mathbb{2}^f(q) = p$, showing that $\mathbb{2}^f$ is onto.

For $\mathbb{2}^- : \mathsf{DL} \to \mathsf{Pri}$, we let $f : A \to B$ so that $\mathbb{2}^f : [B, \mathbb{2}] \to [A, \mathbb{2}]$. If f is onto, then $\mathbb{2}^f$ is an embedding, for the same reason as in Pos. If f is an embedding and $p : A \to \mathbb{2}$ is a prime filter, then by the prime filter theorem there is a prime filter $q : B \to \mathbb{2}$ containing $\{f(x) \mid x \in p\}$ and disjoint from $\{f(x) \mid x \notin p\}$. Therefore $q \circ f = p$, ie, $\mathbb{2}^f(q) = p$, showing that $\mathbb{2}^f$ is onto. □

5.5.2 Duality of Relations

Before we can state and prove Theorem 5.5.9 about the equivalence of DL and Pri relations, we need to describe the set-up summarised in (5.22).

Given a function, or deterministic program, $f : X \to Y$ there are two natural ways of associating a relation to f. The weakening closed relation given by the 'hypergraph' $f_* = \{(fx, y) \mid fx \le y\}$ and the co-weakening closed relation given by the 'hypograph' $f^* = \{(y, fx) \mid y \le fx\}$.

If f is Scott-continuous then the hypergraph is closed whereas the hypograph does not have a similar good property. This is one reason we choose to work with the hypergraph on the side of spaces. Technically, this means that the relation associated to f will be $f_* = \lambda x, y \,.\, Y(fx, y)$.

Dually, f will be mapped to $\mathbb{2}^f : \mathbb{2}^Y \to \mathbb{2}^X$. We turn this into a relation by stipulating

$$a \subseteq f^{-1}(b)$$

or, equivalently, $f[a] \subseteq b$, which agrees with (5.9). This means that the relation associated to a $g : B \to A$ in DL is given by g^* which is

$$g^*(a, b) = A(a, gb)$$

Recalling that extensions of a contravariant functor are contravariant on 2-cells, see (5.7), we obtain

$$\begin{tikzcd} \overline{\mathsf{Pri}}^{\mathrm{co}} \arrow[r, bend left, "\overline{\mathbb{2}}"] & \overline{\mathsf{DL}} \arrow[l, bend left, "\overline{\mathbb{2}}"] \\ \mathsf{Pri} \arrow[u, "(-)_*"] \arrow[r, bend left, "\mathbb{2}^-"] & \mathsf{DL}^{\mathrm{op}} \arrow[u, "(-)^*"'] \arrow[l, bend left, "\mathbb{2}^-"] \end{tikzcd} \tag{5.22}$$

which is in accordance with the left-hand diagram before Remark 5.4.6.

The functor $\overline{\mathsf{Pri}}^{\mathrm{co}} \to \overline{\mathsf{DL}}$ tabulates a relation r as a span

$$X \xleftarrow{p} R \xrightarrow{q} Y$$

and maps it to the cospan

$$\mathbb{2}^X \xrightarrow{\mathbb{2}^p} \mathbb{2}^R \xleftarrow{\mathbb{2}^q} \mathbb{2}^Y$$

which in turn gives rise to a relation

$$\overline{\mathbb{2}}(r) = (\mathbb{2}^q)^* \cdot (\mathbb{2}^p)_* : \mathbb{2}^X \looparrowright \mathbb{2}^Y. \tag{5.23}$$

This agrees with the definition of $\overline{\mathbb{2}}(r)$ as a functor on Pos in (5.8), but we need to be aware that here $\mathbb{2}^X$ refers to the set of Priestley-maps from X to the Priestley space $\mathbb{2}$.

The functor $\overline{\mathsf{DL}} \to \overline{\mathsf{Pri}}^{\mathrm{co}}$ is defined in the same way on relations. In detail, it tabulates a relation r as a span

$$A \xleftarrow{p} R \xrightarrow{q} B$$

and maps it to the cospan

$$\mathbb{2}^A \xrightarrow{\mathbb{2}^p} \mathbb{2}^R \xleftarrow{\mathbb{2}^q} \mathbb{2}^B$$

which in turn gives rise to a relation

$$\overline{2}(r) = (2^q)^* \cdot (2^p)_* : 2^A \looparrowright 2^B. \tag{5.24}$$

This again agrees with the definition of $\overline{2}(r)$ as a functor on Pos in (5.8), but now 2^A refers to the set of distributive lattice morphisms from A to the distributive lattice 2.

Theorem 5.5.9 *The equivalence*

$$\begin{tikzcd} \mathsf{Pri} \arrow[r, bend left, "2^-"] & \mathsf{DL}^{\mathrm{op}} \arrow[l, bend left, "2^-"] \end{tikzcd}$$

extends to an equivalence of categories of relations

$$\begin{tikzcd} \overline{\mathsf{Pri}}^{\mathrm{co}} \arrow[r, bend left, "\overline{2}"] & \overline{\mathsf{DL}} \arrow[l, bend left, "\overline{2}"] \end{tikzcd} \tag{5.25}$$

where $2 : \overline{\mathsf{Pri}}^{\mathrm{co}} \to \overline{\mathsf{DL}}$ *is defined by* (5.23) *and* $2 : \overline{\mathsf{DL}} \to \overline{\mathsf{Pri}}^{\mathrm{co}}$ *is defined by* (5.24).

Proof To prove that (5.25) is well-defined, use again Remark 5.4.6 and proceed as in the proof of Theorem 5.4.10. Property 1 (preservation of maps) follows from the fact that if two Pos (or DL) relations are adjoint in Pos then they are adjoint in Pri (or DL). Property 2 (preservation of exact squares) is Lemma 5.5.5. Property 3 (mapping surjections to embeddings) is Lemma 5.5.8.

It remains to show that (5.25) is an equivalence of categories. Let r be a Pri-relation and $(p, q) = Graph(r)$. Let (p', q') be the comma of the cospan $(2^p, 2^q)$. We have to show that $Rel(p, q) = Rel(2^{p'}, 2^{q'})$. But this follows from $p = 2^{2^p}$ and $q = 2^{2^q}$ (due to Priestley duality) and the square

$$\begin{tikzcd} & \cdot \arrow[dl, "2^{2^p}"'] \arrow[dr, "2^{2^q}"] & \\ \cdot \arrow[dr, "2^{p'}"'] & & \cdot \arrow[dl, "2^{q'}"] \\ & \cdot & \end{tikzcd}$$

being exact. The latter, in turn, is a consequence of 2 preserving exact squares and the comma-square of (p', q') being exact. The other direction, starting with a DL-relation r, is proved in the same way. $\square$

The next proposition allows us to compute the dual of a relation by dualising the legs of a representing span even if it is not weakening closed.

Proposition 5.5.10 *If two (not-necessarily weakening closed) spans in* DL *or* Pri *represent the same weakening relation, then their dual cospans do so as well.*

Proof The proof is the same as for Remark 5.4.12 and uses that commas and cocommas are exact (see Corollary 5.5.6) and that duality preserves exactness (see Lemma 5.5.5). □

5.5.3 Examples

Recall that in the category Pos, we characterised cocommas as collages. In particular, in the cocomma (j, k) of a span $(A \leftarrow R \rightarrow B)$, the maps j and k are embeddings. Intuitively, this means that the quotient of $A + B$ by R cannot add inequations to A or to B. The next example shows that we cannot say the same about cocommas of bounded distributive lattices.

Example 5.5.11 Let $(A \leftarrow R \rightarrow B)$ be a span in DL with $R = A \times B$ the total relation. Intuitively, the cocomma of the span should be the trivial DL since R forces the top of A to be below the bottom of B. That this is indeed the case is most easily seen using 2-dimensional duality (Sect. 5.2.4) to compute the cocomma of R as the dual of the graph of the dual relation (=the dual of the comma of the dual of the span of R). Indeed, the dual of $(A \leftarrow R \rightarrow B)$ is a cospan injecting into the disjoint union of the dual of A and the dual of B. It follows from the disjointness that the comma of this cospan is the empty relation. Its dual is the cospan in DL that has the one-element DL as its apex.

The example above depends crucially on working with *bounded* distributive lattices. It will be of interest to look into the duality of not-necessarily-bounded distributive lattices in the future.

We continue with some examples around the Cantor space, which is a Priestley space with a discrete ordering. The Cantor space is homeomorphic to $2^{\mathbb{N}}$ with the product topology, homeomorphic to the Stone dual of the free Boolean algebra over the set $\mathbb{N}$, and homeomorphic to the "middle-third" subspace of the unit interval.

Example 5.5.12 (*The ordered Cantor space*) Let X be the middle-third Cantor space and $X \leftarrow \leq \rightarrow X$ the order inherited form the real numbers. According to (5.9), the dual $\sqsubseteq = \overline{\mathbb{2}}(\leq)$ is given by $a \sqsubseteq b$ iff $a \subseteq b$.

The following proposition shows that we can recover the distributive lattice dual to the Priestley space $(X, \leq)$ in a natural way from the dual of $\leq$. For the definition of an inserter see Remark 5.2.9.

Proposition 5.5.13 *Let $(X, \leq)$ be Priestley space. Consider $\leq$ as a weakening relation between order-discrete Priestley spaces (i.e., Stone Spaces). Then the inserter of the dual of $\leq$ is the distributive lattice dual to $(X, \leq)$.*

Proof Recall that the distributive lattice dual to $(X, \leq)$ is given by the upper clopens of X, hence is a sublattice of the Boolean algebra of clopens $\mathbb{2}^X$ dualising X. We only need to show that this sublattice arises as the inserter of $j, k : \mathbb{2}^X \rightrightarrows \mathbb{2}^{\leq}$, where

(j, k) is the cospan dual to the span $X \leftarrow \leq \rightarrow X$. We use that inserters in distributive lattices are computed as inserters in Pos. The inserter of (j, k) is the set of clopens $a \in 2^X$ such that $j(a) \subseteq k(a)$, that is, such that $\{(x, y) \mid x \in a \;\&\; x \leq y\} \subseteq \{(x, y) \mid y \in a \;\&\; x \leq y\}$, which is the set of upwards closed clopens. $\square$

The proposition can also be proved more categorically. Since $(X, \leq)$ is the quotient (= coinserter) of X by $\leq$, the dual of $(X, \leq)$ must be the inserter of the dual of X by the dual of $\leq$.

We can summarise the previous example and proposition as follows. The reflexive elements (see also Examples 5.4.14 and 5.4.15) of the dual of X, that is those clopens a for which $\leq[a] \subseteq a$, form the dual of the Priestley space $(X, \leq)$. We next consider what happens if we start from an ordered Stone space that is not a Priestley space, an example due to Stralka.

Example 5.5.14 (*The ersatzkette* Stralka 1980) Let X be the middle-third Cantor space and $x \leq y$ be the relation that holds whenever x is the left-hand and y the right-hand endpoint of a middle-third gap. The dual $\sqsubseteq = \overline{2}(\leq)$ is given by $a \sqsubseteq b$ iff $a \subseteq b$ and b strictly extends a on the right.

The next example is at the heart of a forthcoming paper on extending Stone type dualities from the zero-dimensional to the compact Hausdorff setting.

Example 5.5.15 (*The unit interval*) Let X be the "middle-third" Cantor space and R the equivalence relation that identifies the endpoints at both sides of a gap. X is a Stone space. The dual of X is the Boolean algebra A of clopens of X. The dual $\prec = \overline{2}(R)$ satisfies $\overline{2}(R)(a, b)$ if and only if the closure of a is contained in b or, equivalently, if a is way-below b. We will develop the general theory at which this example is hinting at in a sequel paper. In a nutshell, the quotient of X by R is homeomorphic to the unit interval, and, at the same time, dual to the 'proximity lattice' $(A, \prec)$. This observation can be extended to a duality for compact Hausdorff spaces and proximity lattices (Moshier, 2004).

While the unit interval is the coinserter (or, because of discreteness, the coequalizer) of X wrt R, the inserter of the dual of R is not dual to the unit interval. The explanation for this mismatch is that in this case the coinserter and the inserter are not computed in dual categories. In forthcoming work we will present a category of algebras in which the inserter of the dual of R is indeed the dual of the unit interval (obviously, the forgetful functor from this category of algebras to Pos cannot preserve inserters and, hence, cannot preserve all Pos-limits).

5.6 Concretely Order-Regular Categories

In Theorem 5.5.9 we extended the duality between distributive lattices and Priestley spaces from maps to relations.

This construction from a duality of maps to a duality of relations is purely category theoretic and does not depend on the particularities of distributive lattices and Priestley spaces. All we need are comma objects and a factorisation system in order to compose relations and a duality of maps that respects this structure in a suitable sense. To work out the precise conditions is the purpose of this section. In the next section we can then prove Theorem 5.7.6 as a category theoretic generalization of Theorem 5.5.9.

The main results of this section are Definition 5.6.1 and Theorems 5.6.9 and 5.6.10 which generalise the approach described in Remark 5.4.6 to categories over Pos.

The general setting are two forgetful functors to Pos and two contravariant functors P, S which are adjoint on the right. In this section we concentrate on axiomatising the properties of V and U and will return to the adjunction in Sect. 5.7.

$$
\begin{tikzcd}
\mathcal{X} \arrow[r, bend left, "P"] \arrow[d, "V"'] & \mathcal{A} \arrow[l, bend left, "S"] \arrow[d, "U"] \\
\mathsf{Pos} & \mathsf{Pos}
\end{tikzcd}
$$

So far we took relations as basic, and spans and cospans as devices to represent relations. This can be transferred to concrete Pos-categories, that is, Pos-categories $\mathcal{C}$ with a forgetful functor

$$U : \mathcal{C} \to \mathsf{Pos}.$$

In particular, a relation $A \looparrowright B$ in $\mathcal{C}$ will be a relation $UA \looparrowright UB$. In order to make sure that a relation also respects the structure of $\mathcal{C}$, we add the requirement that $UA \looparrowright UB$ can be represented by a span in $\mathcal{C}$. Equivalently, we can say that a $\mathcal{C}$-relation $A \looparrowright B$ is a subobject of $A \times B$ that is upward closed in $A^{\mathrm{op}} \times B$.

Since the forgetful functors will be P-faithful, there is at most one relation in $\mathcal{C}$ over any $UA \looparrowright UB$ and the order between the relation is inherited from Pos.

To make sure that relations in $\mathcal{C}$ compose as they do in Pos, we ask $\mathcal{C}$ to have comma objects and factorisations preserved by U.

5.6.1 Concretely Order-Regular Categories

The following definition details the assumptions sketched above. For notions such as P-faithful, finite limits, (*Onto*, *Emb*), etc. see Sect. 5.2.1. For weakening-closed embedding spans, exact squares, etc. see Sect. 5.3.[8]

[8] We could follow a more abstract approach in which one defines a calculus of relations via a given set of squares declared to be exact. Then one only needs to require existence of enough exact squares as well as functors preserving them. But this would require the development of a theory that would distract from the duality theory we are interested in here. In all of our examples, relations

Definition 5.6.1 A Pos-functor $U : \mathcal{C} \to \mathsf{Pos}$, or just the category $\mathcal{C}$ if U is understood, satisfying the following properties will be called a **concretely order-regular category**.

- U is P-faithful, that is, order-preserving and order-reflecting on homsets.
- $\mathcal{C}$ has and U preserves finite limits in the Pos-enriched sense.
- $\mathcal{C}$ has a factorisation system $(\mathcal{E}, \mathcal{M})$ such that $U\mathcal{E} = \mathit{Onto}$ and $U\mathcal{M} = \mathit{Emb}$ and for all $(\mathit{Onto}, \mathit{Emb})$-factorisations $Uf = e \circ m$ there are unique $e' \in \mathcal{E}$ and $m' \in \mathcal{M}$ such that $Ue' = e$ and $Um' = m$.

Remark 5.6.2 The third item can be replaced by the stronger requirement that $\mathcal{C}$ has a P-regular/P-mono factorisation system given by $(U^{-1}\mathit{Onto}, U^{-1}\mathit{Emb})$. This would make sure that concretely order-regular categories are order-regular and still include Priestley spaces since the image of a continuous map between Priestley spaces is closed and, therefore, a Priestley space.

Remark 5.6.3 Definition 5.6.1 allows us to lift terminology from Pos to $\mathcal{C}$. For example,

- A surjection/embedding in $\mathcal{C}$ is an arrow f such that Uf is a surjection/embedding in Pos.
- A span $(p : W \to A, q : W \to B)$ in $\mathcal{C}$ is **weakening-closed** if (Up, Uq) is weakening-closed in Pos. The span (p, q) is an **embedding-span** if the image of $\langle p, q\rangle : W \to A \times B$ under U is an embedding in Pos.
- A square is **exact** in $\mathcal{C}$ if its image under U is exact in Pos. It follows that U (by definition) preserves exact squares.

Example 5.6.4
- All order-regular categories in the sense of Kurz & Velebil (2017), Definition 3.18 are concretely order-regular categories under mild conditions, see Kurz & Velebil, 2017, Theorem 5.13. This includes all quasi-varieties of ordered algebras as well as ordered compact Hausdorff spaces such as Priestley spaces.
- All regular categories are order-regular categories with discrete homsets. This includes the categories of compact Hausdorff spaces or Stone spaces and the category of Boolean algebras.

Definition 5.6.5 ($\mathcal{C}$*-relation*) Let $U : \mathcal{C} \to \mathsf{Pos}$ be a concretely order-regular category and $A, B \in \mathcal{C}$. A U-relation, or simply, a $\mathcal{C}$-relation, $A \looparrowright B$ is an isomorphism class of weakening closed embedding spans $A \leftarrow \bullet \to B$, or equivalently, an upward closed P-mono subobject of $A^{\mathrm{op}} \times B$.

Definition 5.6.6 Given a concretely order-regular category $U : \mathcal{C} \to \mathsf{Pos}$, the extension

$$\mathsf{Rel}(U) : \mathsf{Rel}(\mathcal{C}) \to \mathsf{Rel}(\mathsf{Pos}) \qquad \text{or shorter} \qquad \overline{U} : \overline{\mathcal{C}} \to \overline{\mathsf{Pos}},$$

are weakening relations in Pos with, possibly, additional properties. And this is what our notion of concretely order-regular captures.

is defined as follows. $\overline{\mathsf{Pos}}$ is the category $\mathsf{Rel}(\mathsf{Pos})$ defined in Sect. 5.2.2. $\overline{\mathcal{C}}$ has the same objects as $\mathcal{C}$ and $\mathcal{C}$-relations as arrows. The order on relations is inherited from Pos.

Remark 5.6.7 Composition in $\overline{\mathcal{C}}$ is associative (and $\overline{\mathcal{C}}$ is a category) since composition of weakening-closed embedding spans can be computed in the base category where it is relational composition. $\overline{U} : \overline{\mathcal{C}} \to \overline{\mathsf{Pos}}$ is a P-faithful functor since the order on arrows in $\overline{\mathcal{C}}$ is inherited from $\overline{\mathsf{Pos}}$.

The next definition generalises the corresponding notions from Pos, see Sect. 5.2.2, to a concretely order-regular category $\mathcal{C}$.

Definition 5.6.8 The functor

$$(-)_* : \mathcal{C} \to \overline{\mathcal{C}}^{\mathrm{co}}$$

takes a map $f : A \to B$ and maps it to the comma object of the cospan (f, id). The functor

$$(-)^* : \mathcal{C}^{\mathrm{op}} \to \overline{\mathcal{C}}$$

takes $f : A \to B$ and maps it to the comma object of (id, f).

Given our assumptions on U, we have that $f_*(a, b) = B(fa, b)$ for $f : A \to B$ and $f^*(a, b) = B(b, fa)$. It is worth emphasising that this means that if $f : A \to B$ is a $\mathcal{C}$-morphism, then the Pos-relations f_* and f^* are also $\mathcal{C}$-relations.

5.6.2 Extending Functors

The following extension theorems generalise (Bilkova et al., 2012, Theorem 4.1). We follow the notation of the survey (Kurz & Velebil 2016, Theorem 3.8) which is summarised in Remark 5.4.6. It states, informally speaking, that a functor extends from maps to relations if it preserves exact squares and maps epis to split epi relations.

Theorem 5.6.9 *Let $U : \mathcal{X} \to \mathsf{Pos}$ be a concretely order-regular category as in Definition 5.6.1. The locally monotone functor $(-)_* : \mathcal{X} \to \overline{\mathcal{X}}^{\mathrm{co}}$ has the following three properties:*

1. *$(-)_*$ preserves maps, that is, every f_* has a right-adjoint in $\overline{\mathcal{X}}$.*
2. *$q_* \cdot p^* = g^* \cdot f_*$ for all exact squares in $\mathcal{X}$*

$$\begin{tikzcd} & UW \arrow[dl, "Up"'] \arrow[dr, "Uq"] & \\ UX \arrow[dr, "Uf"'] & \leq & UY \arrow[dl, "Ug"] \\ & UZ & \end{tikzcd} \tag{5.26}$$

3. $e_* \cdot e^* = \mathrm{Id}$ *for all surjections* e *in* $\mathcal{X}$.

Moreover, the functor $(-)_*$ *is universal w.r.t. these three properties in the following sense: if* $\mathcal{K}$ *is any concretely order-regular category to give a locally monotone functor* $H : \overline{\mathcal{X}}^{\mathrm{co}} \to \mathcal{K}^{\mathrm{co}}$ *is the same as to give a locally monotone functor* $F : \mathcal{X} \to \mathcal{K}^{\mathrm{co}}$ *with the following three properties:*

1. *Every* Ff *has a right adjoint in* $\mathcal{K}$*, denoted by* $(Ff)^r$.
2. $Fq \cdot (Fp)^r = (Fg)^r \cdot Ff$ *for all exact squares as in* (5.26).
3. $Fe \cdot (Fe)^r = \mathrm{Id}$ *for all epise.*

Proof Since composition of $\mathcal{X}$-relations in $\overline{\mathcal{X}}$ is the same as the composition of the underlying relations in $\overline{\mathsf{Pos}}$, the properties 1–3 of $(-)_*$ follow from the corresponding facts on Pos. For the universal property, given F, we define $H(f_*) = Ff$ and on a general relation R we let

$$H(R) = H(cR_* \cdot dR^*) = F(cR) \cdot F(dR)^r.$$

In the case that R is the tabulation of f_*, we have $H(R) = F(cR) \cdot F(dR)^r = \mathrm{id}^r \cdot Ff = f_*$, because the square defining R as the comma-object of the cospan (f_*, id) is exact and because F satisfies property 2. A similar argument shows that H preserves identities. To show that H preserves composition, note that if R, S are relations in $\overline{\mathcal{X}}$, then applying F to the diagram (which abbreviates $R \cdot S$ to RS)

$$\begin{tikzcd}
 & & {} \arrow[ddrr, bend left, "cRS"] \arrow[ddll, bend right, "dRS"'] & & \\
 & & {} \arrow[u, two heads, "e"'] \arrow[dl, "dP"'] \arrow[dr, "cP"] & & \\
 & {} \arrow[dl, "dS"'] \arrow[dr, "cS"] & & {} \arrow[dl, "dR"'] \arrow[dr, "cR"] & \\
{} & & {} & & {}
\end{tikzcd} \tag{5.27}$$

we obtain $H(R \cdot S)$ as the relation represented by the outside span and $HR \cdot HS$ as the relation obtained from composing the bottom zig-zag. These two are the same because F satisfies properties 2 and 3. To show that H is locally monotone, let $R \subseteq S$ in $\overline{\mathcal{X}}$, that is, there is f in $\mathcal{X}$ such that $dR = dS \circ f$ and $cR = cS \circ f$. Then we calculate in $\mathcal{K}$

$$\begin{aligned}
H(R) &= F(cR) \cdot F(dR)^r \\
&= F(cS \circ f) \cdot F(dS \circ f)^r \\
&= F(cS) \cdot Ff \cdot Ff^r \cdot F(dS)^r \\
&\leq F(cS) \cdot F(dS)^r \\
&= H(S)
\end{aligned}$$

We have shown that $\overline{\mathcal{X}} \to \mathcal{K}$ is locally monotone. Hence $\overline{\mathcal{X}}^{\mathrm{co}} \to \mathcal{K}^{\mathrm{co}}$ is as well. $\square$

There is a dual version of the theorem. Since we need it later, we write it out in detail for reference.

Theorem 5.6.10 *Let $U : \mathcal{A} \to \mathsf{Pos}$ be an concretely order-regular category as in Definition 5.6.1. The locally monotone functor $(-)^* : \mathcal{A}^{\mathrm{op}} \to \overline{\mathcal{A}}$ has the following three properties:*

1. *Every f^* has a left-adjoint f_* in $\overline{\mathcal{A}}$.*
2. *$q_* \cdot p^* = g^* \cdot f_*$ for all exact squares in $\mathcal{A}$*

$$\begin{tikzcd} & UW \arrow[dl, "Up"'] \arrow[dr, "Uq"] & \\ UA \arrow[dr, "Uf"'] \arrow[rr, phantom, "\leq"] & & UB \arrow[dl, "Ug"] \\ & UC & \end{tikzcd} \tag{5.28}$$

3. *$e_* \cdot e^* = \mathrm{Id}$ for all surjections e in $\mathcal{A}$.*

Moreover, the functor $(-)^*$ is universal w.r.t. these three properties in the following sense: if $\mathcal{K}$ is any Pos-category to give a locally monotone functor $H : \overline{\mathcal{A}} \to \mathcal{K}$ is the same as to give a locally monotone functor $F : \mathcal{A}^{\mathrm{op}} \to \mathcal{K}$ with the following three properties:

1. *Every Ff has a left adjoint in $\mathcal{K}$, denoted by $(Ff)_l$.*
2. *$(Fq)_l \cdot Fp = Fg \cdot (Ff)_l$ for all exact squares as in (5.28).*
3. *$(Fe)_l \cdot Fe = \mathrm{Id}$ for all epise.*

Proof To aid future calculations, we emphasise some of the places where notation changes wrt to the proof of Theorem 5.6.9. Given F and $f : B \to A$, we define $H(f^*) = Ff : FA \to FB$ and for a general relation R we let

$$H(R) = H(cR_* \cdot dR^*) = F(cR)_l \cdot F(dR).$$

In the case that R is the tabulation of f^*, we have $H(R) = F(cR)_l \cdot F(dR) = Ff \cdot \mathrm{id}_l = Ff$. The computation showing that H is locally monotone runs as follows. Let $R \subseteq S$ in $\overline{\mathcal{A}}$, that is, there is f in $\mathcal{A}$ such that $dR = dS \circ f$ and $cR = cS \circ f$. Then we calculate in $\mathcal{K}$

$$\begin{aligned} H(R) &= F(cR)_l \cdot F(dR) \\ &= F(cS \circ f)_l \cdot F(dS \circ f) \\ &= F(cS)_l \cdot Ff_l \cdot Ff \cdot F(dS) \\ &\leq F(cS)_l \cdot F(dS) \\ &= H(S) \end{aligned}$$

showing that H is locally monotone. □

Remark 5.6.11 From the point of view of relations, the two theorems are the same. In both cases, we extend a functor to a relation R by tabulating the relation as $R = cR_* \cdot dR^*$ and applying the functor to the legs. We spelled them out both for reference in the next section.

Remark 5.6.12 In the previous two theorems, if the category on which the functor F is defined has exact cocommas, or enough exact squares, then we can drop the condition 3. Indeed let (p, q) and (r, s) be two composable spans. Let (u, v) be the comma of (q, r). Let (x, y) be the graph of the composition $(p, q); (r, s)$. To show that the extension to relations of F preserves composition, we need to show that $(Fp, Fq); (Fr, Fs)$ and (Fx, Fy) represent the same relation. Let (j, k) be a cospan completing (x, y) and (pu, qv) to exact squares. Since F preserves exact squares, all of $(Fp, Fq); (Fr, Fs)$ and $(F(pu), F(qv))$ and (Fx, Fy) represent the same relation.

5.6.3 Examples

In this section we illustrate Definition 5.6.5 of $\mathcal{C}$-relations by a range of examples. In particular, we will build some new dualities of categories where objects are equipped with additional structure in the form of $\mathcal{C}$-relations for various categories $\mathcal{C}$. We will instantiate these general constructions with the duality of Pri and DL-relations of Theorem 5.5.9. We start with an observation about completely distributive lattice relations.

Example 5.6.13 The functor $\overline{2} : \mathsf{Rel}(\mathsf{Pos})^{\mathrm{co}} \to \mathsf{Rel}(\mathsf{Pos})$ from Theorem 5.4.10 defined by $R \mapsto \{(a, b) \mid R[a] \subseteq b\}$ induces order-isomorphisms $\overline{2}_{X,Y} : \mathsf{Rel}(\mathsf{Pos})(X, Y)^{\mathrm{op}} \to \mathsf{Rel}(\mathsf{CDL})(2^X, 2^Y)$ where CDL is the category of completely distributive lattices.

For the remainder of this section, we let $U : \mathcal{X} \to \mathsf{Pos}$ and $V : \mathcal{A} \to \mathsf{Pos}$ be concrete order-regular categories. We start by generalising Example 5.4.14, noting that a reflexive and transitive relation is a monad in the category of relations and that an interpolative relation below the identity is a comonad.

Example 5.6.14 Let $H : \mathcal{X} \to \mathcal{A}$ be a contravariant functor preserving relations[9] that H extends to $\overline{H} : \mathsf{Rel}(\mathcal{X}) \to \mathsf{Rel}(\mathcal{A})^{\mathrm{co}}$. Since $\overline{H}$ is locally monotone, it maps monads (comonads) in $\mathsf{Rel}(\mathcal{X})$ to comonads (monads) in $\mathsf{Rel}(\mathcal{A})$.

Definition 5.6.15 ($\mathcal{C}$-Rel, $\mathcal{C}$-Pre, $\mathcal{C}$-Ipl) If $\mathcal{C}$ is a concretely order-regular category, then we denote by $\mathcal{C}$-Rel the category that has pairs (C, R) as objects where $C \in \mathcal{C}$

[9] A functor preserves relations if it preserves exact squares and factorisations. If the categories in question have enough exact squares, preservation of exact squares is enough.

and $R \subseteq C \times C$ is a $\mathcal{C}$-relation and arrows $f : C, R) \to (C', R')$ are functions $f : C \to C'$ such that $xRy \Rightarrow f(x)R'f(y)$ for all x, y in the underlying poset of C. $\mathcal{C}$-Pre and $\mathcal{C}$-Ipl are the full subcategories of $\mathcal{C}$-Rel of preorders and interpolative relations, respectively, see Example 5.6.14.

Example 5.6.16 [10] Stone-Rel and BA-Rel as well as Stone-Pre and BA-Ipl are dually equivalent.

To keep the exposition easy, we now specialise to the example above. But Theorems 5.6.17 and 5.6.18 below transfer to dual equivalences (F, G) that rely on dualising objects other than $\mathbb{2}$, see Remark 5.4.3.

Theorem 5.6.17 *Under the standing assumptions of this subsection, the category* $\mathcal{X}$-Rel *is dually equivalent to the category* $\mathcal{A}$-Rel.

Proof Exploiting Theorem 5.5.9, we only have to show that the dualising functor $\mathbb{2}^{-}$: Stone-Rel $\to$ BA-Rel, and its converse $\mathbb{2}^{-}$: BA-Rel $\to$ Stone-Rel, preserve homomorphisms. To this end, going back to (5.10), we consider

$$
\begin{tikzcd}
 & R \arrow[dl, "p"'] \arrow[dr, "q"] & & & R' \arrow[dl, "p'"'] \arrow[dr, "q'"] & \\
X \arrow[dr] \arrow[rrr, dotted, bend right, "f"'] & \leq & X \arrow[dl] \arrow[rrr, dotted, bend right, "f"'] & X' \arrow[dr, "a'"'] & \leq & X' \arrow[dl, "b'"] \\
 & \mathbb{2} & & & \mathbb{2} &
\end{tikzcd}
\tag{5.29}
$$

Assuming $xRy \Rightarrow f(x)R'f(y)$, we have to show that $(a', b') \in \overline{\mathbb{2}}(R') \Rightarrow (\mathbb{2}^{f}(a'), \mathbb{2}^{f}(b')) \in \overline{\mathbb{2}}(R)$. In other words, we have to show that if $\forall w \in R \,.\, \exists w' \in R' \,.\, fp(w)) \leq p'(w') \;\&\; q'(w') \leq fq(w)))$ and if $a'p' \leq b'q'$ then $a'fp(w) \leq b'fq(w)$ for all $w \in R$. This is straightforward. $\square$

Using Example 5.4.14, or Example 5.6.14, we can specialise this to preorders.

Theorem 5.6.18 *The category of preordered Stone spaces is dually equivalent to the full subcategory of* BA-Rel *in which objects* (A, R) *where* R *is interpolative (= idempotent below the identity).*

Theorems 5.6.17 and 5.6.18, transfer, mutatis mutandis, to other dual equivalences than Stone and BA including those that rely on other dualising objects.

[10] Stone-Pos also deserves attention.

5.7 Extending Equivalences and Adjunctions

We are interested in extending contravariant adjunctions and equivalences of Pos-categories from maps to relations. In the case of adjunctions, for Theorem 5.7.11, we need to appeal to the framed bicategories of Shulman (2008). We therefore treat the easier case of equivalences first. Theorem 5.7.6 is a direct generalization of Theorem 5.5.9 and we recommend to read Sect. 5.5.2 before reading this one.

5.7.1 Extending Equivalences to Categories of Relations

Let $U : \mathcal{A} \to \mathsf{Pos}$ and $V : \mathcal{X} \to \mathsf{Pos}$ be two concretely order-regular categories, see Definition 5.6.1.

Given a dual equivalence $F : \mathcal{X} \to \mathcal{A}$ and $G : \mathcal{A} \to \mathcal{X}$, we will extend it to $\overline{\mathcal{X}} = \mathsf{Rel}(\mathcal{X})$ and $\overline{\mathcal{A}} = \mathsf{Rel}(\mathcal{A})$ in Theorem 5.7.6. The plan is to apply Theorems 5.6.9 and 5.6.10 to the situation

$$
\begin{tikzcd}
\overline{\mathcal{X}}^{\mathrm{co}} \arrow[r, bend left, "\overline{F}"] & \overline{\mathcal{A}} \arrow[l, bend left, "\overline{G}"] \\
\mathcal{X} \arrow[u, "(-)_*"] \arrow[r, bend left, "F"] & \mathcal{A}^{\mathrm{op}} \arrow[u, "(-)^*"'] \arrow[l, bend left, "G"]
\end{tikzcd}
$$

To obtain $\overline{F}$ from Theorem 5.6.9, we define the functor

$$\mathcal{X} \to \overline{\mathcal{A}}$$

as mapping arrows $(f : X \to Y)$ to relations $Ff^* : FX \looparrowright FY$. That is, we have $(a, b) \in Ff^*$ iff $a \leq Ff(b)$.

Note that Ff^* has a left-adjoint in $\overline{\mathcal{A}}$ and hence a right adjoint $(Ff)^r = Ff_*$ in $\overline{\mathcal{A}}^{\mathrm{co}}$ as required by Theorem 5.6.9.

For the condition that $(F-)^*$ preserves exact squares, given an exact square in $\mathcal{X}$

$$
\begin{tikzcd}
& W \arrow[dl, "p"'] \arrow[dr, "q"] & \\
X \arrow[dr, "f"'] & \leq & Y \arrow[dl, "g"] \\
& C &
\end{tikzcd}
\tag{5.30}
$$

we need $Fq \cdot (Fp)^r = (Fg)^r \cdot Ff$ in $\overline{\mathcal{A}}^{\mathrm{co}}$, which is in $\overline{\mathcal{A}}$

$$Fq^* \cdot Fp_* = Fg_* \cdot Ff^* \tag{5.31}$$

as in Lemma 5.5.5 for the case of Priestley spaces and distributive lattices.

We also need that for all epis e in $\mathcal{X}$ we have $Fe \cdot (Fe)^r = \mathrm{Id}$ in $\overline{\mathcal{A}}^{\mathrm{co}}$, which is in $\overline{\mathcal{A}}$

$$Fe^* \cdot Fe_* = \mathrm{Id}, \tag{5.32}$$

which holds iff F maps surjections to embeddings, as in Lemma 5.5.8 for the case of distributive lattices and Priestley spaces.

Following exactly the same line of reasoning as for $\overline{F}$ above, to obtain $\overline{G} : \overline{\mathcal{A}} \to \overline{\mathcal{X}}^{\mathrm{co}}$ from Theorem 5.6.10, we let the functor

$$\mathcal{A}^{\mathrm{op}} \to \overline{\mathcal{X}}^{\mathrm{co}}$$

be given by mapping arrows $g : A \to B$ in $\mathcal{A}$ to relations $Gg_* : GB \nrightarrow GA$. That is, we have $(y, x) \in Gg_*$ iff $Gg(y) \leq x$. Note that Gg_* has a left adjoint

$$(Gg)_l = Gg^*$$

in $\overline{\mathcal{X}}^{\mathrm{co}}$, as required by Theorem 5.6.10. In order to verify that G satisfies the assumptions of Theorem 5.6.10, given an exact square

$$\begin{tikzcd} & W \arrow[dl, "p"'] \arrow[dr, "q"] & \\ A \arrow[dr, "f"'] \arrow[rr, phantom, "\leq"] & & B \arrow[dl, "g"] \\ & C & \end{tikzcd} \quad , \tag{5.33}$$

we need to check that $(Gq)_l \cdot Gp = Gg \cdot (Gf)_l$ in $\overline{\mathcal{X}}^{\mathrm{co}}$, which is in $\overline{\mathcal{X}}$

$$Gq^* \cdot Gp_* = Gg_* \cdot Gf^* \; . \tag{5.34}$$

We also need to check that for all epis e in $\mathcal{A}$ we have in $\overline{\mathcal{X}}^{\mathrm{co}}$

$$(Ge)_l \cdot Ge = \mathrm{Id} \; ,$$

which is in $\overline{\mathcal{X}}$

$$Ge^* \cdot Ge_* = \mathrm{Id} \; , \tag{5.35}$$

which holds iff G maps surjections to embeddings.

To summarize, we have the following corollaries of Theorems 5.6.9 and 5.6.10 about the situation depicted in

$$\begin{tikzcd} \overline{\mathcal{X}}^{\mathrm{co}} \arrow[r, bend left, "\overline{F}"] & \overline{\mathcal{A}} \arrow[l, bend left, "\overline{G}"] \\ \mathcal{X} \arrow[u, "(-)_*"] \arrow[r, bend left, "F"] & \mathcal{A}^{\mathrm{op}} \arrow[u, "(-)^*"'] \arrow[l, bend left, "G"] \end{tikzcd}$$

Proposition 5.7.1 *Let $\mathcal{X}$ and $\mathcal{A}$ be concretely order-regular categories (Definition 5.6.1). If a contravariant functor $F : \mathcal{X} \to \mathcal{A}$ preserves exact squares in the sense that $Fq^* \cdot Fp_* = Fg_* \cdot Ff^*$ for all exact squares as in (5.30) and if F takes surjections to embeddings, then F extends uniquely to a (covariant) functor $\overline{\mathcal{X}}^{\mathrm{co}} \to \overline{\mathcal{A}}$. A relation $r : X \looparrowright Y$ is mapped to $\overline{F}r : FX \looparrowright FY$ given by*

$$(a, b) \in \overline{F}r \;\;\Leftrightarrow\;\; Fp(a) \leq_{FW} Fq(b)$$

where $(p : W \to X, q : W \to Y)$ is a tabulation of r. $\overline{F}r$ is tabulated by the comma object of the cospan (Fp, Fq). In case that the relation is a map, that is, in case that $r = f_*$ for some $f : X \to Y$ this simplifies to

$$(a, b) \in \overline{F}(f_*) \;\;\Leftrightarrow\;\; a \leq_{FX} Ff(b).$$

Proof We know from Theorem 5.6.9 (with H being $\overline{F}$ and Ff being Ff^*) that $\overline{F}(r) = Fq^* \cdot Fp_*$, that is, $\overline{F}r(a, b) = FW(Fp(a), Fq(b))$. In case $r = f_*$, since the square defining (p, q) is exact, we have $\overline{F}(f_*)(a, b) = Ff^*(a, b) = FX(a, Ff(b))$. □

Remark 5.7.2 If $\mathcal{X}$ is a category of spaces and $Ff = 2^f = f^{-1}$, then $r : X \looparrowright Y$ is mapped to $\overline{F}r : FX \looparrowright FY$ such that, see Proposition 5.4.2,

$$(a, b) \in \overline{F}r \quad\Longleftrightarrow\quad (\, x \in a \;\&\; xry \;\Rightarrow\; y \in b \,)$$

which we may write in Hoare-triple notation as

$$\{a\}r\{b\}.$$

In case $r : X \looparrowright Y$ is a map $f : X \to Y$, that is, if $r = f_*$, which is $r(x, y) = Y(fx, y)$, then this can be written as

$$a \subseteq f^{-1}b.$$

The next result is analogous to Proposition 5.7.1, but worth spelling out for future reference.

Proposition 5.7.3 *Let $\mathcal{X}$ and $\mathcal{A}$ be concretely order-regular categories (Definition 5.6.1). If a contravariant functor $G : \mathcal{A} \to \mathcal{X}$ preserves exact squares* *in the sense that* $Gq^* \cdot Gp_* = Gg_* \cdot Gf^*$ *for all exact squares as in* (5.33) *and if* G *takes surjections to embeddings, then* G *extends uniquely to a (covariant) functor* $\overline{\mathcal{A}} \to \overline{\mathcal{X}}^{co}$. *A relation* $r : A \looparrowright B$ *is mapped to* $Gr : GA \looparrowright GB$ *given by*

$$(x, y) \in \overline{G}r \;\Leftrightarrow\; Gp(x) \leq_{GW} Gq(y)$$

where $(p : W \to A, q : W \to B)$ *is a tabulation of* r. Gr *is tabulated by the comma object of the cospan* (Gq, Gp). *In case that the relation is a map, that is, in case that* $r = g^*$ *for some* $g : B \to A$ *this simplifies to*

$$(x, y) \in \overline{G}(g^*) \;\Leftrightarrow\; x \leq_{GA} Gg(y)$$

Proof We know from Theorem 5.6.10 (with H being $\overline{G}$ and Fg being Gg_*) that $\overline{G}(r) = Gq^* \cdot Gp_*$, that is, $\overline{G}r(x, y) = GW(Gp(x), Gq(y))$. In case $r = g^*$, because the square defining (p, q) being exact, we have $\overline{G}(g_*)(x, y) = Gg_*(x, y) = GB(Gg(x), y)$. □

Remark 5.7.4 If $\mathcal{A} = \mathsf{DL}$ and $Gg = 2^g$, then a relation $\vdash : A \looparrowright B$ is mapped to $\overline{G}(\vdash) : GA \looparrowright GB$ such that for prime filters x, y

$$(x, y) \in \overline{G}(\vdash) \iff (a \in x \;\&\; a \vdash b \;\Rightarrow\; b \in y).$$

In the words of Remark 5.7.2, $G(\vdash)$ is the largest relation r making the Hoare triple $\{a\}r\{b\}$ true.

In case $\vdash : A \looparrowright B$ is a map $g : B \to A$, that is, if $\vdash = g^*$, which means $(a \vdash b) \Leftrightarrow (a \leq_A g(b))$, then this can be written as $x \leq_{GA} Gg(y)$ which translates as a statement about prime filters into

$$x \subseteq g^{-1}(y)$$

Before proving that dual equivalences extend from maps to relations, we need to check that the following holds.

Lemma 5.7.5 *Let dually equivalent* $F : \mathcal{X} \to \mathcal{A}$, $G : \mathcal{A} \to \mathcal{X}$ *satisfy the assumptions of Propositions* 5.7.1 *and* 5.7.3. *Let* $(p : W \to X, q : W \to Y)$ *be a span representing the relation* $r : X \looparrowright Y$. *Then* $\overline{G}\,\overline{F}r$ *is represented by* (GFp, GFq).

Proof $\overline{F}r$ is represented by (Fq, Fp). Let (p', q') be its comma object. Because $((p', q'), (Fq, Fp))$ is an exact square, and G preserves exact squares, we know that (Gq', Gp') and (GFp, GFq) represent the same relation. □

Combining Propositions 5.7.1 and 5.7.3 we obtain the following extension theorem.

Theorem 5.7.6 *Let* $\mathcal{X}$ *and* $\mathcal{A}$ *be concretely order-regular categories (Definition* 5.6.1*). Let* $F : \mathcal{X} \to \mathcal{A}$ *and* $G : \mathcal{A} \to \mathcal{X}$ *be a dual equivalence of contravariant functors satisfying the assumptions of Propositions* 5.7.1 *and* 5.7.3, *namely* *preservation of exact squares and the mapping of surjections to embeddings. Then F and G extend to an equivalence $\overline{F} : \overline{\mathcal{X}}^{\mathrm{co}} \to \overline{\mathcal{A}}$ and $\overline{G} : \overline{\mathcal{A}} \to \overline{\mathcal{X}}^{\mathrm{co}}$. Restricting this equivalence to maps as in*

$$
\begin{tikzcd}
\overline{\mathcal{X}}^{\mathrm{co}} \arrow[r, bend left, "\overline{F}"] & \overline{\mathcal{A}} \arrow[l, bend left, "\overline{G}"] \\
\mathcal{X} \arrow[u, "(-)_*"] \arrow[r, bend left, "F"] & \mathcal{A}^{\mathrm{op}} \arrow[u, "(-)^*"'] \arrow[l, bend left, "G"]
\end{tikzcd}
$$

gives back the dual equivalence (F, G).

Proof We have to show that the unit and counit are natural wrt relations. Using the previous lemma, it is enough to consider diagrams such as

$$
\begin{tikzcd}
A \arrow[rr, dotted, bend left, "R"] \arrow[d] & W \arrow[l] \arrow[r] \arrow[d] & B \arrow[d] \\
GFA \arrow[rr, dotted, bend right, "\overline{GFR}"'] & GFW \arrow[l] \arrow[r] & GFB
\end{tikzcd}
\tag{5.36}
$$

where the upper span tabulates a relation R and the vertical arrows are the unit of $F \dashv G$. The inner squares commute by naturality wrt maps, which implies that the outer rectangle (with the dotted horizontal arrows) commutes since the vertical arrows are isos. □

Remark 5.7.7 The previous proof relies on the units being isos. This is where we cannot weaken from dual equivalence to dual adjunctions. We will see how to deal with this with the help of double categories in the next section.

Remark 5.7.8 Let $F : \mathcal{X} \to \mathcal{A}$ and $G : \mathcal{A} \to \mathcal{X}$ be a dual equivalence of concretely order-regular categories. Then F, G preserve exact squares if and only if cocommas in $\mathcal{X}$ and $\mathcal{A}$ are exact. For "only if", note that the dual of a cocomma square in $\mathcal{A}$ is exact (due to being a comma square). It then follows from the functor preserving exactness that the cocomma square itself must be exact as well. For "if", consider an exact square (p, q, j, k) on one side with span (p, q) and cospan (j, k). Let (j', k') be the cocomma of (p, q) and (p', q') be the comma of (j', k'). The squares (p, q, j', k') and (p', q', j', k') are, respectively, cocomma and comma squares by definition. Then (p, q, j, k) is also a comma square. The dual squares are then also comma and cocomma squares, respectively. Since comma and cocomma squares are exact, so is the dual of (p, q, j, k).

5.7.2 *Extending Adjunctions to Double Categories of Relations*

In this section, we are going to extend adjunctions

$$\begin{tikzcd} \mathcal{X} \arrow[r, bend left, "F"] & \mathcal{A}^{\mathrm{op}} \arrow[l, bend left, "G"] \end{tikzcd}$$

to the corresponding categories of relations. As we noted above, the commutativity of Diagram (5.36) in $\mathsf{Rel}(\mathcal{X})$ depends on the unit of the adjunction being an isomorphism. Accordingly, in general, adjunctions on categories of maps do not extend to adjunctions on categories of relations. This problem can be solved by amalgamating the category of relations and the category of maps into a so-called weak double category (Grandis & Paré, 1999). As shown in Grandis & Paré (2004) this makes it possible to extend adjunctions to relations and various other structures such as spans/cospans and distributors. An excellent account can be found in the recent book by Grandis (2020). Framed bicategories are special weak double categories and we will rely on Shulman (2008) in the following.

Framed Bicategories. Framed bicategories (Shulman, 2008) allow us to have both $\mathcal{C}$ and $\mathsf{Rel}(\mathcal{C})$ in one structure, see also Example 2.6 in Shulman (2008). Since we only need a very special case of framed bicategories in this paper, we do not detail the general definition and only explain how any concretely order-regular category $\mathcal{C}$ gives rise to a framed bicategory $\S\mathcal{C}$.

Framed bicategories are special double categories (Grandis & Paré, 2004). Informally speaking, a 2-cell in a double category is a square

$$\begin{tikzcd} A \arrow[r, "R"] \arrow[d, "f"'] \arrow[dr, phantom, "\subseteq"] & B \arrow[d, "g"] \\ C \arrow[r, "S"'] & D \end{tikzcd}$$

where, in our examples, the horizontal arrows are relations, the vertical arrows are maps, and the 2-cell represents a subset-relation as indicated.In other words, forgetting the horizontal structure of $\S\mathcal{C}$ gives back $\mathcal{C}$ and forgetting the vertical structure of $\S\mathcal{C}$, we obtain $\mathsf{Rel}(\mathcal{C})$. Importantly, it is the double category theoretic view which gives us the right notion of functor and adjunction. The technical point where this matters can be seen if we go back to (5.36) and note that the unit of an adjunction $F \dashv G$ is not, in general, natural wrt relations. From a double category theoretic point of view, it suffices that the outer rectangle of (5.36) commutes up to a 2-cell.

More technically, we can define, ignoring issues of size, a double category as an internal category (Borceux, 1994, Chap. 8)

$$\mathbb{D}_1 \underset{cod}{\overset{dom}{\rightrightarrows}} \mathbb{D}_0$$

in Cat.Note that this point of view breaks the symmetry between 'vertical' arrows, which are arrows in $\mathbb{D}_0$, and 'horizontal' arrows, which are objects in $\mathbb{D}_1$. The internal composition in the $\mathbb{D}_i$ is vertical composition and the external composition of $\mathbb{D}_1$ is horizontal composition. Finally, a double category is a framed bicategory if every vertical arrow can be represented by horizontal arrows in a suitable way, see Grandis & Paré 2004, Sects. 1.2 and 1.3) and Shulman (2008), Theorems 4.1 and A.2) for details. This gives a double category theoretic axiomatisation of the two ways (5.1) and (5.2) of embedding maps into relations.

For our purposes, it suffices to know that the construction described in the next proposition is a framed bicategory. This then allows us to use that framed bicategories form a strict 2-category and, therefore, come with a native notion of adjunction. As it turns out, this notion of adjunction is precisely the one we need in Theorem 5.7.11 to prove that adjunctions extend from maps to relations.

We will write $\S\mathcal{C}$ for the framed bicategory of relations of the category $\mathcal{C}$. A framed bicategory is a double category (strict for us) with the additional property that for every horizontal 1-cell R and every pair (f, g) of vertical 1-cells as in

$$\begin{tikzcd} A \arrow[r, dotted, "R(f,g)"] \arrow[d, "f"'] & B \arrow[d, "g"] \\ C \arrow[r, "R"'] & D \end{tikzcd}$$

there is a unique cartesian lifting of R along (f, g). In our special case, the cartesian lifting (also known as the restriction) of $R : C^{op} \times D \to \mathbb{2}$ along (f, g) will be the relation $R(f, g)$, defined by mapping (a, b) to $R(f(a), g(b))$.

Remark 5.7.9 For the reader who wants to understand in detail how framed adjunctions apply to our setting, we give a brief guide to the notation of Shulman (2008). A, B are objects and f, g are vertical 1-cells (maps) and M, N are horizontal 1-cells (relations). We will write 1 for identity arrows dropping the usual subscript of $1_A : A \to A$ so that $A(1, 1)$ is the identity relation on A. In Shulman (2008), Definition 1, the horizontal 1-cell U_A is $A(1, 1)$, the 2-cell U_f records the fact that $A(1, 1) \leq B(f, f)$, that is, that $f : A \to B$ is monotone. Our notation for the horizontal composition $M \odot N$ is $M; N$ or $N \cdot M$. The restriction f^*Mg^*, that is, the cartesian lifting of M along (f, g), is $M(f, g)$, or, equivalently, $g^* \cdot M \cdot f_*$.[11] The extension $f_!Mg_!$, that is, the op-cartesian lifting of M along (f, g), is $g_* \cdot M \cdot f^*$. The base change object ${}_fB$ is $B(f, 1) = f_*$ and B_f is $B(1, f) = f^*$.

[11] Shulman uses $(-)^*$ to denote a cartesian lifting while we use $(-)^*$ for the embedding Pos $\to$ Rel(Pos).

Proposition 5.7.10 *Let $\mathcal{C}$ be a concretely order-regular category. Then there is a framed bicategory $\S\mathcal{C}$ that has the same objects as $\mathcal{C}$, that has the arrows of $\mathcal{C}$ as vertical arrows, and that has the arrows of $\mathsf{Rel}(\mathcal{C})$ as the horizontal arrows. The 2-cells are squares*

$$\begin{tikzcd} A \arrow[r, "S"] \arrow[d, "f"'] & B \arrow[d, "g"] \\ C \arrow[r, "R"'] & D \end{tikzcd} \tag{5.37}$$

such that $S \le R(f, g)$, or, equivalently, any of $g_* \cdot S \subseteq R \cdot f_*$ or $g_* \cdot S \cdot f^* \subseteq R$ or $S \subseteq g^* \cdot R \cdot f_*$.

Proof With the notation of the remark above, it is immediate to verify condition (iii) of Shulman (2008), Theorem 4.1. $\square$

We will write

$$\S\mathcal{C}^{\mathrm{co}} \qquad \text{and} \qquad \S\mathcal{C}^{\mathrm{op}}$$

for the framed bicategories that are the same as $\S\mathcal{C}$ but have, respectively, reversed 2-cells and reversed vertical 1-cells.

Extension Theorems. In the following proposition we assume that we have an adjunction $F \dashv G : \mathcal{A}^{\mathrm{op}} \to \mathcal{X}$ with F and G satisfying the assumptions that allow us to apply Propositions 5.7.1 and 5.7.3 in order to obtain extensions $\S F$ and $\S G$.

Theorem 5.7.11 *Let $\mathcal{X}$ and $\mathcal{A}$ be concretely order-regular categories and let $F \dashv G : \mathcal{A}^{\mathrm{op}} \to \mathcal{X}$ be an adjunction with both F and G preserving exact squares and mapping surjections to embeddings. Define the extensions $\S F$ and $\S G$ on 0- and 1-cells as F and G and on 2-cells by tabulation as in Proposition 5.7.1 for $\S F$ and as in Proposition 5.7.3 for $\S G$. Then these extensions*

$$\begin{tikzcd} \S\mathcal{X}^{\mathrm{co}} \arrow[r, bend left, "\S F"] & \S\mathcal{A} \arrow[l, bend left, "\S G"] \end{tikzcd}$$

constitute an adjunction of framed bicategories. Moreover, if F and G are an equivalence, so are $\S F$ and $\S G$.

Proof First, we have to check that $\S F$ and $\S G$ are framed functors. Defining them on objects and vertical 1-cells as F and G and on horizontal 1-cells as $\overline{F}$ and $\overline{G}$ as in Propositions 5.7.1 and 5.7.3, $\S F$ and $\S G$ are strong framed functors in the sense of Shulman (2008), Definitions 6.1, 6.14. It remains to see that the units $\eta : \mathrm{Id} \to GF$ and $\varepsilon : \mathrm{Id} \to FG$ of the adjunction extend to framed transformations (Shulman, 2008, Definitions 6.15, 6.16). Since our 2-cells are posetal, all 2-cell diagrams between the same 1-cells commute. So it suffices to show that for all relations $R : A \to B$ we have a 2-cell $(\eta_A, \eta_B) : R \Rightarrow GFR$, that is, $(\eta_B)_* \cdot R \subseteq GFR \cdot (\eta_A)*$ and this follows from the two squares in (5.36) commuting. $\square$

As a corollary we obtain a result in the same spirit as the equivalence Theorem 5.7.6. But, technically, they are different theorems, because Theorem 5.7.6 is about categories where relations are arrows, whereas Corollary 5.7.12 is about framed bicategories where relations are objects parameterised by maps.

Corollary 5.7.12 *Let $\mathcal{X}$ and $\mathcal{A}$ be concretely order-regular categories and let $F \dashv G : \mathcal{A}^{\mathrm{op}} \to \mathcal{X}$ be a dual equivalence with both F and G preserving exact squares and mapping surjections to embeddings. Then there is an equivalence between the framed bicategories $\S\mathcal{X}^{\mathrm{co}}$ and $\S\mathcal{A}^{\mathrm{op}}$, determined by the action of F and G on vertical arrows.*

The next theorem shows that the adjunction 'homming into $\mathbb{2}$' extends to relations. It only works for the framed bicategory $\S\mathsf{Pos}$ and has no analogue in terms of $\mathsf{Rel}(\mathsf{Pos})$.

Corollary 5.7.13 $\mathbb{2}^- \dashv \mathbb{2}^- : \S\mathsf{Pos}^{\mathrm{op}} \to \S\mathsf{Pos}$ *is a framed (and op-framed) adjunction.*

5.7.3 Examples

We exhibit two further dualities that satisfy the assumptions of Theorems 5.7.6 and 5.7.11. We start with some remarks on semi-lattices in ordered categories.

Pos-algebras (or Pri-algebras) have monotonic (or monotonic continuous) operations. But if the operations themselves determine a partial order, for example, if one of the operations is associative and idempotent, the underlying partial order does not have to coincide with the algebraically determined order. For example, it is possible to have a lattice in Pos for which the lattice order is not the same as the underlying order (take a lattice with discrete underlying set). So care needs to be taken in specifying how the poset order and derived order relate.

In Set, a semilattice is only conventionally spoken of being a meet or join semilattice depending on intuition. In Pos, a semilattice may actually be a meet or join semilattice (or neither) according to whether the underlying poset order coincides with the order defined by the lattice operation, or its opposite. Thus we call a unital semilattice $(X, *, e)$ in Pos a *unital meet semilattice* if $x \leq y$ coincides with $x = x * y$, and e is the maximal element. Likewise we call it a *unital join semilattice* if $x \leq y$ coincides with $x * y = y$, and e is the minimal element.

To show that the conditions of Theorems 5.7.6 and 5.7.11 are satisfied for a particular natural duality, the key step is to verify that the functors mediating the duality preserve exact squares.

We use the notation of the proof of Lemma 5.5.5.

5.7.3.1 Hofmann-Mislove-Stralka Duality

Hofmann-Mislove-Stralka duality (1974) establishes that the duals of unital meet semilattices in Pos are unital join semilattices in Pri, where the semilattice order and Priestley order coincide. So take a *Hofmann-Mislove-Stralka space*, or HMS space, to be a unital join semilattice in Pri.

Suppose we have an exact square (5.15) in meet semilattices. We must show that the dual square is exact in HMS spaces.

Since $\mathbb{2}^{-}$ preserves order on morphisms, $\mathbb{2}^{p} \circ \mathbb{2}^{f} \leq \mathbb{2}^{q} \circ \mathbb{2}^{g}$. Consider some $a \in \mathbb{2}^{X}, b \in \mathbb{2}^{Y}$ so that $\mathbb{2}^{p}(a) \leq \mathbb{2}^{q}(b)$. Then $f[a_{+}]$, see (5.20), is a filter. So it corresponds to an element c of $\mathbb{2}^{Z}$, which by construction satisfies $a \leq \mathbb{2}^{f}(c)$. Also if $\mathbb{2}^{g}(c)(y) = 1$, then there is some x so that $a(x) = 1$ and $f(x) \leq g(y)$. So by exactness of the given square, pick w so that $x \leq p(w)$ and $q(w) \leq y$. Hence $1 = a(x) \leq a(p(w)) \leq b(q(w)) \leq b(y)$. We have shown that $a \leq \mathbb{2}^{f}(c)$ and $\mathbb{2}^{g}(c) \leq b$, that is, the dual square in HMS is exact.

In the other direction, suppose we have an exact square (5.15) in HMS spaces. Again $\mathbb{2}^{-}$ preserves order on morphisms, so $\mathbb{2}^{p} \circ \mathbb{2}^{f} \leq \mathbb{2}^{q} \circ \mathbb{2}^{g}$.

In an HMS space a closed ideal is principal. This follows from the following observations. As Priestley spaces, HMS spaces are bitopologically spectral spaces. That is, (i) the upper opens constitute a spectral topology, as do the lower opens, (ii) the Priestley order is the specialization order for the upper open topology, and is the converse of the specialization order for the lower topology, and (iii) the Priestley topology is the join of these two spectral topologies. In particular, the upper open topology is sober. So specialization is a dcpo. Suppose I is a closed ideal. Since it is a downset, it is closed in the upper open set topology. Suppose $C \cup D \subseteq I$ for two closed sets C and D. If $x \in I \setminus C$ and $y \in I \setminus D$, then $x \wedge y \in I \setminus (C \cup D)$. So I is an irreducible closed, and must be the closure (in the upper open set topology) of a point.

Suppose $\mathbb{2}^{p}(a) \leq \mathbb{2}^{q}(b)$. Then $f[a_{+}]$ and $g[b_{-}]$, see (5.20) and (5.21), must be disjoint. For suppose not. Then for some x and y, $a(x) = 1$, $f(x) \leq g(y)$, and $b(y) = 0$. By exactness, there is a w so that $x \leq p(w)$ and $q(w) \leq y$. But then $a(x) \leq b(y)$, contradicting $a(x) = 1$ and $b(y) = 0$.

Since f is continuous, and a_{+} is clopen, $f[a_{+}]$ is compact. And since $g[b_{-}]$ is a principal ideal, $g[b_{-}] = {\downarrow} g(y_{*})$ for some $y_{*} \in b$.

For each $x \in a_{+}$, $f(x) \not\leq g(y_{*})$. So there is a clopen ideal I_{x} separating them. That is, $g(y_{*}) \in I_{x}$ and $f(x) \notin I_{x}$. The complements of these I_{x}'s form an open cover of $f[a_{+}]$. So finitely many suffice, and the intersection of the corresponding clopen ideals contains $g(y_{*})$, and is disjoint from $f[a_{+}]$. This intersection is itself a clopen ideal determining an HMS morphism $c \in \mathbb{2}^{Z}$. Clearly, $a \leq \mathbb{2}^{f}(c)$ and $\mathbb{2}^{g}(c) \leq b$ directly by the construction.

5.7.3.2 Banaschewski Duality

Banaschewski (1976) shows, in effect, that the topological duals of posets are bounded distributive lattices in Pri where the underlying order coincides with the lattice order—we call such spaces *Banaschewski spaces*.

Suppose (5.15) is an exact square in Banaschewski spaces. Then $\mathbb{2}^p \circ \mathbb{2}^f \leq \mathbb{2}^q \circ \mathbb{2}^g$ in Pos.

Suppose $\mathbb{2}^p(a) \leq \mathbb{2}^q(b)$. By the same argument as in HMS spaces, $g[b_-]$ is a principal ideal and $f[a_+]$ is a principal filter. Let y_* be the generator of $g[b_-]$ and x_* be the generator of $f[a_+]$. Then $f(x_*) \not\leq g(y_*)$ by exactness of the given square. So there is a closed prime ideal separating them.

Suppose (5.15) is an exact square in Pos. Then $\mathbb{2}^p \circ \mathbb{2}^f \leq \mathbb{2}^q \circ \mathbb{2}^g$ in Banaschewski spaces. If $\mathbb{2}^p(a) \leq \mathbb{2}^q(b)$, then $f[a_+]$ is an up-set, $g[b_-]$ is a down-set, and the two are disjoint. So $f[a_+]$ determines an c element of $\mathbb{2}^C$ that satisfies $a \leq \mathbb{2}^f(c)$ by construction. Clearly, $\mathbb{2}^g(c)(y) = c(g(y)) \leq b(y)$ for every $y \in Y$.

5.8 Conclusion

We showed how to extend an equivalence or adjunction from maps to relations. In more detail, Theorem 5.7.6 extends a dual equivalence of maps to a dual equivalence of relations, while Theorem 5.7.11 extends a dual adjunction (or equivalence) of maps to a dual adjunction (or equivalence) of the framed bicategory of relations.

The general framework is that of regular categories in a suitable order-enriched sense. Roughly speaking, the categories in question must have forgetful functors that preserve order-enriched limits and preserve regular factorisations; and the adjoint functors must preserve exact squares and regular factorisations.

In our experience, to exhibit a particular example of an adjunction or equivalence satisfying these conditions, most of the work will go into verifying preservation of exact squares, see Lemma 5.5.5 for our main example. It is worth noting that the proofs involving the dualising object $\mathbb{2}$, always follow the same common outline inherited from Pos, with the particularities of the situation entering only in one specific place, see Lemma 5.5.5 and the proofs of Sect. 5.7.3 for specific examples.

In a sequel paper, we will apply the duality of relations in order to extend zero-dimensional dualities to continuous ones in a systematic way. As we have seen here, dualities such as the one between ordered Stone spaces and distributive lattices can be extended from maps to relations. Once we have relations, we can split idempotents and then restrict to maps again, obtaining a non-zero dimensional duality.

For future investigations, two important questions concern other dualising objects than $\mathbb{2}$. First, while staying inside order-enriched categories, we plan to integrate our work here into the theory of natural dualities as described by Clark & Davey (1998) and to specialise Theorems 5.7.6 and 5.7.11 to this setting. Also possible relationships with Jónnson-Tarski duality (1952) and the theory of canonical extensions (Venema, 2006; Dunn et al., 2005) should be explored.

Second, we want to know whether our approach can be extended to other enrichments than $\mathbb{2}$ as for example Lawvere metric spaces (Lawvere, 1973). In particular, it would be interesting to see whether this could find applications to stochastic relations as studied, for example, in Doberkat (2007) and Panangaden (2009).

Another question is how much of the theory developed in this paper can be salvaged for functors that do not preserve exact squares. Looking back to Theorem 5.7.11, even without the assumption of preservation of exact squares, we are in a situation similar to orthogonal adjunction in the double category of pseudo double categories with lax and colax double functors as in Sect. 5.3 of Grandis & Paré (2004).

From a category theoretic point of view, there is the question how much of the theory of regular categories transfers to order-regular categories. While P-varieties feature prominently in our work, recent work by Abramsky and coauthors (Abramsky et al. 2017; Abramsky & Shah 2018 on game comonads suggests potential examples of enriched *co*varieties. In particular, one could have a look at games for continuous model theory, which has been given a category theoretic foundation recently by Cho (2020).

There is also a long list of more specific questions. For example, as discussed after Example 5.5.11, it should be interesting to look at dual relations of not-necessarily-bounded distributive lattices. Or a wide range of other dualities, for that matter. Finally, there are a number of technical questions, for example whether cocommas are exact in all order-regular categories or how an explicit characterisations of cocommas in various algebraic categories including distributive lattices would look like.

Returning to more fundamental questions, this paper focussed on heterogeneous relations $A \looparrowright B$ in the context of order-enriched algebra. An investigation into homogeneous relations $A \looparrowright A$ in the presence of order-preserving as well as order-reversing operations is one important topic for future investigation, in particular in connection with some recent work in proof theory of Greco et al. (2020). Another question is whether our approach can be extended to relations $A_1 \times \dots A_n \looparrowright B_1 \times \dots B_m$.

References

Abramsky, S. (1991). Domain theory in logical form. *Annals of Pure and Applied Logic, 51*(1–2), 1–77. https://www.cs.ox.ac.uk/files/295/dtlf.pdf

Abramsky, S., & Shah, N. (2018). Relating structure and power: Comonadic semantics for computational resources. CSL.

Abramsky, S., Dawar, A., & Wang, P. (2017). The pebbling comonad in finite model theory. LICS.

Bakhtiari, Z., & Hansen, H. H. (2017). Bisimulation for weakly expressive coalgebraic modal logics, CALCO. https://drops.dagstuhl.de/opus/volltexte/2017/8050/pdf/LIPIcs-CALCO-2017-4.pdf

Banaschewski, B. (1976). Remarks on dual adjointness. *Nordwestdeutsches Kategorienseminar, Math.-Arbeitspapiere, Teil A: Math. Forschungspapiere, 7*, 3–10.

Barr, M. (1970). Relational algebras. *Lecture Notes in Mathematics, 137,* 39–55. https://www.math.mcgill.ca/barr/papers/relalgs.pdf

Bilkova, M., Kurz, A., Petrisan, D., & Velebil, J. (2012). Relation Liftings on preorders and posets. arXiv:1210.1433.

Bilkova, M., Kurz, A., Petrisan, D., & Velebil, J. (2013). Relation lifting, with an application to the many-valued cover modality. *Logical Methods in Computer Science, 9*(4). https://arxiv.org/pdf/1307.4682.pdf

Bloom, S., & Wright, J. (1983). P-varieties - a signature independent characterization of varieties of ordered algebras. *Journal of Pure and Applied Algebra, 29,* 13–58. https://core.ac.uk/download/pdf/81944428.pdf

Bonsangue, M., van Breugel, F., & Rutten, J. (1998). Generalized metric spaces: Completion, topology, and powerdomains via the Yoneda embedding. *Theoretical Computer Science, 193*(1–2), 1–51.

Borceux, F. (1994). *Handbook of categorical algebra 1*. Cambridge University Press.

Bourke, J. (2010). *Codescent objects in 2-dimensional universal algebra*. Ph.D. Thesis, University of Sydney.

Bourke, J., & Garner, R. (2013). Two-dimensional regularity and exactness. https://arxiv.org/pdf/1304.5275.pdf.

Carboni, A., & Vitale, E. M. (1998). Regular and exact completions. *Journal of Pure and Applied Algebra, 125,* 79–116. https://core.ac.uk/download/pdf/82574355.pdf.

Cho, S. (2020). Categorical semantics of metric spaces and continuous logic *The Journal of Symbolic Logic,*. https://arxiv.org/pdf/1901.09077.

Clark, D., & Davey, B. (1998). *Natural dualities for the working algebraist*. Cambridge University Press.

Dahlqvist, F., & Kurz, A. (2017). The positivication of coalgebraic logics. CALCO. https://arxiv.org/pdf/1812.07288.pdf

Davey, B., & Priestley, H. (2002). *Introduction to lattices and order*. Cambridge University Press.

de Groot, J., Hansen, H. H., & Kurz, A. (2020). Logic-Induced Bisimulations. AiML. https://arxiv.org/pdf/2008.09238.

Doberkat, E.-E. (2007). *Stochastic relations*. Chapman & Hall.

Dunn, M., Gehrke, M., & Palmigiano, A. (2005). Canonical extensions and relational completeness of some substructural logics. *The Journal of Symbolic Logic, 70*(3), 713–740.

Freyd, P., & Scedrov, A. (1990). *Categories*. Allegories: North-Holland.

Galatos, N., & Jipsen, P. (2020). The structure of generalized BI-algebras and weakening relation algebras. *Algebra Universalis, 81,* 35. https://digitalcommons.chapman.edu/scs_articles/667/.

Garner, R., & Lack, S. (2012). Lex colimits. *Journal of Pure and Applied Algebra, 126*(6), 1372–1396.

Goguen, J., Thatcher, J., Wagner, E., & Wright, J. (1977). Initial algebra semantics and continuous algebras. *Journal of ACM, 24*(1), 68–95. https://arxiv.org/pdf/1107.0778.pdf.

Grandis, M. (2020). *Higher dimensional categories - from double categories to multiple categories*. World Scientific.

Grandis, M., & Paré, R. (1999). Limits in double categories, *Cahiers de Topologie et Géométrie Différentielle, 40*(3), 162–220. http://www.numdam.org/article/CTGDC_1999__40_3_162_0.pdf.

Grandis, M., & Paré, R. (2004). Adjoint for double categories, *Cahiers de Topologie et Géométrie Différentielle, 45*(3), 193–240. http://archive.numdam.org/article/CTGDC_2004__45_3_193_0.pdf.

Greco, G., Richard, V., Moortgat, M., & Tzimoulis, A. (2020). Lambek-Grishin calculus: Focusing, display and full polarization. arXiv:2011.02895.

Guitart, R. (1980). Relations et carrés exacts. *Annales Mathematiques du Quebec, IV*(2), 103–125.

Gutierres, G., & Hofmann, D. (2013). Approaching metric domains. *Applied Categorical Structures, 21*(6), 617–650.

Hansen, H. H., Kupke, C., & Pacuit, E. (2009). Neighbourhood structures: Bisimilarity and basic model theory. *Logical Methods in Computer Science, 5*(2). https://arxiv.org/pdf/0901.4430.

Hermida, C. (2011). A categorical outlook on relational modalities and simulations. *Information & Computation, 209*(12), 1505–1517. https://www.sciencedirect.com/science/article/pii/S0890540111001428/pdf.

Hermida, C., & Jacobs, B. (1998). Structural induction and coinduction in a fibrational setting. *Information & Computation, 145*(2), 107–152. https://www.sciencedirect.com/science/article/pii/S0890540198927250/pdf.

Hofmann, K. H., Mislove, M., Michael, & Stralka, A. (1974). *The Pontryagin duality of compact O-dimensional semilattices and its applications*. Springer.

Jipsen, P. (2017). Relation algebras, idempotent semirings and generalized bunched implication algebras, RAMiCS. https://digitalcommons.chapman.edu/cgi/viewcontent.cgi?article=1568&context=scs_articles.

Johnstone, P. T. (1982). *Stone spaces*, Cambridge University Press.

Jónsson, B., & Tarski, A. (1952). Boolean algebras with operators. *American Journal of Mathematics, 74*(1), 127–162. https://www.jstor.org/stable/2372074.

Jung, A., & Moshier, M. A. (2006). *On the bitopological nature of Stone duality*. University of Birmingham, School of Computer Science, Technical Report CSR-06-13.

Jung, A.,& Sünderhauf, P. (1996). On the duality of compact vs. open, *Annals of the New York Academy of Sciences, 806,* 214–230. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.47.662&rep=rep1&type=pdf.

Jung, A., Kegelmann, M., & Moshier, M. A. (1999). Multi lingual sequent calculus and coherent spaces. *Fundamenta Informaticae, 37*(4), 369–412.

Kelly, G. M. (1982). *Basic concepts of enriched category theory*. *London mathematical society lecture notes series* (Vol. 64). http://www.tac.mta.ca/tac/reprints/articles/10/tr10.pdf.

Kelly, G. M., & Power, A. J. (1993). Adjunctions whose counits are coequalizers, and presentations of finitary enriched monads, *Journal of Pure and Applied Algebra, 89,* 163–179. https://www.sciencedirect.com/science/article/pii/0022404993900928/pdf.

Kurz, A., & Velebil, J. (2016). Relation lifting, a survey. *Journal of Logarithms Algebra Method Program, 85*(4), 475–499. https://alexhkurz.github.io/papers/kv-relation-lifting.pdf.

Kurz, A., & Velebil, J. (2017). Quasivarieties and varieties of ordered algebras: Regularity and exactness. *Mathematical Structures in Computer Science, 27*(7), 1153–1194.

Lack, S. (1999). A note on the exact completion of a regular category, and its infinitary generalizations. *Theory and Applications of Categories,5*(3), 70–80. http://www.tac.mta.ca/tac/volumes/1999/n3/5-03abs.html.

Lane, S. M. (1971). *Categories for the working mathematician*. Springer.

Lawvere, F. W. (1973). Metric spaces, generalized logic and closed categories. *Rendiconti del Seminario Matematico e Fisico di Milano, XLIII,* 135–166. http://tac.mta.ca/tac/reprints/articles/1/tr1.pdf.

Moshier, M. A. (2004). On the relationship between compact regularity and Gentzen's cut rule. *Theoretical Computer Science, 316*, 113–136.

Moss, L. (1999). Coalgebraic logic. *Annals of Pure and Applied Logic, 96,* 277–317. https://www.sciencedirect.com/science/article/pii/S0168007298000426/pdf.

Panangaden, P. (2009). *Labelled Markov processes*. Imperial College Press.

Pigozzi, D. (2004). Partially ordered varieties and quasivarieties. http://orion.math.iastate.edu/dpigozzi/.

Priestley, H. A. (1970). Representation of Distributive Lattices by means of ordered Stone Spaces. *Bulletin of the London Mathematical Society, 2*, 186–190.

Rutten, J. (1996). Elements of generalized ultrametric domain theory. *Theoretical Computer Science, 170*(1–2), 349–381. https://www.sciencedirect.com/science/article/pii/S0304397596807110/pdf.

Rutten, J. (1998). Relators and metric bisimulations. *Electronic Notes in Theoretical Computer Science, 11,* 252–258. https://www.sciencedirect.com/science/article/pii/S1571066104000635/pdf.

Scott, D. (1970). Outline of a mathematical theory of computation. In Proceedings of 4th annual Princeton conference on information sciences and systems (pp. 169–176). https://www.cs.ox.ac.uk/files/3222/PRG02.pdf.

Scott, D. (1976). Data types as lattices. *SIAM Journal on Computing, 5*(3), 522–587. http://www.cs.ox.ac.uk/files/3287/PRG05.pdf.

Shulman, M. (2008). Framed bicategories and monoidal fibrations. *Theory and Applications of Categories, 20*(18), 650–738. https://arxiv.org/pdf/0706.1286.pdf.

Smyth, M. B. & Plotkin, G. D. (1982). The category-theoretic solution of recursive domain equations. *SIAM Journal on Computing, 11*(4), 761–783. https://core.ac.uk/download/pdf/204746035.pdf.

Smyth, M. B. (1992). Stable compactification I. *Journal of the London Mathematical Society, 45,* 321–340. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.105.3358&rep=rep1&type=pdf.

Stone, M. H. (1937). Topological representations of distributive lattices and Brouwerian logics. *Casopis pro pesiovani matematiky a fysiky, 67,* 1–25. http://dml.cz/bitstream/handle/10338.dmlcz/124080/CasPestMatFys_067-1938-1_1.pdf.

Stralka, A. (1980). A partially ordered space which is not a Priestly space. *Semigroup Forum, 20*, 293–297.

R. Street: Fibrations in bicategories. *Cahiers de Topologie et Géométrie Différentielle, 21*(2), 111–160. http://archive.numdam.org/article/CTGDC_1980__21_2_111_0.pdf.

Street, R. (1974). Fibrations and Yoneda's lemma in a 2-category. *Lecture Notes in Mathematics, 420*, 104–133.

Street, R. (1982). Two-dimensional sheaf theory. *Journal of Pure and Applied Algebra, 24*, 251–270.

Stubbe, I. (2007). Towards "dynamic domains": Totally continuous cocomplete Q-categories. *Theoretical Computer Science, 373*(1–2), 142–160.

Trnková, V. (1980). General theory of relational automata. *Fundamenta Informaticae, 3*(2), 189–234.

Venema, Y. (2006). Algebras and coalgebras. In *Handbook of modal logic*. Elsevier.

Wagner, K. R. (1994). Solving Recursive Domain Equations with Enriched Categories, Ph.D. thesis, Carnegie Mellon Univeristy. http://ra.adm.cs.cmu.edu/anon/usr/anon/1994/CMU-CS-94-159.ps.

Waszkiewicz, P. (2002). Quantitative continuous domains, Ph.D. thesis, University of Birmingham.

Worrell, J. (2000). Coinduction for recursive data types: partial orders, metric spaces and Ω-categories. *Electronic Notes in Theoretical Computer Science, 33*, 337–356. https://www.sciencedirect.com/science/article/pii/S1571066105803561.

Part II
Game Semantics

Chapter 6
The Far Side of the Cube

An Elementary Introduction to Game Semantics

Dan R. Ghica

Abstract Game-semantic models usually start from the core model of the prototypical language PCF, which is characterised by a range of combinatorial constraints on the shape of plays. Relaxing each such constraint usually corresponds to the introduction of a new language operation, a feature of game semantics commonly known as the *Abramsky Cube*. In this presentation we relax all such combinatorial constraints, resulting in the most general game model, in which all the other game models live. This is perhaps the simplest set up in which to understand game semantics, so it should serve as a portal to the other, more complex, game models in the literature. It might also be interesting in its own right, as an extremal instance of the game-semantic paradigm.

6.1 Game Semantics and Definability

Thus we begin to develop a semantic taxonomy of constraints on strategies mirroring the presence or absence of various kinds of computational features.

(Abramsky and McCusker 1996)

A *denotational semantics* models a programming language by translating it into a mathematical *semantic domain*. This approach was pioneered by Scott and Strachey (1971), motivated by reasoning about program and compiler correctness. For mathematical and philosophical reasons it makes sense to define this translation function *compositionally* from the structure of the language syntax. This translation function will not be, except in the most trivial cases, an *injection* since the same semantic concept can have multiple syntactic representations. This is part of the challenge and attraction of understanding languages. On the other hand, also for mathematical and philosophical reasons, it is preferable if the translation function is a *surjection*, meaning that every semantic concept has a syntactic representation, as pointed out

D. R. Ghica (✉)
University of Birmingham, Birmingham, England
e-mail: dan@ghica.net

A. Palmigiano and M. Sadrzadeh (eds.), *Samson Abramsky on Logic and Structure in Computer Science and Beyond*, Outstanding Contributions to Logic 25,
https://doi.org/10.1007/978-3-031-24117-8_6

by Lawvere (1969). Informally speaking, this simply means that there is no '*junk*' in the semantics or, conversely, that there are no missing elements in the syntax. This property is called *definability*.

There are other, more basic, requirements that a denotational semantics must meet. It must be *sound*, meaning essentially that distinct syntactic entities are not wrongly identified (for example *true* and *false*, or 1 and 0), and it must be *adequate* meaning that *terminating* and *non-terminating* programs are not mistakenly identified. Definability, soundness, and adequacy together establish that the translation is mathematically precise: it will translate all and only terms which are *equivalent* in the syntax into *equal* mathematical objects. This ideal situation is called *full abstraction*. And, as many ideal situations, it turns out to be difficult to achieve, primarily because of failure of definability. This was first pointed out by Plotkin (1977) in the case of PCF, a simple yet surprisingly challenging functional language.

The failure of full abstraction indicates a mismatch between syntax and the semantic domain, which can be resolved in two ways. The first one is to enrich the syntax with the missing operations, a course of action taken in *loc. cit.*, to wit, by adding a 'parallel-or' operator. If we think of syntax as mere notation for semantic concepts, which we may hold as essential, this would seem the right course of action. There is however a second way to mend the gap, by removing from the semantics those objects that have no syntactic expression. This may seem a somewhat surprising concession to the preeminence of syntax, but it is more than that. Solving full abstraction, which means solving definability, is a challenging litmus test to the power of a semantic methodology. This difficult mathematical problem, in the context of any non-trivial (and not contrived) programming language, remained open for about two decades. Hyland and Ong (2000) give an excellent scholarly account of the quest for answering this question (Sect. 1.3, *loc. cit.*).

The solution to the problem of definability was brought about by *game semantics*. For a tutorial introduction, history and overview of the subject the reader is referred to Abramsky and McCusker (1999); Ghica (2009); Murawski and Tzevelekos (2016). Of particular interest to us is the (Hyland & Ong, 2000) model, which along with Abramsky et al. (2000), is one of the original game-semantic models for PCF which achieves definability. It also introduces a style of game semantics, based on so-called *pointer sequences*, which proved to be very successful because of its flexibility. Using this style of game semantics, Abramsky and McCusker (1996) gave the first fully abstract model of Reynold's (1981) intensely studied functional-imperative language *Idealised Algol* (IA)—O'Hearn and Tennent (1997) collect these and other key papers on the semantics of this language.

The relation between PCF and IA is a very interesting one. Syntactically and operationally IA is a superset of PCF, to which it adds *local state*. Despite this close connection, denotational models of IA differed significantly from those of PCF in terms of their mathematical structure. It was considered essential that the structure of the semantic domain mimics the structure of the store, something that was postulated by Reynolds (1981) as one of the basic principles of the language: "*5. The language should obey a stack discipline, and its definition should make this discipline obvious.*" Here 'obvious' means that it should be part of the domain equations. This imperative led (Oles, 1983) to formulate an influential model based on functor categories. Tennent and Ghica (2000) give a survey of the evolution of IA models.

However, since IA lives inside PCF it was likely that the full abstraction problems for the two are connected. And, indeed, the fully abstract model of IA followed shortly that of PCF. Even though the IA model is less celebrated than that for PCF, which solved a long-standing open problem of high stature, two of its features foreshadowed the coming success and dominance of the game-semantic methodology.

The first achievement of the IA model was rather technical. Part of the methodology of denotational semantics mandates that *equivalent* syntactic phrases are mapped into *equal* mathematical objects. This was achieved by Milner (1977) using so-called *term-model constructions*, starting from the syntax and applying quotients. But such models do not make semantic reasoning any easier. They are a form of sweeping under the rug. Game models for PCF are not syntactic, but they use a form of quotienting which was found by some to be objectionable in the same way as term models. The subtle relationship between definablity and full abstraction and a critique of game models in this context is discussed by Curien (2007).

Moreover, as Loader (2001) showed soon after, term equivalence for finitary PCF is not decidable, so it was unlikely that the semantic domain of PCF was going to consist of neat mathematical objects. The IA game model put this debate to rest by providing a language interpretation in which no quotienting is required and thus eliminating a significant, if somewhat obscure, objection to games-based models.

The second achievement of the IA model was more subtle but at the same time more consequential. The model of IA was, in some sense, as close to the model of PCF as the syntax of IA is close to that of PCF. Both are interpreted in, essentially, the same semantic domain and the difference is a mere tweak. Even though the IA game model was foreshadowed by some earlier models, such as the object model of Reddy (1996), the similarity between it and that of PCF was striking, and it suggested that small tweaks to the game model can lead to models for diverse languages, starting from a common fundamental game model. The final paragraph of the paper (the version which appeared as a part of O'Hearn and Tennent (1997)) states that:

> Another point for further investigation is suggested by the following diagram:
>
> $$\begin{tikzcd} & \neg I \wedge \neg B & \\ I \wedge \neg B \arrow[ur] & & \neg I \wedge B \arrow[ul] \\ & I \wedge B \arrow[ul] \arrow[ur] & \end{tikzcd}$$
>
> Here I denotes innocence and B the bracketing condition [...] and very successfully capture pure functional programming. As we have seen in the present paper, the category of knowing (but well bracketed) strategies captures IA. If we conversely retain innocence but weaken the bracketing condition then we get a model of PCF extended with non-local control operators.

> *Thus we begin to develop a semantic taxonomy of constraints on strategies mirroring the presence or absence of various kinds of computational features.*

The '*innocence*' and '*bracketing*' conditions mentioned above are the relatively small adjustments that the PCF game model requires in order to lead to full abstraction for other languages. The lattice of conditions above subsequently received new dimensions, and was dubbed a *cube* by Abramsky and McCusker (1999), which was then commonly referred to as '*Abramsky's Cube*'. Exploring the various vertices of this (hyper)cube led to the development of many interesting and useful semantic models. Even though other methods such as *trace semantics* also led to the development of fully abstract models for non-trivial languages (Jeffery & Rathke, 2005) it is fair to say that ultimately game semantics became the dominant paradigm, thanks in no small part to the guidance and inspiration provided by the Abramsky Cube.

Beyond the Cube, Beyond Definability

The methodology of game semantics was naturally guided by its history, energised by the quest for PCF definability. This meant that the first game semantic model, that of PCF, was also in some sense the most highly constrained game semantic model. Other models are then derived by relaxing some of the constraints. This is perhaps paradoxical: Why does the simplest language (PCF) have the most complicated model? This is, again, because of definability. In a simple language relatively few semantic objects are definable. The constraints on the model are intended to rule out certain objects by deeming them to be '*illegal*'. It is the remaining, legal, ones which are syntactically definable. As the language becomes richer, some of these semantic constraints can be relaxed. But this should lead to an obvious question: What if we relax all the constraints? Or, rather, what if we relax all the constraints that we can relax without making the model fall apart? What model lies at the top of the cube (or lattice, rather) of constraints. This is what the current essay will attempt to answer.

Since the game model on display here is simple, we aim for this to be a self-contained, accessible, and elementary introduction to game semantics. This presentation will be done in the style of Gabbay and Ghica (2012) which will allow us to streamline some of the basic proofs of properties of game semantics. Arguably, the model we present here can be seen as an ur-model, at least for *call-by-name* programming languages. Understanding it should give an easier access ramp to the rich, diverse, and mathematically sophisticated world of game semantics.

A final caveat: for pedagogical reasons this paper is trying to tell a story. Like any story, it will sometimes omit certain technicalities for the sake of painting a coherent picture. The picture is, hopefully, more compelling even if it is not complete. Even though the model we are constructing, and the syntax we are recovering from it, are indeed 'extremal' it does not mean that all programming languages—not even all call-by-name programming languages—live in this model. The greatest missing beast from our zoo garden is *control.* There are many varieties of control operators, studied in the context of game semantics starting with Laird (1997). They all require different tweaks of the underlying game model, which we do not address.

6.2 Game Semantics, an Interaction Semantics

6.2.1 Arenas, Plays, Strategies

The terminology of game semantics guides the intuition towards the realm of game theory. Indeed, there are methodological and especially historical connections between game semantics and game theory, but they are neither direct nor immediate. The games involved are rooted in logic and reach programming languages via the Curry-Howard nexus. They are not essential for a working understanding of game semantics as a model of programming languages, so we will not describe them here. But if we were to be pushed hard to give a game-theoretic analogy, the ones to keep in mind are not the quantitative games of economics but rather last-player-wins games such as Nim.

It is more helpful to think of game semantics as an interaction, or dialogue, between two agents, rather than a game. The dialogue is between a *term* t, i.e. a piece of programming language code, and its *context* $\mathcal{C}[-]$, i.e. the rest of the code. By placing the term in context we create an executable program $\mathcal{C}[t]$. During execution, certain interactions such as function calls and returns, or variable access, will happen. These are the interactions that are organised into a game model.

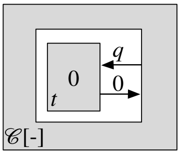
This interaction is asymmetric. One agent (P) represents the term and the other (O) represents an abstract and general context in which the term can operate.[1] The interaction consists of sequences of events called *moves*, which can be seen as either calls, called *questions*, or returns, called *answers*. A sequence of events, with some extra structure to be discussed later, is called a *play* and it corresponds to the sequence of interactions between the term and the context in one given program run. The set of all such possible plays, for all possible contexts, is called a *strategy* and it gives the *interpretation* of the term. The strategy of a term can be constructed inductively on its syntax, from basic strategies for the atomic elements and a suitable notion of composition to be discussed later.

Before we proceed, a caveat. The structure of a game semantics is dictated by the evaluation strategy of the language and its type structure. Call-by-name games are quite differently structured than call-by-value games. Hereby we shall assume a call-by-name evaluation strategy and simple type discipline of base types and functions. The reason is didactic, as these games are easier to present. Having understood game

[1] The names stand for '*Proponent*' and '*Opponent*' even though there is nothing being proposed, and there is no opposition to it. The names are historical artefacts. We might as well call them '*Popeye*' and '*Olive*'. Same applies to 'move', 'play', and 'strategy'.

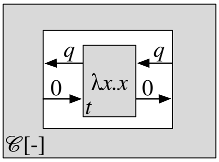
semantics in this simple setting, understanding other more complex setups should be easier.

Let us consider a most trivial example, the term consisting of the constant 0. The way this term can interact with any context is via two moves: a question (q) corresponding to the context interrogating the term, and an answer (0) corresponding to the term communicating its value to the context. The sequence $q \cdot 0$ is the only possible play, therefore the strategy corresponding to the set of plays $\{q \cdot 0\}$ is the interpretation of the term 0.



This behaviour is 'at the interface' in the sense that any other term evaluating to 0, such as $7 - 7$ or $4 \times 2 - 8$ would exhibit the same interaction.

Let us consider a slightly less trivial example, the identity function over natural numbers $\lambda x.x : nat \to nat$. The context can call this function, but also the function will enquire about the value of its argument x. Lets call these questions q and q'. The context can answer to q' with some value n and the term will answer to q with the same value n. Even though the answers carry the same value they are different moves, so we will write n and n' to distinguish them, where the prime is a syntactic tag. Plays in the strategy interpreting the identity over natural numbers have shape $q \cdot q' \cdot n' \cdot n$. Equivalent terms such as $(\lambda x.x)(\lambda x.x)$ exhibit identical interactions.



Let us now define the concepts more rigorously.

Definition 1 (*Arena*) An *arena* is a tuple $\langle M, Q, I, O, \vdash \rangle$ where

- M is a set of *moves*.
- $Q \subseteq M$ is a set of *questions*; $A = M \setminus Q$ is the set of *answers*.
- $O \subseteq M$ is a set of *O-moves*; $P = M \setminus O$ is the set of *P-moves*.
- $I \subseteq Q \cap O$ is a set of *initial moves*;
- $\vdash \subseteq M \times M$ is an *enabling* relation such that if $m \vdash n$ then

  (e1) $m \in Q$
  (e2) $m \in O$ if and only if $n \in P$
  (e3) $n \notin I$.

An arena represents the set of moves associated with a type, along with the structure discussed above (questions, answers, O, P). Additionally, the arena introduces the concept of *enabling* relation, which records the fact that certain moves are causally related to other moves. Enabling requires certain preliminary conditions:

(e1) Only questions can enable other moves, which could be interpreted by the slogan '*all computations happen because of a function call*'.
(e2) P-moves enable O-moves and *vice versa*. Game semantics records behaviour *at the interface* so any action from the context enables an action of the term, and the other way around.
(e3) There is a special class of O-questions called *initial moves*. These are the moves that are allowed to kick off an interaction, so do not need to be enabled.

The informal discussion above can be made more rigorous now.

Example 2 Let $\mathbf{1} = \{\star\}$. The arena of natural numbers is $N = \langle \mathbf{1} \uplus \mathbb{N}, \mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{1} \times \mathbb{N}\rangle$.

More complex arenas can be created using product $\times$ and arrow $\Rightarrow$ constructs. Let

$$inl : M_A \to M_A + M_B$$
$$inr : M_B \to M_A + M_B$$

where $+$ is the co-product of the two sets of moves. We lift the notation to relations, $R + R' \subseteq (A + A') \times (B + B')$:

$$inl(R) = \{(inl(m), inl(n)) \mid (m, n) \in R\}$$
$$inr(R') = \{(inr(m), inr(n)) \mid (m, n) \in R'\}.$$

Definition 3 (*Arena product and arrow*) Given arenas $A = \langle M_A, Q_A, O_A, I_A, \vdash_A\rangle$ and $B = \langle M_B, Q_B, O_B, I_B, \vdash_B\rangle$ we construct the *product arena* as

$$A \times B = \big\langle M_A + M_B, Q_A + Q_B, O_A + O_B, I_A + I_B, \vdash_A + \vdash_B\big\rangle$$

and the *arrow arena* as

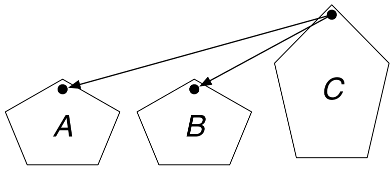
$$A \Rightarrow B = \big\langle M_A + M_B, Q_A + Q_B, P_A + O_B, inr(I_B), \vdash_A + \vdash_B \cup\, inr(I_B) \times inl(I_A)\big\rangle.$$

If we visualise the two arenas as DAGs, with the initial moves as sources and with the enabling relation defining the edges, then the product arena is the disjoint union of the two DAGs and the arrow arena is the grafting of the A arena at the roots of the B arena, but with the O-P polarities reversed.

Since arenas will be used to interpret types we can anticipate by noting that

Proposition 4 (Currying) *For any arenas A, B, C the arenas $A \times B \Rightarrow C$ and $A \Rightarrow B \Rightarrow C$ are isomorphic.*

Proof Both arena constructions correspond to the DAG below.



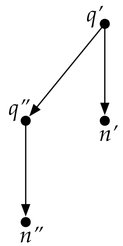
The isomorphism is a node-relabeling isomorphism induced by the associativity isomorphism of the co-product. □

We also note that

Proposition 5 (Unit) *The arena $I = \langle \emptyset, \emptyset, \emptyset, \emptyset \rangle$ is a unit for product, i.e. for any arena A, $A \times I$, $I \times A$, $I \Rightarrow A$ are isomorphic to A.*

The isomorphism is a re-tagging of moves.

Example 6 We talked earlier about the arena for the type $nat \to nat$. Let $inl(m) = m''$ and $inr(m) = m'$, where $'$ and $''$ are syntactic tags. The arena $Nat \Rightarrow Nat$ is represented by the DAG below



We already mentioned that for the identity all plays have the shape $q'q''n''n'$. We note that in this particular play all move occurrences are preceded by an enabling move. The move corresponding to the term returning a value n' can happen because the context initiated a play q'. The term can ask for the value of the argument q'' also because q has happened earlier. Each move occurrence is *justified* by an enabling move, according to $\vdash$, occurring earlier. The enabling relation defines the causal skeleton of the play.

Let us further consider another term in the same arena $\lambda x.x + x$. How does this term interact with its context?

1. the context initiates the computation
2. the term asks for the value of x
3. the contest returns some m
4. the term asks again for the value of x
5. the context returns some n
6. the term returns to the context $m + n$.

The reader familiar with call-by-value may be rather confused as to why the context returns first an m and then an n. This is because in call-by-name arguments are thunks. In some languages the thunks may contain side-effects, which means that repeat evaluations may yield different values.

Looking at the arena, this interaction corresponds to the play $q'q''m''q''n''p'$, where $p = m + n$. The causal structure of this play is a little confusing. There are two occurrences of q'', the first one preceding both m'' and n'' and the second one only n''. It should be that the first occurrence of q'' enables m'' and the second enables n'', to reflect the proper call-and-return we might expect in a programming language. In order to do that the plays will be further instrumented with names called *pointers*. Each question has a symbolic address, and is paired with the address of some other move, called *enabling move*. The fully instrumented play is called a *pointer sequence*.

Let us use ϵ for the empty sequence, $\cdot$ for sequence concatenation, $\sqsubseteq$ for sequence prefix ($u \sqsubseteq u \cdot v$) and $\Subset$ for the sub-sequence relation ($u \Subset v \cdot u \cdot w$). If unambiguous we may represent concatenation simply as juxtaposition (uv). Let $\mathbb{A}$ be a set of '*names*'.

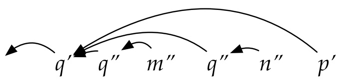
Definition 7 (*Pointer sequence*) Given a set M, a *pointer sequence* p is a sequence $p \in (M \times \mathbb{A} \times \mathbb{A})^*$ such that for all $q \cdot (m, a, b) \sqsubseteq p$, for all $(m', a', b') \Subset q$, $b \neq a'$ and $b \neq b'$.

We call J_M the set of pointer sequences over M. We write the triples (m, a, b) as $ma\langle b\rangle$. The addresses $a, b, c, \ldots \in \mathbb{A}$ are just names, and the notation $\langle b\rangle$ means that the name b is '*fresh*', i.e. not used earlier in the sequence. We sometimes write this condition as $b \# q$. In general we will employ the Barendregt name convention that if two names a, b are denoted by distinct variables they are distinct $a \neq b$. Answers never justify (in these games) so we may write their unused name as _ or we may omit the whole $\langle \_ \rangle$ component altogether. In a pointer sequence, by a *move occurrence* we mean the move along with the justifier a and, if it is the case, the pointer name $\langle b\rangle$, taken as a whole. We write J_A for the set of pointer sequences over the moves of an arena A.

Going back to the earlier example, the interaction corresponds to the following pointer sequence:

$$q'a\langle b\rangle \cdot q''b\langle c\rangle \cdot m''c \cdot q''b\langle d\rangle \cdot n''d \cdot p'b,$$

noting that a is the only name without a previous binder, and is used by the initial question. If we were to represent the pointers graphically, the sequence above would be:



In general, if we draw a pointer sequence we will omit the 'dangling' pointers from the diagram.

The *pointer sequence* represents not only the actions, that is the calls and returns, but also what calls correspond to what returns and even what calls are caused by other calls. In a sequential language, for terms up to order three (in the type hierarchy counting the depth of nesting of exponentials) the pointers can be actually uniquely reconstructed from the sequence itself. Otherwise the justification pointers are necessary.

Definition 8 (*Play*) A pointer sequence $p \in J_A$ is said to be a *play* when:

- for any $p' \cdot ma\langle c\rangle \sqsubseteq p$, $p' \neq \epsilon$, there exists $q \in Q_A, b \in \mathbb{A}$ such that $qb\langle a\rangle \sqsubseteq p'$ and $q \vdash_A m$.
- if $qa\langle b\rangle \sqsubseteq p$ then $q \in I_A$.

Above, it is implicit that $m \in M_A$, and $a, b, c \in \mathbb{A}$.

We write the set of plays of an arena A as $\mathbf{P}_A$. They represent computations which are *causally sensible*, so that pointers are consistent with the enabling relation. The behaviour of a function that would return or ask for its argument without being itself called is, for example, not causally sensible and its corresponding interactions are thus not plays.

Let $\pi : \mathbb{A} \to \mathbb{A}$ be bijections representing *name permutations*, and define *renaming actions* of a name permutation on a pointer sequence over arena A as

$$\pi \bullet \epsilon = \epsilon, \quad \pi \bullet (p \cdot ma\langle b\rangle) = (\pi \bullet p) \cdot (m\, \pi(a)\, \langle \pi(b)\rangle)$$

Proposition 9 *If $p \in \mathbf{P}_A$ then $\pi \bullet p \in \mathbf{P}_A$, for any bijection $\pi : \mathbb{A} \to \mathbb{A}$.*

The proofs are elementary. If they are also straightforward we will leave them as an exercise.

A *strategy* in an arena A is any set of plays which is prefix closed, closed under choices of pointer names (*equivariant*), and closed over O-moves.

Definition 10 (*Strategy*) A strategy over an arena $\sigma : A$ is a set of plays such that for any $p \in \sigma$ the following properties hold:

prefix-closed If $p' \sqsubseteq p$ then $p' \in \sigma$
O-closed If $p \cdot m \in \mathbf{P}_A$ for some $m \in O_A$ then $p \cdot m \in \sigma$
equivariance For any permutation π, $\pi \bullet p \in \sigma$.

The conditions above have intuitive explanations. Prefix-closure is a natural condition on trace semantics, going back to Hoare (1978) model of CSP and beyond, to common encodings of trees in set theory. It has a clear causal motivation in the sense that a trace semantics is a history of behaviour, and any history must be prefix-closed.

The O-closure condition reflects the fact that a term has no control over which one of a range of possible next moves the context might choose to play. Finally, pointer equivariance is akin to an alpha-equivalence on plays, motivated by the fact that pointer names are not observable, so the choice of particular names in a play is immaterial. Name, equivariance and related concepts and reasoning principles are comprehensively studied by Pitts (2013).

In this most general setting names introduced by moves are required to be fresh. In more constrained settings it can be determined that a name is no longer to be used, because any use of that name would violate the constraints. In fact most game models in the literature have this property, with the model presented here being an exception. If a name is no longer usable than it is possible to introduce a notion of '*scope*' for that name, raising the possibility of name reuse. Gabbay et al. (2015) give a formulation of game models where pointer names are scoped.

Note that it is equivalently possible to present strategies as a next-move function from a play to a P-move (or set of P-moves) along with the justification infrastructure, indicating the next move in the play. If $\mathcal{P}$ is the power-set, then

$$\hat{\sigma} : \mathbf{P}_A \to \mathcal{P}(P_A \times \mathbb{A}^2),$$

such that for all $p \in \mathbf{P}_A$, and $(m, a, b) \in \hat{\sigma}(p)$, $p \cdot ma\langle b\rangle \in \mathbf{P}_A$.

Proposition 11 *Let* $\hat{\sigma} : \mathbf{P}_A \to \mathcal{P}(P_A \times \mathbb{A}^2)$ *be a next-move function, and let* σ *be the smallest set such that*

Empty: $\epsilon \in \sigma$
P-move: *if* $p \in \sigma$ *and* $ma\langle b\rangle \in \hat{\sigma}(p)$ *then* $p \cdot ma\langle b\rangle \in \sigma$
O-move: *if* $p \in \sigma$ *and* $p \cdot ma\langle b\rangle \in \mathbf{P}_A^O$ *then* $p \cdot ma\langle b\rangle \in \sigma$.

The set σ *is a strategy.*

Proof The set σ is prefix-closed by construction. Adding all O-moves wherever legal ensures O-completeness. Equivariance holds from general principles. □

When we specify a next-move function, if the result is a singleton set $\{ma\langle b\rangle\}$ we may simply write $ma\langle b\rangle$, and if $\hat{\sigma}(p) = \emptyset$ we may omit that case from the definition.

We will sometimes define a strategy directly and some other times via the next-move function, whichever is more convenient.

Definition 12 Given a set of plays over an arena A, $\sigma \subseteq P_A$, let us write $\mathrm{strat}(\sigma)$ for the least strategy including σ.

We will sometimes abuse the notation above by applying it to a set of sequences of moves, in the case that the pointer structure can be unambiguously reconstructed.

Example 13 In arena *Nat*,

$$\sigma_0 = \mathrm{strat}(q \cdot 0) = \mathrm{strat}(qa\langle b\rangle \cdot 0b) = \{\epsilon, qa\langle b\rangle, qa\langle b\rangle \cdot 0b \mid a, b \in \mathbb{A}\}.$$

The next-move function is

$$\hat{\sigma}_0(qa\langle b\rangle) = 0a.$$

6.2.2 Examples of Strategies

If we consider programming languages in the style of Landin (1966), i.e. the simply-typed lambda calculus with additional operations, such additional operations can be defined in the unrestricted game model. Let us consider several examples.

6.2.2.1 Arithmetic

Any arithmetic operator $\circledast : nat \to nat \to nat$ is interpreted by a strategy over the arena $Nat \Rightarrow Nat \Rightarrow Nat$. Let us tag the moves of the first Nat arena with $-_1$, the moves of the second with $-_2$ and leave the third un-tagged (the trivial tag). Then the interpretation of the operator is in most cases

$$\sigma_{\circledast} = \text{strat}\,(\{qq_1m_1q_2n_2p \mid m, n, p \in \mathbb{N} \wedge p = m \circledast n\})$$

Note that in the case of division, or any other operation with undefined values, the strategy must include those cases explicitly:

$$\sigma_{\div} = \text{strat}\,(\{qq_1m_1q_2n_np \mid m, n \neq 0, p \in \mathbb{N} \wedge p = m \div n\} \cup \{qq_1m_1q_20_2 \mid m \in \mathbb{N})$$

Following the 0 O-answer, there is no way P can continue, i.e. $\hat{\sigma}_{\div}(qq_1m_1q_20_2) = \emptyset$.

From this point of view sequencing can be seen as a degenerate operator which evaluates then forgets the first argument, then evaluates and return the second

$$\sigma_{seq} = \text{strat}\,(\{qq_1m_1q_2n_2n \mid m, n \in \mathbb{N}\})$$

Of course, sequencing commonly involves commands com which are degenerate, single-value, data types which are constructed just like the natural numbers but using a singleton set instead of $\mathbb{N}$.

The flexibility of the strategic approach also gives an easy interpretation to shortcut (lazy) arithmetic operations:

$$\sigma_{\times} = \text{strat}\,(\{qq_1m_1q_2n_2p \mid m \neq 0, n, p \in \mathbb{N} \wedge p = m \times n\} \cup \{qq_10_10\})$$

This comes in handy when implementing an if-then-else operator (over natural numbers), in arena $Bool \Rightarrow Nat \Rightarrow Nat \Rightarrow Nat$, where $Bool$ is the arena of Booleans, with $M_{Bool} = \mathbb{B} = \{tt, f\!f\}$:

$$\sigma_{if} = \text{strat}\left(\{qq_1tt_1q_2n_2n \mid n \in \mathbb{N}\} \cup \{qq_1f\!f_1q_3n_3n \mid n \in \mathbb{N}\}\right)$$

These strategies are found in the original PCF model of Hyland and Ong (2000).

6.2.2.2 Non-determinism

A non-deterministic Boolean choice operator $chooseb : \mathbb{B}$ is interpreted by the strategy

$$\sigma_{chooseb} = \mathrm{strat}(\{q \cdot tt, q \cdot \mathit{ff}\})$$

where two P-answers are allowed. This can be extended to probabilistic choice by adding a probability distribution over the strategy.

Note that the flexibility of the strategic approach can allow the definition of computationally problematic operations such as unbounded non-determinism $choosen : \mathbb{N}$,

$$\sigma_{choosen} = \mathrm{strat}(\{qn \mid n \in \mathbb{N}\})$$

Nondeterministic and probabilistic game semantics have been studied by Harmer and McCusker (1999) and Danos and Harmer (2000), respectively. In terms of the Abramsky Cube, these games lead to definability via the relaxation of the determinism condition, which means that the strategy function can result in more than one possible move. By contrast, deterministic language strategies respond with at most one move for any given play.

6.2.2.3 State

In order to model state we first need to find an appropriate arena to model assignable variables. In the context of call-by-name it is particularly easy to model *local* (bloc) variables ($new\ x\ in\ t$, where x is the variable name and t the term representing the variable block). It turns out that new need not be a term-former but it can be simply a higher order language constant $new : (var \to T) \to T$ where T is some language ground type.

The type of variables var can be deconstructed following an 'object oriented' approach initially proposed by Reynolds (1981). A variable must be readable $der : var \to nat$ and assignable $asg : var \to nat \to nat$. Since no other operations are applicable, we can simply define $var = nat \times (nat \to nat)$ which means $der = proj_1$, $asg = proj_2$, so assignment behaves like in C, returning the assigned value. We will see later how projections are uniformly interpreted by strategies.

What is interesting is the interpretation of the new operation in arena $(Var \Rightarrow Unit) \Rightarrow Unit'$. With the decomposition above in mind and for the sake of clarity we call the moves in Var as follows:

Read request : rd
Value read : $val(n)$ for $n \in Nat$
Write request : $wr(n)$ for $n \in Nat$
Value written : $ok(n)$ for $n \in Nat$.

We define this strategy using the next-move function. The strategy will include copy-cat moves between $Unit$ and $Unit'$ along with stateful moves.

If a read (rd) move is played by O then P will respond with $val(n)$ where n is the last move it played before, which can be a read value $val(n)$ or a written acknowledgment $ok(n)$.

$$\hat{\sigma}_{new}(p \cdot m(n)a \cdot rd\, b\langle c\rangle) = val(n)c, \quad m \in \{val, ok\}.$$

If a write $wr(n)$ move is played by O then P will always acknowledge it with $ok(n)$.

$$\hat{\sigma}_{new}(p \cdot wr\, a\langle b\rangle) = val(n)b$$

These strategies were introduced by Abramsky and McCusker (1996) in the game model of IA. On the Abramsky Cube these strategies relax the '*innocence*' condition of the strategy function which states that for any play p there is a (usually smaller) play computed from it, called '*the view*' $\lceil p \rceil$ such that $\hat{\sigma}(p) = \hat{\sigma}(\lceil p \rceil)$. In other words the term has a '*restricted memory*' of the play in choosing the next move. This subtle condition has been studied extensively by Danos and Harmer (2001); Harmer and Laurent (2006); Harmer et al. (2007).

6.2.2.4 Control

If-then-else is a very simple control operator, but more complex ones can be defined. The family of control operators is large, so let us look at a simple one, $catch$: $(com_1 \to nat_2) \to option\ nat$ where the type $option\ nat$ is interpreted in an arena constructed just like *Nat* but using $\mathbb{N} + 1$ instead of $\mathbb{N}$. The extra value indicates an error result (ϕ). Just like in the case of state, the construct $catch(\lambda x.t)$ can be sugared as $escape\ x\ in\ t$. If x is used in t then the enclosing $catch$ returns immediately with ϕ, otherwise it returns whatever t returns.

The strategy is

$$\sigma_{catch} = \text{strat}\,(\{qq_2n_2n \mid n \in \mathbb{N}\}) \cup \{qq_2q_1\phi\})$$

Game semantics for languages with control have been initially studied by Laird (1997). In the Cube, these strategies relax the '*bracketing*' constraint of the PCF model, which requires questions and answers to nest like well-matched brackets.

6.2.2.5 Concurrency

As our final example we will consider *parallel composition* of commands, par : $com_1 \to com_2 \to com$. In the case of this strategy, which represents a function which executes its arguments *asynchronously* all interleavings of the two argument executions are acceptable:

$$\sigma_{par} = \text{strat}\,(q \cdot (q_1 a_1 \mid q_2 a_2) \cdot a)\,,$$

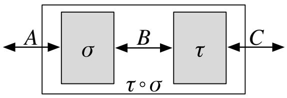
where $p \mid q$ is the set of all interleavings of two sequences.

The constraint relaxed by this strategy is the *alternation* of O/P moves, and was first studied by Laird (2001) in the context of game semantics for process calculi and by Ghica and Murawski (2008) in the context of game semantics for programming languages.

6.2.3 Composing Strategies

In the previous section we looked at strategies interpreting selected language constants. In order to construct an interpretation of terms, denotationally, strategies need to *compose*.

The intuition of composing a strategy $\sigma : A \Rightarrow B$ with a strategy $\tau : B \Rightarrow C$ is to use arena B as an interface on which in a first instance σ and τ will synchronise their moves.



After that, the moves of B will be hidden, resulting in a strategy $\sigma;\tau : A \Rightarrow C$. In order to preserve proper justification of plays all pointers that factor through hidden moves will be 'extended' so that the hiding of the move will not leave them dangling. In order to define composition some ancillary operations are required.

The first one is deleting moves while extending the justification pointers over the deleted moves, in order to preserve justification (Gabbay & Ghica, 2012).

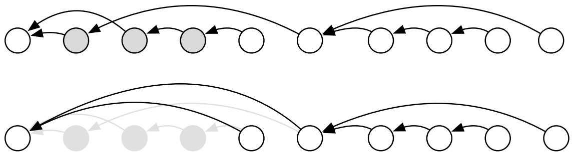
Definition 14 (*Deletion*) Let $X \subseteq M$ be sets. For a pointer sequence $p \in J_M$ we define *deletion* inductively as follows, where we take $(p', \pi) = p \downharpoonleft X$:

$$\begin{aligned}
\epsilon \downharpoonleft X &= (\epsilon, id) & \\
(p \cdot ma\langle b\rangle) \downharpoonleft X &= (p' \cdot m \cdot \pi(a)\langle b\rangle, \pi) & \text{if } m \notin X \\
(p \cdot ma\langle b\rangle) \downharpoonleft X &= (p', (\pi \mid b \mapsto \pi(a))) & \text{if } m \in X
\end{aligned}$$

The result of a deletion is a pointer sequence along with a function $\pi : \mathbb{A} \to \mathbb{A}$ which represents the chain of pointers associated with deleted moves. Informally, by $p \downharpoonleft X$ we will understand the first projection applied to the resulting pair. Since deletion only removes names, it is immediate that $p \downharpoonleft X$ is a well-formed pointer sequence.

Proposition 15 *If $X \subseteq M$ and $p \in J_M$ then $p \downharpoonleft X \in J_{M \setminus X}$.*

For example, the removal of the grayed-out moves in the diagrammatic representation of the play below results in a sequence with reassigned pointers:



Note that in general the deletion of an arbitrary set of moves from a play does not result in another play, since the pointers may be reassigned in a way that is not consistent with the enabling relation. There is however an important special situation:

Proposition 16 *Given arenas A, B if $p \in \mathbf{P}_{A \Rightarrow B}$ then $p \downharpoonright A \in \mathbf{P}_B$.*

Proof Since the enabling relation is a DAG, no A-move enables any B-move. Thus, in a play p there can be no pointer from an A-move occurrence to a B-move occurrence. Therefore when the A-move occurrences are deleted no pointer reassignment is required, so the result is a B-play. □

The second operation is the selection of 'hereditary' sub-plays, i.e. all the moves that can be reached from an initial set of moves following the justification pointers (Gabbay & Ghica, 2012).

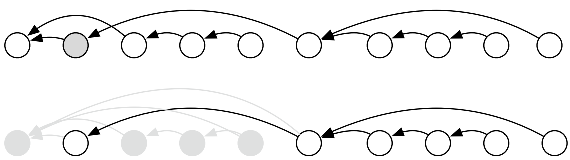
Definition 17 (*Hereditary justification*) Let $X \subseteq M$ be sets. For a pointer sequence $p \in J_M$ we define the *hereditarily justified* sequence $p \upharpoonright X$ recursively as below, where we take $p \upharpoonright X = (p', X')$:

$$
\begin{aligned}
\epsilon \upharpoonright X &= (\epsilon, X) && \\
(p \cdot ma\langle b\rangle) \upharpoonright X &= (p' \cdot ma\langle b\rangle, X \cup \{b\}) && \text{if } a \in X \\
(p \cdot ma\langle b\rangle) \upharpoonright X &= (p', X) && \text{if } a \notin X
\end{aligned}
$$

The result of a hereditary justification is a pointer sequence along with a set of names $X \subseteq \mathbb{A}$ which represents the addresses of selected questions. Informally, by $p \upharpoonright X$ we will understand the first projection applied to the resulting pair. Since hereditary justification only removes names, it is immediate that $p \upharpoonright X$ is a well-formed pointer sequence.

Proposition 18 *If $X \subseteq M$ and $p \in J_M$ then $p \upharpoonright X \in J_X$.*

For example, the hereditary justification of the grey move in the diagrammatic representation of the play below results in the sequence below:



In general the hereditary justification of an arbitrary set of moves from a play does not result in another play, since it may result in a sequence that does not start with an initial move. The special situation is:

Proposition 19 *Given arenas A, B if $p \in \mathbf{P}_{A \Rightarrow B}$, with $ma\langle b\rangle \sqsubseteq p$, $m \in I_A$ then $p \upharpoonright \{b\} \in \mathbf{P}_A$.*

Proof Since the enabling relation is a DAG, no A-move enables any B-move. In a play p thus there can be no pointer from an A-move occurrence to a B-move occurrence. Therefore when the hereditarily justified sequence from an A-initial move is selected it will result in an A-play. □

Propositions 16 and 19 are technically important, and they are consistent with the intuitive model of computation we relied on. What they say is that in a play corresponding to a function of type $A \Rightarrow B$, the sequences associated just with the argument A or just with the result B are also plays. In other words, when the behaviour of a function is causally sensible, both the arguments and the body of the function are going to be causally sensible. In both cases the DAG structure of the arena is essential, since it allows no enabling from the argument back to the function body.

We now have the requisite operations to define the *interaction* and, finally, the *composition* of strategies.

Definition 20 (*Interaction*) Given sets of pointer sequences $\sigma \subseteq J_M, \tau \subseteq J_N$ their *interaction* is defined as

$$\sigma \mathop{||}_{M,N} \tau = \{p \in J_{M \cup N} \mid p \downharpoonright (M \setminus N) \in \tau \wedge p \downharpoonright (N \setminus M) \in \sigma\}.$$

A good intuition for interaction is of two strategies synchronising their actions on the shared moves $M \cap N$.

Observation. This definition will be used to compute the interaction of strategies $\sigma : A \Rightarrow B$ and $\tau : B \Rightarrow C$, but we will ignore the issue of tagging of moves as they participate in the definition of composite arenas, and we will just assume the underlying sets of moves are disjoint. This is not technically correct because the arenas can be equal, case in which the tagging is essential to disambiguate the co-products. In which case the formalisation of tagging, de-tagging and re-tagging is routine and tedious and may obscure the main points. We are sacrificing some formality for clarity.

Example 21 Let $\tau = \mathrm{strat}(qq'm'(m+1))$ denote a function that increments its argument and $\sigma = \mathrm{strat}(q'q''n''(2n)')$ a function that doubles its argument, $\sigma, \tau : Nat \Rightarrow Nat$. Their interaction $\sigma \,||\, \tau$ is a set of sequences of the form $qq'q''m''(2m)'(2m+1)$. As written below, the flow of time is top to bottom and each is lined up with the arena in which it occurs.

$$
\begin{array}{ccc}
Nat'' \overset{\sigma}{\Rightarrow} & Nat' \overset{\tau}{\Rightarrow} & Nat \\
\hline
 & & q \\
 & q & \\
q & & \\
m & & \\
 & 2m & \\
 & & 2m+1
\end{array}
$$

Example 22 As defined, a strategy can interact with another strategy only once. Let $\tau = \mathrm{strat}(qq'm'q'n'(m+n))$ denote a function that evaluates its argument twice and returns the sum of received values, and let $\sigma = \mathrm{strat}(q'0')$ be the strategy for constant 0. The interaction $\sigma \mid\mid \tau$ cannot proceed successfully because removing the untagged moves representing the result of τ leaves sequences of the shape $q'm'q'n'$ which are not in σ no matter what values m, n take.

$$
\begin{array}{ccc}
I \overset{\sigma}{\Rightarrow} & Nat' \overset{\tau}{\Rightarrow} & Nat \\
\hline
 & & q \\
 & q & \\
 & 0 & \\
 & ? &
\end{array}
$$

Definition 23 (*Iteration*) Given a set of pointer sequences $\sigma \in J_M$ its iteration on $N \subseteq M$ is the set of pointer sequences

$$!_N\sigma = \{p \in J_M \mid \forall ma\langle b\rangle \sqsubseteq p.m \in N \Rightarrow p \restriction \{b\} \in \sigma\}$$

A good intuition of iteration is a strategy interleaving its plays. The definition says that if we select moves from an identified subset N and we trace the hereditarily justified plays, they are all in the original set. We can think of each $p \restriction \{b\}$ as untangling the 'thread of computation' associated with move $ma\langle b\rangle$ from the interleaved sequence.

Example 24 Using interaction with the iterated strategy for 0 in Example 22 is now possible, so $!\sigma \mid\mid \tau = qq'0'q'0'0$.

$$
\begin{array}{ccc}
I \overset{\sigma}{\Rightarrow} & Nat' \overset{\tau}{\Rightarrow} & Nat \\
\hline
 & & q \\
 & q & \\
 & 0 & \\
 & q & \\
 & 0 & \\
 & & 0
\end{array}
$$

Note that for iteration to have the desired effect it is essential that strategies are equivariant. Consider the situation if the strategy were non-equivariant.

$$\sigma_0 = \{\epsilon, qa\langle b\rangle, qa\langle b\rangle \cdot 0b\}$$

where $a, b \in \mathbb{A}$ are fixed. Iterated pointer sequences such as $qa\langle b\rangle \cdot 0b \cdot qa\langle b\rangle \cdot 0b$ are not well formed because the second occurrence of $\langle b\rangle$ is no longer fresh. However, because the strategy is equivariant we can choose other names when iterating, so that $qa\langle b\rangle \cdot 0b \cdot qc\langle d\rangle \cdot 0d$ is legal.

Composition is iterated interaction with the synchronisation moves internalised and hidden.

Definition 25 (*Composition*) Given strategies $\sigma : A \Rightarrow B, \tau : B \Rightarrow C$ we defined their *composition* as $\sigma; \tau = (!_{I_B}\sigma \underset{M_{A\Rightarrow B}, M_{B\Rightarrow C}}{||} \tau) \downharpoonleft M_B$.

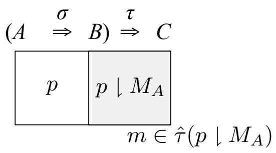
We also use $\tau \circ \sigma \stackrel{def}{=} \sigma; \tau$.

The definition above is the usual *extensional* presentation of strategy composition, which has the slight disadvantage of eliding some of the computational and operational flavour of the games-based approach. An equivalent *intensional* definition can be given using the strategy functions to compute the next move.

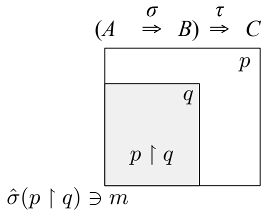
Definition 26 Given strategies $\sigma : A \Rightarrow B, \tau : B \Rightarrow C$ we define their *interaction function* as $\widehat{\sigma \lightning \tau} : \mathbf{P}_{(A\Rightarrow B)\Rightarrow C} \to \mathcal{P}(M_{(A\Rightarrow B)\Rightarrow C} \times \mathbb{A}^2)$,

$$\widehat{\sigma \lightning \tau}(p) = \hat{\tau}(p \downharpoonleft M_A) \cup \bigcup_{\substack{qa\langle b\rangle \sqsubseteq p \\ q \in I_B}} \hat{\sigma}(p \upharpoonright qa\langle b\rangle)$$

As in the case of the extensional definition, the definition is asymmetrical. Unlike the extensional definition the intensional definition makes some features of composition clearer. The first one is that the behaviour of the composite strategy in the second component $(B \Rightarrow C)$ only depends on the history of the play as restricted to that component, $p \downharpoonleft M_A$. In other words, the strategy τ does not 'see' what σ is up to.



The second one is that the behaviour of the composite strategy in the first component $(A \Rightarrow B)$ is restricted not only to just the history of the play in that component, but also to each 'thread' of the strategy $p \upharpoonright q$, going back to some initial move $q \in I_B = I_{A\Rightarrow B}$.



This definition also makes it more apparent that there is an implicit nondeterminism when a move occurs in the shared arena B, as both σ and τ can continue playing independently. We will see in Sect. 6.3.1.2 that this has some important consequences.

To arrive at composition itself we need to hide the moves in the interface arena B from each $\widehat{\sigma \between \tau}(p)$. This concludes the detour into the intensional presentation of strategy composition.

Proposition 27 *Given strategies $\sigma : A \Rightarrow B, \tau : B \Rightarrow C$ their composition is also a strategy $\sigma; \tau : A \Rightarrow C$.*

Proof Equivariance is preserved by all operations above, from the general principles of the theory of nominal sets (Pitts, 2013).

Prefix-closure and O-closure are immediate by unwinding the definitions.

It remains to show that the sequences are valid plays of $A \Rightarrow C$. We already know the pointer sequences are well formed. The first step is to show that $!_{I_B}\sigma \underset{M_{A\Rightarrow B}, M_{B\Rightarrow C}}{||} \tau$ is a strategy in arena $(A \Rightarrow B) \Rightarrow C$, which is immediate from definitions. The second step is to show that $\downharpoonleft M_B$ gives plays in $A \Rightarrow C$, which is true because pointers from C-moves to B-moves to A-moves are replaced by pointers from C-moves directly to A-moves. □

6.3 Game Semantic Models

6.3.1 Cartesian Closed Categories

In this essay we are focussing on programming languages that build on the (call-by-name) lambda calculus, so we will focus on games which can model it. Instead of relating directly the syntactic and the semantic models, it is standard to use an abstract mathematical model expressed in terms of *category theory*.[2] For our language, this model is known as a *Cartesian closed category* (Lambek & Scott, 1988). It is an important model, beautiful in its simplicity, deeply connected with type-theoretical

[2] For the current presentation a minimal familiarity with the basic concepts of this topic is required; accessible introductions and tutorials abound (e.g. Abramsky & Tzevelekos, 2011; Milewski, 2018)).

and logical aspects of computations. Baez and Stay (2009) give a fascinating account of these connections.

We will start by attempting to identify a category of games where objects are arenas $A, B, \ldots$ and morphisms $\sigma : A \to B$ are strategies $\sigma : A \Rightarrow B$. The product of two objects is the arena product $A \times B$, the terminal object is the empty arena I, and the exponential object is the arrow arena, $B^A = A \Rightarrow B$.

Composition is well defined (Proposition 27), but we still need to verify its required categorical properties: *associativity* and the existence of an *identity* strategy for composition for all arenas.

6.3.1.1 Associativity

We show the proof of composition in detail, for didactic reasons. The justification pointers can be an awkward mathematical structure, and the original formulation, based on numerical indices into the sequence is particularly unfortunate since any operation on indices requires re-indexing. As a consequence, the original proof of associativity is rather informal. The use of *names* as a representation for pointers eliminates the need for re-indexing and can allow proofs that are both rigorous and elementary. The only challenge of the proof lies in the careful unpacking of several layers of complicated definitions, but once this bureaucracy is dealt with the reasoning is obvious. Other proofs in this presentation are similarly elementary, if tedious. Avoiding these complications can be achieved, but at the cost of some significant additional mathematical sophistication (Castellan, Clairambault, & Rideau, 2017).

Proposition 28 (Associativity) *For any three strategies $\sigma : A \Rightarrow B$, $\tau : B \Rightarrow C$, $\upsilon : C \Rightarrow D$, $(\sigma; \tau); \upsilon = \sigma; (\tau; \upsilon)$.*

Proof Elaborating the definitions, the LHS is

$$(!_C((!_B\sigma \underset{AB,BC}{||} \upharpoonright B) \underset{ABC,CD}{||} (\nu \upharpoonright B)) \upharpoonright C \quad = (!_C((!_B\sigma \underset{AB,BC}{||} \tau) \underset{ABC,CS}{||} \nu) \upharpoonright BC \tag{6.1}$$

There are no B-moves in $(!_C((!_B\sigma \underset{AB,BC}{||} \upharpoonright B)$, so we can extend the scope of $\upharpoonright B$.

Elaborating the definitions, the RHS is

$$(!_B\sigma \underset{AB,BD}{||} (!_C\tau \underset{BC,CD}{||} \nu) \upharpoonright C) \upharpoonright B \tag{6.2}$$

$$= (!_B(\sigma \upharpoonright C) \underset{AB,BD}{||} (!_C\tau \underset{BC,CD}{||} \nu) \upharpoonright C) \upharpoonright B \tag{6.3}$$

$$= (!_B\sigma \underset{AB,BD}{||} (!_C\tau \underset{BC,CD}{||} \nu) \upharpoonright C) \upharpoonright BC \tag{6.4}$$

Equation 6.3 is true because there are no C-moves in σ, so $\sigma \upharpoonright C = \sigma$.

Equation 6.4 is true because $\upharpoonright C$ distributes over concatenation.

Therefore, it is sufficient to show the expressions in Eqs. 6.1 and 6.3 are equal.

Let $p \in !_C((!_B\sigma \underset{AB,BC}{||} \tau) \underset{ABC,CS}{||} \nu$ is equivalent, by definition with

$$p \upharpoonright D \in !_C(!_B\sigma \underset{AB,BC}{||} \tau) \tag{6.5}$$

$$\wedge\ p \upharpoonright AB \in \nu \tag{6.6}$$

By elaborating the definitions:

$$\text{Proposition 5} \Leftrightarrow \forall ma\langle b\rangle.m \in I_C \Rightarrow p \upharpoonright D \upharpoonright \{b\} \in !_B\sigma \underset{AB,BC}{||} \tau \tag{6.7}$$

$$\Leftrightarrow p \upharpoonright D \upharpoonright \{b\} \upharpoonright C \in !_B\sigma \tag{6.8}$$

$$\wedge\ p \upharpoonright D \upharpoonright \{b\} \upharpoonright A \in \tau \Leftrightarrow p \upharpoonright \{b\} \upharpoonright A \in \tau \tag{6.9}$$

The equivalence in Eq. 6.9 holds because once we restrict to moves hereditarily justified by a C-move ($m \in I_C$), the removal of D-moves has no effect since the hereditarily justified play is restricted to arenas ABC.

Elaborating the definition yet again,

$$\text{Proposition 8} \Leftrightarrow \forall nc\langle d\rangle \sqsubseteq p \upharpoonright D \upharpoonright \{b\} \upharpoonright C.n \in I_B \tag{6.10}$$

$$\Rightarrow p \upharpoonright D \upharpoonright \{b\} \upharpoonright C \upharpoonright \{d\} \in \sigma \Leftrightarrow p \upharpoonright \{d\} \in \sigma \tag{6.11}$$

Because restricting to the hereditarily justified play of a B-move ($n \in I_B$) makes the other restrictions irrelevant.

To summarise,

$$p \in !_C((!_B\sigma \underset{AB,BC}{||} \tau) \underset{ABC,CS}{||} \nu \Leftrightarrow$$
$$\forall ma\langle b\rangle.m \in I_C \Rightarrow p \upharpoonright \{b\} \in \tau \wedge \forall nc\langle d\rangle \sqsubseteq p \upharpoonright \{b\}.n \in I_B \Rightarrow p \upharpoonright \{d\} \in \sigma, \tag{6.12}$$

This was the more difficult case.

$p \in !_B\sigma \underset{AB,BD}{||} (!_C\tau \underset{BC,CD}{||} \nu)$ is equivalent to the same conditions as in Eq. 6.12 simply by elaborating the definitions, except that Proposition 6.9 appears as $p \upharpoonright A \upharpoonright \{b\} \in \tau$, but $\upharpoonright, \upharpoonright$ commute in this case. □

Composition is not only associative but also monotonic with respect to the inclusion ordering:

Proposition 29 (Monotonicity) *If* $\sigma \subseteq \sigma'$ *then for any* τ, υ, $\sigma; \tau \subseteq \sigma'; \tau$ *and* $\upsilon; \sigma \subseteq \upsilon; \sigma'$.

The proof is immediate, since all operations involved are monotonic.

6.3.1.2 Identity

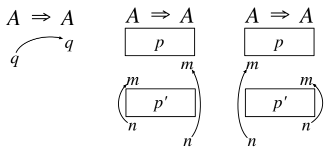
The second challenge is the formulation of an appropriate notion of *identity* strategy for any arena. A candidate for identity $\kappa_A : A_0 \Rightarrow A_1$ is a strategy which immediately replicates O-moves from A_0 to A_1 and vice versa, while preserving the pointer structures—a so-called *copy-cat strategy*.

Definition 30 (*Copy-cat*) For any arena A we defined κ_A as

$$\hat{\kappa}_A(q_1 a\langle b\rangle) = q_0 b\langle c\rangle$$
$$\hat{\kappa}_A\left(p \cdot m_i a\langle b\rangle \cdot m_j c\langle d\rangle \cdot p' \cdot n_j d\langle e\rangle\right) = n_i b\langle f\rangle$$

where $i \neq j \in \{0, 1\}$.

Graphically, the strategy can be represented informally as:



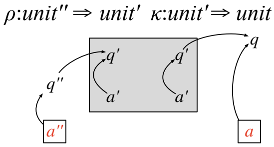
where copies of the same arena (A), initial question (q) or move (m, n) are indicated by using the same variable.

A copy-cat strategy has 'the same behaviour' in both components:

Proposition 31 $\kappa_A \upharpoonright A_0 = \kappa_A \upharpoonright A_1$*, up to a relabeling of moves.*

Proposition 32 $\kappa_A : A_0 \Rightarrow A_1$ *is a strategy.*

However, κ_A is, perhaps surprisingly, not a unit for composition. Consider for example $A = unit$ and the strategy $\rho : unit'' \Rightarrow unit' = \{q'q''a'a''\}$. This strategy, whereby the argument acknowledges termination after the body of the function is akin to a process-forking call. The interaction of $\rho; \kappa$ is shown below:



The synchronisation in the shared arena and the ordering of moves in κ_{unit} impose no particular order between the moves highlighted in the diagram (a, a''), because of the inherent nondeterminism of strategy composition. Therefore the play $qq''a''a$ is in the composition but absent from the original strategy ρ.

However, even though κ_A is not in general an identity for composition, it is *idempotent*, i.e. it is an identity when composed with itself.

Proposition 33 *The* κ_A *strategy is* idempotent, *that is for any arena A we have that* $\kappa_A = \kappa_A; \kappa_A$.

The proof is similar to that of Proposition 36.

Identifying an idempotent morphism means that we can now construct a proper category using the so-called *Karoubi envelope* construction (Balmer & Schlichting, 2001).

Proposition 34 *There exists a category of games in which objects are arenas A, identities at A are* copy-cats κ_A *and morphisms* $\sigma^\dagger : A \to B$ *are* saturated strategies $\sigma^\dagger = \kappa_A; \sigma; \kappa_B$, *where* $\sigma : A \Rightarrow B$ *is a strategy.*

We call strategies $\sigma^\dagger$ *saturated* because through composition with the copy-cat strategies κ new behaviours are added. We will discuss this further when we talk about definability (Sect. 6.4).

6.3.1.3 Cartesian Closed Structure

To be able to model at least call-by-name lambda calculus the category above needs to be Cartesian closed. And, indeed,

Proposition 35 (CCC) *The category of games and saturated strategies is Cartesian closed:*

Terminal object *is the arena* I *with no moves* $M_I = \emptyset$
The product *of two arenas* A_1, A_0 *is the arena* $A_1 \times A_0$ *with projections* $\pi_i : A_1 \times A_0 \to A_i'$, $\pi_i = \kappa_{A_i}$.
The exponential *of two arenas* A, B *is the arena* $A \Rightarrow B$ *with*

Evaluation morphism $ev_{A,B} : (A_0 \Rightarrow B_0) \times A_1 \to B_1$ *with* $ev_{A,B} = \varepsilon^\dagger_{A,B}$ *where*

$$\varepsilon_{A,B}(pm) = \hat{\kappa}_A(pm) \text{ if } m \in A_i$$
$$\varepsilon_{A,B}(pm) = \hat{\kappa}_B(pm) \text{ if } m \in B_i.$$

Transpose *of any strategy* $\sigma : A \times B \to C$ *is the strategy* $\lambda\sigma : A \to (B \Rightarrow C)$ *where* λ *is a re-tagging of moves.*

Proof **Terminal object**: The only strategy $! : A \to I$ contains the empty play since no move in A is enabled.

Product: For every object B and pair of morphisms $\sigma_i^\dagger : B \to A_i$ the product of the two morphisms $\langle \sigma_1^\dagger, \sigma_0^\dagger \rangle = \sigma_1^\dagger \cup \sigma_0^\dagger$. The fact that $\sigma_i^\dagger = \langle \sigma_1^\dagger, \sigma_0^\dagger \rangle; \pi_i$ is immediate because π_i is essentially a (saturated) copy-cat which preserves σ_i and removes σ_{1-i} as none its moves are enabled.

Exponential: Evaluation is a combination of copy-cat strategies, on the A and B components, respectively. The *re-tagging* defining λ is induced by the two isomorphic ways in which the coproduct can associate. We leave the details as an exercise to the reader.

□

6.3.2 Interpreting PCF

Proposition 35 along with Proposition 29 show that the category of saturated games is a model for call-by-name lambda calculus with recursion. For the sake of simplicity we will leave recursion aside and concentrate on the recursion-free language. The interpretation is the standard one.

Let θ stand for the types of PCF. Let β be the base-types of the language (naturals, booleans, unit, etc.) and let $\theta \to \theta$ be the only type-forming construct. The constants of the language are the base-type constants together with base-type operations (arithmetic and logic) and if-then-else. The other term formers are lambda-abstraction $(\lambda x.t)$ and application $(t\, t')$. Let us write $fv(t)$ for the free variables of a term t, defined as usual.

We use typing judgments of the form $\Gamma \vdash t : \theta$, with $\Gamma = x_0 : \theta_0, \ldots, x_n : \theta_n$ a set of variable type assignments, $fv(t) \subseteq \mathrm{dom}(\Gamma)$. The judgment is read as "if variables x_i have types θ_i as given by Γ, then t has type θ." The judgments are checked using the following rules, expressed in natural deduction style:

$$\frac{}{\Gamma, x : \theta \vdash x : \theta} \qquad \frac{\Gamma \vdash t' : \theta \to \theta' \qquad \Gamma \vdash t : \theta}{\Gamma \vdash t'\, t : \theta'} \qquad \frac{x : \theta, \Gamma \vdash t' : \theta'}{\Gamma \vdash \lambda x.t' : \theta \to \theta'}$$

The interpretation function is written as $[\![-]\!]$. Types are interpreted as arenas:

$$[\![bool]\!] = Bool, \text{etc.} \qquad [\![\theta \to \theta']\!] = [\![\theta]\!] \Rightarrow [\![\theta']\!].$$

Variable type assignments are interpreted as products:

$$[\![\emptyset]\!] = I, \qquad [\![x : \theta, \Gamma]\!] = [\![\theta]\!] \times [\![\Gamma]\!].$$

Terms $\Gamma \vdash t : \theta$ are interpreted as strategies on arena $[\![\Gamma]\!] \Rightarrow [\![\theta]\!]$, inductively on the (unique) derivation of the type judgment. The interpretation of the constants as strategies was already given in the preceding sections. Variables, abstraction and application are interpreted canonically using the categorical recipe:

$$\begin{aligned}
[\![x : \theta, \Gamma \vdash x : \theta]\!] &= \pi_0 : [\![\theta]\!] \times [\![\Gamma]\!] \to [\![\theta]\!] \\
[\![\Gamma \vdash t'\, t : \theta']\!] &= \langle [\![\Gamma \vdash t' : \theta \to \theta']\!], [\![\Gamma \vdash t : \theta]\!] \rangle ; ev_{[\![\theta]\!],[\![\theta']\!]} \\
[\![\Gamma \vdash \lambda x.t' : \theta \to \theta']\!] &= \lambda [\![x : \theta, \Gamma \vdash t' : \theta']\!].
\end{aligned}$$

6.4 Definability

The saturated unrestricted model described here contains many behaviours which are not syntactically definable in PCF. A simple example would be the non-deterministic coin-flip strategy ($\sigma_{flip} = \text{strat}\{q \cdot true, q \cdot false\}$). In this section we will determine a syntactic extension for PCF which restores definability. One may think of it as an 'axiomatisation' of the game-semantic model.

The saturation of strategies might be worrying since it involves a loss of control over the order in which certain moves occur. Can we still have languages with sequencing or synchronisation? The property below gives a positive answer.

Proposition 36 (Synchronisation) *Let* $\sigma = \text{strat}\{p \cdot ma\langle b\rangle \cdot nc\langle d\rangle \cdot p'\} : A \Rightarrow B$. *Then,* $p \cdot nc\langle d\rangle \cdot ma\langle b\rangle \cdot p' \in \sigma^\dagger$ *if and only if* $m \in P_{A\Rightarrow B}$ *or* $n \in O_{A\Rightarrow B}$. *This also holds if* $a = c$.

Proof We need to consider all combinations for m and n to be O or P moves in $A \Rightarrow B$, and also whether they occur in A or in B. Because κ always copy-cats O moves to P moves, and because in this case the P move occurs after the O move ultimately it does not matter whether κ_A or κ_B does the copying, so for this argument it does not matter whether m, n occur in A or B. Let us assume they occur in B

1. $m \in O_B \subseteq O_{A\Rightarrow B}$ and $n \in P_B \subseteq P_{A\Rightarrow B}$. When composing σ with κ_B the polarities of the moves in arena $B \Rightarrow B'$ are reversed, $m \in P_{B\Rightarrow B'}$, $n \in O_{B\Rightarrow B'}$. As usual, we use B, B' to distinguish the two occurrences of arena B in the composite arena $B \Rightarrow B'$. This means that m is necessarily the copy-image of a B' move occurring earlier and n will be copied into a B' move occurring later. Following the hiding of the arena B the order of the moves m and n must necessarily stay the same.
2. $m \in O_B \subseteq O_{A\Rightarrow B}$ and $n \in O_B \subseteq P_{A\Rightarrow B}$. When composing σ with κ_B the polarities of the moves in arena $B \Rightarrow B'$ are reversed, $m \in P_{B\Rightarrow B'}$, $n \in P_{B\Rightarrow B'}$. This means that m is necessarily the copy-image of a B' move occurring earlier and so is n. Both these moves are O-moves in arena $B \Rightarrow B'$, and they may occur in any order since κ_B accepts both orders. So after hiding the arena B in composition both orders of m, n may be present in the composite strategy.
3. All other cases are similar to the previous case (2).

□

This Proposition gives in fact a rational reconstruction of the *permutative saturation* condition in asynchronous game semantics used by Laird (2005); Ghica and Murawski (2008) and more broadly in semantics of asynchronous communication (Udding, 1986). The reason that we can permute all sequences of moves $m \cdot n$ in which m does not justify n, unless m is an O-move and n a P-move is, intuitively, that P should always be able to synchronise on O. This is reflected, mathematically, by the fact that the saturated copy-cat κ will always copy O-moves into P-moves. The Proposition is also useful because it will allow us to follow quite closely the definability procedure used by Ghica and Murawski (2008).

It is important to emphasise at this stage that the simple game model we use here takes an '*angelic*' perspective on termination. We mentioned earlier the $\sigma_{flip} : Bool$ non-deterministic strategy. Let us consider a 'non-deterministic projection' strategy $\sigma_\pi : Unit \times Unit' \to Unit'', \sigma_\pi = \text{strat}\{q'' \cdot q \cdot a \cdot a'', q'' \cdot q' \cdot a' \cdot a''\}$, along with the '*diverging*' unit strategy $\sigma_\Omega = \{q\} : Unit$ and the non-diverging strategy at the same type $\sigma_{nop} = \text{strat}\{q \cdot a\} : Unit$. It follows that $\langle \sigma_\Omega, \sigma_{nop} \rangle; \sigma_\pi = \sigma_{nop}$. In words, the non-deterministic choice between a responsive and a non-responsive strategy will always be the responsive strategy, hence the 'angelic' moniker. More precise models can include separately the non-responsive plays, called *divergences*, similar to the trace models of Brookes et al. (1984), as adapted to game semantics by Harmer and McCusker (1999).

Another important feature of a game model is whether it is *extensional*, like that of Abramsky and McCusker (1996) or *intensional*, like that of Hyland and Ong (2000). Two strategies $\sigma_1, \sigma_2 : I \to A$ are said to be *equivalent* $\sigma_1 \equiv \sigma_2$ if for all *test strategies* $\tau : A \to Unit$, $\sigma_1; \tau = \sigma_2; \tau$, i.e. $\sigma_1; \tau = \sigma_2; \tau = \sigma_\Omega$ or $\sigma_1; \tau = \sigma_2; \tau = \sigma_{nop}$.

Proposition 37 *Two strategies* $\sigma_1, \sigma_2 : I \to A$ *are* equivalent $\sigma_1 \equiv \sigma_2$ *if and only if their saturations are equal* $\sigma_1^\dagger = \sigma_2^\dagger$.

Proof Equal strategies are equivalent, obviously. If two strategies are not equal then there is a play p in their symmetric difference. Composition with $\text{strat}\{q \cdot p \cdot a\} : A \to Unit$ will yield σ_Ω for the strategy not containing p in its saturation, and σ_{nop} for the other one. □

This makes the model extensional.

Note that unlike IA or concurrent IA (ICA) (Ghica & Murawski, 2008) saturated strategies are not characterised by their 'set of complete plays', i.e. those plays in which the initial question is answered. This is because strategies may contain plays such as $q \cdot a \cdot q' \cdot a'$ in which moves happen after the initial move was answered. Unlike intensionality, which makes the model harder to use directly, the fact that strategies are not characterised by complete plays is not problematic from a technical point of view.

We are now ready to address the question of definability: what syntax do we need so that any strategy, or rather its saturated version, is the denotation of some term. We will follow the definability procedure of ICA, since the two models are similar enough. The ingredients are:

State. ICA has *local variables*, which are used to record and test the order of execution of moves by associating each move either with writing to the state, or reading from the state.

Semaphores. ICA also has *local split binary semaphores* to achieve synchronisation between sub-plays, which can be seen as '*threads*'. The type of semaphores is isomorphic to $Unit_1 \times Unit_2$ and have strategy $\sigma_{sem} : (Sem \Rightarrow Unit) \Rightarrow Unit'$, $\sigma_{sem} = \text{strat}\{q'q(q_1a_1q_2a_2)^*aa'\}$, strictly alternating between the two components of the semaphore type. Intuitively, one represents a '*grab*' action, and the other a '*release*'.

Concurrency. ICA has a *static parallelism* strategy $\sigma_{par} : \mathit{Unit}' \Rightarrow \mathit{Unit}'' \Rightarrow \mathit{Unit}$, $\sigma_{par} = \text{strat}\{qq'q''a'a''a\}$. It interleaves its arguments in any order but it only terminates when both arguments terminate. This gives a '*fork/join*' structure to all strategies, in the sense that only unanswered questions may justify, and they may only be answered when all the questions they justify have been themselves answered. This is the construct we drop, and we replace it with a simpler, dynamic concurrency strategy $\sigma_{run} : \mathit{Unit}' \Rightarrow \mathit{Unit}$ which allow its argument to finish before or *after* the initial question is answered. In other words, $\sigma_{run} = \text{strat}\{qq'aa'\}$.

With this rather minor change, the definability procedure for ICA can be replicated, giving us a complete syntax for the model. The only possibly significant distinction between the two models is in the fact that ICA strategies are characterised by complete plays, which means that a play can be safely removed from a strategy if the initial question is not answered. This could have a potential impact on definability because we could, for example, allow several P-moves to occur and, subsequently, if we decide the play is not proceeding in a desired play we can simply stop playing in it. Such an artifice would not work in this model. However, the definability argument for ICA does not make use of it.

For instance, the strategy for *catch* in Sect. 6.2.2.4 can be defined in this model, but not in the ICA model because in the case the exception is raised it breaks the fork/join discipline by providing the answer to the initial question before pending questions have been answered. However, a word of caution. Operationally, the definability argument and therefore the reconstruction of *catch* creates a large number of concurrent threads, killing off those plays that evolve in an undesirable direction by introducing divergences, which are subsequently hidden by the angelic notion of observation. This reconstruction of the strategy is artificial, of little practical importance.

6.5 Conclusion

This presentation is meant primarily as a didactic exercise, presenting a game model where the plays are governed by no combinatorial restrictions such as bracketing, alternation, innocence, etc. The structure we preserve is the proper justification of plays, which we take it to have essentially a causal rationale. We also preserve certain closure conditions of strategies which also have causal or temporal motivations, while adding a new one via the Karoubi envelope construction: permutative asynchronous saturation. Behind this construction lies a technical reason, the fact that the 'intuitive' copy-cat strategy does not do the job of being a unit for composition with arbitrary strategies. However, saturation is consistent with similar closure conditions tracing back to the early literature on asynchronous concurrency. Of course, saturation could have been stipulated as a required closure property of strategies, as Ghica and Murawski (2008) do. The choice to present it as an 'add-on' property is also didactically motivated. The direct intuition behind saturation is, arguably, not as compelling as that behind, say, prefix-closure or equivariance, so there could be a risk of baffling the newcomer to game semantics. But there is also some deliberate methodological candor in showing how the game-semantic sausage is made. Game semantics is sometimes fiddly. Its power is derived from its ability to model the sometimes messy reality of programming languages. Saturation is an example of a condition that can be reconstructed by backtracking from mathematical considerations.

Methodologically, this semantics harkens back to the early days of semantic programming languages when the denotational approach was deemed sufficient. In time, operational semantics became the dominant specification paradigm, with denotational models seen more like characterisations of observational equivalence, rather than prima facie specifications. And, indeed, some of the more abstract denotational models, such as functor-category models of state (O'Hearn, Power, & Takeyama, 1999), are far removed from the mechanics of evaluation. In contrast, game models, especially when strategies are presented via a next-move function, are close enough to evaluation to allow for quantitative modeling (Ghica, 2005) or semantics-directed compilation (Ghica, 2007). In fact operational semantics and game semantics are eminently compatible, since the latter can be used to give a compositional formulation to the former via so-called '*system-level game semantics*' (Ghica & Tzevelekos, 2012).

Finally, Abramsky's idea of a *semantic cube* is compelling and inspirational beyond definability, and beyond game semantics. In semantics of programming languages the idea of a taxonomy of behaviours is usually automatically set in the context of types. But types are largely related to the input-output (extensional) shape of computation, whereas the features that form the dimensions of the Abramsky cube concern the intensional shape of computation, and it cuts across the type discipline. Indeed, all languages we have considered in this essay, extensions of PCF, satisfy the simply-typed discipline. This deeper semantic taxonomy which we associate with the Abramsky Cube has direct relevance on equational reasoning, as pointed out for examble by Dreyer et al. (2012). More recent work by Ghica et al. (2019), which presents a model of computation based on hyper-graphs, shows how these metaphorical shapes of computation can be realised as actual shapes of graphs arising in the process of computation. These all suggest that the ideas behind Abramsky's Cube are likely to exert a lasting influence on the study of programming languages.

Acknowledgements Much of this material represents a tidying up of notes for courses taught at research summers schools, in particular the JetBrains Summer School, St. Petersburg (2016) and the Oregon Programming Languages Summer School (2018).

References

Abramsky, S., & McCusker, G. (1996). Linearity, sharing and state: A fully abstract game semantics for idealized Algol with active expressions. *Electronic Notes in Theoretical Computer Science, 3*, 2–14 (1996). https://doi.org/10.1016/S1571-0661(05)80398-6 (Also appeared in O'Hearn and Tennent (1997)).

Abramsky, S., & McCusker, G. (1999). Game semantics. In U. Berger, & H. Schwichtenberg (Eds.), *Computational Logic* (pp. 1–55). Berlin, Heidelberg, Springer Berlin Heidelberg. ISBN 978-3-642-58622-4. https://doi.org/10.1007/978-3-642-58622-4_1

Abramsky, S., & Tzevelekos, N. (2011). Introduction to categories and categorical logic. arXiv:1102.1313

Abramsky, S., Jagadeesan, R., & Malacaria, P. (2000). Full abstraction for PCF. *Information and Computation, 163*(2), 409–470. https://doi.org/10.1006/inco.2000.2930.

Baez, J. C., & Stay, M. (2009). Physics, topology, logic and computation: A Rosetta stone. In B. Coecke (Ed.), *New structures for physics* (Vol. 813, pp. 95–174). Lecture notes in physics. Berlin: Springer 2011. arXiv:0903.0340

Balmer, P., & Schlichting, M. (2001). Idempotent completion of triangulated categories. *Journal of Algebra, 236*(2), 819–834. ISSN 0021-8693. https://doi.org/10.1006/jabr.2000.8529

Brookes, S. D., Hoare, C. A. R., & Roscoe, A. W. (1984). A theory of communicating sequential processes. *Journal of the ACM, 31*(3), 560–599. https://doi.org/10.1145/828.833

Castellan, S., Clairambault, P., Rideau, S., & Winskel, G. (2017). Games and strategies as event structures. *Logical Methods in Computer Science, 13*(3). https://doi.org/10.23638/LMCS-13(3:35)2017

Curien, P. (2007). Definability and full abstraction. *Electronic Notes in Theoretical Computer Science, 172*, 301–310. https://doi.org/10.1016/j.entcs.2007.02.011

Danos, V., & Harmer, R. (2000). Probabilistic game semantics. In *15th Annual IEEE Symposium on Logic in Computer Science, Santa Barbara, California, USA, 26–29 June 2000* (pp. 204–213). https://doi.org/10.1109/LICS.2000.855770

Danos, V., & Harmer, R. (2001). The anatomy of innocence. In *Computer Science Logic, 15th International Workshop, CSL 2001. 10th Annual Conference of the EACSL, Paris, France, 10–13 Sept 2001, Proceedings* (pp. 188–202). https://doi.org/10.1007/3-540-44802-0_14

Dreyer, D., Neis, G., & Birkedal, L. (2012). The impact of higher-order state and control effects on local relational reasoning. *Journal of Functional Programming, 22*(4–5), 477–528. https://doi.org/10.1017/S095679681200024X

Gabbay, M., & Ghica, D. R. (2012). Game semantics in the nominal model. *Electronic Notes in Theoretical Computer Science, 286,* 173–189. https://doi.org/10.1016/j.entcs.2012.08.012.

Gabbay, M. J., Ghica, D. R., & Petrisan, D. (2015). Leaving the nest: Nominal techniques for variables with interleaving scopes. In *24th EACSL Annual Conference on Computer Science Logic CSL, 7–10 Sept 2015, Berlin, Germany* (pp. 374–389). https://doi.org/10.4230/LIPIcs.CSL.2015.374

Ghica, D. R. (2005). Slot games: A quantitative model of computation. In *Proceedings of the 32nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2005, Long Beach, California, USA, 12–14 Jan 2005* (pp. 85–97). https://doi.org/10.1145/1040305.1040313

Ghica, D. R. (2007). Geometry of synthesis: A structured approach to VLSI design. In *Proceedings of the 34th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2007, Nice, France, 17–19 Jan 2007* (pp. 363–375). https://doi.org/10.1145/1190216.1190269

Ghica, D. R. (2009). Applications of game semantics: From program analysis to hardware synthesis. In *Proceedings of the 24th Annual IEEE Symposium on Logic in Computer Science, LICS 2009, 11–14 Aug 2009, Los Angeles, CA, USA* (pp. 17–26). https://doi.org/10.1109/LICS.2009.26

Ghica, D. R., & Murawski, A. S. (2008). Angelic semantics of fine-grained concurrency. *Annals of Pure and Applied Logic, 151*(2–3), 89–114. https://doi.org/10.1016/j.apal.2007.10.005.

Ghica, D. R., & Tzevelekos, N. (2012). A system-level game semantics. *Electronic Notes in Theoretical Computer Science, 286,* 191–211. https://doi.org/10.1016/j.entcs.2012.08.013.

Ghica, D. R., Muroya, K., & Ambridge, T. W. (2019). Local reasoning for robust observational equivalence. *CoRR*. arXiv:1907.01257

Harmer, R., Laurent, O. (2006). The anatomy of innocence revisited. In *FSTTCS 2006: Foundations of Software Technology and Theoretical Computer Science, 26th International Conference, Kolkata, India, 13–15 Dec 2006, Proceedings* (pp. 224–235). https://doi.org/10.1007/11944836_22

Harmer, R., & McCusker, G. (1999) A fully abstract game semantics for finite nondeterminism. In *14th Annual IEEE Symposium on Logic in Computer Science, Trento, Italy, 2–5 July 1999* (pp. 422–430). https://doi.org/10.1109/LICS.1999.782637

Harmer, R., Hyland, M., & Melliès, P. (2007). Categorical combinatorics for innocent strategies. In *22nd IEEE Symposium on Logic in Computer Science (LICS 2007), 10–12 July 2007, Wroclaw, Poland, Proceedings* (pp. 379–388). https://doi.org/10.1109/LICS.2007.14

Hoare, C. A. R. (1978). Communicating sequential processes. In *The origin of concurrent programming* (pp. 413–443). Berlin: Springer.

Hyland, J. M. E., & Ong, C. L. (2000). On full abstraction for PCF: I, II, and III. *Information and Computation, 163*(2), 285–408. https://doi.org/10.1006/inco.2000.2917.

Jeffrey, A., & Rathke, J. (2005). Java Jr: Fully abstract trace semantics for a core java language. In *Programming Languages and Systems, 14th European Symposium on Programming, ESOP 2005, Edinburgh, UK, 4–8 April 2005, Proceedings* (pp. 423–438). https://doi.org/10.1007/978-3-540-31987-0_29

Laird, J. (1997). Full abstraction for functional languages with control. In *Proceedings, 12th Annual IEEE Symposium on Logic in Computer Science, Warsaw, Poland, June 29–July 2, 1997* (pp. 58–67). https://doi.org/10.1109/LICS.1997.614931

Laird, J. (2001). A game semantics of idealized CSP. In S. D. Brookes, & M. W. Mislove (Eds.), *Seventeenth Conference on the Mathematical Foundations of Programming Semantics, MFPS 2001, Aarhus, Denmark, 23–26 May 2001*, *Electronic notes in theoretical computer science* (Vol. 45, pp. 232–257). Elsevier. https://doi.org/10.1016/S1571-0661(04)80965-4

Laird, J. (2005). A game semantics of the asynchronous *pi*-calculus. In *Concurrency Theory, 16th International Conference, San Francisco, CA, USA, 23–26 Aug 2005* (pp. 51–65). https://doi.org/10.1007/11539452_8

Lambek, J., & Scott, P. J. (1988). *Introduction to higher-order categorical logic* (Vol. 7). Cambridge University Press.

Landin, P. J. (1966). The next 700 programming languages. *Communications of the ACM, 9*(3), 157–166. https://doi.org/10.1145/365230.365257.

Lawvere, F. W. (1969). Adjointness in foundations. *Dialectica, 23*(3–4), 281–296. https://doi.org/10.1111/j.1746-8361.1969.tb01194.x.

Loader, R. (2001). Finitary PCF is not decidable. *Theoretical Computer Science, 266*(1–2), 341–364. https://doi.org/10.1016/S0304-3975(00)00194-8.

Milewski, B. (2018) *Category theory for programmers*. Blurb.

Milner, R. (1977). Fully abstract models of typed lambda-calculi. *Theoretical Computer Science, 4*(1), 1–22. https://doi.org/10.1016/0304-3975(77)90053-6.

Murawski, A. S., & Tzevelekos, N. (2016). Nominal game semantics. *Foundations and Trends in Programming Languages, 2*(4), 191–269. https://doi.org/10.1561/2500000017.

O'Hearn, P., & Tennent, R. (1997). *ALGOL-like languages*. Birkhäuser.

O'Hearn, P. W., Power, J., Takeyama, M., & Tennent, R. D. (1999). Syntactic control of interference revisited. *Theoretical Computer Science, 228*(1–2), 211–252. https://doi.org/10.1016/S0304-3975(98)00359-4.

Oles, F. (1983). Type algebras, functor categories, and block structure. *DAIMI Report Series, 12*(156). https://doi.org/10.7146/dpb.v12i156.7430. A version appeared in O'Hearn and Tennent (1997).

Pitts, A. M. (2013). *Nominal sets: Names and symmetry in computer science* (Vol. 57). Cambridge University Press.

Plotkin, G. D. (1977). LCF considered as a programming language. *Theoretical Computer Science, 5*(3), 223–255. https://doi.org/10.1016/0304-3975(77)90044-5.

Reddy, U. S. (1996). Global state considered unnecessary: An introduction to object-based semantics. *LISP and Symbolic Computation, 9*(1), 7–76.

Reynolds, J. C. (1981). The essence of ALGOL. In J. W. de Bakker, & J. C. van Vliet (Eds.), *Proceedings of the International Symposium on Algorithmic Languages* (pp. 345–372). Amsterdam. Also appeared in O'Hearn and Tennent.

Scott, D., & Strachey, C. (1971). Toward a mathematical semantics for computer languages. Technical report PRG-6, Oxford Programming Research Group.

Tennent, R. D., & Ghica, D. R. (2000). Abstract models of storage. *Higher-Order and Symbolic Computation, 13*(1/2), 119–129. https://doi.org/10.1023/A:1010022312623.

Udding, J. T. (1986). A formal model for defining and classifying delay-insensitive circuits and systems. *Distributed Computing, 1*(4), 197–204. ISSN 1432-0452. https://doi.org/10.1007/BF01660032

Chapter 7
An Axiomatic Account of a Fully Abstract Game Semantics for General References

Jim Laird and Guy McCusker

Abstract We present an analysis of the game semantics of general references introduced by Abramsky, Honda and McCusker which exposes the algebraic structure of the model. Using the notion of *sequoidal category*, we give a coalgebraic definition of the denotational semantics of storage cells of arbitrary type. We identify further conditions on the model which allow an axiomatic presentation of the proof that finite elements of the model are definable by programs, in the style of Abramsky's *Axioms for Definability*.

Keywords Semantics of programming languages · Game semantics · Computational effects · Computational adequacy · Full abstraction · Higher-order store

7.1 Introduction

Game semantics emerged in the 1990s as a novel approach to denotational semantics of programming languages. The game paradigm made a striking breakthrough with the construction of fully abstract models of PCF, in three independent pieces of work by Abramsky, Jagadeesan and Malacaria (AJM), Hyland and Ong (HO), and Nickau. The significance of these was recognised with the Alonzo Church Award in 2017. The inherent sequentiality of plays in games enabled this new style of semantics to capture notions of sequential composition precisely, thus eliminating "junk" elements such as parallel-or which prevented domain-theoretic models from achieving full abstraction. But that was not the whole story. In order to obtain accurate games models of PCF-

This paper is electronically published in Electronic Notes in Theoretical Computer Science URL: www.elsevier.nl/locate/entcs

J. Laird · G. McCusker (✉)
Department of Computer Science, University of Bath, Bath, UK
e-mail: gam23@bath.ac.uk

J. Laird
e-mail: jl317@bath.ac.uk

A. Palmigiano and M. Sadrzadeh (eds.), *Samson Abramsky on Logic and Structure in Computer Science and Beyond*, Outstanding Contributions to Logic 25,
https://doi.org/10.1007/978-3-031-24117-8_7

style functional computation, the behaviour of strategies needed to be constrained carefully. In the case of the HO model, three constraints were identified:

Bracketing Moves were marked as *questions* and *answers*, and the question and answer moves were required to nest like matching parentheses

Visibility Each move carried a pointer to a previous move; these pointers were required to land in a particular sub-sequence of the play so far, called the *view*

Innocence A strategy's decision on what move to make could only depend upon the view.

In the presence of these three conditions, and similar ones in the case of the other games models, it became possible to establish the key *definability* result, showing that each finite strategy is the denotation of a term of PCF. To prove such a result, a close analysis of the possible behaviours of strategies under these constraints was given. Abramsky went on to describe the finer structure of the games models which allows this analysis to be carried out, proposing an axiomatisation of the definability theorem.

After the breakthrough results on PCF, it was discovered—largely in work of Abramsky and his co-authors and students, including the two authors of this paper—that discarding these constraints led to new game semantic models, and moreover that the removal of each constraint corresponded in a precise way to the addition of a certain kind of *computational effect* to the programming language:

- relaxing the bracketing condition corresponds to the addition of first class continuations (Laird, 1997)
- relaxing the condition of innocence corresponds to the addition of basic imperative features—the ability to store and retrieve unstructured data (Abramsky & McCusker, 1996, 1997)
- relaxing the visibility condition corresponds to the addition of higher-order store, that is, the ability to store and retrieve data of function types as well as unstructured data (Abramsky et al., 1998).

(The flexibility of game semantics was also demonstrated by other authors, for example Honda and Yoshida (1997) who showed how a reconfiguration of the structure of games being played led to a model of *call-by-value* computation.)

Abramsky's work on game semantics is characterized by a keen eye for the fine structure of the models. The AJM model was described in terms of an underlying model of intuitionistic linear logic, which was refined further in the axiomatic presentation of definability (Abramsky, 2000). Similarly, the semantics of flat storage was first presented using a linear model. The third of these results, which received the LICS Test of Time Award in 2018, stands out in this respect. As originally described, the model appears quite difficult to grasp from a structural point of view. The essential new element—the higher-order reference cell—is defined by hand and described in a monolithic fashion, and key results such as soundness must be proved by detailed manipulation of this definition. Contrast this with, for example, the elegance of a soundness proof for a model of the lambda-calculus in a cartesian closed category.

The other key properties of games models—computational adequacy and definability of compact elements—present similar issues. Adequacy for the model of higher-order store eludes standard methods such as logical relations due in part to its self-referential nature. The proofs of definability for PCF give a kind of canonical form for the strategies in each of its games models: their common structure was distilled by Abramsky into an axiomatic characterization of the categorical properties required to decompose each finite strategy into smaller parts. However, definability for the models of effects was proved not by a direct analysis but by reduction to the PCF case. In each case it is possible to show that a strategy in the unconstrained model may be factorised as the composition of a PCF strategy with an archetypical effectful strategy: a control operator, a storage cell, or a higher-order reference cell.

There has been considerable study and development in game semantics since the 1990s. One line of work focusses on the fine structure of games models, going beyond the connectives of intuitionistic linear logic to expose aspects that are particular to games. When working with games, the fact that moves are played one at a time makes analysis via linear logic tempting, but the correspondence is not quite perfect—games models tend to exhibit degeneracies or idiosyncracies with respect to linear logic. While it is possible to coerce games models into a linear logic shape, a different line of enquiry is to take the models as they present themselves and search for the natural logic that explains their structure. This point of view has led to Polarised Linear Logic (Olivier, 2002), Tensorial Logic (Melliès & Tabareau, 2010), and the first author's notion of sequoidal category (Laird, 2003).

In this paper we return to the game semantics of higher-order references and show how it may be presented in a principled way after all. The first step in this direction was taken by Laird (2003), who observed that the reference cell does arise in algebraic way, once one has access to the notion of sequoid $A \oslash B$. The connective $\oslash$ captures a fundamental aspect of game semantics, which is that the next move has to happen somewhere! Roughly, the game $A \oslash B$ is played as a parallel composition of A and B, just as in the common interpretation of the linear logic $A \otimes B$, but with the additional constraint that the first move must happen in A. (It is worth noting that Abramsky's axiomatic definability argument makes use of a relative of this idea, without enshrining it in a connective.) In models without the visibility condition, the functor $- \oslash A$ has the remarkable property of being both left and right adjoint to the internal hom $A \to -$. This means that a morphism of type $(A \to B) \to C$ is interchangeable with one of type $B \to (C \oslash A)$, so that supplying an argument (here, of type A) to a function is equivalent to producing it directly to the environment. This is exactly what higher-order references allow: by storing terms on the heap, they can be exported beyond their lexical scope. This observation allows for an algebraic presentation of the game semantics of general references. More recently, Laird (2019) took this further and showed how coalgebraic methods could be used to define and reason about dynamically allocated local reference cells.

Our contribution in this paper is to draw all of these ideas together, giving a reconstruction of the games model of general references and its extension with first-class continuations. We use the sequoid to prove soundness of the operational semantics, give a novel proof of computational adequacy by extending the model to count reduction steps directly and provide for the first time an axiomatic approach to the proof of definability.

7.1.1 Overview

Before embarking on a detailed presentation of these ideas, we present in this section a high-level overview of the landscape, with the hope of highlighting the most significant aspects of our axiomatic account of references. As explained above, the central tool is the sequoid connective $\oslash$. It captures a key aspect of game semantics and of computation: the order in which actions take place. The simple idea of refining the product type $A \times B$ to $A \oslash B$ in which A must be accessed before B underpins our algebraic analysis of the game semantics of store.

7.1.1.1 Linearity, the Sequoid, and a Coalgebraic View of Local Store

Abramsky's work on game semantics of store emphasised from the beginning the significance of *linearity*, understood simply as the distinction between invoking a program once or multiple times. In the pure functional case, every invocation of a program yields the same result. By contrast, when dealing with stateful programs, successive invocations of the same program may yield different results. Linear logic, in providing a logical distinction between resources that may be used once and those that may be used repeatedly, provides a way to analyse this situation at the level of types. (Similar observations of the connection between linearity and state were present in the work of Wadler (1990) and Reddy (1993)). This is exactly what was done in Abramsky and McCusker (1996, 1997). There, one may encounter a linear strategy $\sigma : A$ which can be invoked only once; its promotion $\sigma^\dagger :!A$ which may be invoked repeatedly but behaves as σ in each invocation; or a general strategy $\tau :!A$ which may be invoked repeatedly but where the behaviour in each invocation depends on the history in all the others. These ideas carry over to the present work, in a slightly different form: we will use separate categories of "well-opened" strategies which can only be started once, and "multi-threaded" strategies. The sequoid provides a vital connection between these two worlds: roughly speaking, a multi-threaded strategy of type A is the same as a single threaded strategy on the "infinite sequoidal power"

$$A \oslash A \oslash A \oslash \cdots$$

This observation forms the basis of our approach to modelling objects with local, encapsulated store. Suppose we have a type of stores, S. A program of type A employing a global store of type S may be interpreted straightforwardly as a morphism of the form

$$S \to A \oslash S$$

which returns an updated store along with its own result. Notice in passing how the sequoid here emphasises that the updated store is only available after the progarm itself has been invoked. The infinite sequoidal power described above can be formalised by seeing the object A as a certain kind of final coalgebra for the operation $A \oslash -$. The final coalgebra

$$A \to A \oslash A$$

is a single step of unfolding A into the infinite sequoidal power. Then the fact that A is a *final* coalgebra gives us a map

$$S \to A$$

which encapsulates the global store S as a local store.

7.1.1.2 Visibility and General References

Further analysis of the algebraic properties of the sequoid yields the final piece of the puzzle allowing us to interpret dynamically allocated local store of arbitrary type. We first mention the unremarkable fact that, in an appropriate category of games, there is an adjunction

$$A \oslash - \dashv A \multimap -$$

This amounts to a refinement of the usual cartesian closed structure which keeps track of the order in which subterms are evaluated.

Now consider the type $(A \multimap B) \oslash A$. For the sake of simplicity let B be a simple object such as the type commands: in game semantics terms this has a single first move q to which player may make a single response a.

In a pure functional setting, a term of type $(A \to B) \times A$ would be a pair of a term $\lambda x.M : (A \to B)$ and a term $N : A$ with no interaction between them. In a setting with access to a store, for example allowing storage of Booleans, there may be communication via setting flags in shared variables, so that evaluation of $\lambda x.M$ can influence the behaviour of N. Nevertheless the two terms have district lexical scope, so for example N cannot access the term that x becomes bound to in evaluation of $\lambda x.M$.

In game semantics terms, this restriction is captured by the *visibility condition*: consider the play

$$\begin{array}{ccccc}
(A & \to & B) & \times & A \\
 & & q & & \\
 & & a & & \\
 & & & & m \\
m & & & &
\end{array}$$

Playing this move m in the leftmost A, and subsequently copying moves back and forth between the instances of A, would amount to exporting the term bound to x into the term N. But the visibility condition makes this impossible: it says that in order to play this move m. Proponent's *view* must contain the justifying move q. The view is a partial history of the play, obtained by omitting moves between an Opponent move and its justifier, and stopping at an initial move. The view after the first m is played contains only that single move, so the play shown above is forbidden.

In the absence of the visibility condition, such play is possible and in fact forms the basis of the games model of higher-order references: the play sketched above is typical of the reference cell: the $A \to B$ component of the illustrated type provides the "write method", which receives a term of type A and stores it; the A component provides the "read method" which retrieves the stored term. The strategy pictured above simply routes requests to access the stored term back to the original storer.

We explain this behaviour algebraically by noticing that, in the absence of the visibility condition, we have a second, much more remarkable adjunction:

$$A \multimap - \dashv A \oslash -$$

Using the counit of this adjunction we can construct a map

$$1 \to (A \multimap B) \oslash A$$

with exactly the behaviour pictured above. From here it is a few simple steps to recover the semantics of higher order store as introduced in [AHM98]: taking the product of this "writing" map with the previously explained final coalgebra $A \to A \oslash A$, which plays the role of the "reading" map, gives us a coalgebra for $A \times (A \multimap B)$:

$$A \longrightarrow (A \oslash A) \times (A \multimap B) \oslash A \cong A \times (A \multimap B) \oslash A.$$

The unique coalgebra map

$$A \longrightarrow A \times (A \multimap B)$$

then provides us with the interpretation of the reference cell.

This reconstruction of the semantics presented in Abramsky et al. (1998) not only exposes the algebraic structure underlying this model of store, it also provides tools for reasoning with it. We exploit properties of the coalgebra structure that provides the reader morphism and the adjunction that provides the writer to establish the soundness of the semantics, in particular when showing that what is written to a cell is returned when a subsequent read takes place.

We note in passing that the categories of games we present in this paper are not identical in their presentation to those of Abramsky et al. (1998) and similar papers. In particular, in order to gain access to the sequoid $A \oslash B$, we allow moves to be enabled by previous moves of the same player: this allows $A \oslash B$ to be defined by stipulating that the initial moves of B are enabled by those of A. This technical difference makes possible our algebraic analysis, without interfering with the model being studied: the games and strategies which are the denotations of types and terms from our programming language are identical to those of Abramsky et al. (1998).

7.1.1.3 Axiomatic Definability and Full Abstraction

The sequoid and the close interplay between its algebraic properties and the construction of our model allow us to conduct an axiomatic decomposition of strategies in the model. The decomposition exhibits any strategy as arising from smaller strategies, combined according to the operations explained above. Since these operations correspond directly to the semantic definitions, this forms the basis of an inductive proof that every strategy is the meaning of a program. This *definability* result leads quickly to a *full abstraction* result for our model.

We illustrate the decomposition with a very simple example. Let o be an object representing an empty type; in the semantics this will be a game with a single opening move only. Consider a non-empty multi-threaded strategy

$$(o \Rightarrow o) \to o.$$

The discussion above lets us see this as a single-threaded strategy

$$(o \Rightarrow o) \to o \oslash o.$$

The sequoid allows us to disentangle the first invocation of the function input from later ones, yielding a strategy

$$(o \Rightarrow o) \oslash (o \Rightarrow o) \to o \oslash o.$$

We note that this strategy is not just single-threaded but *linear*: it only plays one initial move on the left hand side. The adjunctions described above let us rearrange this into a linear strategy

$$o \Rightarrow (o \Rightarrow o) \to (o \Rightarrow o) \Rightarrow o$$

which is equivalent to

$$(o \times o \Rightarrow o) \to (o \Rightarrow o) \Rightarrow o.$$

A linear strategy of this type must arise by applying $- \Rightarrow o$ to one of type

$$(o \Rightarrow o) \to o \times o.$$

We may now use induction to find a term which defines this strategy, and turn that term into one which defines the original strategy by inverting the decomposition steps above, using the syntax of our language.

7.1.1.4 Computational Adequacy

The final piece of the puzzle to complete our account of the sharp correspondence between programs with general references and strategies in our model is *computational adequacy*. Using the algebraic structure of the model we have established that execution steps in the language correspond to equality in the model; computational adequacy is a partial converse, saying that if the model indicates that a program produces a value, then the execution of that program does indeed return a value. That is, it states that the computational evaluation mechanism is adequate for the denotational model.

In higher-order languages, adequacy can be difficult to establish, having much in common with the problem of strong normalization in type theories. A range of techniques are available, most employing some combination of logical relations or predicates (Plotkin, 1977) and continuity or finiteness properties of the denotational model (Pitts, 1996). The case of general references seems particularly challenging. Reference cells introduce infinite behaviour which is not readily reducible to a finite approximant; reference cells may be generated dynamically; and the type system of the language offers little support for analysing this behaviour: the type of a reference cell is very simple and disguises the complex behaviour that may arise. One may hope that the algebraic analysis of references offered by our work sheds light on this problem, but unfortunately to date we have not been able to exploit it in this way. Instead we offer a new approach to computational adequacy which makes explicit in the model the fact that successful executions are finite. By instrumenting the denotational model so that a "value" in the model contains information about how many execution steps are taken in computing it, we obtain a built-in induction measure. Computational adequacy with respect to the instrumented model is then straightforward to prove. It remains to show that the original model may be recovered from the instrumented one, and for this we may indeed make use of logical relations supported by the type structure of the language.

7.1.1.5 Structure of the Paper

We begin in Sect. 7.2 by introducing the syntax and semantics of a prototypical programming language incorporating dynamically generated local storage of arbitrary type. Section 7.3 sets up the categories of games and strategies in which the semantics of this language will be given. In Sect. 7.4 we define the denotational semantics "by hand", essentially as was done in Abramsky et al. (1998).

We begin our algebraic account of this semantics in Sect. 7.5 by introducing the notion of sequoidal category and sequoidal CCC, and establishing the existence of the required structure in our categories of games. Section 7.6 explains how this may be used to define the semantics of the reference cell, making use of the additional coalgebraic properties explained above. In Sect. 7.7 we exploit this analysis to prove the soundness of the model, and extend this to computational adequacy via the instrumentation approach in Sect. 7.8.

Our final contribution is the algebraic decomposition of strategies set out in Sect. 7.9. In this section we make use of the decomposition to show that an extension of our programming language with first-class control operators may be given a *fully abstract* denotational semantics using games.

7.2 Syntax

We work with an economical language based on the call-by-value simply-typed lambda-calculus with both product and function types. To the standard syntax of this language we add constants for arithmetic and conditionals, as in PCF for example, and a single term former for generating reference cells: the term $\mathsf{ref}\, V$ evaluates to a new reference cell initialised to hold the value V.

The types of the language are standard:

$$A, B ::= \mathsf{unit} \mid \mathsf{nat} \mid A \times B \mid A \to B$$

Note that we do not introduce a dedicated type for reference cells. A cell storing values of type A will be accessed by two functions, one of type $\mathsf{unit} \to A$ for retrieving the value stored in the cell, and one of type $A \to \mathsf{unit}$ for updating the stored value. In light of this, we adopt the abbreviation

$$\mathsf{var}[A] \triangleq (\mathsf{unit} \to A) \times (A \to \mathsf{unit})$$

The terms are then given by

$$\begin{aligned} M, N, P ::= \; & x \mid MN \mid \lambda x^A.M \\ \mid \; & () \mid \langle M, N\rangle \mid \pi_0 M \mid \pi_1 M \\ \mid \; & n \mid \mathsf{succ}\, M \mid \mathsf{ifzero}\; M\; N\; P \\ \mid \; & \mathsf{ref}_A M \end{aligned}$$

where x is drawn from a countably infinite set of identifiers and n ranges over the natural numbers. Among these terms we specify the *values*

$$V ::= x \mid \lambda x^A.M \mid \langle V, V\rangle \mid n$$

(We will frequently omit the type annotations on abstractions and the ref construct to avoid notational clutter.)

An imperative variable (i.e. a value of type $\mathsf{var}[A]$) is dereferenced by "unthunking" its first projection, and assigned with a value by application of its second projection. We introduce the corresponding syntactic sugar:

$$!M \triangleq (\pi_1 M)()$$
$$M := N \triangleq (\pi_2 M)N$$

We will also make use of standard abbreviations for sequencing constructs: $\mathsf{let}\, x = M \,\mathsf{in}\, N$ abbreviates $(\lambda x.N)M$, and when M has type unit we may also write $M; N$ for the same term.

Note that we do not have any explicit syntax for recursion. In fact, the imperative features of our language make it possible to encode recursive functions by manipulation of the heap. For example, the fixed point of a higher-order function $F : (A \to B) \to (A \to B)$ is given by the term

$$\mathsf{let}\ f = \mathsf{ref}\, \lambda a.b \ \mathsf{in}\ f := \lambda a.F(!f)a; !f$$

(where b is any term of type B).

This example begins to demonstrate the great expressive power of reference cells. Their presence has a dramatic impact on the possible flows of information through a program, because they allow terms to be communicated outside of the static lexical scopes which seem to contain them. For instance, consider a term of the form

$$\lambda f.f(\lambda x.M)(\lambda y.N)$$

The term that f becomes bound to may supply arguments to $\lambda x.M$ and $\lambda y.N$, which are bound to x and y. In the lambda-calculus, there would be no way for the term N to access the term to which x is bound, because $\lambda x.M$ and $\lambda y.N$ are in distinct branches of the Böhm tree. However, in our extended language, communication via the store can make this possible: for example

$$\lambda f.f(\lambda x.v := x)(\lambda y.!v)$$

Viewed through the lens of game semantics, this removal of the barrier of lexical scoping corresponds precisely to the invalidation of the *visibility* constraint on plays which we will describe in the next section.

Type judgements, locations and configurations

Typing judgements for terms take the form

$$x_1 : A_1, \ldots, x_n : A_n \vdash M : A$$

Contexts $x_1 : A_1, \ldots, x_n : A_n$ may be empty and are identified up to permutation. Each variable may appear at most once. We let Γ, Δ range over contexts. Typing rules for the simply-typed lambda-calculus with products and constants are standard. The cell-generation term is typed as follows:

$$\frac{\Gamma \vdash M : A}{\Gamma \vdash \mathsf{ref}\, M : \mathsf{var}[A]}$$

The operational semantics is given in terms of *stores* which record the values stored in the heap. A store is a finite mapping from *locations* to values, together with their types. We write a store as a list of triples $\mathcal{S} = (l_1, V_1, A_1), \ldots (l_n, V_n, A_n)$. Each location provides two handles for accessing it: for a location l we specify two identifiers, written as $\mathsf{set}(l)$ and $\mathsf{get}(l)$, which we use to store and retrieve values in that location. We adopt the convention that these identifiers may never be λ-bound.

Accordingly, the *type* of a store as above is the context $\Gamma_{\mathcal{S}} \triangleq \mathsf{get}(l_1) : \mathsf{unit} \to A_1, \mathsf{set}(l_1) : A_1 \to \mathsf{unit}, \ldots, \mathsf{get}(l_n) : \mathsf{unit} \to A_n, \mathsf{set}(l_n) : A_n \to \mathsf{unit}$. It is well-typed if and only if each stored value is well-typed, that is, for each i the term

$$\Gamma_{\mathcal{S}} \vdash V_i : A_i$$

is well-typed. A *configuration* is a pair $(\mathcal{S}, M : T)$ of a store and a term; it is well-typed if both $\mathcal{S}$ and

$$\Gamma_{\mathcal{S}} \vdash M : T$$

are well-typed.

We write $\mathcal{S}[l \mapsto V]$ for the store obtained by adding to $\mathcal{S}$ the mapping $l \mapsto V$ (together with the type of V); this may either override an existing mapping in $\mathcal{S}$ or extend the domain of $\mathcal{S}$.

Operational semantics is given by small-step reductions on configurations $(\mathcal{S}; M : T) \to (\mathcal{S}'; M' : T)$. We typically omit the type information and parentheses for brevity.

Standard reductions for the call-by-value lambda-calculus with arithmetic carry the store along unchanged:

$$\begin{aligned}
\mathcal{S}; (\lambda x.M)V &\to \mathcal{S}; M[V/x] \\
\mathcal{S}; \mathsf{succ}\ n &\to \mathcal{S}; (n+1) \\
\mathcal{S}; \mathsf{ifzero}\ 0\ M\ N &\to \mathcal{S}; M \\
\mathcal{S}; \mathsf{ifzero}\ (n+1)\ M\ N &\to \mathcal{S}; N
\end{aligned}$$

The reductions for assignment and dereferencing make use of the store:

$$\begin{aligned}
\mathcal{S}; \mathsf{get}(l)() &\to \mathcal{S}; V && \text{where } \mathcal{S}(l) = V \\
\mathcal{S}; \mathsf{set}(l)(V) &\to \mathcal{S}[l \mapsto V : A]; ()
\end{aligned}$$

where A is the type of V.

Evaluating $\mathsf{ref}\ V$ generates a new location and initializes its contents in the store with V. This implicitly extends the context with new get and set handles, which are returned in a pair.

$$\mathcal{S}; \mathsf{ref}\, V \to \mathcal{S}[l \mapsto V : A]; \langle \mathsf{get}(l), \mathsf{set}(l\ \ )\rangle \qquad \text{where } l \text{ is fresh and } A \text{ is the type of } V$$

Reductions may be carried out within evaluation contexts. The evaluation contexts are given by

$$E ::= - \mid EN \mid (\lambda x.M)E \mid \langle E, N\rangle \mid \langle V, E\rangle \mid \pi_i E \mid \mathsf{succ}\ E \mid \mathsf{ifzero}\ E\ M\ N \mid \mathsf{ref}\ E$$

and evaluation within such a context is allowed by the rule

$$\frac{\mathcal{S}; M \to \mathcal{S}'; M'}{\mathcal{S}; E[M] \to \mathcal{S}'; E[M']}$$

7.3 Games and Strategies

The denotational semantics of our language is based on a category of games and strategies, which is ultimately based on the notion of *arena* (a kind of two-player game) introduced by Hyland and Ong (2000). Games are played by two players who play *moves*, and may associate to each move a *justification pointer* to a previously played move. Our definitions depart from the original ones of Hyland and Ong by allowing moves to be justified by previous moves of the same player. This will enable us to define the *sequoid* connective which plays a central role in our semantic analysis of the reference cell.

Definition 7.3.1 An *arena* A is a directed acyclic graph G_A, with a labelling λ_A of its nodes, partitioning it into sets of *Proponent* and *Opponent* moves.

G_A is presented as a set of nodes (moves) M_A, a set of specified source nodes $M_A^I \subseteq M_A$ (initial moves) and edge-relation $\vdash_A \subseteq M_A \times M_A \backslash M_A^I$ (enabling), and λ_A as a function from M_A into $\{O, P\}$.

An arena is *bipartite* if non-initial Opponent moves are enabled by player moves and vice-versa (note that we do not require this property of all arenas). Two key constructions on arenas are:

- Involution—$(G, \lambda)^\perp = (G, \overline{\lambda})$ (the same graph with player/Opponent labelling swapped over).
- Disjoint union of labelled graphs—$(G, \lambda) \uplus (G', \lambda') = (G \uplus G', [\lambda, \lambda'])$.

Definition 7.3.2 A *justified sequence* is a sequence equipped (via a partial endofunction on its set of prefixes) with at most one "justification pointer" from each move in the sequence to one which strictly precedes it (its justifier).

A *legal sequence* on the arena A is a finite justified sequence over M_A in which each move:

- Occurs at an odd position in the sequence if and only if it is an Opponent move.
- Has no justification pointer if it is initial, and has a justification pointer to a move which enables it in A, otherwise.

A (deterministic) strategy σ on an arena A is a non-empty, even-prefix-closed set of even-length legal sequences on A such that $sa, sb \in \sigma$ implies $a = b$.

Given some relation R between the legal sequences of A and B, a legal sequence on $A^{\perp} \uplus B$ is a copycat up to R if the projections of every even prefix to A and B are related in R. A strategy is a copycat up to R if it consists of all such copycat sequences.

Definition 7.3.3 Let $\mathcal{G}$ be the category in which objects are arenas and the morphisms from A to B are strategies on the game $A^{\perp} \uplus B$.

- The identity morphism on the arena A is the copycat strategy of the identity relation on legal sequences of A.
- The composition of morphisms is their "parallel composition with hiding" as strategies—given $\sigma : A \to B$ and $\tau : B \to C, \sigma; \tau$ consists of the legal sequences which are restrictions to $A \Rightarrow C$ of justified sequences t over the moves $M_A + M_B + M_C$ such that $t{\upharpoonright}A, B$ is in σ and $t{\upharpoonright}B, C$ is in τ.

Disjoint union of arenas equips $\mathcal{G}$ with symmetric monoidal structure: on morphisms, $\sigma \otimes \tau : A \otimes B \to C \otimes D$ is the set of even-length legal sequences for which the restrictions of each even prefix to $A \to C$ and $B \to D$ are in σ and τ respectively.

The arenas $(A \uplus B)^{\perp} \uplus C$ and $A^{\perp} \uplus (B^{\perp} \uplus C)$ are the same up to associativity of the disjoint sum, giving a natural equivalence between $\mathcal{G}(A \otimes B, C)$ and $\mathcal{G}(A, B^{\perp} \otimes C)$—i.e.

Proposition 7.3.4 *$\mathcal{G}$ is compact closed.*

We will see that the trace operator induced by compact closure plays a key role in interpreting the self-referencing nature of the store.

7.3.1 A Cartesian Closed Category of Games

$\mathcal{G}$ does not have cartesian products, in general (as it is compact closed, they would have to be biproducts). To define a cartesian closed category of games and strategies from which to construct a model of our λ-calculus based programming language, we restrict to certain *single-threaded* strategies, for which we may define an operation of copying *ad libitum*.

Definition 7.3.5 An arena A is *well-opened* if all of its initial moves are Opponent moves.

A legal sequence is *single-threaded* if it contains at most one initial Opponent move. A morphism between well-opened arenas is single-threaded if it consists of single-threaded legal sequences.

Observe that if arenas A, B and C are well-opened, then the precomposition of a multi-threaded (i.e. not necessarily single-threaded) morphism $\tau : A \to B$ with a single-threaded morphism $\sigma : B \to C$ yields a single-threaded morphism $\sigma; \tau : A \to C$.

Definition 7.3.6 Given a single-threaded morphism $\sigma : A \to B$, let $\sigma^\dagger : A \to B$ be the multi-threaded morphism consisting of legal sequences on $A^\perp \uplus B$ which are the interleaving of some $s_1, \ldots, s_n \in \sigma$.

Let $\mathcal{G}_w$ be the category in which objects are well-opened arenas, and morphisms from A to B are single-threaded morphisms from A to B.

- The single-threaded identity on A, $\mathsf{id}^w_A : A \to A$, is the restriction of the multi-threaded identity on A to single-threaded sequences.
- The composition of $\sigma : A \to B$ with $\tau : B \to C$ in $\mathcal{G}_w$ is $\sigma^\dagger; \tau$.

For any indexed family of well-opened arenas $\{B_i \mid i \in I\}$, the set of non-empty single-threaded sequences on $A^\perp \uplus \biguplus_{i \in I} B_i$ is (up to relabelling) the disjoint union over I of non-empty single threaded sequences on the $A^\perp \uplus B_i$. This yields a natural equivalence between $\mathcal{G}_w(A, \biguplus_{i \in I} B_i)$ and $\prod_{i \in I} \mathcal{G}_w(A, B_i)$—i.e. $\biguplus_{i \in I} B_i$ is the cartesian product of the B_i in $\mathcal{G}_w$.

Definition 7.3.7 Given graphs G_A, G_B, let $G_A \lhd G_B$ be the “graft” of G_B onto the root nodes (initial moves) of G_A—i.e. $(M_A + M_B, M^I_A, (\vdash_A + \vdash_B) \cup (M^I_A \times M^I_B))$.

For well-opened arenas A and B, let $A \Rightarrow B$ be the graft of $A^\perp$ onto B, $(G_B \lhd G_A, [\lambda_B, \overline{\lambda_A}])$.

There is a bijection between single threaded sequences on $(A \uplus B)^\perp \uplus C$ and on $A^\perp \uplus (B \Rightarrow C)$, obtained by adding a justification pointer from each initial move of B into the unique initial move of C. This yields a natural equivalence between $\mathcal{G}_w(A \times B, C)$ and $\mathcal{G}_w(A, B \Rightarrow C)$ making $\Rightarrow$ the internal hom—i.e.

Lemma 7.3.8 *$\mathcal{G}_w$ is cartesian closed.*

Remark 7.3.9 $(\_)^\dagger$ acts as an identity-on-objects (monoidal) functor from $(\mathcal{G}_w, \times)$ into $(\mathcal{G}, \otimes)$. Thus for each well-opened arena A, we have a commutative comonoid $(A, \delta^\dagger_A, \epsilon^\dagger_A)$, where $\delta_A : A \to A \uplus A$ is the (single-threaded) diagonal for the cartesian product in $\mathcal{G}_w$ and $\epsilon_A : A \to 1$ is the terminal map. Thus $(\_)^\dagger$ is a $1-1$ correspondence between the single-threaded strategies on $A \to B$, and those multi-threaded strategies which are *comonoid morphisms* from A to B in $\mathcal{G}$ with respect to this structure; cf. Harmer and McCusker (1999).

7.4 Denotational Semantics

We may now outline the semantics of our programming language. To interpret the call-by-value λ-calculus, we require a strong monad on the category of "pre-arenas" obtained by applying the $\mathsf{Fam}(\_)$ construction (small co-product completion) to $\mathcal{G}_w$ as in Abramsky and McCusker (1998).

Definition 7.4.1 For any category $\mathcal{C}$, $\mathsf{Fam}(\mathcal{C})$ is the category of set-indexed families of objects of $\mathcal{C}$, which has as morphisms from $\{A_i \mid i \in I\}$ to $\{B_j \mid j \in J\}$, the pairs $\langle f : I \to J, \{\phi_i : A_i \to B_{f(i)} \mid i \in I\}\rangle$ of a re-indexing function and a family of morphisms in $\mathcal{C}$. The composition of $\langle f : I \to J, \{\psi_i\}_{i \in I}\rangle$ with $\langle g : J \to K, \{\psi_j\}_{j \in J}\rangle$ is $\langle f; g, \{\phi_i; \psi_{f(i)}\}_{i \in I}\rangle$.

This has co-products, given by the disjoint union of indexed families, and if $\mathcal{C}$ is Cartesian closed then so is $\mathsf{Fam}(\mathcal{C})$, with the distributive product:

$$\{A_i \mid i \in I\} \times \{B_j \mid j \in J\} \triangleq \{A_i \times B_j \mid \langle i, j\rangle \in I \times J\}$$

and internal hom:

$$\{A_i \mid i \in I\} \Rightarrow \{B_j \mid j \in J\} = \{\Pi_{i \in I}(A_i \Rightarrow B_{f(i)}) \mid f : I \to J\}$$

We will treat $\mathcal{G}_w$ as a full subcategory of $\mathsf{Fam}(\mathcal{G}_w)$ by identifying the arena A with the singleton family containing it.

Any strong monad $\mathbf{T} : \mathsf{Fam}(\mathcal{G}_w) \to \mathsf{Fam}(\mathcal{G}_w)$ yields a model of the functional part of our language in its Kleisli category, by setting the interpretations of the ground types unit and nat to be the corresponding coproducts of the terminal object (empty arena)—i.e. types denote the objects: a context $\Gamma = x_1 : S_1, \ldots, x_n : S_n$

$$\begin{array}{ll} [\![\mathsf{unit}]\!] = \{1\} & [\![\mathsf{nat}]\!] = \{1 \mid n \in \mathbb{N}\} \\ [\![S_1 \times S_2]\!] = [\![S_1]\!] \times [\![S_2]\!] & [\![S_1 \to S_2]\!] = [\![S_1]\!] \Rightarrow \mathbf{T}[\![S_2]\!] \end{array}$$

denotes the product $[\![S_1]\!] \times \ldots \times [\![S_n]\!]$ and the term $\Gamma \vdash M : S$ denotes a morphism $[\![M : T]\!]_\Gamma : [\![\Gamma]\!] \to \mathbf{T}[\![S]\!]$.

We will give further conditions on $\mathbf{T}$ which will enable us to soundly interpret the imperative part (general references) in due course. Our leading example of such a monad, for which we will prove further adequacy and full abstraction results, corresponds to a *lifted sum* construction—for each family of arenas $\{A_i\}_{i \in I}$, we have a well-opened arena $\Sigma\{A_i \mid i \in I\}$ with a single initial Opponent move (called q for "question"), enabling distinct Proponent moves ("answers") a_i for each $i \in I$, beneath each of which is the corresponding arena A_i. We can define the structure of a strong monad on Σ directly: as the name suggests, it is a (weak) coproduct—it comes with family of injection strategies $\eta = \{\mathsf{inj}_i : A \to \Sigma_{i \in I} A_i\}$ and a co-pairing operation sending a family of strategies $\{\sigma_i : A_i \to \Sigma B \mid i \in I\}$ to $[\sigma_i \mid i \in I] :$

$\{\Sigma_i A_i \to \Sigma B\}$. The (natural) distributivity morphism $d : B \times \Sigma_{i \in I} A_i \to \Sigma_{i \in I}(B \times A_i)$ gives the monadic strength.

However, we can also specify the monadic structure of Σ via the observation that in the absence of any bracketing condition, it is equivalent to a *continuations monad* on $\mathsf{Fam}(\mathcal{G}_w)$:

Definition 7.4.2 For any object A of a cartesian closed category $\mathcal{C}$, let $\neg_A$ be the continuations functor $\_ \Rightarrow \{A\} : \mathsf{Fam}(\mathcal{C})^{op} \to \mathsf{Fam}(\mathcal{C})$. The duality (right and left adjunction) of $\neg$ and $\neg^{op}$ is enriched over $\mathsf{Fam}(\mathcal{C})$ and thus resolves a strong monad $\neg_A \neg_A : \mathsf{Fam}(\mathcal{C}) \to \mathsf{Fam}(\mathcal{C})$.

Taking our answer object to be the arena o with a single (initial) Opponent move gives us a continuations functor $\neg : \mathsf{Fam}(\mathcal{G}_w)^{op} \to \mathsf{Fam}(\mathcal{G}_w)$ (we drop the subscript) such that $\neg\neg\{A_i\}_{i \in I} = \neg \prod_{i \in I}(\neg A_i)$ is the arena $\sum\limits_{i \in I} A_i$ (up to relabelling of moves). (We will subsequently use this observation to interpret first-class continuations in the Kleisli category of Σ).

To complete the denotational semantics of our language it suffices to give the denotation of $\mathsf{ref}_S : S \to \mathtt{var}[S]$ as a morphism from $[\![S]\!]$ to $\mathbf{T}(\mathbf{T}[\![S]\!] \times ([\![S]\!] \Rightarrow \mathbf{T}1))$ in $\mathsf{Fam}(\mathcal{G}_w)$. This is derived from a multi-threaded strategy in $\mathcal{G}$ which behaves as a generic reference cell by correctly matching up requests to read and write. We will later construct this map in an algebraic fashion, using the structure of the category rather than by direct manipulation of justified sequences and strategies. Our construction will deliver a family of maps $\mathsf{cell}_{A,B} : A \times B \to A \times (A \Rightarrow B)$ that exhibit reference-cell-like behaviour, parameterised not only in the type A of the information being stored and the value initially stored, but also in the return type B of the assignment command, and in the behaviour associated to a complete assignment. Concretely, this is defined as a multi-threaded strategy which:

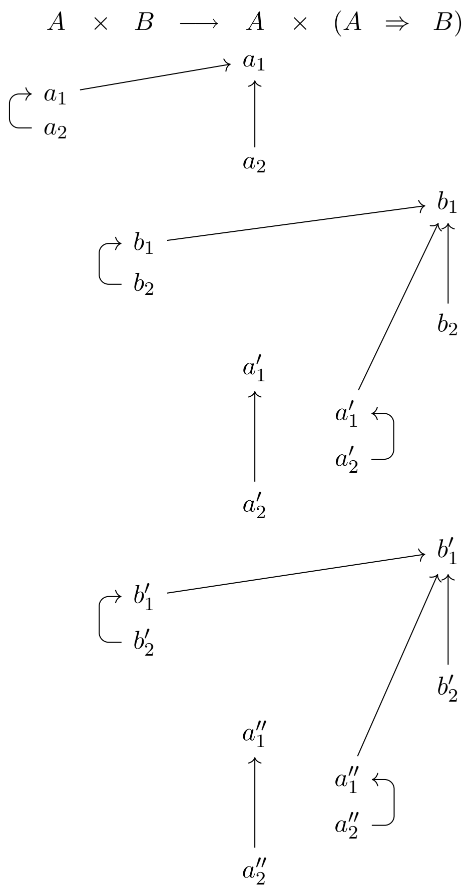
- Plays copycat between the positive and negative occurrences of B—corresponding to a generic response to a write request
- In any thread which is opened before a write request has been received, plays copycat between the occurences of A on the left and right side of $\to$—responding to a read request by returning the initial state
- In any thread which is opened after a write request has been received, plays copycat between the positive and negative occurrences of A in $A \times (A \Rightarrow B)$, adding a justification pointer from each initial move of A in $A \Rightarrow B$ to the most recent initial Opponent move in B.

A typical play in this strategy is illustrated in Fig. 7.1.

Using this strategy, we can recover the game semantics of the reference cell introduced in Abramsky et al. (1998), as follows. We assume that the monad $\mathbf{T}$ factors through the inclusion of $\mathcal{G}_w$ in $\mathsf{Fam}(\mathcal{G}_w)$ (i.e. for each family of arenas A, $\mathbf{T}A$ is a singleton family), and define the denotation of $\mathtt{ref}_A : A \to \mathtt{var}[A]$ to be (the currying of) the morphism

$$[\![A]\!] \stackrel{\eta_A \times \eta_1}{\to} \mathbf{T}[\![A]\!] \times \mathbf{T}1 \stackrel{\mathsf{cell}_{\mathbf{T}[\![A]\!],\mathbf{T}1}}{\to} \mathbf{T}[\![A]\!] \times (\mathbf{T}[\![A]\!] \Rightarrow \mathbf{T}1) \stackrel{\mathbf{T}[\![A]\!] \times (\eta_A \Rightarrow \mathbf{T}1)}{\to} [\![\mathtt{var}[A]]\!] \stackrel{\eta_{[\![\mathtt{var}[A]]\!]}}{\to} \mathbf{T}[\![\mathtt{var}[A]]\!]$$

Fig. 7.1 A typical play of the generic cell



Note that here we have specialised the return type of the assignment command to be $\mathbf{T}1$ and the behaviour of the successful write to be $\eta_{\mathbf{T}1}$. To illustrate the semantics of references we give an example play from the denotation of the term $f\langle \lambda x.v := x, \lambda y.\,!\,v\rangle$: unit in context

$$v : \mathsf{var}[\mathsf{unit} \to \mathsf{unit}],\ f : ((\mathsf{unit} \to \mathsf{unit}) \to \mathsf{unit}) \times (\mathsf{unit} \to \mathsf{unit} \to \mathsf{unit}) \to \mathsf{unit}$$

Fixing the monad $\mathbf{T}$ as the lifted sum monad, Fig. 7.2 shows a play in which the denotation of this term interacts with the cell strategy, which supplies the moves corresponding to v. We have used write, ok and read to refer to the moves corresponding to reading and writing to the cell, to distinguish from the moves corresponding to interactions with the *contents* of the cell, which remain q and a.

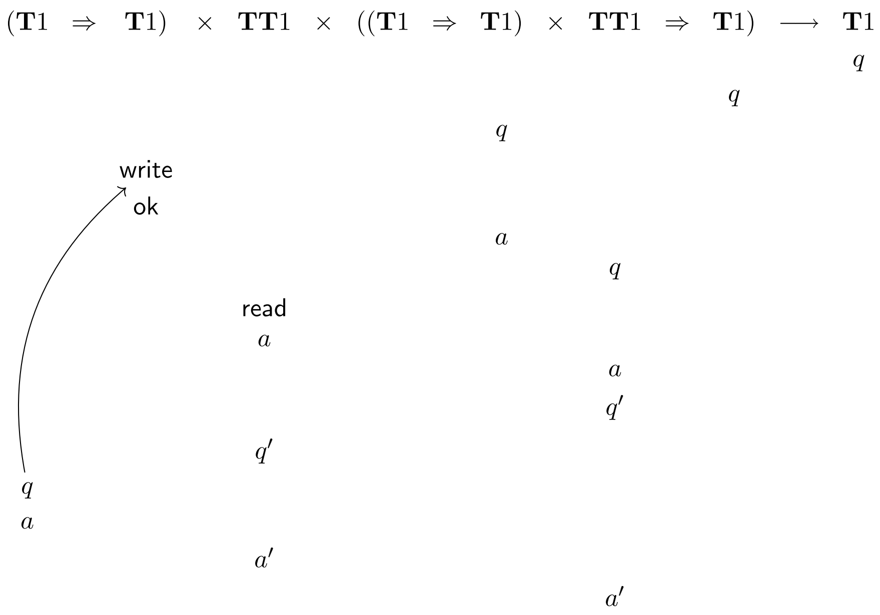


Fig. 7.2 Illustrating the use of the cell

7.5 Sequoidal Categories

We now have a complete description of our denotational semantics. However, its informal nature presents some difficulties, in particular for proving that it is sound. A more precise combinatorial description of the cell strategy would be of little help here—what we really need is to extend the categorical structure which allows us to soundly interpret higher-order functions in our model (cartesian closure, strong monads) to also capture the statefulness manifest in strategies such as the cell. We now introduce the notion of sequoidal category for this purpose.

Definition 7.5.1 An *action* of a symmetric monoidal category $\mathcal{C}$ is given by a category $\mathcal{L}$ and a strong monoidal functor from $\mathcal{C}$ into $\mathcal{L}^{\mathcal{L}}$ (the monoidal category of endofunctors on $\mathcal{L}$ with composition). In other words, a functor $\_ \oslash \_ : \mathcal{L} \times \mathcal{C} \to \mathcal{L}$ with isomorphisms $\lambda_{X,B,C} : X \oslash (B \otimes C) \cong (X \oslash B) \oslash C$ and $\kappa_X : X \oslash I \cong X$ satisfying the coherence equations for a strong monoidal functor (Janelidze & Kelly, 2001). We write $(\mathcal{C}, \otimes)$ for the canonical action of $\mathcal{C}$ on itself.

By a weak morphism between the $\mathcal{C}$-actions $(\mathcal{L}, \oslash)$ and $(\mathcal{L}', \oslash')$ we mean a functor $J : \mathcal{L} \to \mathcal{L}'$ with a natural transformation $\omega_{X,A} : JX \oslash' A \to J(X \oslash A)$ such that:

$$
\begin{tikzcd}
JX \oslash' (A \otimes B) \arrow[r, "\omega_{X,A\otimes B}"] \arrow[d, "\lambda'_{JX,A,B}"] & J(X \oslash (A \otimes B)) \arrow[dr, "J(\lambda_{X,A,B})"] & \\
(JX \oslash A) \oslash' B \arrow[r, "(\omega_{X,A} \oslash' B)"] & J(X \oslash A) \oslash' B \arrow[r, "\omega_{X \oslash A, B}"] & J((X \oslash A) \oslash B)
\end{tikzcd}
\qquad
\begin{tikzcd}
JX \oslash' I \arrow[r, "\omega_{X,I}"] \arrow[dr, "\kappa'_{JX}"'] & J(X \oslash I) \arrow[d, "J(\kappa_X)"] \\
& JX
\end{tikzcd}
$$

It is *strong* if ω is an isomorphism, and *strict* if it is the identity.

Definition 7.5.2 A *sequoidal category* is a tuple $(\mathcal{C}, \mathcal{L}, \oslash, J)$ given by a symmetric monoidal category $(\mathcal{C}, I, \otimes)$, a $\mathcal{C}$-action $(\mathcal{L}, \oslash)$ and a weak morphism of $\mathcal{C}$-actions (J, ω) from $(\mathcal{L}, \oslash)$ to $(\mathcal{C}, \otimes)$ (the canonical action of $\mathcal{C}$ on itself.)

We now identify sequoidal structure on our category of games $\mathcal{G}$.

Definition 7.5.3 Let $\mathcal{G}_l$ be the lluf subcategory of $\mathcal{G}_w$ consisting of linear morphisms, where $\tau : A \to B$ is *linear* if it satisfies the following conditions:

Totality For every non-empty strategy $\sigma : B \to C, \tau; \sigma$ is non-empty. (Concretely, this requires that for every initial move $b \in M_B^I$ there exists $a \in M_A^I$ such that $ba \in \tau$.)

Affineness $\mathsf{id}_A^w; \tau = \tau$. (Concretely, every sequence in τ contains at most one initial move from A. Note that the composition ; is that of $\mathcal{G}$.)

Composing the inclusion of $\mathcal{G}_l$ in $\mathcal{G}_w$ with $(\_)^\dagger : \mathcal{G}_w \to \mathcal{G}$ gives an identity-on-objects functor $J^\dagger : \mathcal{G}_l \to \mathcal{G}$. We define a monoidal action of $\mathcal{G}$ on $\mathcal{G}_l$ as follows:

Definition 7.5.4 Given arenas A and B, $A \oslash B \triangleq (G_A \lhd G_B, [\lambda_A, \lambda_B])$. That is, the arena $A \oslash B$ is defined in the same way as $B \Rightarrow A$ except that Player/Opponent labelling is not swapped in B, so that for well-opened B we have $B \Rightarrow A = A \oslash B^\perp$.

On morphisms, $\sigma \oslash \tau$ consists of those legal sequences on $(A \oslash B)^\perp \uplus (C \oslash D)$ such that the restrictions to $A^\perp \uplus C$ and $B^\perp \uplus D$ are in σ and τ, respectively.

7.5.1 Sequoidal Cartesian Closed Categories

We have defined sequoidal structure on $\mathcal{G}$; to interpret general references, we need to integrate it with the cartesian closed structure on the category of single-threaded strategies. Fortunately, the observation that the sequoid is dual to the internal hom for this category provides a neat way to do this.

Definition 7.5.5 A right dual for a $\mathcal{C}$-action $(\mathcal{L}, \oslash)$ is a $\mathcal{C}^{op}$-action $\multimap$ on $\mathcal{L}$ such that for each A in $\mathcal{C}$, $A \multimap \_$ is right dual to $\_ \oslash A$ in the monoidal category $\mathcal{L}^{\mathcal{L}}$—i.e. right adjoint to the endofunctor $\_ \oslash A$. It is a *commuting dual* if there is a natural isomorphism $\theta_{A,B} : A \multimap (\_ \oslash B) \cong (A \multimap \_) \oslash B$ such that $\theta_{A,A}; \epsilon_A$ and $\eta_A; \theta_{A,A}$ are the counit and unit of an adjunction $A \multimap \_ \dashv \_ \oslash A$.

Definition 7.5.6 A sequoidal closed category is given by a tuple $(\mathcal{C}, \mathcal{L}, \oslash, J)$ where

(i) $\mathcal{C}$ is a symmetric monoidal closed category.
(ii) $(\mathcal{L}, \oslash)$ is a $\mathcal{C}$-action with a commuting dual, $(\mathcal{L}, \multimap)$.
(iii) J is a strong morphism of $\mathcal{C}^{op}$ actions from $(\mathcal{L}, \multimap)$ to $(\mathcal{C}, \multimap)$.

To justify the terminology, observe that by uncurrying $J(\eta_{X,B}) : JX \to B \multimap J(X \oslash B)$ (where η is the unit of the adjunction $B \multimap \_ \dashv \_ \oslash B$) the above data determine a natural transformation $\omega_{X,B} : (JX \otimes B) \to J(X \oslash B)$ making (J, ω) a lax morphism of $\mathcal{C}$-actions from $(\mathcal{L}, \oslash)$ to $(\mathcal{C}, \otimes)$. So we can equivalently say that a sequoidal category is closed if it arises in this way.

We may remark that the sequoidal category $(\mathcal{C}, \mathcal{C}, \otimes, I)$ given by the canonical action of $\mathcal{C}$ on itself is closed if and only if $\mathcal{C}$ is compact closed (i.e. $(\_)^{\perp} \otimes \_$ is a dual $\mathcal{C}$-action to $\otimes$).

Definition 7.5.7 A sequoidal closed category is a sequoidal CCC if the symmetric monoidal structure on $\mathcal{C}$ is cartesian, and the image under J of a specified finite product structure on $\mathcal{L}$ (i.e. $J(X \times Y) = JX \times JY$).

We can now show that the sequoidal structure on $\mathcal{G}$ corresponds (via the identity-on-objects functor $(\_)^{\dagger}$) to sequoidal closed structure on $\mathcal{G}_w$. Observe that the internal hom functor $\_ \Rightarrow \_ : \mathcal{G}_w^{op} \times \mathcal{G}_w \to \mathcal{G}_w$ restricts to a functor from $\mathcal{G}_w^{op} \times \mathcal{G}_l$ to $\mathcal{G}_l$ (i.e. if τ is a linear morphism then $\sigma \Rightarrow \tau$ is a linear for any σ).

This gives an action of $\mathcal{G}_w^{op}$ on $\mathcal{G}_l$ such that the inclusion $J : \mathcal{G}_l \to \mathcal{G}_w$ is a strict map of $\mathcal{G}^{op}$ actions. As we have noted, the sequoid satisfies $B \Rightarrow A = A \oslash B^{\perp}$. These actions are therefore (commuting) duals—there are evident natural isomorphisms:

$$\frac{\mathcal{G}_l(A \oslash B, C)}{\mathcal{G}_l(A, B \Rightarrow C)} \qquad \frac{\mathcal{G}_l(B \Rightarrow A, C)}{\mathcal{G}_l(A, C \oslash B)}$$

The disjoint union of arenas is a cartesian product in G_l (which is preserved by the inclusion functor) making $(\mathcal{G}, \mathcal{G}_l, J, \oslash)$ a sequoidal CCC.

7.6 Deriving the Cell, Coalgebraically

In a sequoidal category, a program with input on A and output on B which can also read from, and write to, a *global state* of type S may be represented as a morphism from $JA \otimes JS$ into $J(B \oslash JS)$ in $\mathcal{C}$. In particular, in a sequoidal CCC we have a *reader* morphism for the global state:

$$r_S \triangleq \delta_S; \omega_S : JS \to J(S \oslash JS)$$

and the corresponding *writer* morphism:

$$w_{B,S} \triangleq \pi_l; \eta_{B,S} : JB \times JS \to J((JS \Rightarrow B) \oslash JS)$$

where $\eta_{B,S} : B \to (JS \Rightarrow B) \oslash JS$ is the counit for the adjunction $JS \Rightarrow \_ \dashv \_ \oslash JS$. In our sequoidal CCC of games, the reader morphism copies the first-opened thread of its input state as its principal output and passes the subsequently-opened threads as its output state. The writer morphism ignores its input state, and copies the argument of the principal output as its output state. (Note the similarity to the key behaviour of the cell strategy.)

Remark 7.6.1 We can relate this to the standard interpretation of global state via the side-effects monad $JS \Rightarrow (\_ \times JS)$. In a sequoidal CCC $(\mathcal{C}, \mathcal{L}, J, \oslash)$ the adjunction between $JS \Rightarrow \_$ and $\_ \oslash JS$ resolves a monad $S \Rightarrow (\_ \oslash S)$ on $\mathcal{L}$, and the pair $(J : \mathcal{L} \to \mathcal{C}, S \Rightarrow \omega_{S,A} : S \Rightarrow (JX \times A) \to JS \Rightarrow J(X \oslash A) \cong J(JS \Rightarrow X \oslash A))$ is a lax morphism of monads into the side effects monad for $\mathcal{C}$.

We relate this interpretation of global state to the local state manifested in the cell strategy by establishing that the reader morphism is the terminal object in a category of coalgebras, which we will now define. Let $\mathcal{G}_{A\oslash}$ be the category in which:

- Objects are $J(A \oslash \_)$-coalgebras—pairs (B, β) of a well-opened arena B and a *single-threaded* morphism $\beta : B \to A \oslash B$.
- A morphism from (B, β) to (C, γ) is a *multi-threaded morphism* $\sigma : B \to C$ in $\mathcal{G}$ such that the following diagram commutes (in $\mathcal{G}$):

$$\begin{tikzcd} B \arrow[r, "\beta"'] \arrow[d, "\sigma"] & A \oslash B \arrow[d, "J(A \oslash \sigma)"] \\ C \arrow[r, "\alpha"] & A \oslash C \end{tikzcd}$$

Note that this not a standard category of coalgebras, because the horizontal morphisms (coalgebras) and vertical morphisms between them come from different categories. We claim that for any well-opened game A, the coalgebra $(A, r_A : A \to A \oslash A)$ is a terminal object in this category. To establish this, we prove the following:

Lemma 7.6.2 *For well-opened arenas S, A, the map sending $\sigma : S \to A$ to $\sigma; r_A$ is an order-isomorphism from $\mathcal{G}(S, A)$ to $\mathcal{G}_w(S, A \oslash A)$.*

Proof Postcomposition with r_A corresponds to the operation on justified sequences in σ which removes left and right tagging from the moves in $A \oslash A$, and any justification pointers from moves which are now initial.

This is a (prefix) order-isomorphism from multi-threaded sequences in $S^\perp \uplus A$ to single-threaded sequences on $S^\perp \uplus A \oslash A$—its inverse adds left-tags to all moves hereditarily justified by the opening move, right-tags to all other moves in A, and justification pointers into the opening move from any right-tagged initial A-moves. The latter therefore corresponds to a map $\hat{\_} : \mathcal{G}_w(S, A \oslash A) \to \mathcal{G}(S, A)$ sucht that $\hat{\tau}; r_A = \tau$ and $\widehat{\sigma; r_A} = \sigma$. $\square$

Proposition 7.6.3 *For any well-opened game A, the coalgebra $(A, r_A : A \to A \oslash A)$ is a terminal object in $\mathcal{G}_{A\oslash}$.*

Proof In other words, for each single-threaded strategy $\sigma : S \to A \oslash S$ in $\mathcal{G}_w$, there exists a unique multi-threaded strategy $(\!|\sigma|\!) : S \to A$ in $\mathcal{G}$ such that $(\!|\sigma|\!); r_A = \sigma; (A \oslash (\!|\sigma|\!))$.

Let $(\!|\sigma|\!)$ be the $\subseteq$-least fixed point of the ($\subseteq$-continuous) map sending $g \in \mathcal{G}(S, A)$ to $\sigma; A \hat{\oslash} g$. Unfolding the fixed point, we have:

$$(\!|\sigma|\!); r_A = \sigma; (A \hat{\oslash} (\!|\sigma|\!)); r_A = \sigma; (A \oslash (\!|\sigma|\!))$$

Conversely, if $f : S \to A$ satisfies $f; r_A = \sigma; (A \oslash f)$ then it is the limit of the same chain of approximants and thus equal to $(\!|\sigma|\!)$, by the following *minimal invariance* property for $\hat{\_}$:

The identity on A is the $\subseteq$-least fixed point of the endofunction on $\mathcal{G}(A, A)$ sending f to $r_A; A \hat{\oslash} f$. $\square$

This coalgebraic property of the sequoid gives us a recipe for interpreting imperative objects with local state:

- Take a *single-threaded* morphism $\sigma : S \to A \oslash S$ in $\mathcal{G}_w$ representing an object of type A with access to global state of type S.
- (S, σ) is an object of $\mathcal{G}_{A\oslash}$ and so there is a unique morphism $(\!|\sigma|\!) : S \to A$ in $\mathcal{G}$, which reads from an initial state of type S and produces a multithreaded output of type A by passing a hidden state between its threads.

We will now apply this recipe to give a coalgebraic definition of our generalized cell strategy

$$\mathsf{cell}_{A,B} : A \times B \to A \times (A \Rightarrow B)$$

by constructing an appropriate coalgebra

$$A \times B \to (A \times (A \Rightarrow B)) \oslash (A \times B).$$

Note that in a sequoidal CCC, the functor $\_ \oslash (A \times B)$ preserves cartesian products (since it is right adjoint to $(A \times B) \Rightarrow \_$)—thus $(A \Rightarrow B \times A) \oslash (A \times B)$ is a cartesian product for $(A \Rightarrow B) \oslash (A \times B)$ and $A \oslash (A \times B)$ in $\mathcal{G}_w$. The required coalgebra can therefore be constructed by pairing the maps

$$A \times B \xrightarrow{r \times B} (A \oslash A) \times B \xrightarrow{\omega} (A \oslash A) \oslash B \cong A \oslash (A \times B)$$

and

$$A \times B \xrightarrow{\langle w, \pi_r \rangle} ((A \Rightarrow B) \oslash A) \times B \xrightarrow{\omega} ((A \Rightarrow B) \oslash A) \oslash B \cong (A \Rightarrow B) \oslash (A \times B).$$

Remark 7.6.4 In addition to the interpretation of objects with local state, we can use the coalgebraic properties of the sequoid to construct models of linear logic. The sequoidal CCCs of games in Laird (2003); Gowers and Laird (2017); Churchill et al. (2011) are based on a general construction of the *cofree commutative comonoid* on a game A from a final coalgebra for the functor $J(A \oslash \_)$ in a sequoidal category which satisfies certain further axioms.

7.7 Soundness of the Denotational Semantics

Fixing the denotation of ref as the cell strategy gives a semantics of our programming language in the Kleisli category of a strong monad $\mathbf{T}$ on $\mathsf{Fam}(\mathcal{G}_w)$, provided $\mathbf{T}$ factorizes through the "inclusion" $\{\_\} : \mathcal{G}_w \to \mathsf{Fam}(\mathcal{G}_w)$ sending each object to the corresponding singleton family. In order to prove that this interpretation is sound we need to strengthen this condition, as follows:

Definition 7.7.1 A strong monad on $\mathsf{Fam}(\mathcal{G}_w)$ with Kleisli triple $(\mathbf{T}, (\_)^*, \eta)$ is sequoidal if:

- $\mathbf{T}$ factorizes via the functor $\{J\_\} : \mathcal{G}_l \to \mathsf{Fam}(\mathcal{G}_w)$—in other words for each object A of $\mathsf{Fam}(\mathcal{G}_w)$ there is an arena $\widehat{\mathbf{T}A}$ such that $\mathbf{T}A = \{\widehat{\mathbf{T}A}\}$, and for each morphism $f : A \to \mathbf{T}B$, there is a linear morphism $\widehat{f} : \widehat{\mathbf{T}A} \to \widehat{\mathbf{T}B}$ such that $f^* : \mathbf{T}A \to \mathbf{T}B = \{\widehat{f}\}$.
- There is a linear "sequoidal strength" natural transformation $s_{A,B} : \widehat{\mathbf{T}}A \oslash B \to \widehat{\mathbf{T}}(A \times \{B\})$ which factorizes the monoidal strength $t_{A,\{B\}} : \mathbf{T}A \times \{B\} \to \mathbf{T}(A \times \{B\})$ via the natural transformation $\omega_{\mathbf{T}A,B} : \mathbf{T}A \times B \to \mathbf{T}A \oslash B$—i.e.:

$$\omega_{\mathbf{T}A,B}; s_{A,B} = t_{A,\{B\}} : \mathbf{T}A \times B \to \mathbf{T}(A \times B)$$

Lemma 7.7.2 *The lifted sum monad Σ on $\mathsf{Fam}(\mathcal{G}_w)$ has a sequoidal factorization.*

Proof IAny choice of answer object from a sequoidal closed category $\mathcal{C}$ gives a sequoidal continuations monad on $\mathsf{Fam}(\mathcal{C})$. In particular, for any morphism $f : \{A_i\}_{i \in I} \to \{B_j\}_{j \in J}$, the morphism $\neg f : \{\prod_{j \in J}(B_J \Rightarrow o) \to \{\prod_{i \in I}(A_i \Rightarrow o)\}$ is linear by definition of a sequoidal CCC—i.e. we may define $\widehat{\neg} : \mathsf{Fam}(\mathcal{C}^{op}) \to \mathsf{Fam}(\mathcal{L})$ which factorizes $\neg$ as $\widehat{\neg}; \{J\_\}$, and thus the monad $\neg\neg : \mathsf{Fam}(\mathcal{C}) \to \mathsf{Fam}(\mathcal{C})$ as $\widehat{\neg\neg}$.

By construction, the strength $t_{A,B} : \neg\neg A \times B \to \neg\neg(A \times B)$ is the currying of $\neg f$ for a morphism $f : (B \times \neg(A \times B) \to \neg A$. Therefore $\widehat{\neg} f \in \mathcal{L}(\neg\neg A, \neg(B \times \neg(A \times B))$. The natural equivalences of $\mathcal{L}(\neg\neg A, \neg(B \times \neg(A \times B)))$ to $\mathcal{L}(\neg\neg A, B \Rightarrow \neg\neg(A \times B))$ (currying) and thus to $\mathcal{L}(\neg\neg A \oslash B, \neg\neg(A \times B)$ therefore yields $s_{A,B}$ such that $t_{A,B} = s_{A,B}; \omega_{A,B}$. □

7.7.1 Semantics of Configurations

We prove that our operational semantics is sound—that if a program reduces to a value then it denotes that value—by defining a semantics for the well-typed configurations of the operational semantics and showing that each of the reduction rules respects this interpretation.

Recall that for any store $\mathcal{S} = (a_1, V_1 : S_1), \ldots, (a_n, V_n : S_n)$, the context $\Gamma_{\mathcal{S}}$ denotes (the singleton family of) the arena $\mathbf{T}[\![S_1]\!] \times ([\![S_1]\!] \Rightarrow \mathbf{T}1) \times \cdots \times \mathbf{T}[\![S_n]\!] \times ([\![S_n]\!] \Rightarrow \mathbf{T}1)$. Since the store is self-referential—each cell holds a value which may refer to any cell, including itself—it is interpreted as an endomorphism on $\Gamma_{\mathcal{S}}$.

Definition 7.7.3 The denotation of the store $\mathcal{S}$ is the multi-threaded morphism

$$[\![\mathcal{S}]\!] : [\![\Gamma_{\mathcal{S}}]\!] \to [\![\Gamma_{\mathcal{S}}]\!] = \delta^{\dagger}_{\Gamma_{\mathcal{S}}}; ([\![V_1 : S_1]\!]^{\dagger}_{\Gamma_{\mathcal{S}}}; \mathsf{vcell}_{S_1} \otimes \cdots \otimes [\![V_n : S_n]\!]^{\dagger}_{\Gamma_{\mathcal{S}}}; \mathsf{vcell}_{S_n})$$

We can then "close the loop" by taking a fixpoint for this endomorphism using the trace operator on $\mathcal{G}$ which arises from its compact closed structure. It will be useful to view this as an instance of a trace operator which may be defined on any sequoidal closed category.

Definition 7.7.4 If $(\mathcal{C}, \mathcal{L}, J, \oslash)$ is sequoidal closed then we may define the trace operator $\mathrm{tr}^{B}_{A,X} : \mathcal{C}(A \otimes B, J(X \oslash B)) \to \mathcal{C}(A, JX)$ sending $f : A \otimes B \to J(X \oslash B)$ to $\mathrm{tr}^{B}_{A,X}(f) = \Lambda(f); J(\epsilon_{B,X})$, where $\epsilon_{B,X} : B \multimap (X \oslash B) \to X$ is the co-unit of the adjunction $B \multimap \_ \dashv \_ \oslash B$.

As we have remarked, the action of a compact closed category on itself is sequoidal closed: in this case, the sequoidal trace operator coincides with the canonical trace operator on a compact closed category. So in our category of games $\mathcal{G}$, we thus have trace operators on $(\mathcal{G}, \otimes)$ and $(\mathcal{G}, \oslash)$ which correspond via the lax morphism of $\mathcal{G}$-actions ω—i.e. given $\sigma : A \otimes B \to C \otimes B$, $\mathrm{tr}^{B}_{A,C}(\sigma; \omega_{C,B})_{\oslash} = \mathrm{tr}^{B}_{A,C}(\sigma)_{\otimes}$. Concretely, $\mathrm{tr}^{B}_{A,C}(\sigma)$ is the restriction to $A \Rightarrow C$ of those sequences in σ for which the projections onto positive and negative occurrences of B are equal.

Definition 7.7.5 A configuration $\mathcal{S}; M : T$ denotes the single-threaded morphism $\mathrm{tr}^{\Gamma_{\mathcal{S}}}_{I,\Gamma_{\mathcal{S}}}([\![\mathcal{S}]\!]; \delta^{\dagger}_{[\![\Gamma_{S}]\!]}); [\![M : T]\!]_{\Gamma_{\mathcal{S}}}$.

Observe that the denotation of a configuration is independent of the ordering of the store—i.e. if θ is a permutation on n, acting on $\mathcal{S}$ then $[\![\theta(\mathcal{S}); M]\!] = [\![\theta(\mathcal{S})]\!]; [\![M]\!]_{\Gamma_{\theta(\mathcal{S})}} = [\![\mathcal{S}]\!]; [\![M]\!]$.

We now need to show that each of the rules of our operational semantics is sound with respect to this interpretation—i.e.

$$\text{If } \mathcal{S}; M : T \longrightarrow \mathcal{S}'; M' : T \text{ then } [\![\mathcal{S}; M : T]\!] = [\![\mathcal{S}'; M : T]\!]$$

For the rules which are independent of the store—i.e. those which have the form $(\mathcal{S}; M : T) \longrightarrow (\mathcal{S}; M' : T)$ where M' does not depend on $\mathcal{S}$—this follows from the fact that $[\![\Gamma_{\mathcal{S}} \vdash M : T]\!] = [\![\Gamma_{\mathcal{S}} \vdash M' : T]\!]$ by virtue of their interpretation in the Kleisli category of a strong monad on a category with coproducts. This leaves the rules for declaring, dereferencing and assigning a location. First, we note that for any evaluation context we have $[\![E[M]]\!]_\Gamma = [\![\Gamma \vdash \texttt{let}\, x = M\, \texttt{in}\, E[x]]\!]_\Gamma$ (assuming x does not occur free in $E[\_]$), and so it suffices to establish soundness for the instances of these rules in which the evaluation context is a `let`-context—i.e. $E[\_] \equiv \texttt{let}\, x\, \texttt{be}\, [\_]\, \texttt{in}\, N$. To prove soundness of the rule for declaring a new location we require the following property, derived from the axioms of a traced monoidal category:

Lemma 7.7.6 *For $f : A \to B$ and $g : A \to A$:*

$$\mathrm{tr}^{A\otimes B}_{I,A\otimes B}(\delta^\dagger_{B\otimes A}; (B \otimes A \otimes \delta_A; (f \otimes g))) = \mathrm{tr}^{A}_{I,A}(g; \delta_A)^\dagger; \delta_A; (f \otimes A)$$

Observing that if $\Gamma \vdash M : T$ then $[\![\texttt{let}\, x\, \texttt{be}\, (\texttt{ref}_T\, V)\, \texttt{in}\, M : T]\!]_\Gamma = \delta^\dagger_{[\![\Gamma_{\mathcal{S}}]\!]}$; $(([\![V]\!]^\dagger; \mathsf{vcell}_R) \otimes [\![\Gamma_{\mathcal{S}}]\!]); [\![M : T]\!]_{\texttt{var}[R],\Gamma}$, we instantiate the above identity with $[\![V]\!]^\dagger; \mathsf{vcell}_R$ for f and $[\![S]\!]$ for g, to give:

Lemma 7.7.7 $[\![\mathcal{S}; \texttt{let}\ x\ \texttt{be}\ (\texttt{ref}_R\, V)\, \texttt{in}\ M : T]\!] = [\![(a, V : R), \mathcal{S}; M [\langle \mathsf{get}(a), \mathsf{set}(a)\rangle/x]]\!]$.

To establish that the rules for reading from and writing to the store are sound using the coalgebraic construction of the cell from the reader and writer morphisms, we introduce some further useful notation to reflect the fact that actions which interact with the store have two effects—producing an observable output, and (potentially) changing its state—and we need to account for both at the same time. Accordingly we define a notion of parameterised composition in a sequoidal CCC which captures the "ouput state" of the first component by passing it forward.

Definition 7.7.8 Given morphisms $f : B \to C$ and $g : A \otimes C \to D$, define

$$\ll f \gg g : A \otimes B \to D \oslash C \triangleq (A \otimes (f; \delta_C)); (g \otimes C); \omega_{D,C}$$

By the definition of sequoidal structure, this satisfies:

Lemma 7.7.9 *Given $f_1 : B_1 \to C_1$, $f_2 : B_2 \to C_2$ and $g : A \otimes C_1 \otimes C_2 \to D$*

$$\ll f_1 \otimes f_2 \gg g = \ll f_1 \gg \ll f_2 \gg g$$

By the correspondence between trace operators on $(\mathcal{G}, \otimes)$ and $(\mathcal{G}, \oslash)$, we can express the semantics of a configuration as:

$$[\![\mathcal{S}; M : T]\!] = \mathrm{tr}^{\Gamma_{\mathcal{S}}}_{I,[\![T]\!]}(\ll [\![\mathcal{S}]\!] \gg [\![M]\!]) : I \to [\![T]\!]$$

We also relate our parameterised composition to the sequentiality of composition implicit in the notion of sequoidal monad via the following lemma:

Lemma 7.7.10 *If* $\ll [\![\mathcal{S}]\!] \gg [\![M]\!] = \ll [\![\mathcal{S}']\!] \gg [\![V]\!]$ *then* $\ll [\![S]\!] \gg [\![\mathtt{let}\ x\ \mathtt{be}\ M\ \mathtt{in}\ N]\!] = [\![\ll [\![S']\!] \gg [\![N[V/x]]\!]$.

Proof By factorizing the monoidal strength through the sequoidal strength, and observing that $\delta^\dagger_A; ((\delta_A; \omega_{A,A}) \otimes A); \omega_{A \oslash A,A} : A \to (A \oslash A) \oslash A = \delta_A; \omega_{A,A}; (A \oslash \delta_A); \lambda_{A,A,A}$, since ω is a lax map of actions and $\delta^\dagger$ is the comultiplication of a comonoid, we can show that:

$$\ll [\![\mathcal{S}]\!] \gg [\![\mathtt{let}\ x\ \mathtt{be}\ M : T\ \mathtt{in}\ N]\!]_{\Gamma_\mathcal{S}} = (\ll [\![S]\!] \gg [\![M]\!]_{[\![\Gamma_\mathcal{S}]\!]}); \\ ([\![T]\!] \oslash \delta_{[\![\Gamma_\mathcal{S}]\!]}); \lambda_{[\![T]\!],[\![\Gamma_\mathcal{S}]\!],[\![\Gamma_\mathcal{S}]\!]}; ((\omega_{[\![T]\!],[\![\Gamma_\mathcal{S}]\!]}; [\![M]\!]^*) \oslash [\![\Gamma_\mathcal{S}]\!]).$$

□

Therefore, to prove that the rules for assignment and dereferencing are sound, it suffices to show that:

$$[\![\ll \mathcal{S}[a_i \mapsto V_i] \gg [\![\mathsf{get}(a_i)\ ()]\!] = \ll [\![\mathcal{S}]\!] \gg [\![V_i]\!] \text{ and } \quad \ll [\![\mathcal{S}]\!] \gg [\![(\mathsf{set}(a_i)\ V)]\!] \\ = \ll [\![\mathcal{S}[a_i \mapsto V]]\!] \gg [\![()]\!].$$

By Lemma 7.7.9, $\ll [\![(a_1, V_1), \ldots, (a_n, V_n)]\!] \gg [\![M]\!] = \ll [\![V_1]\!]^\dagger; \mathsf{vcell} \gg \ldots \ll [\![V_n]\!]^\dagger; \mathsf{vcell} \gg [\![M]\!]$, and so (since the meaning of a configuration is independent of the ordering of the store) it suffices to prove the following two lemmas.

Lemma 7.7.11 $\ll [\![V]\!]^\dagger; \mathsf{vcell}_{[\![T]\!]} \gg [\![\mathsf{get}(l)\ ()]\!] = \ll [\![V]\!]^\dagger; \mathsf{vcell}_{[\![T]\!]} \gg [\![V]\!]$.

Proof As a morphism between the $J(A \oslash \_)$-coalgebras $\langle w_A, r_A\rangle$ and $\mathsf{r}_{\mathtt{var}[A]}$ in $\mathcal{G}_{A\oslash}$, vcell_A satisfies $\mathsf{vcell}_A; r_{\mathtt{var}[a]} = \langle w_A, r_A\rangle; (\mathtt{var}[A] \oslash \mathsf{vcell}_A)$, and thus the cell satisfies $\mathsf{vcell}_A; r_{\mathtt{var}[A]}; (\pi_l \oslash \mathtt{var}[A]) : A \to A \oslash \mathtt{var}[A] = r_A; (A \oslash \mathsf{vcell}_A)$. □

Lemma 7.7.12 *For any values* $\Gamma_\mathcal{S} \vdash U, V : A$*:*

$$\ll [\![U]\!]^\dagger_\Gamma; \mathsf{vcell}_S \gg [\![\mathsf{set}(l)\ V]\!]_\Gamma = \ll [\![V]\!]^\dagger; \mathsf{vcell}_T \gg [\![()]\!]$$

Proof By definition of the cell as a morphism in our category of coalgebras, $((\mathsf{vcell}_A; \mathsf{r}_A; (\pi_2 \oslash \mathtt{var}[A])) = \Lambda(\eta_1 \times \mathsf{vcell}); \omega_{\Sigma 1,\mathtt{var}[A]}; \theta$ where $\theta : A \Rightarrow (B \oslash \mathtt{var}[A]) \to (A \Rightarrow B) \oslash C$ is the commutation isomorphism for the sequoid. Thus $\ll [\![U]\!]^\dagger; \mathsf{vcell} \gg [\![\mathsf{set}(l)\ V]\!] = \langle [\![U]\!], [\![V]\!]\rangle^\dagger; (\eta_1 \times \mathsf{vcell}_A); \omega_{\Sigma 1,\mathtt{var}[A]} = \ll [\![V]\!]^\dagger; \mathsf{vcell} \gg [\![()]\!]$ as required □

We have thus established soundness of the the rules for dereferencing and assignment in full generality, and therefore for the whole operational semantics.

Proposition 7.7.13 *If* $(\mathcal{S}; M) \to (\mathcal{S}'; M')$ *then* $[\![\mathcal{S}; M]\!] \to [\![\mathcal{S}'; M']\!]$.

7.8 Adequacy

Our aim in this section is to prove a partial converse to the soundness result, *computational adequacy*. From now on we fix $\mathbf{T}$ to be the lifted sum monad. For this model, will show that if $[\![\mathcal{S}; M : T]\!] \neq \bot$ then $\mathcal{S}; M : T \rightarrow^* \mathcal{S}'; V : T$ for some store $\mathcal{S}'$ and value V.

The self-referential nature of the higher-order store makes it difficult to find a logical relation on which to base an adequacy proof in typical style, following Plotkin. However, because strategies represent rather directly the operational behaviour of the programs which denote them, game semantics offers a more direct approach to proving computational adequacy, which we shall describe here.

Our proof will proceed by considering an instrumented version of the semantics in which the number of reduction steps that a configuration takes in computing a value is explicitly counted (cf. "step-indexed logical relations"). We may then prove adequacy for the instrumented semantics by induction on this count, and transfer the adequacy result to the original model by establishing a correspondence between the instrumented model and the original. This approach is similar in spirit to those of Tzevelekos (2009) and Laird (2003), where instrumentation is used to establish a finite bound on the number of times that reference cells may be read in the course of computation. Our formulation differs in that it does not depend on the details of the language being considered—the reference cells are part of the problem but not an essential part of the solution.

In the denotational model, the move from the original to the instrumented semantics can be viewed in two steps. First, we replace every instance of the monad $\mathbf{T}$ with $\mathbf{T}(- \times \mathbb{N})$. This carries a monad structure with unit

$$A \xrightarrow{\eta} \mathbf{T}A \xrightarrow{\mathbf{T}\langle \mathsf{id}, 0\rangle} \mathbf{T}(A \times \mathbb{N})$$

and multiplication

$$\mathbf{T}(\mathbf{T}(A \times \mathbb{N}) \times \mathbb{N}) \xrightarrow{\mathbf{T}\tau} \mathbf{T}^2(A \times \mathbb{N} \times \mathbb{N}) \xrightarrow{\mu} \mathbf{T}(A \times \mathbb{N} \times \mathbb{N}) \xrightarrow{\mathbf{T}(\mathsf{id} \times +)} \mathbf{T}(A \times \mathbb{N}).$$

(This is an instance of Gurr's complexity monad constructor (Gurr, 1990).) Having done this, we can insert into the semantic definitions a morphism tick given by

$$\mathbf{T}(A \times \mathbb{N}) \xrightarrow{\mathbf{T}(\mathsf{id} \times \mathsf{succ})} \mathbf{T}(A \times \mathbb{N})$$

wherever we would like the instrumented model to count a reduction step. Computational adequacy for the instrumented semantics can then be established by induction on the number of ticks, and it then remains to relate the instrumented semantics to the original one.

Rather than develop the categorical machinery to express this instrumentation at a general level, we decompose the instrumented semantics via a translation of the source syntax into itself: the instrumented semantics is then the (ordinary) semantics of translated terms.

The translation comes in two parts. Values V are translated to $\overline{V}$ while general terms M are translated to M^*. The associated types are also translated:

$$\begin{aligned}\overline{\mathsf{unit}} &= \mathsf{unit}\\ \overline{\mathsf{nat}} &= \mathsf{nat}\\ \overline{A \to B} &= \overline{A} \to B^*\\ \overline{A \times B} &= \overline{A} \times \overline{B}\\ A^* &= \overline{A} \times \mathsf{nat}\end{aligned}$$

Observe that this translation does indeed implement the intended construction on monads: for any type A, $[\![\overline{A}]\!]$ may be obtained from $[\![A]\!]$ by replacing each instance of $\mathbf{T}$ with $\mathbf{T}(- \times \mathbb{N})$ as indicated above; similarly one obtains $\mathbf{T}[\![A^*]\!]$ from $\mathbf{T}[\![A]\!]$.

We extend the translation of values and types to stores: if $\mathcal{S}$ is the store $(l_1, V_1, T_1), \ldots, (l_n, V_n, T_n)$ then $\overline{\mathcal{S}} = (l_1, \overline{V_1}, \overline{T_1}), \ldots, (l_n, \overline{V_n}, \overline{T_n})$.

Terms are translated in such a way that a value

$$\Gamma_{\mathcal{S}}, x_1 : B_1, \ldots, x_m : B_m \vdash V : A$$

gives rise to a translated term

$$\Gamma_{\overline{\mathcal{S}}}, x_1 : \overline{B_1}, \ldots, x_m : \overline{B_m} \vdash \overline{V} : \overline{A}$$

and a general term in the same context gives rise to

$$\Gamma_{\overline{\mathcal{S}}}, x_1 : \overline{B_1}, \ldots, x_m : \overline{B_m} \vdash M^* : A^*$$

The translation on terms is as follows. For values we have

$$\begin{aligned}\overline{x} &= x\\ \overline{n} &= n\\ \overline{()} &= ()\\ \overline{\langle V, V'\rangle} &= \langle \overline{V}, \overline{V'}\rangle\\ \overline{\lambda x.M} &= \lambda x.M^*\\ \overline{l} &= l\end{aligned}$$

and so on. For general terms we let $V^* = \langle \overline{V}, 0\rangle$ and

$$(MN)^* = \mathsf{let}\ \langle f, n_1\rangle = M^*\ \mathsf{in}\ \mathsf{let}\ \langle a, n_2\rangle = N^*\ \mathsf{in}\ \mathsf{let}\ \langle b, n_3\rangle = f a\ \mathsf{in}\ \langle b, n_1 + n_2 + n_3 + 1\rangle$$

In the denotational semantics, this amounts to taking the usual semantics for these terms, using the the monad $\mathbf{T}(- \times \mathbb{N})$, and post-composing with the tick morphism.

The translation respects substitution:

Lemma 7.8.1 *For all values V, V' and terms M of appropriate types we have*

$$\overline{V[V'/x]} = \overline{V}[\overline{V'}/x]$$
$$M[V'/x]^* = M^*[\overline{V'}/x]$$

Proof By induction on the structure of V and M. □

The operational semantics of instrumented version of a term corresponds closely to that of the original term, as shown by this lemma.

Lemma 7.8.2 *Let $\mathcal{S}$; M be a configuration. If $\mathcal{S}, M \rightarrow^* \mathcal{S}', V$ then $\overline{\mathcal{S}}$; $M^* \rightarrow^* \overline{\mathcal{S}'}$; $\langle \overline{V}, n\rangle$ for some n; and conversely if $\overline{\mathcal{S}}$; $M^* \rightarrow^* \mathcal{S}''$; V' then $\mathcal{S}$; $M \rightarrow^* \mathcal{S}'$; V for $\mathcal{S}'$; V such that $\mathcal{S}'' = \overline{\mathcal{S}'}$ and $V' = \langle \overline{V}, n\rangle$ for some n.*

Proof Straightforward induction on the length of the reduction sequence. □

Computational adequacy for instrumented terms may now be obtained by induction on the value computed in the counter. Given a configuration $\mathcal{S}$; M, if $[\![\overline{\mathcal{S}}; M^*]\!] \neq \bot$ then $[\![\overline{\mathcal{S}}; M^*]\!]$ contains at least a two-move sequence $q \cdot (a, n)$ where n is a natural number. We refer to this number as the *measure* of the configuration. Using this measure, we may prove:

Lemma 7.8.3 *For any configuration $\mathcal{S}$; M, if $[\![\overline{\mathcal{S}}; M^*]\!] \neq \bot$ then $\mathcal{S}$; $M \rightarrow^* \mathcal{S}'$; V for some $\mathcal{S}'$; V.*

Proof By induction on the measure of $\mathcal{S}$; M. First note that if the measure is zero then by definition of the translation M^*, M must already be a value. For terms that are not values, the compositionality of the translation allows us to apply the inductive hypothesis to obtain the result. The case of application $M_1 M_2$ is illustrative. If $[\![\overline{\mathcal{S}}; (M_1 M_2)^*]\!] \neq \bot$ with measure n then we must have $[\![\overline{\mathcal{S}}; M_1^*]\!] \neq \bot$ with measure $n_1 < n$. By inductive hypothesis $\mathcal{S}$; $M_1 \rightarrow^* \mathcal{S}'$; $\lambda x.N$ and

$$[\![\overline{\mathcal{S}}; (M_1 M_2)^*]\!] = [\![\overline{\mathcal{S}'}; \mathsf{let}\ \langle a, n_2\rangle = M_2^*\ \mathsf{in}\ \mathsf{let}\ \langle b, n_3\rangle = (\lambda x.N^*)a\ \mathsf{in}\ \langle b, n_1 + n_2 + n_3 + 1\rangle]\!]$$

Hence $[\![\overline{\mathcal{S}'}; M_2^*]\!] \neq \bot$ with some measure $n_2 < n$. Again the inductive hypothesis tells us that $\mathcal{S}'$; $M_2 \rightarrow^* \mathcal{S}''$; V and then

$$[\![\overline{\mathcal{S}}; (M_1 M_2)^*]\!] = [\![\overline{\mathcal{S}''}; \mathsf{let}\ \langle b, n_3\rangle = (\lambda x.N^*)\overline{V}\ \mathsf{in}\ \langle b, n_1 + n_2 + n_3 + 1\rangle]\!].$$

Hence $[\![\overline{\mathcal{S}''}; (\lambda x.N^*)\overline{V}]\!] \neq \bot$ with measure $n_3 < n$. But this is equal to $[\![\overline{\mathcal{S}''}; (N[V/x])^*]\!]$ so we can apply the inductive hypothesis a final time to deduce that $\overline{\mathcal{S}''}$; $N[V/x] \rightarrow^* \mathcal{S}'''$; V' which completes the argument in this case. □

It just remains to show that the denotational semantics of instrumented terms corresponds in the expected way to that of the original terms; essentially we must show that the instrumentation does not interfere with the overall interpretation of configurations.

Each type A is interpreted as a family of arenas $[\![A]\!] = \{A_i \mid i \in I\}$ indexed by some set I. Note that the indexing set of $[\![A]\!]$ is equal to that of $[\![\overline{A}]\!]$ for every type A: for the base types this is trivial; it is clearly preserved by the product; and all function types have singleton index sets because the lifted sum monad always yields a singleton family. Using this observation we inductively define a function erase_A from the moves of $\mathbf{T}[\![A^*]\!]$ to those of $\mathbf{T}[\![A]\!]$ which deletes the instrumentation. We simultaneously define $\mathsf{erase}'_{A,i}$ which erases the instrumentation from each member $[\![\overline{A}]\!]_i$ of the family $[\![\overline{A}]\!]$ to give moves of $[\![A]\!]_i$. The moves of $\mathbf{T}[\![A^*]\!]$ are the initial question q, a move (i, n) for each i in the indexing set of $[\![A]\!]$, and $\mathbb{N}$-copies of the moves of $[\![\overline{A}]\!]_i$. The function erase_A takes q to q, each (i, n) to i, and applies $\mathsf{erase}'_{A,i}$ to the other moves, collapsing the $\mathbb{N}$-many copies to one. For base types erase' is the identity; for products, $\mathsf{erase}'_{A\times B,(i,j)} = \mathsf{erase}'_{A,i} + \mathsf{erase}'_{B,j}$; for function types $\mathsf{erase}'_{A\to B,*} = \Sigma_i(\mathsf{erase}'_{A,i} + \mathsf{erase}_B.)$

This function lifts to one on justified sequences, and hence to a function on sets of justified sequences. Now for any configuration $\mathcal{S}; M$ we would like to show that

$$[\![\mathcal{S}; M]\!] = \mathsf{erase}([\![\overline{\mathcal{S}}; M^*]\!])$$

from which adequacy would follow. At first sight this appears straightforward: all the constructions of the instrumented semantics reduce to those of the regular semantics when erase is applied. However, erase does not preserve composition of strategies, except in special circumstances which we now explain, and which will suffice for our purposes.

Let A, B, C and A^*, B^*, C^* be arenas. Let $\mathsf{erase}_A : M_{A^*} \to M_A$ (resp. erase_B, erase_C) be functions on moves, and let $\sigma : A \to B$, $\tau : B \to C$, $\sigma^* : A^* \to B^*$, $\tau^* : B^* \to C^*$ be strategies. Let us write $\mathsf{erase}(\sigma^*)$ for the set of justified sequences obtained by applying erase_A or erase_B as appropriate to each move in each sequence in σ^*, and similarly $\mathsf{erase}(\tau^*)$.

Say that σ^* erases to σ if $\mathsf{erase}(\sigma^*) = \sigma$, and that σ^* strongly erases to σ if moreover for all $sab \in \sigma$, $t \in \sigma^*$ with $\mathsf{erase}(t) = s$, and for any ta' such that $\mathsf{erase}(ta') = sa$, there exists $ta'b' \in \sigma^*$ such that $\mathsf{erase}(ta'b') = sab$; and make similar definitions for τ, τ^*.

Lemma 7.8.4 *If σ^* erases to σ and τ^* erases to τ then* $\mathsf{erase}(\sigma^*; \tau^*) \subseteq \sigma; \tau$.

If σ^* strongly erases to σ and τ^* strongly erases to τ then $\sigma^*; \tau^*$ strongly erases to $\sigma; \tau$.

Proof Suppose σ^* erases to σ and τ^* erases to τ.

Every sequence in $\mathsf{erase}(\sigma^*; \tau^*)$ is of the form $\mathsf{erase}(s)$ for some $s \in \sigma^*; \tau^*$. Such an s arises from a sequence u in $M_{A^*} + M_* + M_{C^*}$ with $u\restriction A^*, B^* \in \sigma^*$ and $u\restriction B^*, C^* \in \tau^*$. But then $\mathsf{erase}(u)$ is a sequence in $M_A + M_B + M_C$ for which

$\mathsf{erase}(u)\upharpoonright A, B \in \sigma$ and $\mathsf{erase}(u)\upharpoonright B, C \in \tau$. Hence $\mathsf{erase}(s) = \mathsf{erase}(u)\upharpoonright A, C \in \sigma; \tau$.

Now suppose that σ^* strongly erases to σ and τ^* strongly erases to τ. We must show that $\sigma; \tau \subseteq \mathsf{erase}(\sigma^*; \tau^*)$.

We show that for every parallel composition sequence u over $M_A + M_B + M_C$ with $u\upharpoonright A, B \in \sigma$ and $u\upharpoonright B, C \in \tau$ there is a u^* over $M_{A^*} + M_* + M_{C^*}$ such that $\mathsf{erase}(u^*) = u$, $u^*\upharpoonright A^*, B^* \in \sigma^*$ and $u^*\upharpoonright B^*, C^* \in \tau^*$, by induction on the length of u. The base case is trivial.

For the inductive step, consider a sequence $umb_1 \ldots b_k n$ where m, n are moves in A or C and the b_i are moves in B. By the inductive hypothesis we have u^* which erases to u. Suppose m is in A and $k \geq 1$ so that b_1 is a move in B. Since $umb_1\upharpoonright A, B \in \sigma$, the strong erasure property means that for any m' with $\mathsf{erase}(u^*m')\upharpoonright A^*, B^* = um\upharpoonright A, B$, we can find b_1' such that $u^*m'b_1'\upharpoonright A^*, B^* \in \sigma^*$ and $\mathsf{erase}(um'b_1') = umb_1$. Now the strong erasure property for τ gives us b_2' such that $u^*m'b_1'b_2'\upharpoonright B^*, C^* \in \tau^*$ and $\mathsf{erase}(um'b_1'b_2') = umb_1b_2$. An inductive argument delivers $um'b_1' \ldots b_k'n'$ with $um'b_1' \ldots b_k'n'\upharpoonright A^*, B^* \in \sigma^*$, $um'b_1' \ldots b_k'n'\upharpoonright B^*, C^* \in \tau^*$, and $\mathsf{erase}(um'b_1' \ldots b_k'n') = umb_1 \ldots b_k n$.

Hence $\sigma^*; \tau^*$ erases to $\sigma; \tau$. The strong erasure property follows by the same reasoning. □

We may now observe that every construction in the instrumented semantics strongly erases to the corresponding construction in the regular semantics. Since strong erasure is preserved by composition, this is enough to establish:

Lemma 7.8.5 *For any configuration $\mathcal{S}; M$, $[\![\overline{\mathcal{S}}; M^*]\!]$ strongly erases to $[\![\mathcal{S}; M]\!]$.*

Theorem 7.8.6 *For any configuration $\mathcal{S}; M$, if $[\![\mathcal{S}; M]\!] \neq \bot$ then $\mathcal{S}; M \longrightarrow^* \mathcal{S}'; V$ for some $\mathcal{S}'; V$.*

Proof By Lemma 7.8.5, if $[\![\mathcal{S}; M]\!] \neq \bot$ then $[\![\overline{\mathcal{S}}; M^*]\!] \neq \bot$. Then by Lemma 7.8.3 $\mathcal{S}; M \longrightarrow^* \mathcal{S}'; V$ for some $\mathcal{S}'; V$. □

7.9 A Fully Abstract Semantics of State and Control

We conclude by precisely locating our category of games within the intensional hierarchy of computational effects by showing that it provides a fully abstract semantics for a programming language which combines references with non-local control, in the form of *first-class continuations*. Maintaining our minimalist approach to syntax, we introduce the latter to our programming language with a constant (value) $\texttt{call\_cc}_S : ((S \to T) \to S) \to S$, corresponding to the *call-with-current-continuation* of Scheme or New Jersey SML, which captures its enclosing continuation as a function of type $S \to T$ and passes it to its argument. In combination with local references, which allow continuations to be stored, this is a powerful control construct—a program can jump backwards to any previously visited point (and forwards again), whether in or out of scope. This corresponds to the lack of constraints in our games model.

To extend our operational semantics to this expanded language, we introduce further syntax—an operation, $\mathtt{throw}_R$, taking an evaluation context $E[\bullet : S] : R$ (where R is the type of the whole program) to a value $\mathtt{throw}_R(E) : S \to T$ for arbitrary T. The small-step rules for evaluating configurations are extended as follows:

$$(\mathcal{S};\, E[\mathtt{call\_cc}_T\ V]) \longrightarrow (\mathcal{S};\, E[V\ \mathtt{throw}(E)])$$
$$(\mathcal{S};\, E[\mathtt{throw}_R(E')\ V]) \longrightarrow (\mathcal{S};\, E'[V])$$

For a program M, we write $M \Downarrow$ if $(\_;\, M) \longrightarrow^* (\mathcal{S};\, V)$ for some store $\mathcal{S}$ and value V.

7.9.1 Denotational Semantics

To extend our model with first-class continuations it suffices to interpret the constant $\mathtt{call\_cc}_S$ as a morphism from $[\![(S \to T) \to S]\!]$ to $\mathbf{T}[\![S]\!]$ in the Kleisli category $\mathsf{Fam}(\mathcal{G}_w)_{\mathbf{T}}$—a strategy which plays copycat between the single positive occurrence of $[\![S]\!]$ and whichever negative occurrence is available (and never plays any (Proponent) moves in $[\![T]\!]$). We can define this more precisely using the fact that the strong monad $\mathbf{T} = \Sigma$ on $\mathsf{Fam}(\mathcal{G}_w)$ is equivalent to the continuations monad $\neg\neg\_$ given by the "answer object" o (the one-move game). Thus there is a morphism $w : o \Rightarrow \neg\neg B$ for any B, and we may construct the denotation of $\mathtt{call\_cc}$ as follows:

$$(A \Rightarrow \neg\neg B) \Rightarrow \neg\neg A \overset{(A\Rightarrow w)\Rightarrow\neg\neg A}{\longrightarrow} \neg A \Rightarrow \neg A \Rightarrow o \cong (\neg A \times \neg A)$$
$$\Rightarrow o \overset{\langle \neg A, \neg A\rangle \Rightarrow o}{\longrightarrow} \neg A \Rightarrow o \cong \neg\neg A$$

A simple route to establish soundness and adequacy of this interpretation is by reduction to the results we have already established for the language without $\mathtt{call\_cc}$, using an alternative presentation in indirect, continuation-passing-style. This may be given as a *CPS translation* into our original language, extended with an empty type 0 (which may be interpreted in $\mathsf{Fam}(\mathcal{G}_w)$ as the empty family, which is the initial object). We further assume a divergent term $\Omega_T : T$ at each type, so that $\mathsf{abort} : 0 \to T \triangleq \lambda x.\Omega_T$ denotes the unique morphism from the initial object into $\Sigma[\![T]\!]$. The CPS translation on types is defined:

$$\overline{B} = B \quad B \in \{\mathsf{unit}, 0, \mathsf{nat}\} \qquad \overline{S \times T} = \overline{S} \times \overline{T} \qquad \overline{S \to T} = (\overline{S} \times (\overline{T} \to 0)) \to 0$$

As we did for the instrumented semantics in the previous section, we give mutually dependent translations of values $\overline{(\_)}$ and general terms $(\_)^*$. Defining $\lambda(x_1, \ldots, x_n).M \triangleq$

$\lambda y.\mathtt{let}\, x_1 = \pi_1(y) \in \ldots \mathtt{let}\, x_n = \pi_n(y) \in M$, a value $x_1 : S_1, \ldots, x_n : S_n \vdash V : T$ is translated as a general term $x_1 : \overline{S_1}, \ldots, x_n \overline{S_n} \vdash \overline{V} : \overline{T}$, where:

$$\overline{x} = x \quad \overline{\lambda x.M} = \lambda(x,k).(M^*\, k) \quad \overline{\langle V_1, V_2\rangle} = \langle \overline{V_1}, \overline{V_2}\rangle$$

$$\overline{\mathtt{ref}_T} = \lambda(x,k).k\ (\mathtt{ref}_{\overline{T}}\, x) \quad \overline{\mathtt{call\_cc}_T} = \lambda(f,k).f\ \langle \lambda(x,y).k\, x, k\rangle$$

A general term $\Gamma \vdash M : T$ is translated as a value $\overline{\Gamma} \vdash M^* : (\overline{T} \to 0) \to 0$, where:

$$V^* = \lambda k.k\ \overline{V} \quad (M\ N)^* = \lambda k.M^*\ \lambda m.(N^*\ \lambda n.m\ \langle n, k\rangle))$$

Observe that since $\Sigma 0$ is (up to relabelling) our answer object o (the one-move game) we have $[\![S \to T]\!] = [\![S]\!] \Rightarrow \Sigma[\![B]\!] \cong ([\![A]\!] \times ([\![B]\!] \Rightarrow o)) \Rightarrow o \cong [\![(S \times (T \to 0)) \to 0]\!]$. Thus for every type T, there is an isomorphism $\phi_T : [\![T]\!] \cong [\![\overline{T}]\!]$. Moreover, by correspondence of the monadic and CPS semantics, this isomorphism preserves the meaning of terms.

Proposition 7.9.1 *For every value $V : T$, $[\![\overline{V}]\!] = \phi_T([\![V]\!])$.*

To show that CPS interpretation also preserves the operational semantics of programs, we extend it to configurations. Fix an identifier $\kappa : R \to 0$ (never bound) which will represent the top-level continuation. A κ-configuration of type T is a pair $(\mathcal{S}; M)$ of a store and program in our target language such that $\kappa : R \to 0, \Gamma_{\mathcal{S}} \vdash V_i : S_i$ for each value in the store, and $\kappa : R \to 0, \Gamma_S \vdash M : T$. Under the small-step rules from Sect. 7.2, a κ-configuration either diverges, reduces to a value $(\mathcal{S}; V : T)$, or reduces to a call to the top-level continuation $(\mathcal{S}; E[\kappa])$. Subject to this further possibility, the proofs of soundness and adequacy from the previous section extend straightforwardly to κ-configurations. In particular (since there are no values of type 0).

Proposition 7.9.2 *For any κ-configuration $(\mathcal{S}; M)$ of type 0, $[\![\mathcal{S}; M]\!] \neq \bot$ if and only if $(\mathcal{S}; M) \longrightarrow^* (\mathcal{S}'; E[\kappa])$ for some $\mathcal{S}'$, E'.*

Returning to the CPS interpretation, we define the interpretation of an evaluation context $\Gamma \vdash E[\bullet : S] : R$ with respect to κ as a term $\kappa : R \to \mathbf{0}, \overline{\Gamma} \vdash \overline{E} : \overline{S} \to 0$, as follows:

$$\overline{\bullet} = \kappa \quad \overline{E[\bullet\ M]} = \lambda x.M^*\ \lambda y.x\ \langle y, \overline{E}\rangle \quad \overline{E[V\ \bullet]} = \lambda x.\overline{V}\ \langle x\ \overline{E}\rangle$$

The CPS translation of values is extended to include $\mathtt{throw}_R$:

$$\overline{\mathtt{throw}_R(E)} = \lambda(x,y).\overline{E}\ x$$

Applying this extended CPS translation to each of the values in the store gives a translation of a well-typed configuration $(\mathcal{S};\, M : R)$ as a κ-configuration $(\overline{\mathcal{S}};\, (M^* \,\kappa) : 0)$, for which the operational semantics tracks that of the source configuration in the following sense:

Proposition 7.9.3 $(\mathcal{S};\, M) \longrightarrow^* (\mathcal{S}';\, V)$ *if and only if* $(\overline{\mathcal{S}};\, M^* \,\kappa) \longrightarrow^* (\overline{\mathcal{S}'};\, \kappa\, \overline{V})$.

Proof We show that:

- $E[V]^* \,\overline{E'}$ β-reduces to $\overline{E'[E]}\,\overline{V}$.
- $(\mathcal{S};\, E[V]) \longrightarrow^* (\mathcal{S}';\, E'[V'])$ if and only if $(\overline{\mathcal{S}};\, \overline{E}\,\overline{V}) \longrightarrow^* (\overline{\mathcal{S}'};\, \overline{E'}\,\overline{V'})$.

□

Hence, by Propositions 7.9.1 and 7.9.2, we obtain soundness and adequacy of the denotational semantics.

Corollary 7.9.4 $M \Downarrow$ *if and only if* $[\![M]\!] \neq \bot$.

7.9.2 Definability and Full Abstraction

We will now show that our model is *fully abstract*. That is, two terms of the programming language denote the same strategy if and only if they are *observationally equivalent*.

Definition 7.9.5 Terms $\Gamma \vdash M, N : T$ are observationally equivalent ($M \approx^{\Gamma}_{T} N$) if for every context $C[\bullet : T]$ which binds all of the variables in Γ:

$$C[M] \Downarrow \iff C[N] \Downarrow$$

To prove full abstraction for our model we need to show that any two different strategies can be distinguished by composition with an observing strategy which is the denotation of a distinguishing context. It suffices to establish a form of definability or universality result—that every finite strategy over a type-object is the denotation of a term. One route to such a result is via *factorization* arguments showing that each finite strategy in our model may be obtained by composition of the cell and `call_cc` strategies with the denotation of a term from the purely functional part (the "innocent and well-bracketed strategies"). Here, we shall instead give a direct proof of definability via a *decomposition* argument showing how each strategy may be deconstructed in a unique way, using the algebraic structure of our category of games. The inverse to each decomposition step is made up of definable operations: abstraction and application of function variables and assignment, dereferencing and declaration of references. So we can use it to give an inductive proof of definability which captures a kind of normal form for stateful programs.

First, note that the isomorphism between types and their CPS translations means that we can reduce definability at the former to definability at the latter. The key point is that the isomorphism $S \to T \cong (S \times (T \to 0)) \to 0$ is definable in our programming language—i.e. it, and its inverse, are denoted by the values:

$$f : S \to T \vdash \lambda(x, y).y\,(f\,x) : (S \times (T \to 0)) \to 0$$
$$g : (S \times (T \to 0)) \to 0 \vdash \lambda x.\mathtt{call\_cc}_T \lambda k.\mathsf{abort}(g\,\langle x, k\rangle).$$

Applying this inductively, we obtain:

Lemma 7.9.6 *For any type T, the isomorphism $\phi_T : [\![T]\!] \cong [\![\overline{T}]\!]$ is the denotation of a value $x : T \vdash \Phi_T : \overline{T}$.*

So for any strategy $\sigma : 1 \to [\![T]\!]$, if $\sigma; \phi_T$ is the denotation of a term $V : \overline{T}$ then σ is the denotation of $\Phi_T[V/x]$.

Our decomposition for strategies at CPS types exploits the sequoidal structure we have already described, which can also be used to express further properties of our games model which will be required. The following property is essentially a form of the "linear function extensionality" axiom of Abramsky (2000).

Lemma 7.9.7 *The functor $\neg : \mathsf{Fam}(\mathcal{G})^{op} \to \mathcal{G}_l$ is fully faithful.*

Proof Given a linear strategy $\sigma : \Pi_{i\in I}(A_i \Rightarrow o) \to \Pi_{j\in J}(B_j \Rightarrow o)$:

- By totality, there is a function $f_\sigma : J \to I$ such that for every $j \in J$, $f_\sigma(j)$ is the unique value in I such that $jf_{\sigma(j)} \in \sigma$.
- For each $j \in J$, the set of justified sequences s such that $jf_\sigma(j)s \in \sigma$ is a even-branching and even-prefix closed subset of $L_{A_{f(j)} \uplus B_j^\perp}$—i.e. a morphism $\sigma_j : B_j \to A_{f(j)}$

Thus σ is the image under $\neg\_$ of a unique morphism $(f_\sigma, \{\sigma_j \mid j \in J\} : \{A_i \mid i \in J\} \to \{B_j \mid j \in J\})$. □

In fact, this result can be strengthened:

Proposition 7.9.8 $\neg : \mathsf{Fam}(\mathcal{G})^{op} \to \mathcal{G}_l$ *is an equivalence of categories.*

Proof Given an arena A, with initial move $m \in M_A^I$, let

$$A_m = .(\{a \in M_A \mid m \vdash_A^+ a\}, \{a \in M_{A_m} \mid m \vdash_A a\}, \vdash_A \cap (M_{A_m} \times M_{A_m}))$$

Then $A \cong \prod_{m\in M_A^I} (A_m \Rightarrow o)$. □

This has the consequence (which we will also use in our decomposition) that objects of the form $o \oslash B$ are *π-atomic*.

Definition 7.9.9 An object A of a category $\mathcal{C}$ is π-atomic if the hom-functor $\mathcal{C}(\_, A) : \mathcal{C}^{op} \to \mathsf{Set}$ preserves coproducts.

Lemma 7.9.10 *The one-move game o is π-atomic in $\mathcal{G}_l$*

Proof By Proposition 7.9.8, this is equivalent to showing that the terminal object is π-atomic in $\mathsf{Fam}(\mathcal{G})^{op}$—i.e. $\mathsf{Fam}(\mathcal{G})(1, \_)$ preserves coproducts, which holds by definition of the Fam construction. □

Observing that in a sequoidal CCC, $\mathcal{L}(\_, o \oslash B)$ is naturally isomorphic to $\mathcal{L}(B \Rightarrow \_, o)$ yields:

Corollary 7.9.11 *Every object of the form $o \oslash B$ is π-atomic.*

A further property that we require for our definability proof is a version of the "linearization of head occurrence" axiom of Abramsky (2000). We have already noted (Lemma 7.6.2) that postcomposition with $r_B : B \to B \oslash B$ is a natural equivalence between $\mathcal{G}(A, B)$ and $\mathcal{G}_w(A, B \oslash B)$ for any well-opened arenas A, B ("right linearization of head occurrence"). We now observe that precomposition with the reader morphism gives a similar equivalence for linear strategies ("left linearization of head-occurrence"). To express this, and our decomposition theorem, we need to consider our categories of games, and the various functors and natural transformations between them as *cpo-enriched*: that is, they have a continuous partial order on their hom-sets (inclusion of strategies), and composition of strategies (and the action of functors) is continuous with respect to this order.

Lemma 7.9.12 *The map from $\mathcal{G}_l(A \oslash A, o \oslash B)_\perp \to \mathcal{G}_w(A, o \oslash B)$ sending $\perp$ to the empty strategy, and $f : A \oslash A \to o \oslash B$ to $r_A; f : A \to o \oslash B$, is an order-isomorphism.*

Proof **Injectivity** For any $f \in \mathcal{G}_l(A \oslash A, o \oslash B)$, $r_A; f$ is non-empty and if $r_A; f = r_A; g$ then $f = \hat{\mathsf{id}}_A; r_A; f = \hat{\mathsf{id}}_A; r_A; g = g$.

Surjectivity Any single-threaded morphism $\sigma : A \to o \oslash B$ is either the empty strategy or else total. In the latter case, $\hat{id}_A; \sigma$ is linear, and satisfies $r_A; \hat{id}_A; \sigma = \sigma$. □

7.9.3 *Decomposition*

Using these properties, together with the sequoidal structure that we have already identified, we may now give a generalized decomposition for morphisms in our model. Let A and B be any well-opened arenas. By Proposition 7.9.8, A is isomorphic to $\neg\{A_i\}_{i \in I}$ and B to $\neg\{B_j\}_{j \in J}$, where $B_j = \neg\{B_{jk} \mid k \in K_j\}$ for each $j \in J$. Let us assume that these are equalities. For $j \in J$ and $l \in L_j \triangleq I + K_j$, let

$$C_{jl} = \begin{cases} B_{jl} & \text{if } l \in K_j \\ A_l & \text{otherwise} \end{cases}$$

so that $C_j \triangleq \neg\{C_{jl}\}_{l \in L_j} = A \times B_j$. Taking $\sum$ to be the lifted sum of cpos:

Proposition 7.9.13 *There is an order isomorphism* $\Phi : \mathcal{G}(A, B) \cong \prod_{j \in J} \sum_{l \in L_j} \mathcal{G}(C_j, B \times C_{jl})$.

Proof

$$\mathcal{G}(A, B) \cong \mathcal{G}_w(A, B \oslash B)$$

by right linearization (Lemma 7.6.2); next,

$$\mathcal{G}_w(A, B \oslash B) \cong \mathcal{G}_w\left(A, \prod_{j \in J}(B_j \Rightarrow o) \oslash B\right) \cong \prod_{j \in J} \mathcal{G}_w(A, (B_j \Rightarrow o) \oslash B)$$

since $\_ \oslash B$ preserves products (as a right adjoint). Then

$$\prod_{j \in J} \mathcal{G}_w(A, (B_j \Rightarrow o) \oslash B) \cong \prod_{j \in J} \mathcal{G}_w(A, B_j \Rightarrow (o \oslash B))$$

since $B_j \Rightarrow \_$ commutes with $\_ \oslash B$;

$$\prod_{j \in J} \mathcal{G}_w(A, B_j \Rightarrow (o \oslash B)) \cong \prod_{j \in J} \mathcal{G}_w(B_j \times A, o \oslash B) = \prod_{j \in J} \mathcal{G}_w(C_j, o \oslash B)$$

by uncurrying;

$$\prod_{j \in J} \mathcal{G}_w(C_j, o \oslash B) \cong \prod_{j \in J} \mathcal{G}_l(C_j \oslash C_j, o \oslash B)_\perp$$

by left linearization (Lemma 7.9.12);

$$\begin{aligned}\prod_{j \in J} \mathcal{G}_l(C_j \oslash C_j, o \oslash B)_\perp &= \prod_{j \in J} \mathcal{G}_l\left(\prod_{l \in L_j}(C_{jl} \Rightarrow o) \oslash C_j, o \oslash B\right)_\perp \\ &\cong \prod_{j \in J} \mathcal{G}_l\left(\prod_{l \in L_j}((C_{jl} \Rightarrow o) \oslash C_j), o \oslash B\right)_\perp\end{aligned}$$

since $\_ \oslash C_j$ preserves products;

$$\begin{aligned}\prod_{j \in J} \mathcal{G}_l\left(\prod_{l \in L_j}((C_{jl} \Rightarrow o) \oslash C_j), o \oslash B\right)_\perp &\cong \prod_{j \in J}\left(\coprod_{l \in L_j} \mathcal{G}_l((C_{jl} \Rightarrow o) \oslash C_j, o \oslash B)\right)_\perp \\ &= \prod_{j \in J} \sum_{l \in L_j} \mathcal{G}_l((C_{jl} \Rightarrow o) \oslash C_j, o \oslash B)\end{aligned}$$

by π-atomicity of $o \oslash B$ (Corollary 7.9.11);

$$\prod_{j \in J} \sum_{l \in L_j} \mathcal{G}_l((C_{jl} \Rightarrow o) \oslash C_j, o \oslash B) \cong \prod_{j \in J} \sum_{l \in L_j} \mathcal{G}_l(B \Rightarrow C_{jl} \Rightarrow o, C_j \Rightarrow o)$$

by adjointness of $B \Rightarrow \_$ and $(\_ \oslash B)$, and $C_j \Rightarrow \_$ and $\_ \oslash C_j$;

$$\prod_{j \in J} \sum_{l \in L_j} \mathcal{G}_l(B \Rightarrow C_{jl} \Rightarrow o, C_j \Rightarrow o) \cong \prod_{j \in J} \sum_{l \in L_j} \mathcal{G}_l((B \times C_{jl}) \Rightarrow o, C_j \Rightarrow o)$$

by uncurrying; and finally by Lemma 7.9.7,

$$\prod_{j \in J} \sum_{l \in L_j} \mathcal{G}_l((B \times C_{jl}) \Rightarrow o, C_j \Rightarrow o) \cong \prod_{j \in J} \sum_{l \in L_j} \mathcal{G}(C_j, B \times C_{jl}).$$

$\square$

Note that the content of this result is really in the proof—we have not only shown the existence of an order isomorphism between certain cpos of morphisms but that it is composed of categorical operations which interpret syntactic constructions in our programming language.

7.9.4 Definability

Using the decomposition lemma we will now prove that every finitary strategy on a finite type object is the denotation of a term of that type. Define the finite continuation types by the grammar: $T ::= \mathsf{unit} \mid T \times (T \rightarrow 0)$.

Note that by Proposition 7.9.8, the objects denoting these types are (up to isomorphism) the finite arenas.

Proposition 7.9.14 *Let $S = S_1 \times \cdots \times S_I$ and $T = T_1 \times \cdots \times T_J$ be finite continuation types. For every finitary strategy $\sigma \in \mathcal{G}([\![S]\!], [\![T]\!])$ there is a term M_σ such that*

$$\sigma; \eta_T = [\![x_1 : S_1, \ldots, x_I : S_I \vdash M : T]\!]$$

Proof By induction on the size of σ.

For each $j \leq J$ (i.e. initial move in $[\![T]\!]$) we define a value $x_1 : S_1, \ldots, x_n : S_n, a : \mathtt{var}[T] \vdash V_j : T_j$.
If $\pi_j(\Phi(\sigma)) = \bot$—i.e. σ has no response to this opening move—then let $V_j \triangleq \lambda x.\Omega_0 : T_j$. Otherwise, $\pi_j(\Phi(\sigma)) = \langle i, \sigma_j \rangle$, for some $i \leq I + K$ (where $T_j = T_{j1} \times \cdots \times T_{jK}$) representing Player's response to the opening move, and $\sigma_j : [\![S]\!] \times [\![T_j]\!] \rightarrow [\![T]\!] \times [\![R_j]\!]$, where $R_j = \begin{cases} T_{jk} & \text{if } i = I + k \\ S_i & \text{otherwise} \end{cases}$.

By induction hypothesis, σ_j is definable as a term $x_1 : S_1, \ldots, x_I : S_I, x_{I+1} : T_{j1}, \ldots, x_{K_j} : T_{jK_j} \vdash N_j : T \times R_j$. Let

$$V_j \triangleq (a := \pi_1(N_j)); \lambda(x_{I+1}, \ldots, x_{I+K_j}).x_i\ \pi_2(N_j)$$

and thus

$$M_\sigma \triangleq \mathtt{new}\, a\ \mathtt{in}\, a := \langle V_1, \ldots, V_J \rangle; !a$$

□

Using Lemma 7.9.6, we may infer that definability holds at all finite (i.e. nat-free) types. A simple way to extend definability to all types is provided by the observation that every non-atomic type denotes the colimit of a chain of embeddings between finite type-objects (by Proposition 7.9.8 every arena has this form). In other words, for every non-atomic type T, there is a chain of finite types $\langle T_i \mid i \in \omega \rangle$ and embedding/projection pairs with $[\![\overline{T}]\!]$ as its colimit. Moreover this colimit is definable—i.e. the embeddings from $[\![T_i]\!]$ into $[\![\overline{T}]\!]$, and corresponding projections, are definable in our programming language.

We define the T_i by induction over T, for which the key step is the observation that $(\mathsf{nat} \times R) \to S$ is the definable colimit of the chain of product types $\langle (R \to S)^i \mid i \in \omega$. Let $\mathsf{inj}_0 \triangleq \lambda x.\lambda(y, z).\Omega_S : \mathsf{unit} \to (\mathsf{nat} \times R) \to S$, and $\mathsf{proj}_0 \triangleq \lambda f.() : (\mathsf{nat} \times R) \to S \to \mathsf{unit}$. The terms

$$\mathsf{inj}_{i+1} \triangleq \lambda x.\lambda(y, z).\mathtt{If0}\ \ y\ \mathtt{then}\,(\pi_1 x\ z)\ \mathtt{else}\,(\mathsf{inj}_n(x)\ \langle \mathtt{pred}(y), z \rangle$$
$$\mathsf{proj}_{n+1} \triangleq f.\langle \lambda k.f\ \langle 0, \rangle, \ldots, \lambda k.f\ \langle i-1, k \rangle$$

denote an embedding-projection pair from $(R \to S)^i$ into $(\mathsf{nat} \times R) \to S$ such that $\bigcup_{i \in \omega} [\![\lambda f.\mathsf{inj}_i\ (\mathsf{proj}_i\ f)]\!] = [\![\lambda f.f]\!]$. Applying this lemma inductively to all continuation types yields:

Lemma 7.9.15 *For every type T, $[\![\overline{T}]\!]$ is the definable colimit of a chain of finite continuation types.*

For any finite strategy σ on $[\![\overline{T}]\!]$ each strategy $[\![\mathsf{proj}_i]\!](\sigma)$ is the denotation of a value $V_i : T_i$, and by cpo-enrichment of the model, $\sigma = \bigcup_{i \in \omega} [\![\mathsf{inj}_i\ V_i]\!]$. So by compactness of σ, there is some $i \in \omega$ such that $\sigma = [\![\mathsf{inj}_i\, V_i]\!]$. Thanks to the definable isomorphism (Lemma 7.9.6) between T and $\overline{T}$, we therefore have:

Proposition 7.9.16 *For every finitary, single-threaded strategy $\sigma : 1 \to T$ there is a value $V : T$ such that $[\![V]\!] = \sigma; \eta_T$.*

This allows us to establish our full abstraction result.

Theorem 7.9.17 *For any terms $\Gamma \vdash M, N : T$, $M \approx_T^\Gamma N$ if and only if $[\![M : T]\!]_\Gamma = [\![N : T]\!]_\Gamma$.*

Proof It is sufficient to consider only closed values, since $M \approx_T^{x_1:S_1,\ldots,x_n:S_n} N$ if and only if $\lambda(x_1, \ldots, x_n).M \approx_{\mathbf{S}\to T} \lambda(x_1, \ldots, x_n).N$. Computational adequacy and soundness of the operational semantics implies equational soundness ($[\![M : T]\!] = [\![N : T]\!]$ implies $M \approx_T N$) by a standard argument.

For the converse (completeness) suppose $[\![M : T]\!] \neq [\![N : T]\!]$. Without loss of generality there exists a sequence $qas \in [\![M : T]\!] : 1 \to \Sigma T$ such that $qas \notin [\![N : T]\!]$. Let $\sigma : [\![T \to \mathsf{unit}]\!]$ be the set of even prefixes of $\mathsf{run}s\mathsf{done}$, where run and done are the inital move in $\Sigma[\![\mathsf{unit}]\!]$ and its response. By Proposition 7.9.16 there is a value $V : T \to \mathsf{unit}$ which denotes $\sigma; \eta_{T\to\mathsf{unit}}$ and so $[\![V\ M]\!] \neq \bot$ and $[V\ N]\!] = \bot$. Hence by soundness and adequacy of the semantics, $V\ M \Downarrow$ but $V\ N \not\Downarrow$, and $M \not\approx_T N$ as required. $\square$

7.10 Conclusions

We have shown how the notion of sequoidal CCC underpins an algebraic analysis of the game semantics of higher-order references introduced in Abramsky et al. (1998), making use of the first author's coalgebraic approach to local state (Laird, 2019). This approach has several advantages over working directly with the game semantics as sets of traces, in particular exposing the structure required to prove the soundness of the model, and, after a little further analysis, allowing a decomposition of the model in the style of Abramsky's *Axioms for Definability* (Abramsky, 2000).

One might hope for more from such an account. Could there be other models which exhibit the same or similar structure? To date the only categories we are aware of that have all the structure needed are games models; but since our work shows that this structure is inherent to the computational phenomenon of higher-order store, it seems likely that other such models are available. For example, sequoidal closed categories can be constructed from compact closed Freyd categories (Sakayori & Tsukada, 2019), which are a categorical model for the π-calculus: this is consistent with a name-passing process interpretation of higher-order references. In another direction, this structure can be used as the basis of a type theory (Laird, 2019); it is reasonable to ask to what extent such a system can be used to perform meaningful reasoning about higher-order imperative programs.

Whatever further discoveries await, we see our work as evidence that the approach advocated by Abramsky, combining the use of particular denotational models with axiomatic analysis of those models, remains a powerful and fruitful way to understand computation.

References

Abramsky, S. (2000). Axioms for definability and full completeness. In G. Plotkin, C. Sterling, & M. Tofte (Eds.), *Essays in honour of robin milner* (pp. 55–75). MIT Press.

Abramsky, S., Honda, K., & McCusker, G. (1998). A fully abstract game semantics for general references. In *Proceedings, Thirteenth Annual IEEE Symposium on Logic in Computer Science* (pp. 334–344). IEEE Computer Society Press.

Abramsky, S., & McCusker, G. (1996). Linearity, sharing and state: A fully abstract game semantics for Idealized Algol with active expressions (extended abstract). In *Proceedings of 1996 Workshop on Linear Logic* (Vol. 3, pp. 2–14). *Electronic Notes in Theoretical Computer Science*. Elsevier.

Abramsky, S., & McCusker, G. (1997). Linearity, sharing and state: A fully abstract game semantics for Idealized Algol with active expressions. In P. W. O'Hearn, & R. D. Tennent (Eds.), *Algol-like languages* (Vol. 2, pp. 297–329). Birkhaüser.

Abramsky, S., & McCusker, G. (1998). Call-by-value games. In M. Nielsen, & W. Thomas (Eds.), *Computer Science Logic: 11th International Workshop Proceedings* (pp. 1–17). Lecture notes in computer science. Berlin: Springer.

Churchill, M., Laird, J., & McCusker, G. (2011). Imperative programs as proofs via game semantics. In *2011 IEEE 26th Annual Symposium on Logic in Computer Science, 26th Annual IEEE Symposium on Logic in Computer Science, 21–24 June 2011* (pp. 65–74). Toronto, Ontario, Canada: IEEE Computer Society Press.

Gowers, W. J., & Laird, J. (2017). Sequoidal categories and transfinite games: A coalgebraic approach to stateful objects in game semantics. In F. Bonchi, & B. König (Eds.), *7th Conference on Algebra and Coalgebra in Computer Science, CALCO 2017, 12–16 June 2017, Ljubljana, Slovenia* (Vol. 72, pp. 13:1–13:17). *LIPIcs*. Schloss Dagstuhl—Leibniz-Zentrum für Informatik.

Gurr, D. (1990). *Semantic frameworks for complexity*. Ph.D. thesis, University of Edinburgh.

Harmer, R., & McCusker, G. (1999). A fully abstract game semantics for finite nondeterminism. In *Proceedings, Fourteenth Annual IEEE Symposium on Logic in Computer Science* (pp. 422–430).

Honda, K., & Yoshida, N. (1997). Game theoretic analysis of call-by-value computation. In P. Degano, R. Gorrieri, & A. Marchietti-Spaccamela (Eds.), *Proceedings, 25th International Colloquium on Automata, Languages and Programming: ICALP '97* (Vol. 1256, pp. 225–236). Lecture notes in computer science. Berlin: Springer.

Hyland, J. M. E., & Ong, C.-H.L. (2000). On full abstraction for PCF: I, II and III. *Information and Computation, 162*(2), 285–408.

Janelidze, G., & Kelly, G. M. (2001). A note on actions of a monoidal category. *Theory and Applications of Categories [Electronic Only], 9*, 61–91.

Laird, J. (1997). Full abstraction for functional languages with control. In *Proceedings, Twelfth Annual IEEE Symposium on Logic in Computer Science* (pp. 58–67). IEEE Computer Society Press.

Laird, J. (2003). A categorical semantics of higher order store. In R. Blute, & P. Selinger (Eds.), *Proceedings, 9th Conference on Category Theory and Computer Science, CTCS 2002*. Electronic notes in theoretical computer science. Elsevier.

Laird, J. (2019). From global to local state, coalgebraically and compositionally. In B. König (Ed.), *Proceedings of the Thirty-Fifth Conference on the Mathematical Foundations of Programming Semantics, MFPS 2019, London, UK, 4–7 June 2019* (Vol. 347, pp. 203–222). *Electronic Notes in Theoretical Computer Science*. Elsevier.

Melliès, P. A., & Tabareau, N. (2010). Resource modalities in tensor logic. *Annals of Pure and Applied Logic, 161*(5), 632–653. (The Third workshop on Games for Logic and Programming Languages (GaLoP)).

Olivier, L. (2002). *Étude de la polarisation en logique*. Theses, Université de la Méditerranée—Aix-Marseille II March.

Pitts, A. M. (1996). Relational properties of domains. *Information and Computation, 127*, 66–90.

Plotkin, G. (1977). LCF considered as a programming language. *Theoretical Computer Science, 5*, 223–255.

Reddy, U. S. (1993). A linear logic model of state. Technical report FP-1993-3, Department of Computer Science, University of Glasgow.

Sakayori, K., & Tsukada, T. (2019). A categorical model of an **i/o**-typed π-calculus. In L. Caires (Ed.), *Programming languages and systems* (pp. 640–667), Cham: Springer International Publishing.

Tzevelekos, N. (2009). Full abstraction for nominal general references. *Logical Methods in Computer Science, 5*(3).

Wadler, P. (1990). Linear types can change the world! In M. Broy, & C. Jones (Edis.), *Programming concepts and methods*. *IFIP TC2 Working Conference on Programming Concepts and Methods*. North-Holland, Sea of Galilee, Israel.

Chapter 8
Deconstructing General References via Game Semantics

Andrzej S. Murawski and Nikos Tzevelekos

Abstract We investigate the game semantics of general references through the fully abstract game model of Abramsky, Honda and McCusker (AHM), which demonstrated that the visibility condition in games corresponds to the extra expressivity afforded by higher-order references with respect to integer references. First, we prove a stronger version of the visible factorisation result from AHM, by decomposing any strategy into a visible one and a single strategy corresponding to a reference cell of type $\mathsf{unit} \to \mathsf{unit}$ (AHM accounted only for finite strategies and its result involved unboundedly many cells). We show that the strengthened version of the theorem implies universality of the model and, consequently, we can rely upon it to provide semantic proofs of program transformation results. In particular, one can prove that any program with general references is equivalent to a purely functional program augmented with a single $\mathsf{unit} \to \mathsf{unit}$ reference cell and a single integer cell. We also propose a syntactic method of achieving such a transformation. Finally, we provide a type-theoretic characterisation of terms in which the use of general references can be simulated with an integer reference cell or through purely functional computation, without any changes to the underlying types.

Keywords Game semantics · General references · Factorisation · Universality · Program transformation

Tribute The results presented in this paper build upon two seminal papers by Samson: the so-called AJM paper (Abramsky et al., 2000), recognised with the Church Award in 2017, and the AHM paper (Abramsky et al., 1998), which received the Test-of-

---

This is an extended and revised version of Murawski and Tzevelekos (2013), which was presented at the 16th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS 2013).

A. S. Murawski (✉)
University of Oxford, Oxford, UK
e-mail: andrzej.murawski@cs.ox.ac.uk

N. Tzevelekos
Queen Mary University of London, London, UK
e-mail: nikos@tzevelekos.org

A. Palmigiano and M. Sadrzadeh (eds.), *Samson Abramsky on Logic and Structure in Computer Science and Beyond*, Outstanding Contributions to Logic 25,
https://doi.org/10.1007/978-3-031-24117-8_8

Time Award at LICS 2018. We carry out our investigations inside the AHM model, where references are modelled as reader-writer pairs. Although this way of modelling references is often referred to as bad, we show that bad variables are not so bad after all. In this paper, they will save the day and emerge as a useful tool for staging program transformation. A second theme of the paper is universality. The AJM model of PCF was shown to be universal in order to highlight its accurracy and domain-theoretic credentials. We extend these results to general references and apply them to deduce the theoretical existence of program transformations. Bad variables are then used to show how one can actually carry them out.

8.1 Introduction

In computer science, references are a programming idiom that allows the programmer to manipulate objects in computer memory. The referenced content can be accessed (dereferenced) or overwritten (updated). The most common sort of reference is that to a ground-type value, such as an integer. However, most modern programming languages allow more complicated values to be referenced. For example, the language ML features general references, where memory locations can contain values of any type, in particular of function type. Higher-order references are a very expressive construct. Among others, they can be used to simulate recursion, objects (Bruce et al., 1999) and aspects (Sanjabi & Ong, 2007). In this paper we investigate higher-order references taking inspiration from their game-semantic model (Abramsky et al., 1998). In particular, we shall provide both semantic and syntactic accounts of how they can be decomposed, which highlight the fact that their full expressive power is already present in their simplest instance, references of type $\mathsf{ref}(\mathsf{unit} \rightarrow \mathsf{unit})$. Although in general the inclusion of higher-order references strictly increases expressivity, we also consider the question whether there are circumstances, delineated by types, where there is no such increase. Finally, we shall also answer the question when references in general, integer-valued and higher-order ones, can be replaced altogether by purely functional computation over the same types.

Game semantics (Abramsky et al., 2000; Hyland & Ong, 2000) is a semantic theory that interprets computation as an exchange of moves between two players: O (environment) and P (program). In the Hyland-Ong style of playing (Hyland & Ong, 2000), moves are equipped with pointers to moves made earlier in the game, giving rise to plays like the one shown below.

$$o_1 \; p_1 \; o_2 \; p_2 \; o_3 \; p_4$$

Existing literature on modelling integer (Abramsky & McCusker, 1997) and higher-order references (Abramsky et al., 1998) showed that the expressive gap between the two paradigms can be captured by a property called *visibility*, which restricts the range of targets (earlier moves) for pointers: moves by P can only point at moves from a restricted fragment of the history, called the *view*. For example, the last move in the play above will violate visibility, because before it is played, the view turns out to be $o_1 \frown p_1 \frown o_3$. Intuitively, the visibility constraint captures the intuition that, without higher-order references, the set of values available to a program is limited and changes dynamically (as captured by the notion of view). In particular, function values that are available at one point, cannot be taken for granted later during the course of computation. In contrast, in presence of higher-order references, such values can be recorded and reused at will. Hence, the visibility condition needs to be relaxed for modelling higher-order references.

A fully abstract game model of an ML-like language with references was presented in Abramsky et al. (1998) and founded on plays that need not obey the visibility condition. As part of the full abstraction argument, the authors showed how to decompose every *finite* strategy into a finite strategy satisfying visibility and *several* strategies corresponding to reference cells of type $\mathsf{unit} \to \mathsf{unit}$ (Proposition 5 in Abramsky et al. (1998)).

As our first contribution, we sharpen the result to *arbitrary* strategies as well as showing that *one* strategy corresponding to a ($\mathsf{unit} \to \mathsf{unit}$)-valued memory cell is sufficient (Theorem 8.14). This brings two benefits. On the theoretical side, one can show universality: any recursively presentable strategy corresponds to a program with higher-order references. Secondly, the refined factorisation result can be applied to denotations of arbitrary programs to yield a powerful expressivity result: any program with higher-order references is equivalent to one of the shape $\mathsf{let}\, u = \mathsf{ref}(\lambda x^{\mathsf{unit}}.\Omega_{\mathsf{unit}})\ \mathsf{in}\ M$, where the ref-constructor in M is restricted to integers (Theorem 8.18).

Our first proof of the result is purely semantic and relies on recursion theory. Consequently, it does not offer much insight into how to transform the use of higher-order references into uses of a ($\mathsf{unit} \to \mathsf{unit}$)-reference cell. Motivated by this, we try to identify semantics-preserving program transformations that will allow us to reprove the same result through syntactic means. The key element of our approach is the bad-variable constructor mkvar, which enables one to create terms of reference types with non-standard behaviour. Although the translation introduces extra occurrences of mkvar, we show how to eliminate them under certain conditions, namely, when the types associated with its free variables and the type of the term do not contain reference types. Note that this allows for arbitrarily complex private uses of general references inside terms as long as the references are not communicated through the program's type interface. From this perspective, mkvar emerges as a useful device for staging program transformation. Altogether our transformations yield an alternative syntactic proof of Theorem 8.18.

In the remainder of the paper, we give a characterization of typing judgements, where term behaviour can be faithfully simulated using integer storage alone. Here is a representative selection of types in typing judgments that turn out to guarantee this property.

$$\ldots, \mathsf{int} \to \cdots \to \mathsf{int}, \ldots \vdash M : \mathsf{int},$$
$$\ldots, (\mathsf{int} \to \cdots \to \mathsf{int}) \to \mathsf{int}, \ldots \vdash M : \mathsf{int} \to \cdots \to \mathsf{int}$$

By highlighting the shape of types in the context, we mean to say that all free identifiers should have types of that form or simpler ones. These are the typing judgments over which there is no distinction in expressive power between integer and higher-order references.

Finally, we show that, as long as terms of the form

$$\ldots, x : \Theta_1, \ldots \vdash M : \beta$$

are considered, where $\beta ::= \mathsf{unit} \mid \mathsf{int}$ and $\Theta_1 ::= \beta \mid \mathsf{ref}(\beta) \mid \beta \to \Theta_1$, the use of higher-order references can be replaced with purely functional computation. That is to say, references do not contribute any expressive power. The last two results are obtained in a semantic way, by referring to game models and associated compositionality and universality results.

8.2 Syntax of the Language

We shall rely on the programming language $\mathcal{L}$ with general references introduced in Abramsky et al. (1998). Its types θ are generated from unit and int using the $\to$ and ref type constructors, as shown below.

$$\theta ::= \quad \mathsf{unit} \quad \mid \quad \mathsf{int} \quad \mid \quad \mathsf{ref}(\theta) \quad \mid \quad \theta \to \theta$$

The syntax is reproduced in Fig. 8.1, where $\oplus$ is meant to cover standard arithmetic operations.

The operational semantics of the language is given in Fig. 8.2. It relies on a countable set L of typed locations. The values of the language are then the locations themselves, (), i, λ-abstractions and $\mathsf{mkvar}(V_1, V_2)$, where V_1, V_2 must be values. The big-step reduction rules have the form $s, M \Downarrow s', V$, where s, s' are stores (par-

$$\frac{}{\Gamma \vdash () : \mathsf{unit}} \qquad \frac{i \in \mathbb{Z}}{\Gamma \vdash i : \mathsf{int}} \qquad \frac{(x : \theta) \in \Gamma}{\Gamma \vdash x : \theta} \qquad \frac{\Gamma \vdash M_1 : \mathsf{int} \quad \Gamma \vdash M_2 : \mathsf{int}}{\Gamma \vdash M_1 \oplus M_2 : \mathsf{int}}$$

$$\frac{\Gamma \vdash M : \mathsf{int} \quad \Gamma \vdash N_0 : \theta \quad \Gamma \vdash N_1 : \theta}{\Gamma \vdash \mathsf{if}\ M\ \mathsf{then}\ N_1\ \mathsf{else}\ N_0 : \theta} \qquad \frac{\Gamma \vdash M : \theta}{\Gamma \vdash \mathsf{ref}(M) : \mathsf{ref}(\theta)} \qquad \frac{\Gamma \vdash M : \mathsf{ref}(\theta)}{\Gamma \vdash !M : \theta}$$

$$\frac{\Gamma \vdash M : \mathsf{ref}(\theta) \quad \Gamma \vdash N : \theta}{\Gamma \vdash M := N : \mathsf{unit}} \qquad \frac{\Gamma \vdash M : \mathsf{unit} \to \theta \quad \Gamma \vdash N : \theta \to \mathsf{unit}}{\Gamma \vdash \mathsf{mkvar}(M, N) : \mathsf{ref}(\theta)}$$

$$\frac{\Gamma, x : \theta \vdash M : \theta'}{\Gamma \vdash \lambda x^{\theta}.M : \theta \to \theta'} \qquad \frac{\Gamma \vdash M : \theta \to \theta' \quad \Gamma \vdash N : \theta}{\Gamma \vdash MN : \theta'} \qquad \frac{\Gamma \vdash M : (\theta \to \theta') \to (\theta \to \theta')}{\Gamma \vdash \mathsf{Y}(M) : \theta \to \theta'}$$

Fig. 8.1 Typing judgments of $\mathcal{L}$

$$\frac{V \text{ is a value}}{s, V \Downarrow s, V} \qquad \frac{M \Downarrow 0 \quad N_0 \Downarrow V}{\mathsf{if}\ M\ \mathsf{then}\ N_1\ \mathsf{else}\ N_0 \Downarrow V} \qquad \frac{i \neq 0 \quad M \Downarrow i \quad N_1 \Downarrow V}{\mathsf{if}\ M\ \mathsf{then}\ N_1\ \mathsf{else}\ N_0 \Downarrow V}$$

$$\frac{M_1 \Downarrow i_1 \quad M_2 \Downarrow i_2}{M_1 \oplus M_2 \Downarrow i_1 \oplus i_2} \qquad \frac{M \Downarrow \lambda x.M' \quad N \Downarrow V' \quad M'[V'/x] \Downarrow V}{MN \Downarrow V}$$

$$\frac{s, M \Downarrow s', \ell \quad s'(\ell) = V}{s, !M \Downarrow s', V} \qquad \frac{s, M \Downarrow s', \ell \quad s', N \Downarrow s'', V}{s, M := N \Downarrow s''(\ell \mapsto V), ()}$$

$$\frac{M \Downarrow \mathsf{mkvar}(V_1, V_2) \quad V_1() \Downarrow V}{!M \Downarrow V} \qquad \frac{M \Downarrow \mathsf{mkvar}(V_1, V_2) \quad N \Downarrow V \quad V_2\, V \Downarrow ()}{M := N \Downarrow ()}$$

$$\frac{M \Downarrow V_1 \quad N \Downarrow V_2}{\mathsf{mkvar}(M, N) \Downarrow \mathsf{mkvar}(V_1, V_2)} \qquad \frac{s, M \Downarrow s', V \quad \ell \notin \mathsf{dom}(s')}{s, \mathsf{ref}(M) \Downarrow s' \cup (\ell \mapsto V), \ell}$$

$$\frac{M \Downarrow \lambda x.M' \quad N \Downarrow V' \quad M'[V'/x] \Downarrow V}{MN \Downarrow V} \qquad \frac{M \Downarrow V}{\mathsf{Y}(M) \Downarrow \lambda x^{\theta}.(V(\mathsf{Y}(V)))x}$$

Fig. 8.2 Operational semantics rules of $\mathcal{L}$

tial functions from L to the set of values) and V is a value. Most of the rules take the form

$$\frac{M_1 \Downarrow V_1 \quad M_2 \Downarrow V_2 \quad \cdots \quad M_n \Downarrow V_n}{M \Downarrow V}$$

which is meant to abbreviate

$$\frac{s_1, M_1 \Downarrow s_2, V_1 \quad s_2, M_2 \Downarrow s_3, V_2 \quad \cdots \quad s_n, M_n \Downarrow s_{n+1}, V_n}{s_1, M_1 \Downarrow s_{n+1}, V}.$$

In particular, this means that the ordering of the hypotheses is significant. Given a closed term $\vdash M : \theta$ we write $M \Downarrow$ if there exist s', V such that $\emptyset, M \Downarrow s', V$.

Definition 8.1 We shall say that two terms $\Gamma \vdash M_1 : \theta$ and $\Gamma \vdash M_2 : \theta$ are contextually equivalent (written $\Gamma \vdash M_1 \cong M_2 : \theta$) if, for any context $C[-]$ such that $C[M_1]$, $C[M_2]$ are closed, we have $C[M_1] \Downarrow$ if and only if $C[M_2] \Downarrow$.

Remark 8.2 $\mathcal{L}$ features the "bad-reference" constructor mkvar in the style of Reynolds (1981). This makes it possible to construct objects of reference types from arbitrary read and write methods. In general this strengthens the discriminating power of contexts, as terms of ref-type can exhibit non-standard behaviour. However, it can be shown that when there are no ref-types in Γ or θ, this extension is inconsequential. At the technical level, this is due to the fact that the corresponding definability argument (Abramsky et al., 1998) need not rely on mkvar then. Our paper also includes a syntactic argument.

Remark 8.3 $\mathcal{L}$ does not feature reference-equality testing as a primitive, as in general it would not make sense in a setting with bad references. Still, it is possible to construct a term that can tell two different locations apart by writing different values to them and testing their content. This is of course conditional on the existence of such values and our ability to distinguish them. In our setting, this method will be applicable to all types $\mathsf{ref}(\theta)$ except when $\theta{=}\mathsf{unit}$.

Remark 8.4 In earlier work, we considered a language called RefML (Murawski & Tzevelekos, 2011) with general references and equality testing for locations, in which bad references could not be created. The above comments imply that the respective notions of contextual equivalence induced by $\mathcal{L}$ and RefML coincide on mkvar-free $\mathcal{L}$-terms $\Gamma \vdash M : \theta$ such that there are no ref-types in Γ or θ. Similarly, one can also say that they converge for RefML-terms $\Gamma \vdash M : \theta$ such that Γ, θ do not contain ref-types and M does not use equality testing for references of type $\mathsf{ref}(\mathsf{unit})$.

We will now define a number of auxiliary terms that will turn out useful in subsequent arguments. As usual, $\mathsf{let}\, x = M \,\mathsf{in}\, N$ stands for $(\lambda x.N)M$. If x does not occur in N, we may also write $M; N$. We also rely on abbreviated notation for nested let's, e.g. $\mathsf{let}\, x, y = M_x, M_y \,\mathsf{in}\, N$ stands for $\mathsf{let}\, x = M_x \,\mathsf{in}\,\mathsf{let}\, y = M_y \,\mathsf{in}\, N$. We shall write Ω_θ for the divergent term $\mathsf{Y}(\lambda f^{\mathsf{unit}\to\theta}.f)\,()$. Also, for any type θ, we define a term $\vdash \mathsf{new}_\theta : \mathsf{ref}(\theta)$, which creates a suitably initialised reference cell.

$$\begin{aligned} \mathsf{new}_{\mathsf{unit}} &\equiv \mathsf{ref}(()) & \mathsf{new}_{\mathsf{ref}(\theta)} &\equiv \mathsf{ref}(\mathsf{mkvar}(\lambda x^{\mathsf{unit}}.\,\Omega_\theta,\, \lambda x^{\theta}.\,\Omega_{\mathsf{unit}})) \\ \mathsf{new}_{\mathsf{int}} &\equiv \mathsf{ref}(0) & \mathsf{new}_{\theta\to\theta'} &\equiv \mathsf{ref}(\lambda x^{\theta}.\Omega_{\theta'}) \end{aligned}$$

8.3 Game Model

The following arguments are couched in the game model of general references due to Abramsky, Honda and McCusker (1998). We use a more direct, yet equivalent, presentation due to Honda and Yoshida (1999).

Definition 8.5 An ***arena*** $A = (M_A, I_A, \vdash_A, \lambda_A)$ is given by

- a set M_A of moves, and a subset $I_A \subseteq M_A$ of *initial* moves,
- a justification relation $\vdash_A \subseteq M_A \times (M_A \setminus I_A)$, and
- a labelling function $\lambda_A : M_A \to \{O, P\} \times \{Q, A\}$

such that $\lambda_A(I_A) = \{PA\}$. Additionally, whenever $m' \vdash_A m$, we have $(\pi_1\lambda_A)(m) \neq (\pi_1\lambda_A)(m')$, and $(\pi_2\lambda_A)(m') = A$ implies $(\pi_2\lambda_A)(m) = Q$, where we write π_1, π_2 for the first and second projections respectively.

The role of λ_A is to label moves as *Opponent* or *Proponent* moves and as *Questions* or *Answers*. We typically write moves as $m, n, \ldots$, or $o, p, q, a, q_P, q_O, \ldots$ when we want to be specific about their kind.

The simplest arena is $0 = (\emptyset, \emptyset, \emptyset, \emptyset)$. Other "flat" arenas are 1 and $\mathbb{Z}$, defined by $M_1 = I_1 = \{\star\}$, $M_{\mathbb{Z}} = I_{\mathbb{Z}} = \mathbb{Z}$. Moreover, given arenas A and B, we can form new arenas $A \otimes B$ and $A \Rightarrow B$ as follows, where $\tilde{I_A}$ stands for $M_A \setminus I_A$, the OP-complement of λ_A is written as $\bar{\lambda}_A$, and i_A, i_B range over initial moves in the respective arenas. We write $\restriction X$ to restrict function domains to X or to restrict relations to X.

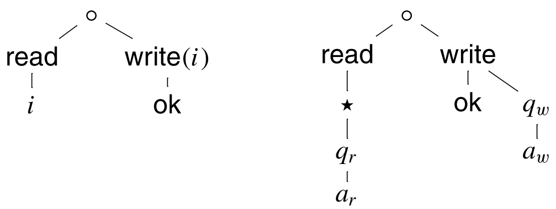
$$\begin{aligned}
M_{A\Rightarrow B} &= \{\star\} \uplus I_A \uplus \tilde{I_A} \uplus M_B \\
I_{A\Rightarrow B} &= \{\star\} \\
\lambda_{A\Rightarrow B} &= [(\star, PA), (i_A, OQ), \bar{\lambda}_A \restriction \tilde{I_A}, \lambda_B] \\
\vdash_{A\Rightarrow B} &= \{(\star, i_A), (i_A, i_B)\} \cup \vdash_A \cup \vdash_B
\end{aligned}$$

$$\begin{aligned}
M_{A\otimes B} &= (I_A \times I_B) \uplus \tilde{I_A} \uplus \tilde{I_B} \\
I_{A\otimes B} &= I_A \times I_B \\
\lambda_{A\otimes B} &= [((i_A, i_B), PA), \lambda_A \restriction \tilde{I_A}, \lambda_B \restriction \tilde{I_B}] \\
\vdash_{A\otimes B} &= (\vdash_A \restriction \tilde{I_A}^{\,2}) \cup (\vdash_B \restriction \tilde{I_B}^{\,2}) \cup \{((i_A, i_B), m) \mid i_A \vdash_A m \vee i_B \vdash_B m\}
\end{aligned}$$

Types of $\mathcal{L}$ can now be interpreted with arenas in the following way.

$$\begin{array}{rcl}
[\![\mathsf{unit}]\!] = 1 & & [\![\mathsf{ref}(\theta)]\!] = (1 \Rightarrow [\![\theta]\!]) \otimes ([\![\theta]\!] \Rightarrow 1) \\
[\![\mathsf{int}]\!] = \mathbb{Z} & & [\![\theta_1 \to \theta_2]\!] = [\![\theta_1]\!] \Rightarrow [\![\theta_2]\!]
\end{array}$$

Example 8.6 $[\![\mathsf{ref}(\mathsf{int})]\!]$ and $[\![\mathsf{ref}(\mathsf{unit} \to \mathsf{unit})]\!]$ have the following respective shapes.



Although arenas model types, the actual games will be played in ***prearenas***, which are defined in the same way as arenas with the exception that initial moves must be O-questions. Given arenas A and B, we can construct the prearena $A \to B$ by setting:

$$\begin{array}{ll}
M_{A\to B} = M_A \uplus M_B & \lambda_{A\to B} = [(i_A, OQ) \cup (\bar{\lambda}_A \restriction \tilde{I_A}),\ \lambda_B] \\
I_{A\to B} = I_A & \vdash_{A\to B} = \{(i_A, i_B)\} \cup \vdash_A \cup \vdash_B\ .
\end{array}$$

A ***justified sequence*** in a prearena A is a finite sequence s of moves of A equipped with pointers subject to the following condition: the first move is initial and has no pointer, but all other moves m have a pointer to an earlier occurrence of a move m' such that $m' \vdash_A m$. We then say that m' *justifies* m. If m is an answer, we also say that m *answers* m'. Given a justified sequence, the last unanswered question will be called *pending*.

Definition 8.7 A ***play*** in A is a justified sequence satisfying alternation (players take turns) and well-bracketing (whenever a player plays an answer, it must answer the current pending question). A ***strategy*** in a prearena A is a non-empty subset σ of even-length plays in A that is closed under the operation of taking even-length prefixes and satisfies determinacy: if $sp_1, sp_2 \in \sigma$ then $sp_1 = sp_2$.

Example 8.8 $\mathsf{cell}_{\mathsf{int}} : 1 \to [\![\mathsf{ref}(\mathsf{int})]\!]$ answers the initial question with $\circ$. Whenever O plays $\mathsf{write}(i)$, it responds with ok. After O plays read, it responds with an integer value present in the latest $\mathsf{write}(i)$ move by O or, if none has been played, with 0. This strategy will model $\vdash \mathsf{ref}(0) : \mathsf{ref}(\mathsf{int})$.

$\mathsf{cell}_{\mathsf{unit}\to\mathsf{unit}} : 1 \to [\![\mathsf{ref}(\mathsf{unit} \to \mathsf{unit})]\!]$ answers the initial question with $\circ$, responds to write and read with ok and $*$ respectively. If O plays q_r justified by an occurrence of $\star$, P plays q_w justified by the last occurrence of ok that precedes the relevant occurrence of $\star$. If none such exists, P has no response. Similarly, if O plays a_w, P will respond with a_r. This strategy will interpret $\vdash \mathsf{ref}(\lambda x^{\mathsf{unit}}.\Omega_{\mathsf{unit}}) : \mathsf{ref}(\mathsf{unit} \to \mathsf{unit})$.

Strategies compose (Honda & Yoshida, 1999), yielding a category of games where objects are arenas and morphisms between objects A and B are strategies in $A \to B$. Let $\Gamma = \{x_1 : \theta_1, \ldots, x_n : \theta_n\}$. We shall write $[\![\Gamma \vdash \theta]\!]$ for the prearena $[\![\theta_1]\!] \otimes \cdots \otimes [\![\theta_n]\!] \to [\![\theta]\!]$ (if $n = 0$ we take the left-hand side to be 1). The game model proposed in Abramsky et al. (1998) interprets a term $\Gamma \vdash M : \theta$ by a strategy in $[\![\Gamma \vdash \theta]\!]$.

We now introduce another condition on plays, known to characterize denotations of terms with ground-type storage only.

Definition 8.9 (*Visibility*) The *view* of a play is inductively defined by:

$$\begin{aligned} view(\epsilon) &= \epsilon \\ view(m) &= m \\ view(s_1\, m\, s_2\, n) &= view(s_1)\, m\, n\,. \end{aligned}$$

A play s satisfies the ***visibility*** condition if, for all even-length prefixes $s'm$ of s, the justifier of m occurs in $view(s')$. A strategy is called ***visible*** if all its plays satisfy visibility.

We can show that in plays the above condition is never violated by answers, because the pending question is always present in the view.

Lemma 8.10 *Let s be a play. If s contains a pending question then it occurs in $view(s)$.*

Proof Induction wrt $|s|$. If $s = \epsilon$ or $s = m$ then the Lemma is obviously true. Suppose $s = s_1\, m\, s_2\, n$. If n is a question then n is pending and the Lemma follows. Otherwise

n is an answer and, if s contains a pending question, it must occur in s_1. Hence, the Lemma follows from IH. $\square$

Proposition 8.11 (Abramsky & McCusker, 1997) *Let* $\Gamma \vdash M : \theta$ *be a term in which applications of the* $\mathsf{ref}(-)$*-constructor are restricted to terms of type* unit *and* int*. Then* $[\![\Gamma \vdash M : \theta]\!]$ *satisfies the visibility condition.*

8.4 Factorisation

We shall next write $!_A$ for the strategy in $A \rightarrow 1$ that responds to the initial move on the left with the unique move on the right. Given strategies $\sigma_i : 1 \rightarrow A_i$ that all respond to the initial question, we write $\langle \sigma_1, \ldots, \sigma_n \rangle$ for the strategy in $1 \rightarrow \bigotimes_{i=1}^{n} A_i$ that responds to the initial move with the tuple containing the individual responses of the n strategies and otherwise behaves like σ_i, depending on the component A_i in which O chooses to play.

Let us recall the factorisation result from Abramsky et al. (1998).

Theorem 8.12 (Abramsky et al., 1998) *Let* $\sigma : A_1 \rightarrow A_2$ *be a finite strategy and* $A = [\![\mathsf{ref}(\mathsf{unit} \rightarrow \mathsf{unit})]\!]$*. There exists n and a visible strategy* $\overline{\sigma} : (\bigotimes_{i=1}^{n} A) \otimes A_1 \rightarrow A_2$ *such that*

$$\langle \tau, \ldots, \tau, \mathsf{id}_{A_1} \rangle ; \overline{\sigma} = \sigma$$

where $\tau = !_{A_1} ; \mathsf{cell}_{\mathsf{unit} \rightarrow \mathsf{unit}}$.

Note that in the result above n may depend on σ. In fact, the proof shows that n can be taken to be (roughly) the length of the longest play in σ.

Remark 8.13 Violations of visibility describe computational scenarios in which a program attempts to call a function that was previously encountered during computation, yet which is not in current scope. The argument from Abramsky et al. (1998) proposes to repair such violations by using free (higher-order) reference variables. Intuitively, they provide an opportunity to record the functional values currently available to the program. A later attempt to access the reference makes it possible to use the required value. In contrast, our argument will take advantage of a single reference cell. We shall also record the scope at each step, but before doing so we will embed the previous value into the current scope, thus allowing backtracking. In this way, the sought value can be found by backtracking to the desired computational step.

Theorem 8.14 (Visible Factorisation) *Let* $\sigma : A_1 \rightarrow A_2$ *be a strategy and* $A = [\![\mathsf{ref}(\mathsf{unit} \rightarrow \mathsf{unit})]\!]$*. There exists a visible strategy* $\overline{\sigma} : A \otimes A_1 \rightarrow A_2$ *such that*

$$\langle \tau, \mathsf{id}_{A_1} \rangle ; \overline{\sigma} = \sigma$$

where $\tau = !_{A_1} ; \mathsf{cell}_{\mathsf{unit} \rightarrow \mathsf{unit}}$.

Proof We shall define $\overline{\sigma}$ to be the least strategy containing the plays from $\{\overline{s} \mid s \in \sigma\}$, where $\overline{s}$ will be defined below by induction on the length of a play. Roughly, $\overline{s}$ will consist of s augmented with moves from A.

- In particular, immediately after each O-move of s we shall insert the sequence $\mathsf{read} \star \mathsf{write}\,\mathsf{ok}$ and, if the move is an answer, it will be followed by a sequence consisting of answers a_w, a_r. Intuitively, each sequence $\mathsf{read} \star \mathsf{write}\,\mathsf{ok}$ corresponds to reading the current value of the reference (represented by A) and updating it with a new value.
- For P-questions, we shall insert $\mathsf{read} \star$ followed by a sequence consisting of questions q_r, q_w *in front of* the P-question. The last q_w will point at the value stored in the reference immediately after the justifier of q was played. P-answers will simply be copied without any extra moves.

We give a precise definition below. The targets of pointers from $\mathsf{read}, \star, \mathsf{write}, \mathsf{ok}$ are obvious so, when discussing pointers, we shall focus on those from q_r, q_w, a_r, a_w.

- $\overline{s\, q_O} = \overline{s}\, q_O\ \mathsf{read} \star \mathsf{write}\,\mathsf{ok}$ (if $|s| > 0$); and $\overline{q_O} = q_O\ \mathsf{write}\,\mathsf{ok}$
- $\overline{s\, q_P} = \overline{s}\ \mathsf{read} \star q_r\ (q_w\, q_r)^k\ q_w\, q_P$
  We take k to be the number of O-moves occurring after the justifier o of q_P in s. Let us list them (in order of occurrence) as $o_k, \ldots, o_1$. Then the ith q_w and q_r in $(q_w\, q_r)^k$ are meant to be justified by respectively write and $\star$ from the $\mathsf{read} \star \mathsf{write}\,\mathsf{ok}$ segment introduced immediately after o_i. The last q_w is justified by write from the $\mathsf{read} \star \mathsf{write}\,\mathsf{ok}$ segment added after o.
  Note that the resultant sequence will satisfy P-visibility, even if q_P may not have. Additionally, the extra O-moves $\star$, ok, q_w are consistent with the behaviour of the $\mathsf{cell}_{\mathsf{unit}\to\mathsf{unit}}$ strategy.

$$\cdots\ o\ \mathsf{r} \star \mathsf{w}\,\mathsf{ok} \cdots o_k\ \mathsf{r} \star \mathsf{w}\,\mathsf{ok} \cdots o_1\ \mathsf{r} \star \mathsf{w}\,\mathsf{ok}\ \mathsf{r} \star q_r\ q_w\ q_r\ \cdots\ q_w\ q_r\ q_w\ q_P$$

- $\overline{s\, a_O} = \overline{s}\, a_O\ \mathsf{read} \star \mathsf{write}\,\mathsf{ok}\ (a_w\, a_r)^{k+1}$
  Suppose a_O answers q_P in s. Then we take k to be the same as in the clause for q_P, i.e. k is the number of O-moves separating q_P's justifier and q_P. The sequence $(a_w\, a_r)^{k+1}$ simply answers all the questions q_w, q_r that were introduced for q_P. Because the pending question of $\overline{s}$ stays the same as that in s, this will yield a valid play. Note also that the O-moves a_r (in response to a_w) are consistent with the $\mathsf{cell}_{\mathsf{unit}\to\mathsf{unit}}$ strategy.

$$\cdots\ \mathsf{r} \star q_r\ (q_w\, q_r)^k\ q_w\, q_P\ \cdots\ a_O\ \ (a_w\, a_r)^{k+1}$$

- $\overline{s\, a_P} = \overline{s}\, a_P$
  As we have already mentioned, the construction of $\overline{s}$ from s preserves the pending question. Hence, the above clause leads to a play.

Consequently, $\overline{\sigma}$ is visible and, because the inserted moves are consistent with $\mathsf{cell}_{\mathsf{unit}\to\mathsf{unit}}$, we have $\langle\, !_{A_1}; \mathsf{cell}_{\mathsf{unit}\to\mathsf{unit}}, \mathsf{id}_{A_1}\,\rangle; \overline{\sigma} = \sigma$. $\square$

The proof above also demonstrates how one can derive $\overline{\sigma}$ from σ in an "effective" way. Next we introduce formal terminology to capture this observation as a formal statement. For a start, we shall assume that prearenas are *effectively given* so, in particular, the set of plays of A, written P_A, is recursively enumerable for any prearena A. We shall say that a strategy in A is ***recursively presentable*** if it is a recursively enumerable subset of P_A. Then our observation amounts to saying that, if σ is recursively presentable, so is $\overline{\sigma}$.

In order to apply the Theorem we need two more results. The first of them is classic and concerns decomposing visible strategies into innocent ones. Innocence (Hyland & Ong, 2000) stipulates that strategy responses be uniquely determined by views. In particular, they have to point at moves in the view, i.e. innocence is a stricter form of visibility.

Theorem 8.15 (Innocent Factorisation (Abramsky & McCusker, 1997)) *Let $\sigma : A_1 \to A_2$ be a visible strategy and $A = [\![\mathsf{ref}(\mathsf{int})]\!]$. There exists an innocent strategy $\hat{\sigma} : A \otimes A_1 \to A_2$ such that $\langle\, !_{A_1}; \mathsf{cell}_{\mathsf{int}}, \mathsf{id}_{A_1}\,\rangle; \hat{\sigma} = \sigma$.*

Note that this result already applies to arbitrary strategies rather than just finite ones. Also, the construction of $\hat{\sigma}$ is effective and shows that $\hat{\sigma}$ is recursively presentable if σ is.

Finally, we prove a universality result for recursively presentable innocent strategies. Universality results were not necessary in research on full abstraction, because their weaker variants phrased for finite (or finitely generated) strategies sufficed. Hence, after the initial ones for call-by-name PCF (Abramsky et al., 2000; Hyland & Ong, 2000), they all but disappeared from subsequent papers.

Theorem 8.16 (Innocent Universality) *Let $\sigma : [\![\Gamma \vdash \theta]\!]$ be a recursively presentable innocent strategy. There exists a* ref*-free term $\Gamma \vdash M : \theta$ such that $[\![\Gamma \vdash M : \theta]\!] = \sigma$. If Γ and θ do not contain occurrences of* ref*-types, then M can be taken to be* mkvar*-free.*

We postpone the proof of Theorem 8.16 until after the next two theorems. By appealing to Theorems 8.14, 8.15 and 8.16 one can deduce Universality.

Theorem 8.17 (Universality) *Let $\sigma : [\![\Gamma \vdash \theta]\!]$ be a recursively presentable strategy. Then there exists $\Gamma \vdash M : \theta$ such that $[\![\Gamma \vdash M : \theta]\!] = \sigma$.*

In fact, in the above statement M can be taken to be of the form

$$\mathsf{let}\ f, x = \mathsf{new}_{\mathsf{unit}\to\mathsf{unit}}, \mathsf{new}_{\mathsf{int}}\ \mathsf{in}\ M'$$

where M' is ref-free. Because the game semantics of a term is recursively presentable, we can conclude the following result.

Theorem 8.18 (Transformation) *Let $\Gamma \vdash M : \theta$. There exists a term*

$$\Gamma, f : \mathsf{ref}(\mathsf{unit} \to \mathsf{unit}), x : \mathsf{ref}(\mathsf{int}) \vdash M' : \theta$$

satisfying the following conditions.

- $\Gamma \vdash M \cong \mathsf{let}\ f, x = \mathsf{new}_{\mathsf{unit}\to\mathsf{unit}}, \mathsf{new}_{\mathsf{int}}\ \mathsf{in}\ M'$.
- *M' is* ref*-free.*
- *If there are no occurrences of* ref *in Γ, θ, then M' is* mkvar*-free.*

Thus, general references in $\mathcal{L}$ can be simulated by two memory cells that store values of type $\mathsf{unit} \to \mathsf{unit}$ and int respectively. Our proof was semantic, but the passage from M to M' can be made effective. However, due to reliance on the universality result, we would need to pass through enumerations of partial recursive functions. This is hardly a reasonable way of transforming programs! In the next section we shall identify several syntactic decomposition principles for general references, which will yield an alternative proof of the Theorem.

Innocent Universality

The remainder of this section is dedicated to proving Theorem 8.16. The argument uses the decomposition lemma for innocent call-by-value strategies from Honda and Yoshida (1999), with very small adjustments to cater for reference types. Recall that innocent strategies form a category $\mathcal{G}_{inn}$. Each innocent strategy σ is determined by its *view-function*:

$$\mathsf{viewf}(\sigma) = \{view(s)p \mid sp \in \sigma\}$$

Conversely, given a set S of even-length plays satisfying determinacy ($\forall sm_1s_1, sm_2s_2 \in S.\ |s|$ odd $\implies m_1 = m_2$) and such that each play in S consists of an odd-length view followed by a single move, we can form a least innocent strategy $\mathsf{strat}(S)$ such that $S \subseteq \mathsf{viewf}(\mathsf{strat}(S))$. Setting $\mathcal{G}^t_{inn}$ to be the luff subcategory of *total strategies* (i.e. strategies in which every initial move is immediately answered), $\otimes$ yields finite products, with the one-element arena 1 being final. Moreover, there is a natural bijection $\Lambda : \mathcal{G}_{inn}(A \otimes B, C) \xrightarrow{\cong} \mathcal{G}^t_{inn}(A, B \Rightarrow C)$, and we write $\mathsf{ev}_{A,B} : (A \Rightarrow B) \otimes A \to B$ for the strategy $\Lambda^{-1}(\mathsf{id}_{A\Rightarrow B})$.

We fix a generic notation $\sharp(\_)$ for coding functions from enumerable sets to ω. For example, $\sharp(i, j)$ encodes the pair (i, j) as a number. Moreover, we represent sequences of types such as $\theta_1, \ldots, \theta_m$ using vector notation $\vec{\theta}$, which we may also employ with explicit indexing (e.g. $\vec{\theta}_i^{\,j} = \theta_i, \theta_{i+1}, \ldots, \theta_j$). Similarly, a sequence of same moves $\star, \ldots, \star$ may be written $\vec{\star}$.

Lemma 8.19 (Decomposition Lemma (DL)) *Let $\theta_1, \ldots, \theta_m, \theta$ be types. Each innocent strategy $\sigma : [\![\theta_1]\!] \otimes \cdots \otimes [\![\theta_m]\!] \to [\![\theta]\!]$ can be decomposed as follows.*

1. *If $\theta_1, \ldots, \theta_m = \vec{\theta}_1^{\,m'}, \mathsf{int}, \vec{\theta}_{m'+2}^{\,m}$ with none of $\theta_1, \ldots, \theta_{m'}$ being* int *then:*

$$\sigma = \{(\vec{\star}_1^{\,m'}, i, \vec{q}_{m'+2}^{\,m})\, s \mid (\vec{\star}_1^{\,m'}, \vec{q}_{m'+2}^{\,m})\, s \in \tau_i\}$$

where, writing i for the unique total strategy into $\mathbb{Z}$ playing i:

$$\tau_i \equiv [\![\vec{\theta}_1^{m'}]\!] \otimes [\![\vec{\theta}_{m'+2}^{m}]\!] \xrightarrow{\mathsf{id}\otimes\langle !;i,\mathsf{id}\rangle} [\![\vec{\theta}_1^{m'}]\!] \otimes \mathbb{Z} \otimes [\![\vec{\theta}_{m'+2}^{m}]\!] \xrightarrow{\sigma} [\![\theta]\!]$$

If none of $\theta_1, \ldots, \theta_m$ is int *then one of the following is the case.*

$\vec{\star}\, a \in \sigma$*, in which case either:*

2. $\theta = \mathsf{unit}$ *and* $\sigma = [\![\vec{\theta}]\!] \xrightarrow{!} 1$*,*
3. $\theta = \mathsf{int}$*,* $i \in \mathbb{Z}$ *and* $\sigma = [\![\vec{\theta}]\!] \xrightarrow{i} \mathbb{Z}$*,*
4. $\theta = \mathsf{ref}(\theta')$ *and* $\sigma = \langle \sigma_1, \sigma_2 \rangle$ *where* $\sigma_i \equiv \sigma; \pi_i$*,*
5. $\theta = \theta' \to \theta''$ *and* $\sigma = \Lambda(\sigma')$ *where:*

$$\sigma' : [\![\vec{\theta}]\!] \otimes [\![\theta']\!] \to [\![\theta'']\!] \equiv \mathsf{strat}\{(\vec{\star}, q_{\theta'})\, s \mid \vec{\star}\, a\, q_{\theta'} s \in \mathsf{viewf}(\sigma)\}$$

6. $\vec{\star}\, q \in \sigma$ *with q played in some* $\theta_l = \mathsf{ref}(\theta_l')$*, in which case* $\sigma \cong \sigma_l$ *where*

$$\sigma_l : [\![\vec{\theta}_1^{l-1}]\!] \otimes (1 \Rightarrow [\![\theta_l']\!]) \otimes ([\![\theta_l']\!] \Rightarrow 1) \otimes [\![\vec{\theta}_{l+1}^{m}]\!] \longrightarrow [\![\theta]\!]$$

is obtained from σ by simply re-associating its initial moves.
7. $\vec{\star}\, q \in \sigma$ *with q played in some* $\theta_l = \theta_l' \to \theta_l''$*, in which case*

$$\sigma = [\![\vec{\theta}]\!] \xrightarrow{\langle \Lambda(\sigma''), \pi_l, \sigma' \rangle} ([\![\theta_l'']\!] \Rightarrow [\![\theta]\!]) \otimes ([\![\theta_l']\!] \Rightarrow [\![\theta_l'']\!]) \otimes [\![\theta_l']\!] \xrightarrow{\mathsf{id}\otimes\mathsf{ev};\, \mathsf{ev}} [\![\theta]\!]$$

where:

$$\sigma' : [\![\vec{\theta}]\!] \to [\![\theta_l']\!] \equiv \mathsf{strat}\{\vec{\star}\, a\, q_l'\, s \mid \vec{\star}\, q\, q_l'\, s \in \mathsf{viewf}(\sigma) \wedge a = q \wedge q_l' \in M_{[\![\theta_l']\!]}\}$$
$$\sigma'' : [\![\vec{\theta}]\!] \otimes [\![\theta_l'']\!] \to [\![\theta]\!] \equiv \mathsf{strat}\{(\vec{\star}, q_l'')s \mid \vec{\star}\, q\, a_l'' s \in \mathsf{viewf}(\sigma) \wedge q_l'' = a_l''\}$$

We fix an enumeration of partial recursive functions such that ϕ_n is the n-th partial recursive function and W_n is the n-th recursively enumerable set. In the rest of this section we closely follow the presentation of Abramsky et al. (2000); the reader is referred thereto for a more detailed exposition of background material.

It is not difficult to show that an innocent strategy σ is recursively presentable iff $\mathsf{viewf}(\sigma)$ is recursively enumerable or, equivalently, represents a partial recursive function (from views of odd-length plays to moves with pointer information). We shall therefore encode an innocent strategy σ by $\sharp(\sigma)$, where—depending on the context—the latter will be seen as the index n such that either $\sigma = W_n$ or $\mathsf{viewf}(\sigma) = \phi_n$. Recursively presentable strategies are closed under composition and therefore form a lluf subcategory $\mathcal{R}_{inn}$ of $\mathcal{G}_{inn}$. Moreover, all our language constructors preserve recursive presentability, thus $\mathcal{R}_{inn}$ is a sound model of the subset of the language which does not use the ref constructor.

List contexts

We say that a set of types T is ***closed*** if whenever $\theta \in T$:

- if $\theta = \theta_1 \to \theta_2$ then $\theta_1, \theta_2 \in T$;
- if $\theta = \mathsf{ref}(\theta')$ then $\mathsf{unit} \to \theta', \theta' \to \mathsf{unit} \in T$.

For the rest of this section let us fix a closed finite set of types T that contains the basic types unit, int. Let us also fix an ordering of T, say $T = T_0, T_1, \ldots, T_n$ such that $T_0 = \mathsf{unit}$ and $T_1 = \mathsf{int}$.

For each i, we encode lists of type T_i as products $\mathsf{int} \times (\mathsf{int} \to T_i)$. In particular, we use the notation

$$z : \mathsf{List}(T_i),\ \Gamma \vdash M : \theta$$

as shorthand for

$$z^L : \mathsf{int},\ z^R : \mathsf{int} \to T_i,\ \Gamma \vdash M : \theta\,.$$

Thus, z^L represents the length of the represented list; for each $1 \le i \le z^L$, the value of the i-th element in the list is represented by $z^R i$. The list can be shortened by simply 'reducing' z^L. For example, for a term $z : \mathsf{List}(T_i),\ \Gamma \vdash M : \theta$ we can form:

$$z : \mathsf{List}(T_i),\ \Gamma \vdash \mathsf{let}\, z^L = z^L{-}1 \,\mathsf{in}\, M : \theta$$

Note that, although the notation seems to suggest differently, the above is unrelated to variable assignment: it stands for $(\lambda z^L.M)(z^L{-}1)$. Moreover, a list can be extended as follows. For terms $z : \mathsf{List}(T_i),\ \Gamma \vdash M : \theta,\ N : T_i$, we define the term

$$z : \mathsf{List}(T_i),\ \Gamma \vdash \mathsf{extend}\, z \,\mathsf{with}\, N \,\mathsf{in}\, M : \theta$$

to be:

$$z : \mathsf{List}(T_i),\ \Gamma \vdash (\lambda z^L.\lambda z^R.\, M)(z^L{+}1)(\lambda x.\, \mathsf{if}\, x = 1 \,\mathsf{then}\, N \,\mathsf{else}\, z^R(x{-}1)) : \theta\,.$$

Our aim will be to construct, for each $\theta \in T$, a *universal term*

$$z_0 : \mathsf{List}(T_0), \ldots, z_n : \mathsf{List}(T_n) \vdash F_\theta : \mathsf{int} \to \theta$$

such that, for every sequence $\vec{\theta} = \theta_1, \ldots, \theta_m$ from T and recursive strategy $\sigma : [\![\theta_1]\!] \otimes \cdots \otimes [\![\theta_m]\!] \to [\![\theta]\!]$, we have $\sigma = [\![\mathsf{deindx}_{\vec{\theta}}\ (F_\theta\, \sharp(\sigma))]\!]$.

To explain what $\mathsf{deindx}_{\vec{\theta}}$ does, let us first define an (effective) indexing function indx which, for any sequence (possibly with repetitions) $\theta_1, \ldots, \theta_m$ of types from T, returns a pair of numbers (i, j) such that $\theta_m = T_i$ and there are j occurrences of T_i in $\theta_1, \ldots, \theta_m$. Suppose now we have such a sequence $\vec{\theta}$ and a term

$$z_0 : \mathsf{List}(T_0),\ z_1 : \mathsf{List}(T_1),\ \ldots,\ z_n : \mathsf{List}(T_n) \vdash M : \theta.$$

We can *de-index* M with respect to $\vec{\theta}$, obtaining the term

$$x_1 : \theta_1, \ldots, x_m : \theta_m \vdash \mathsf{deindx}_{\vec{\theta}}\, M : \theta,$$

defined by

$$\mathsf{deindx}_{\vec{\theta}}\, M \equiv \mathsf{let}\, z \vec{=} \perp \mathsf{in}\, (\mathsf{extend}\, z_{l_m}\, \mathsf{with}\, x_m\, \mathsf{in}\, (\ldots\ (\mathsf{extend}\, z_{l_1}\, \mathsf{with}\, x_1\, \mathsf{in}\, M)))$$

where, for each $1 \le i \le m$, $\theta_i = T_{l_i}$, and

$$\mathsf{let}\, z \vec{=} \perp \mathsf{in}\, N \equiv \mathsf{let}\, z_0^L = 0, z_0^R = \lambda x.\Omega\, \mathsf{in}\, (\ldots\ (\mathsf{let}\, z_n^L = 0, z_n^R = \lambda x.\Omega\, \mathsf{in}\, N))\,.$$

Universal terms

We present a recursive version (for types in T) of the decomposition lemma.

Lemma 8.20 *There exist partial recursive functions*

$$D, H : \omega \rightharpoonup \omega \text{ and } B : \omega \times \omega \rightharpoonup \omega$$

such that for any $\theta_1, \ldots, \theta_m, \theta \in T$ and recursively presentable strategy $\sigma : [\![\theta_1]\!] \otimes \cdots \otimes [\![\theta_m]\!] \to [\![\theta]\!]$:

$$D(\sharp(\sigma)) = \begin{cases} i & \text{if } \sigma \text{ falls within the } i\text{-th DL case (i.e. of Lemma 19)} \\ \perp & \text{otherwise} \end{cases}$$

$$B(\sharp(\sigma), i) = \begin{cases} \sharp(\tau_i) & \text{if } \sigma \text{ and } \tau_i \text{ are related as in first DL case} \\ \perp & \text{otherwise} \end{cases}$$

$$H(\sharp(\sigma)) = \begin{cases} i & \text{if } \sigma, i \text{ are related as in third DL case} \\ \sharp(\sharp(\sigma_1), \sharp(\sigma_2)) & \text{if } \sigma, \sigma_1, \sigma_2 \text{ are related as in fourth DL case} \\ \sharp(\sigma') & \text{if } \sigma, \sigma' \text{ are related as in fifth DL case} \\ \sharp(l_1, l_2, \sharp(\sigma_l)) & \text{if } \sigma, \theta_l, \sigma_l \text{ are related as in sixth DL case} \\ & \text{and } \mathsf{indx}\,(\theta_1, \ldots, \theta_l) = (l_1, l_2) \\ \sharp(l_1, l_2, \sharp(\sigma'), \sharp(\sigma'')) & \text{if } \sigma, \theta_l, \sigma', \sigma'' \text{ are related as in seventh DL} \\ & \text{case and } \mathsf{indx}\,(\theta_1, \ldots, \theta_l) = (l_1, l_2) \end{cases}$$

Proof Note that we assume that the type of σ is represented in $\sharp(\sigma)$, which allows us to examine type-related information via $\sharp(\sigma)$ and compute $\mathsf{indx}\,(\theta_1, \ldots, \theta_l)$, where needed.

$D(\sharp(\sigma))$ returns 1 if any of the θ_i's is int, otherwise it applies $\phi_{\sharp(\sigma)}$ to the unique initial move of $[\![\vec{\theta}]\!]$ and returns the number corresponding to the result. For B, given $\sharp(\sigma), i$, membership in τ_i is r.e. as follows. For any play s, we add i to its initial move and check whether the resulting play is a member of σ. Thus we obtain ϕ_n such that $s \mapsto \phi_n(s, \sharp(\sigma), i)$ is the characteristic function of τ_i. By an application of the S-m-n theorem we obtain $\sharp(\tau_i)$. For H, we argue along the same lines. $\square$

Since (call-by-value) PCF is Turing complete, there are closed PCF-terms $\tilde{D}, \tilde{H}$: $\mathsf{int} \to \mathsf{int}$ and $\tilde{B} : \mathsf{int} \to \mathsf{int} \to \mathsf{int}$ that represent each one of the above functions. We now proceed to define the universal terms. Let us write $\theta = T_{l(\theta)} \to T_{r(\theta)}$ whenever $\theta \in T$ is of arrow type, and $T_{rd(\theta)} = \mathsf{unit} \to \theta'$, $T_{wr(\theta)} = \theta' \to \mathsf{unit}$ when $\theta = \mathsf{ref}(\theta')$ (so $l, r, rd, rw : T \to \{0, \dots, n\})$. Recall that $T_1 = \mathsf{int}$ (hence, the term below refers to z_1).

Definition 8.21 For each $\theta \in T$ we define an open term $z_0 : \mathsf{List}(T_0), \dots, z_n : \mathsf{List}(T_n) \vdash F_\theta : \mathsf{int} \to \theta$ by mutual recursion as follows.[1]

$$
\begin{aligned}
F_\theta \equiv \lambda k^{\mathsf{int}}.\ & \mathsf{if}\ z_1^L \neq 0\ \mathsf{then\ let}\ x = z_1^R\, z_1^L,\ z_1^L = z_1^L - 1\ \mathsf{in}\ F_\theta(\tilde{B}\, k\, x)\\
& \mathsf{else\ case}\ (\tilde{D}\, k)\ \mathsf{of}\\
& \quad 2:\ ()\\
& \quad 3:\ \tilde{H}k\\
& \quad 4:\ \mathsf{let}\ \sharp(k_1, k_2) = \tilde{H}k\ \mathsf{in\ mkvar}\ (F_{\mathsf{unit}\to T_{u(\theta)}}\, k_1)\ (F_{T_{u(\theta)}\to\mathsf{unit}}\, k_2)\\
& \quad 5:\ \lambda y^{T_{l(\theta)}}.\ \mathsf{extend}\ z_{l(\theta)}\ \mathsf{with}\ y\ \mathsf{in}\ F_{T_{r(\theta)}}(\tilde{H}\, k)\\
& \quad 6:\ \mathsf{let}\ \sharp(i_1, i_2, k) = \tilde{H}k\ \mathsf{in\ case}\ i\ \mathsf{of}\\
& \qquad 1:\ \dots\\
& \qquad \vdots\\
& \qquad j:\ \mathsf{extend}\ z_{rd(T_j)}\ \mathsf{with}\ \lambda y^{\mathsf{unit}}.\,!(z_j^L i_2)\ \mathsf{in}\\
& \qquad\qquad \mathsf{extend}\ z_{wr(T_j)}\ \mathsf{with}\ \lambda y^{T_{l(T_{wr(T_j)})}}.\ (z_j^L i_2) := y\ \mathsf{in}\ F_\theta k\\
& \qquad \vdots\\
& \qquad n:\ \dots\\
& \qquad \mathsf{otherwise}:\ \Omega\\
& \quad 7:\ \mathsf{let}\ \sharp(i_1, i_2, k_1, k_2) = \tilde{H}k\ \mathsf{in\ case}\ i_1\ \mathsf{of}\\
& \qquad 1:\ \dots\\
& \qquad \vdots\\
& \qquad j:\ \mathsf{extend}\ z_{r(T_j)}\ \mathsf{with}\ (z_j^L\, i_2)(F_{T_{l(T_j)}} k_1)\ \mathsf{in}\ F_\theta\, k_2\\
& \qquad \vdots\\
& \qquad n:\ \dots\\
& \qquad \mathsf{otherwise}:\ \Omega\\
& \quad \mathsf{otherwise}:\ \Omega
\end{aligned}
$$

[1] Recall that $\mathsf{let}\ z_1^L = z_1^L - 1\ \mathsf{in}\ N$ is *not* assignment, but shorthand for $(\lambda z_1^L.N)(z_1^L - 1)$.

The construction of F_θ follows closely the decomposition of σ along the DL. We can show, by induction on the length of plays, the following.

Proposition 8.22 (Universality of F) *Suppose* $\theta_1, \ldots, \theta_m, \theta \in T$ *and let* $\vec{\theta} = \theta_1, \ldots, \theta_m$. *For every recursively presentable strategy* $\sigma : [\![\theta_1]\!] \otimes \cdots \otimes [\![\theta_m]\!] \to [\![\theta]\!]$, $\sigma = [\![\mathsf{deindx}_{\vec{\theta}}\ (F_\theta\ \sharp(\sigma))]\!]$.

We can now prove innocent universality.

Proof (Theorem 8.16) For any such $\sigma : [\![\Gamma \vdash \theta]\!]$, by the previous Proposition and setting T to be the least closed set of types containing $\mathsf{cod}(\Gamma) \cup \{\theta\}$, it suffices to take M to be $\mathsf{deindx}_{\vec{\theta}}\ (F_\theta\ \sharp(\sigma))$.

By construction (Definition 8.21), M is ref-free. Moreover, if T is ref-free then M does not contain occurrences of mkvar, as the only cases in Definition 8.21 which introduce it are 4 and 6, which require a reference type in the output or the context respectively. Now note that if Γ and θ are ref-free then so is T. $\square$

8.5 Syntactic Transformation

In this section we give an alternative proof of Theorem 8.18 based on a syntactic transformation of terms from $\mathcal{L}$.

Note that $\mathsf{ref}(M)$ is equivalent to $\mathsf{let}\, x = \mathsf{new}_\theta \,\mathsf{in}\, (x := M; x)$ for a suitable θ. Consequently, w.l.o.g. we can assume that the only occurrences of $\mathsf{ref}(\cdots)$ inside terms are those associated with new_θ. Similarly, we assume that terms do not contain fixed-point subterms, as these can be simulated using higher-order reference cells (Abramsky et al., 1998).

Next we show new_θ can be decomposed using instances of new at simpler types. Ultimately, this will allow us to replace any occurrences of $\mathsf{ref}(M)$ with $\mathsf{new}_{\mathsf{unit}\to\mathsf{unit}}$ and $\mathsf{new}_{\mathsf{int}}$. The mkvar constructor is central to the transformations.

The following equivalence is shown formally by comparing strategies corresponding to each term. Intuitively, it is valid because on assignment M_w indirectly records the assigned value g in f. On dereferencing, M_r ensures that the latest value of f is accessed and the corresponding value g applied to the right argument through the internal references x_1 and x_2.

Lemma 8.23 (Decomposition of $\mathsf{ref}(\theta_1 \to \theta_2)$) *For all* θ_1, θ_2, $\vdash \mathsf{new}_{\theta_1\to\theta_2} \cong \mathsf{let}\, f, x_1, x_2 = \mathsf{new}_{\mathsf{unit}\to\mathsf{unit}}, \mathsf{new}_{\theta_1}, \mathsf{new}_{\theta_2} \,\mathsf{in}\, \mathsf{mkvar}(M_r, M_w) \colon \mathsf{ref}(\theta_1 \to \theta_2)$*, where*

$$M_r \equiv \lambda y^{\mathsf{unit}}.\mathsf{let}\, h = !f \,\mathsf{in}\, \lambda z^{\theta_1}.\, (x_1 := z;\ h();\ !x_2),$$
$$M_w \equiv \lambda g^{\theta_1\to\theta_2}.\ f := (\lambda y^{\mathsf{unit}}.\, x_2 := g(!x_1)).$$

Proof Let $M \equiv \mathsf{mkvar}(M_r, M_w)$ with M_r, M_w defined as above. We need to show $\vdash \mathsf{new}_{\theta_1\to\theta_2} \cong \mathsf{let}\, f, x_1, x_2 = \mathsf{new}_{\mathsf{unit}\to\mathsf{unit}}, \mathsf{new}_{\theta_1}, \mathsf{new}_{\theta_2} \,\mathsf{in}\, M$. By full abstraction, it suffices to show:

$$[\![\mathsf{let}\ f, x_1, x_2 = \mathsf{new}_{\mathsf{unit}\to\mathsf{unit}}, \mathsf{new}_{\theta_1}, \mathsf{new}_{\theta_2}\ \mathsf{in}\ M]\!] = [\![\mathsf{new}_{\theta_1\to\theta_2}]\!]$$

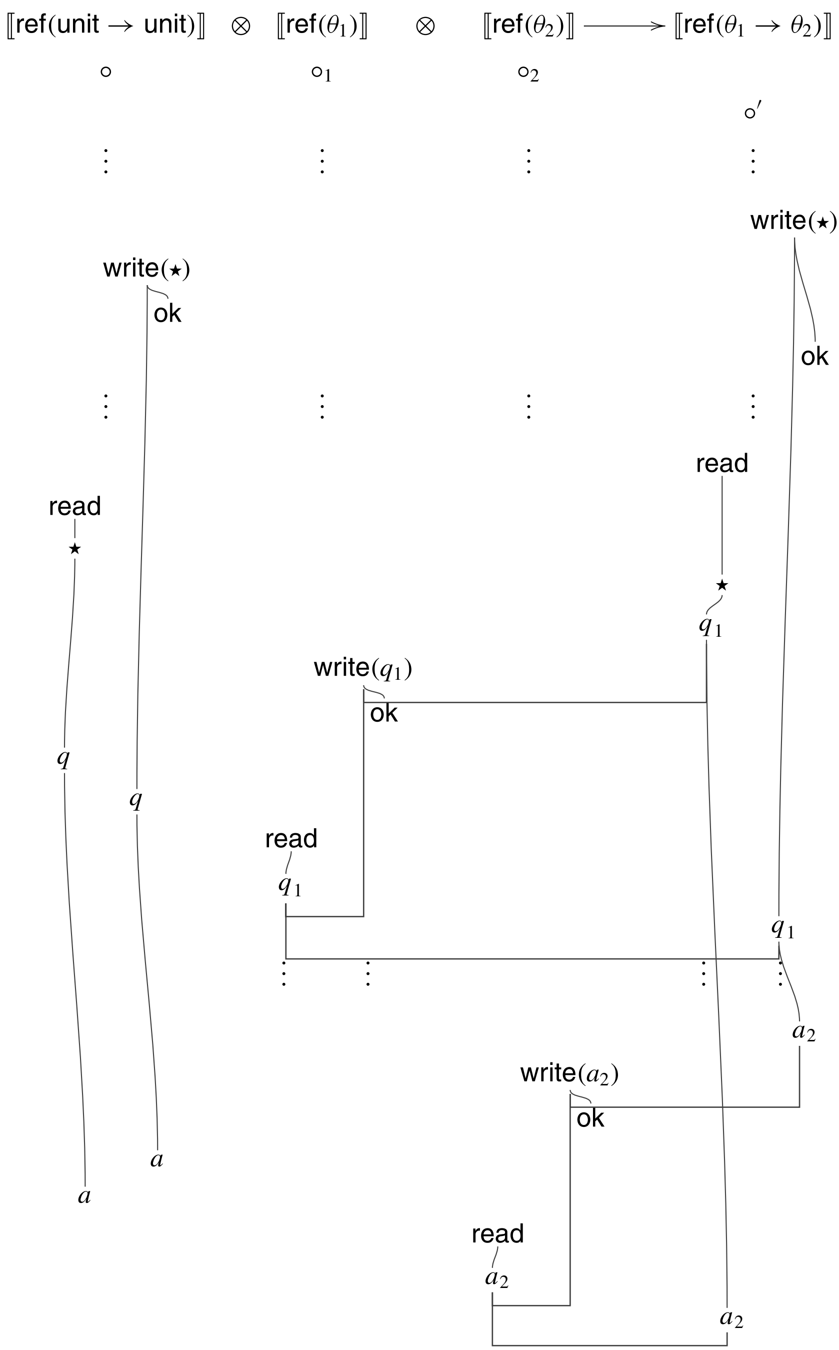
Setting $A = [\![\mathsf{ref}(\mathsf{unit} \to \mathsf{unit})]\!]$, $A_1 = [\![\mathsf{ref}(\theta_1)]\!]$, $A_2 = [\![\mathsf{ref}(\theta_2)]\!]$ and $B = [\![\mathsf{ref}(\theta_1 \to \theta_2)]\!]$, so that $[\![M]\!] : A \otimes A_1 \otimes A_2 \to B$, we define the strategy:

$$\sigma' = \{ s \in [\![M]\!] \mid \star (s \upharpoonright A) \in [\![\mathsf{new}_{\mathsf{unit}\to\mathsf{unit}}]\!] \wedge \star(s \upharpoonright A_i) \in [\![\mathsf{new}_{\theta_i}]\!]\ (i = 1, 2) \}$$

In order to prove the lemma, it suffices to show $\sigma' \upharpoonright B = [\![\mathsf{new}_{\theta_1\to\theta_2}]\!]$, where the latter projection replaces the initial move $(\circ, \circ_1, \circ_2)$ of $A \times A_1 \times A_2$ with $\star$. Let us examine the plays of σ'. Each play starts with the initial move and an answer: $(\circ, \circ_1, \circ_2)\, \circ'$. At that point, if O asks to read then the strategy asks back read under $\circ$ (playing via its read component $[\![M_r]\!]$). Since we assume that f plays like $[\![\mathsf{new}]\!]$, it replies with a value $\star$, which then P copies on the right-hand-side. This exchange of moves can be repeated, until O plays a question q_1 pointing to one of the $\star$'s on the right, at which point P writes q_1 under $\circ_1$ (because of $x_1 := z$) and, once an ok is received (which is the case, as we assume that the strategy $[\![\mathsf{new}]\!]$ is playing under $\circ_1$), the strategy plays q under the corresponding read on the left (because of $h()$). Now, since O follows $[\![\mathsf{new}]\!]$ under $\circ$, there will be no response to q, and the play stops there. Diagrammatically:

$$\begin{array}{ccccccc}
[\![\mathsf{ref}(\mathsf{unit} \to \mathsf{unit})]\!] & \otimes & [\![\mathsf{ref}(\theta_1)]\!] & \otimes & [\![\mathsf{ref}(\theta_2)]\!] & \multimap & [\![\mathsf{ref}(\theta_1 \to \theta_2)]\!] \\
\circ & & \circ_1 & & \circ_2 & & \\
 & & & & & & \circ' \\
 & & & & & & \mathsf{read} \\
\mathsf{read} & & & & & & \\
\star & & & & & & \\
 & & & & & & \star \\
\vdots & & & & & & \vdots \\
 & & & & & & \mathsf{read} \\
\mathsf{read} & & & & & & \\
\star & & & & & & \\
 & & & & & & \star \\
\vdots & & & & & & \vdots \\
 & & & & & & q_1 \\
 & & \mathsf{write}(q_1) & & & & \\
 & & \mathsf{ok} & & & & \\
q & & & & & &
\end{array}$$

Hiding the moves on the left, we obtain the desired $[\![\mathsf{new}]\!]$ behaviour. Now, in case a write occurs on the right before a read, a typical play will look as follows,



where the curved lines are justification pointers, and the polygonal ones are copycat links. Thus, when a write happens on the right, the strategy responds (via its write component $[\![\lambda g^{\theta_1 \to \theta_2}. f := (\lambda z^{\mathsf{unit}}. x_2 := g(!x_1))]\!]$) with a $\mathsf{write}(\star)$ move under $\circ$ (because of $f := \cdots$). This is repeated until a read move is played under $\circ'$, at which point σ' asks back read under $\circ$ and copies back the response with a $\star$ (playing as dictated by $[\![\lambda y^{\mathsf{unit}}. \mathsf{let}\, h = !f \,\mathsf{in}\, \lambda z^{\theta_1}. x_1 := z; h(); !x_2]\!]$). If O plays a question q_1 under $\star$ (possibly after several iterations of the write/read moves played so far), then P responds by writing q_1 under $\circ_1$, and creates a copycat link between these two last moves (because of $x_1 := z$), at which point O will answer ok. P now asks q beneath the $\star$ which came as a response to his read (because of $h()$), to which O responds by copying q under the previous write (since O plays like $[\![\mathsf{new}]\!]$ under $\star$). At this point, σ' plays as dictated by its write component and, as q corresponds to z in $f := (\lambda z^{\mathsf{unit}}. x_2 := g(!x_1))$, P needs to read under $\circ_1$, receive a response and open a copycat link between it and the value of g at that point, that is, the corresponding write on the right. O, playing as the $[\![\mathsf{new}]\!]$ strategy, will give the response q_1 and copycat between it and the last write before it. These three copycat links give us copycat behaviour between the two q_1s on the right. Once an answer a_2 is received on the right, it is written by σ' under $\circ_2$ (because of $x_2 := g(!x_1)$), and a copycat link is created. O responds with ok, and this is copied by P as the result of f, which O copycats under the open read for $\circ$. We are now back in $x_1 := z; h(); !x_2$, with $h()$ having returned, so now P plays a read under $\circ_2$, receives an answer a_2, which opens a copycat link with the previous write, and plays it in turn on the right, opening a new copycat link. These last links create a copycat behaviour between the two a_2's on the right. Taking stock of the copycat links, and hiding the moves on the left hand side of the diagram, we obtain the desired $[\![\mathsf{new}]\!]$ behaviour. □

For the next equivalence, instead of storing a reference of type θ, we store the associated read and write methods, of types $\mathsf{unit} \to \theta$ and $\theta \to \mathsf{unit}$ respectively, which is what references r and w are used for.

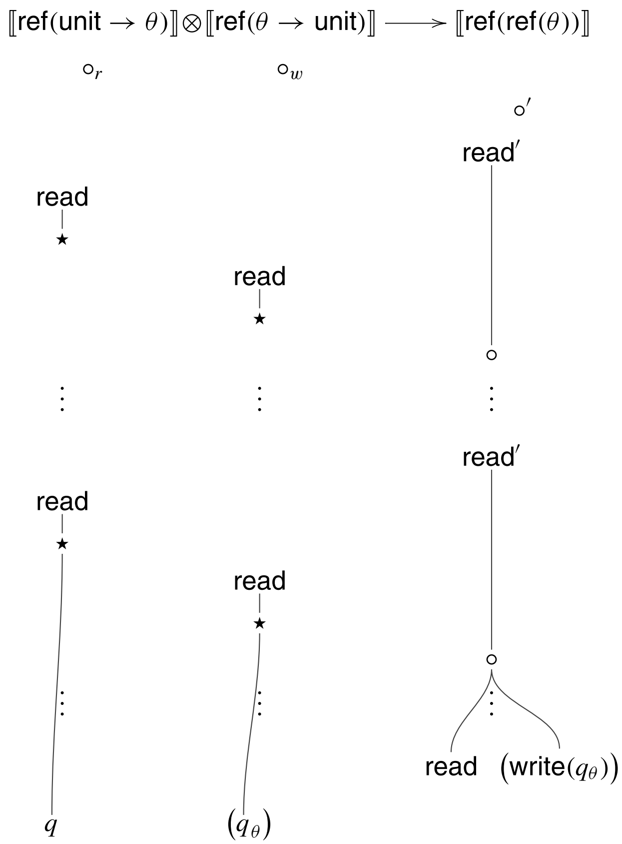
Lemma 8.24 (Decomposition of $\mathsf{ref}(\mathsf{ref}(\theta))$) *For any* θ, $\vdash \mathsf{new}_{\mathsf{ref}(\theta)} \cong \mathsf{let}\, r, w = \mathsf{new}_{\mathsf{unit}\to\theta}, \mathsf{new}_{\theta\to\mathsf{unit}} \,\mathsf{in}\, \mathsf{mkvar}(M_r, M_w) \colon \mathsf{ref}(\mathsf{ref}(\theta))$ *for all* θ, *where*

$$M_r \equiv \lambda z^{\mathsf{unit}}. \mathsf{mkvar}(!r, !w),$$
$$M_w \equiv \lambda g^{\mathsf{ref}(\theta)}. (r := (\lambda z^{\mathsf{unit}}. !g);\ \ w := (\lambda z^{\theta}. g := z)).$$

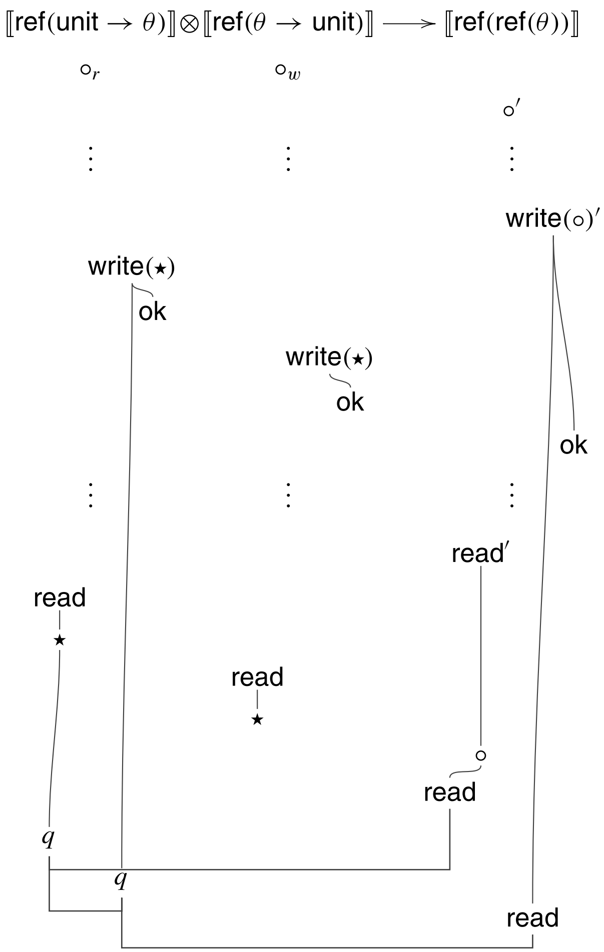
Proof Let $M \equiv \mathsf{mkvar}(M_r, M_w)$ with M_r, M_w defined above. We need to show $\mathsf{new}_{\mathsf{ref}(\theta)} \cong \mathsf{let}\, r, w = \mathsf{new}_{\mathsf{unit}\to\theta}, \mathsf{new}_{\theta\to\mathsf{unit}} \,\mathsf{in}\, M$. Setting $A_r = [\![\mathsf{ref}(\mathsf{unit} \to \theta)]\!]$, $A_w = [\![\mathsf{ref}(\theta \to \mathsf{unit})]\!]$ and $B = [\![\mathsf{ref}(\mathsf{ref}(\theta))]\!]$, so that $[\![M]\!] : A_r \otimes A_w \to B$, we define the strategy:

$$\sigma' = \{s \in [\![M]\!] \mid \star\,(s \restriction A_r) \in [\![\mathsf{new}_{\mathsf{unit}\to\theta}]\!] \wedge \star(s \restriction A_w) \in [\![\mathsf{new}_{\theta\to\mathsf{unit}}]\!]\,\}$$

In order to prove the lemma, it suffices to show $\sigma' \restriction B = [\![\mathsf{new}_{\mathsf{ref}(\theta)}]\!]$, where the latter projection replaces the initial move $(\circ_r, \circ_w)$ of $A_r \times A_w$ with $\star$. Each play of σ' starts with the initial move and an answer: $(\circ_r, \circ_w)\,\circ'$. If O asks read' next, the strategy asks back read, first under $\circ_r$ and then under $\circ_w$ (playing via its read component $[\![M_r]\!]$). Since we assume that r, w play like $[\![\mathsf{new}]\!]$, they both reply with a value $\star$, after which P plays the value $\circ$ on the right-hand-side (which corresponds to the produced mkvar value). This exchange of moves can be repeated, until O plays a question pointing to one of the $\circ$'s on the right, either a read or some $\mathsf{write}(q_\theta)$, at which point P plays q or q_θ under the corresponding $\star_r$ or $\star_w$ respectively. In both cases, since O follows $[\![\mathsf{new}]\!]$ under $\circ_r$ and $\circ_w$, there will be no response and the play stops there. Diagrammatically:



Hiding the moves on the left, we obtain the desired $[\![\mathsf{new}]\!]$ behaviour. Now, in case a write occurs on the right before a read, a typical play will look as follows (and similarly if O asks some $\mathsf{write}(q_\theta)$ under $\circ$ on the right-hand-side).



Thus, when a write$(\circ)'$ move is played on the right, the strategy responds (via its write component $[\![\lambda g^{\mathsf{ref}(\theta)}.\, r := (\lambda z^{\mathsf{unit}}.\, !g);\ w := (\lambda z^{\theta}.\, g := z)]\!]$) with a couple of write$(\star)$ moves under $\circ_r$ and $\circ_w$ respectively (because of $r := \cdots$ and $w := \cdots$). O plays like new in both r and w and therefore responds with ok in both cases, and then P plays ok under write$(\circ)'$. This is repeated until a read move is played under $\circ'$, at which point σ' asks back read under $\circ_r$ and $\circ_w$ (playing as dictated by $[\![\lambda z^{\mathsf{unit}}.\, \mathsf{mkvar}(!r, !w)]\!]$). O at this point answers both questions with $\star$, and P plays an answer $\circ$ in the right. Now, if O examines the value of the reference in $\circ$, by asking read beneath it, then P will query the value if the function stored in r, by playing q under the leftmost $\star$ and opening a copycat link between that last two moves (as $\mathsf{mkvar}(!r, !w)$ dictates that the reads be delegated to r). O now will copy that question to the last precedent write and open a copycat link between them. Now P is playing according to his write component again and therefore copies that question under write$(\circ)'$, playing read, and opens a copycat link between the last two moves. The three copycat links give us copycat behaviour between the two read's on the right. A similar scenario takes place if O asks $\mathsf{write}(q_\theta)$ on the right, only that now the succeeding q will be played under the read which was played beneath $\circ_w$. Taking stock of the copycat links, and hiding the moves on the left hand side of the diagram, we obtain the desired $[\![\mathsf{new}]\!]$ behaviour. □

The last Lemma is easy to verify by reference to the game model. It illustrates the rather strange status of type $\mathsf{ref}(\mathsf{unit})$ in $\mathcal{L}$, in particular the fact that it is not possible to compare reference names (of type $\mathsf{ref}(\mathsf{unit})$) in the language.

Lemma 8.25 $\vdash \mathsf{new}_{\mathsf{unit}} \cong \mathsf{mkvar}(\lambda x^{\mathsf{unit}}.\,(), \lambda x^{\mathsf{unit}}.\,()) : \mathsf{ref}(\mathsf{unit})$.

The last three Lemmas imply the following corollary.

Corollary 8.26 *For any $\Gamma \vdash M : \theta$ there exists $\Gamma \vdash M'' : \theta$ such that $\Gamma \vdash M \cong M'' : \theta$ and occurrences of the* ref *constructor in M'' are restricted to terms of the form* $\mathsf{new}_{\mathsf{unit}\to\mathsf{unit}}$ *or* $\mathsf{new}_{\mathsf{int}}$.

In the result above, $\mathsf{new}_{\mathsf{unit}\to\mathsf{unit}}$ and $\mathsf{new}_{\mathsf{int}}$ are allowed to occur multiple times. In what follows we shall show that one occurrence of each suffices.

Lemma 8.27 *There exist* ref*-free terms M, N such that*

$$\vdash \lambda x^{\mathsf{unit}}.\,\mathsf{new}_{\mathsf{unit}\to\mathsf{unit}} \cong \mathsf{let}\, f, x = \mathsf{new}_{\mathsf{int}\to\mathsf{unit}}, \mathsf{new}_{\mathsf{int}}\, \mathsf{in}\, M : \mathsf{unit} \to \mathsf{ref}(\mathsf{unit} \to \mathsf{unit}),$$
$$\vdash \lambda x^{\mathsf{unit}}.\,\mathsf{new}_{\mathsf{int}} \cong \mathsf{let}\, x = \mathsf{new}_{\mathsf{int}}\, \mathsf{in}\, N : \mathsf{unit} \to \mathsf{ref}(\mathsf{int}).$$

Proof We can encode an unbounded number of references of type $\mathsf{ref}(\mathsf{unit} \to \mathsf{unit})$ with a reference f of type $\mathsf{ref}(\mathsf{int} \to \mathsf{unit})$ by giving to each $(\mathsf{unit} \to \mathsf{unit})$-valued reference a unique integer identifier i, and encoding the value of the ith such reference as $\lambda v^{\mathsf{unit}}.\,(!f)i$. We use the internal variable x to count the number of generated references, so as to assign them unique identifiers. Thus, M can be taken to be $\lambda z^{\mathsf{unit}}.\mathsf{let}\, i = !x\, \mathsf{in}\, (x := !x + 1); \mathsf{mkvar}(M_r, M_w)$, where $M_r \equiv \lambda u^{\mathsf{unit}}.\mathsf{let}\, h = !f\, \mathsf{in}\, \lambda v^{\mathsf{unit}}.\, h\, i$ and $M_w \equiv \lambda g^{\mathsf{unit}\to\mathsf{unit}}.\,\mathsf{let}\, g' = !f\, \mathsf{in}\, f := (\lambda y^{\mathsf{int}}.\,\mathsf{if}\, y = i\, \mathsf{then}\, g()\, \mathsf{else}\, g' y)$.

For the second part, assume a standard encoding $\mathsf{G}(-) : \mathbb{Z}^* \to \mathbb{Z}$ of lists of integers into integers such that $\mathsf{G}(\epsilon) = 0$. Clearly, one can construct closed PCF terms $\mathsf{len} : \mathsf{int} \to \mathsf{int}, \mathsf{add} : \mathsf{int} \to \mathsf{int} \to \mathsf{int}, \mathsf{proj} : \mathsf{int} \to \mathsf{int} \to \mathsf{int}$ and $\mathsf{upd} : \mathsf{int} \to \mathsf{int} \to \mathsf{int} \to \mathsf{int}$ such that, for all $s \in \mathbb{Z}^*$ and $i, j \in \mathbb{Z}$:

$$\mathsf{len}\, \mathsf{G}(s) \Downarrow |s|\,, \quad \mathsf{add}\, \mathsf{G}(s)\, i \Downarrow \mathsf{G}(si)\,, \quad \mathsf{proj}\, \mathsf{G}(s)\, j \Downarrow s_j\,, \quad \mathsf{upd}\, \mathsf{G}(s)\, j\, i \Downarrow \mathsf{G}(s[j \mapsto i])\,,$$

where $|s|$ is the length of s, s_j is the jth element of s, and $s[j \mapsto i]$ is the list s with its jth element changed to i. We can then keep track of an unbounded number of integer-valued references by taking N to be

$$\lambda z^{\mathsf{unit}}.\, x := \mathsf{add}\, (!x)\, 0; \mathsf{let}\, j{=}\mathsf{len}(!x)\, \mathsf{in}\, \mathsf{mkvar}(\lambda z^{\mathsf{unit}}.\, \mathsf{proj}\, (!x)\, j, \lambda i^{\mathsf{int}}.\, x := \mathsf{upd}\, (!x)\, j\, i).$$

□

Now we are ready to give a syntactic proof of the first two claims of Theorem 8.18.

Proof Given $\Gamma \vdash M : \theta$, from Corollary 8.26 we can obtain an equivalent term M'', in which occurrences of ref are restricted to $\mathsf{new}_{\mathsf{unit}\to\mathsf{unit}}$ and $\mathsf{new}_{\mathsf{int}}$. Observe that M'' is thus equivalent to $\mathsf{let}\, h = \lambda x^{\mathsf{unit}}.\mathsf{new}_{\mathsf{unit}\to\mathsf{unit}}\,\mathsf{in}\, M_1$, where $M_1 \equiv M''[h()/\mathsf{new}_{\mathsf{unit}\to\mathsf{unit}}]$ and the only occurrences of ref in M_1 are those of $\mathsf{new}_{\mathsf{int}}$. Applying Lemmata 8.27, 8.25 and 8.23, M'' is further equivalent to $\mathsf{let}\, f = \mathsf{new}_{\mathsf{unit}\to\mathsf{unit}}\,\mathsf{in}\, M_2$, where the only occurrences of ref in M_2 are those of $\mathsf{new}_{\mathsf{int}}$. Finally, noting that M_2 is equivalent to $\mathsf{let}\, h' = \lambda x^{\mathsf{unit}}.\mathsf{new}_{\mathsf{int}}\,\mathsf{in}\, M_3$, where M_3 is ref-free, and invoking Lemma 8.23 we can conclude that M_2 is equivalent to $\mathsf{let}\, x = \mathsf{new}_{\mathsf{int}}\,\mathsf{in}\, M_4$, where M_4 is ref-free. Now we can take M' (from the statement of the Theorem) to be $\mathsf{let}\, f, x = \mathsf{new}_{\mathsf{unit}\to\mathsf{unit}}, \mathsf{new}_{\mathsf{int}}\,\mathsf{in}\, M_4$. □

The final claim in Theorem 8.18 stipulates that if the typing context of the term M contains no reference types then the resulting term M' does not involve mkvar. This claim is proven in the remainder of this section and constitutes Corollary 8.33.

Note that the decompositions presented so far in this section relied on the availability of mkvar and the term M' from the above proof will in general contain many occurrences of mkvar. We devote the remainder of this section to showing that when Γ and θ are ref-free, all the occurrences of mkvar can actually be eliminated. To that end, we shall rely on a notion of canonical form, defined below.

$$\begin{aligned}\mathbb{C} ::= {} & () \mid x^{\mathsf{int}} \mid \mathsf{mkvar}(\lambda u^{\mathsf{unit}}.\mathbb{C}, \lambda v^{\theta}.\mathbb{C}) \mid \lambda x^{\theta}.\mathbb{C} \mid \mathsf{if}\,\mathbb{C}\,\mathsf{then}\,\mathbb{C}\,\mathsf{else}\,\mathbb{C} \mid \\ & \mathsf{let}\, y = i\,\mathsf{in}\,\mathbb{C} \mid \mathsf{let}\, y = \mathbb{C} \oplus \mathbb{C}\,\mathsf{in}\,\mathbb{C} \mid \mathsf{let}\, y = !x\,\mathsf{in}\,\mathbb{C} \mid \mathsf{let}\, y = (x := \mathbb{C})\,\mathsf{in}\,\mathbb{C} \mid \\ & \mathsf{let}\, y = x\,\mathbb{C}\,\mathsf{in}\,\mathbb{C} \mid \mathsf{let}\, y = \mathsf{ref}(\mathbb{C})\,\mathsf{in}\,\mathbb{C}\end{aligned}$$

The canonical forms enjoy the following property.

Lemma 8.28 *For any $\Gamma \vdash M : \theta$ without fixed points, there exists a term $\mathbb{C}_M$ in canonical form such that $\Gamma \vdash M \cong \mathbb{C}_M : \theta$. Moreover, $\mathbb{C}_M$ can be effectively found and the conversion does not add any occurrences of* ref.

In short, given M, we need to produce an equivalent canonical form $\mathbb{C}_M$. Let us begin by considering several special cases.

Lemma 8.29 *Lemma 8.28 holds for identifiers x^{θ}.*

Proof We reason by induction on the structure of θ.

x^{unit}: $\quad \mathbb{C}_x \equiv ()$
x^{int}: $\quad \mathbb{C}_x \equiv x$
$x^{\mathsf{ref}(\theta)}$: $\quad \mathbb{C}_x \equiv \mathsf{mkvar}(\lambda u^{\mathsf{unit}}.\mathsf{let}\, y = !x\,\mathsf{in}\,\mathbb{C}_y,\ \lambda v^{\theta}.\mathsf{let}\, y = (x := \mathbb{C}_v)\,\mathsf{in}\,())$
$x^{\theta_1\to\theta_2}$: $\quad \mathbb{C}_x \equiv \lambda z^{\theta_1}.\mathsf{let}\, y = x\,\mathbb{C}_z\,\mathsf{in}\,\mathbb{C}_y$

□

Now we show another auxiliary result, also a special case of Lemma 8.28.

Lemma 8.30 *Lemma 8.28 holds for terms of the form* $\mathsf{let}\, z =_{\theta} \mathbb{C}_1\,\mathsf{in}\,\mathbb{C}_2$.

Proof We reason by induction on θ.

- Let us start from $\theta \equiv \mathsf{unit}, \mathsf{int}$ and reason by induction on $\mathbb{C}_1$. There are two base cases.

  - $\mathbb{C}_1 \equiv ()$. Then $\mathsf{let}\, z = \mathbb{C}_1 \,\mathsf{in}\, \mathbb{C}_2$ is simply equivalent to $\mathbb{C}_2$.
  - $\mathbb{C}_1 \equiv x^{\mathsf{int}}$. Then $\mathsf{let}\, z = \mathbb{C}_1 \,\mathsf{in}\, \mathbb{C}_2$ is equivalent to $\mathbb{C}_2[x/z]$.

  The remaining cases to consider are

  - $\mathbb{C}_1 \equiv \mathsf{if}\, \mathbb{C}_1' \,\mathsf{then}\, \mathbb{C}_2' \,\mathsf{else}\, \mathbb{C}_3'$,
  - $\mathbb{C}_1 \equiv \mathsf{let}\, y = \dots \,\mathsf{in}\, \mathbb{C}_1'$.

  For the former, observe that

  $$\mathsf{let}\, z = (\mathsf{if}\, \mathbb{C}_1' \,\mathsf{then}\, \mathbb{C}_2' \,\mathsf{else}\, \mathbb{C}_3') \,\mathsf{in}\, \mathbb{C}_2$$

  is equivalent to

  $$\mathsf{if}\, \mathbb{C}_1' \,\mathsf{then}\, (\mathsf{let}\, z = \mathbb{C}_2' \,\mathsf{in}\, \mathbb{C}_2) \,\mathsf{else}\, (\mathsf{let}\, z = \mathbb{C}_3' \,\mathsf{in}\, \mathbb{C}_2)$$

  and invoke IH for the smaller terms $\mathbb{C}_2'$ and $\mathbb{C}_3'$.
  For the latter, note that

  $$\mathsf{let}\, z = (\mathsf{let}\, y = \dots \,\mathsf{in}\, \mathbb{C}_1') \,\mathsf{in}\, \mathbb{C}_2$$

  is equivalent to

  $$\mathsf{let}\, y = \cdots \,\mathsf{in}\, (\mathsf{let}\, z = \mathbb{C}_1' \,\mathsf{in}\, \mathbb{C}_2).$$

  Invoking IH for $\mathbb{C}_1'$ yields the result.
- Assume $\theta \equiv \mathsf{ref}(\theta')$. Again we reason by induction on the structure of $\mathbb{C}_1$.

  - Suppose $\mathbb{C}_1 \equiv \mathsf{mkvar}(\lambda u.\mathbb{C}_1', \lambda v.\mathbb{C}_1'')$. Note that then $\mathsf{let}\, z = \mathbb{C}_1 \,\mathsf{in}\, \mathbb{C}_2$ is equivalent to $\mathbb{C}_2[\mathsf{mkvar}(\lambda u.\mathbb{C}_1', \lambda v.\mathbb{C}_1'')/z]$ (β-reduction for values). Since $\mathbb{C}_2$ is canonical, z can only occur in it in subterms of the shape $\mathsf{let}\, y = !z \,\mathsf{in}\, \mathbb{C}''$ or $\mathsf{let}\, y = (z := \mathbb{C}') \,\mathsf{in}\, \mathbb{C}''$. Note that $!\mathsf{mkvar}(\lambda u.\mathbb{C}_1', \lambda v.\mathbb{C}_1'')$ is equivalent to $\mathbb{C}_1'$, whereas $\mathsf{mkvar}(\lambda u.\mathbb{C}_1', \lambda v.\mathbb{C}_1'') := \mathbb{C}'$ to $\mathsf{let}\, v =_{\mathsf{unit}} \mathbb{C}' \,\mathsf{in}\, \mathbb{C}_1''$, which has an equivalent canonical counterpart, say, $\mathbb{C}_1'''$, by the outer IH. Consequently, the only obstacles to proving the Lemma for $\mathsf{let}\, z = \mathbb{C}_1 \,\mathsf{in}\, \mathbb{C}_2$ are terms of the form $\mathsf{let}\, y =_{\theta'} \mathbb{C}_1' \,\mathsf{in}\, \mathbb{C}$ or $\mathsf{let}\, y =_{\mathsf{unit}} \mathbb{C}_1''' \,\mathsf{in}\, \mathbb{C}_2$. These can be successively removed by appealing to the outer IH.
  - If $\mathbb{C}_1 \equiv \mathsf{if}\, \mathbb{C}_1' \,\mathsf{then}\, \mathbb{C}_2' \,\mathsf{else}\, \mathbb{C}_3'$ or $\mathbb{C}_1 \equiv \mathsf{let}\, y = \dots \,\mathsf{in}\, \mathbb{C}_1'$ then we can repeat the previous arguments from the case $\theta \equiv \mathsf{unit}, \mathsf{int}$.

- Assume $\theta \equiv \theta_1 \to \theta_2$. Again we shall rely by induction on the structure of $\mathbb{C}_1$.

  - Suppose $\mathbb{C}_1 \equiv \lambda x^{\theta_1}.\mathbb{C}_1'$. Note that then $\mathsf{let}\, z = \mathbb{C}_1 \,\mathsf{in}\, \mathbb{C}_2$ is equivalent to $\mathbb{C}_2[\lambda x.\mathbb{C}_1'/z]$ (β-reduction for values). Since $\mathbb{C}_2$ is canonical, z can only occur in it in subterms of the shape $\mathsf{let}\, y = z\, \mathbb{C}' \,\mathsf{in}\, \mathbb{C}''$. Hence, following the substitution, we shall obtain subterms of the form $\mathsf{let}\, y = (\lambda x.\mathbb{C}'_1)\mathbb{C}' \,\mathsf{in}\, \mathbb{C}''$. Note that $(\lambda x.\mathbb{C}'_1)\mathbb{C}'$ is equivalent to $\mathsf{let}\, x =_{\theta_1} \mathbb{C}' \,\mathsf{in}\, \mathbb{C}'_1$, which by (outer) IH (for θ_1) is equivalent to a canonical form, say, $\mathbb{C}'''$. Thus we can transform each subterm mentioned above into $\mathsf{let}\, y =_{\theta_2} \mathbb{C}''' \,\mathsf{in}\, \mathbb{C}''$. But, by outer IH for θ_2, such subterms also have canonical counterparts.

– If $\mathbb{C}_1 \equiv \mathsf{if}\, \mathbb{C}'_1 \,\mathsf{then}\, \mathbb{C}'_2 \,\mathsf{else}\, \mathbb{C}'_3$ or $\mathbb{C}_1 \equiv \mathsf{let}\, y = \ldots \,\mathsf{in}\, \mathbb{C}'_1$ then we can repeat the previous arguments from the case $\theta \equiv \mathsf{unit}, \mathsf{int}$.

□

Now we are ready to prove Lemma 8.28 for arbitrary terms.

Proof We shall reason by induction on the syntax.

- $\mathbb{C}_{()} \equiv ()$
- $\mathbb{C}_i \equiv \mathsf{let}\, y = i \,\mathsf{in}\, y$
- $\mathbb{C}_x$ (Lemma 8.29)
- $\mathbb{C}_{M_1 \oplus M_2} \equiv \mathsf{let}\, y = \mathbb{C}_{M_1} \oplus \mathbb{C}_{M_2} \,\mathsf{in}\, y$
- $\mathbb{C}_{\mathsf{if}\, M \,\mathsf{then}\, M_1 \,\mathsf{else}\, M_2} \equiv \mathsf{if}\, \mathbb{C}_M \,\mathsf{then}\, \mathbb{C}_{M_1} \,\mathsf{else}\, \mathbb{C}_{M_2}$
- $\mathbb{C}_{\lambda x.M} \equiv \lambda x.\mathbb{C}_M$
- $\mathbb{C}_{M := N}$ can be obtained by applying Lemma 8.30 to $\mathsf{let}\, x = \mathbb{C}_M \,\mathsf{in}\, (\mathsf{let}\, y = (x := \mathbb{C}_N) \,\mathsf{in}\, \mathbb{C}_y)$.
- $\mathbb{C}_{!M}$ can be obtained by applying Lemma 8.30 to $\mathsf{let}\, x = \mathbb{C}_M \,\mathsf{in}\, (\mathsf{let}\, y = !x \,\mathsf{in}\, \mathbb{C}_y)$.
- $\mathbb{C}_{MN}$ can be obtained by applying Lemma 8.30 to $\mathsf{let}\, x = \mathbb{C}_M \,\mathsf{in}\, (\mathsf{let}\, y = \mathbb{C}_N \,\mathsf{in}\, \mathsf{let}\, z = x\, \mathbb{C}_y \,\mathsf{in}\, \mathbb{C}_z)$.
- $\mathbb{C}_{\mathsf{ref}(M)} \equiv \mathsf{let}\, y = \mathsf{ref}(\mathbb{C}_M) \,\mathsf{in}\, \mathbb{C}_y$
- $\mathbb{C}_{\mathsf{mkvar}(M,N)}$ is obtained by applying Lemma 8.30 to $\mathsf{let}\, x = \mathbb{C}_M \,\mathsf{in}\, \mathsf{let}\, y = \mathbb{C}_N \,\mathsf{in}\, \mathsf{mkvar}(\mathbb{C}_x, \mathbb{C}_y)$.

□

It turns out that canonical subterms of canonical terms have types drawn from a rather restricted set. We make this statement precise below.

Definition 8.31 Given a type θ, the sets $\mathsf{PST}(\theta)$ (of positive subtypes of θ) and $\mathsf{NST}(\theta)$ (of negative subtypes of θ) are defined respectively as follows. Let us write $\mathsf{ST}(\theta)$ for $\mathsf{PST}(\theta) \cup \mathsf{NST}(\theta)$.

$$\begin{aligned}
\mathsf{PST}(\mathsf{unit}) &= \{\mathsf{unit}\} & \mathsf{PST}(\mathsf{ref}(\theta)) &= \mathsf{ST}(\theta) \cup \{\mathsf{ref}(\theta)\}\\
\mathsf{PST}(\mathsf{int}) &= \{\mathsf{int}\} & \mathsf{PST}(\theta_1 \to \theta_2) &= \mathsf{NST}(\theta_1) \cup \mathsf{PST}(\theta_2) \cup \{\theta_1 \to \theta_2\}\\
\mathsf{NST}(\mathsf{unit}) &= \emptyset & \mathsf{NST}(\mathsf{ref}(\theta)) &= \mathsf{ST}(\theta)\\
\mathsf{NST}(\mathsf{int}) &= \emptyset & \mathsf{NST}(\theta_1 \to \theta_2) &= \mathsf{PST}(\theta_1) \cup \mathsf{NST}(\theta_2)
\end{aligned}$$

Given a canonical form $\mathbb{C}$ such that $\Gamma \vdash \mathbb{C} : \theta$, let $\mathsf{RT}(\mathbb{C})$ stand for the set of types θ' such that $\mathbb{C}$ contains an occurrence of $\mathsf{ref}(\mathbb{C}')$, where $\mathbb{C}'$ is of type θ'. It turns out that the types in $\mathsf{RT}(\mathbb{C})$ together with types present in the original typing judgment determine types of canonical subterms, as made precise below.

Lemma 8.32 *Suppose $\Gamma \vdash \mathbb{C} : \theta$. Let*

$$\begin{aligned}\mathbf{L} &= (\textstyle\bigcup_{(x:\theta_x)\in\Gamma} \mathsf{PST}(\theta_x)) \cup \mathsf{NST}(\theta) \cup (\bigcup_{\theta_r \in \mathsf{RT}(\mathbb{C})} \mathsf{ST}(\mathsf{ref}(\theta_r))) \cup \{\mathsf{unit}, \mathsf{int}\},\\ \mathbf{R} &= (\textstyle\bigcup_{(x:\theta_x)\in\Gamma} \mathsf{NST}(\theta_x)) \cup \mathsf{PST}(\theta) \cup (\bigcup_{\theta_r \in \mathsf{RT}(\mathbb{C})} \mathsf{ST}(\theta_r)) \cup \{\mathsf{unit}, \mathsf{int}\}.\end{aligned}$$

Then, for any subterm $\mathbb{C}'$ of $\mathbb{C}$ which is also in canonical form, we have $\Gamma' \vdash \mathbb{C}' : \theta'$, where $\mathsf{cod}(\Gamma') \subseteq \mathbf{L}$ and $\theta' \in \mathbf{R}$.

The proof of this lemma is postponed to after the next corollary.

Corollary 8.33 *Suppose $\Gamma \vdash \mathbb{C} : \theta$, $\mathsf{RT}(\mathbb{C}) = \{\mathsf{int}, \mathsf{unit} \to \mathsf{unit}\}$ and Γ, θ are* ref*-free. Then $\mathbb{C}$ does not contain any occurrences of* mkvar.

Proof Because $\mathsf{RT}(\mathbb{C}) = \{\mathsf{int}, \mathsf{unit} \to \mathsf{unit}\}$, by Lemma 8.32, $\mathbb{C}$ can only contain mkvar if $(\bigcup_{(x:\theta_x)\in\Gamma} \mathsf{NST}(\theta_x)) \cup \mathsf{PST}(\theta)$ contains a ref-type. Since Γ and θ are ref-free this cannot be the case. □

This completes a syntactic proof of Theorem 8.18.

Remark 8.34 Note that Lemma 8.27 may reintroduce fixed points into the language, because it relies on numerical operations defined in PCF. We can still reduce terms containing such definitions to canonical form by assuming that the required operations are primitive (represented by $\oplus$). If this is not desirable then, after the elimination of mkvar under the above assumption, we can put back the PCF definitions without jeopardizing the result (mkvar is not available in PCF).

It remains to prove Lemma 8.32. Given $\mathsf{X} \in \{\mathsf{P}, \mathsf{N}\}$, we shall write $\mathsf{XST}(\theta)$ to refer to $\mathsf{PST}(\theta)$ or $\mathsf{NST}(\theta)$. Moreover, we assume $\overline{\mathsf{P}} = \mathsf{N}$ and $\overline{\mathsf{N}} = \mathsf{P}$. Then the definitions of $\mathsf{PST}(\theta)$ and $\mathsf{NST}(\theta)$ imply the following result.

Lemma 8.35 • *If $\mathsf{ref}(\theta) \in \mathsf{XST}(\theta)$ then $\theta \in \mathsf{PST}(\theta)$ and $\theta \in \mathsf{NST}(\theta)$.*
• *If $\theta_1 \to \theta_2 \in \mathsf{XST}(\theta)$ then $\theta_1 \in \overline{\mathsf{X}}\mathsf{ST}(\theta)$ and $\theta_2 \in \mathsf{XST}(\theta)$.*

The Lemma entails the following properties of **L** and **R**.

Lemma 8.36 *1. If $\mathsf{ref}(\theta) \in \mathbf{L}$ then $\theta \in \mathbf{L}$ and $\theta \in \mathbf{R}$.*
2. If $\mathsf{ref}(\theta) \in \mathbf{R}$ then $\theta \in \mathbf{L}$ and $\theta \in \mathbf{R}$.
3. If $\theta_1 \to \theta_2 \in \mathbf{L}$ then $\theta_1 \in \mathbf{R}$ and $\theta_2 \in \mathbf{L}$.
4. If $\theta_1 \to \theta_2 \in \mathbf{R}$ then $\theta_1 \in \mathbf{L}$ and $\theta_2 \in \mathbf{R}$.

Now we prove Lemma 8.32.

Proof We reason by induction on the number $|\mathbb{C}| - |\mathbb{C}'|$, where $\mathbb{C}'$ ranges over subterms of $\mathbb{C}$ in canonical form. Note that this amounts to showing that the property is inherited by subterms in canonical form.

In the base case, namely $\mathbb{C}' \equiv \mathbb{C}$, the Lemma is valid, because $\Gamma \vdash \mathbb{C} : \theta$ and, for any $(x : \theta_x) \in \Gamma$, we have $\theta_x \in \mathsf{PST}(\theta_x) \subseteq \mathbf{L}$ as well as $\theta \in \mathsf{PST}(\theta) \subseteq \mathbf{R}$.

Many of the inductive cases take advantage of the closure properties relating **L** and **R** from the preceding Lemma.

$\mathbb{C}' \equiv \mathsf{mkvar}\,(\lambda u^{\mathsf{unit}}.\mathbb{C}_1)\,(\lambda v.\mathbb{C}_2)$: Since canonical forms do not contain variables of type unit, for $\mathbb{C}_1$ the lemma follows from IH and the fact that $\mathsf{ref}(\theta) \in \mathbf{R}$ implies $\theta \in \mathbf{R}$. For $\mathbb{C}_2$ we need to appeal to the fact that $\mathsf{ref}(\theta) \in \mathbf{R}$ implies $\theta \in \mathbf{L}$, and that $\mathsf{unit} \in \mathbf{R}$.

$\mathbb{C}' \equiv \lambda x^{\theta}.\mathbb{C}_1$: Here, in addition to invoking IH we need to observe that $\theta_1 \to \theta_2 \in \mathbf{R}$ implies $\theta_1 \in \mathbf{L}$ and $\theta_2 \in \mathbf{R}$.

$\mathbb{C}' \equiv \mathsf{if}\,\mathbb{C}_1\,\mathsf{then}\,\mathbb{C}_2\,\mathsf{else}\,\mathbb{C}_3$: For $\mathbb{C}_2$ and $\mathbb{C}_3$, the Lemma follows from the inductive hypothesis applied to $\mathbb{C}'$. For $\mathbb{C}_1$, we additionally need to observe that $\mathsf{int} \in \mathbf{R}$.

$\mathbb{C}' \equiv \mathsf{let}\,y = i\,\mathsf{in}\,\mathbb{C}_1$: The result for $\mathbb{C}_1$ follows from IH and $\mathsf{int} \in \mathbf{L}$.

$\mathbb{C}' \equiv \mathsf{let}\,y = \mathbb{C}_1 \oplus \mathbb{C}_2\,\mathsf{in}\,\mathbb{C}_3$: For $\mathbb{C}_1$ and $\mathbb{C}_2$ we appeal to IH and $\mathsf{int} \in \mathbf{R}$. For $\mathbb{C}_3$, we need IH and $\mathsf{int} \in \mathbf{L}$.

$\mathbb{C}' \equiv \mathsf{let}\,y = !x\,\mathsf{in}\,\mathbb{C}_1$: Note that $\mathsf{ref}(\theta'') \in \mathbf{L}$ implies $\theta'' \in \mathbf{L}$.

$\mathbb{C}' \equiv \mathsf{let}\,y = (x := \mathbb{C}_1)\,\mathsf{in}\,\mathbb{C}_2$: IH suffices for $\mathbb{C}_2$. For $\mathbb{C}_1$, invoke IH and observe that $\mathsf{ref}(\theta'') \in \mathbf{L}$ implies $\theta'' \in \mathbf{R}$.

$\mathbb{C}' \equiv \mathsf{let}\,y = x\,\mathbb{C}_1\,\mathsf{in}\,\mathbb{C}_2$: Observe that $\theta_1 \to \theta_2 \in \mathbf{L}$ entails $\theta_1 \in \mathbf{R}$ and $\theta_2 \in \mathbf{L}$. Then invoke IH.

$\mathbb{C}' \equiv \mathsf{let}\,y = \mathsf{ref}(\mathbb{C}_1)\,\mathsf{in}\,\mathbb{C}_2$: Let $\mathsf{ref}(\theta_1)$ be the type of $\mathsf{ref}(\mathbb{C}_1)$. Then $\theta_1 \in \mathsf{RT}(\mathbb{C})$ and thus $\mathsf{ref}(\theta_1) \in \mathbf{L}$. From this the lemma follows for $\mathbb{C}_2$. For $\mathbb{C}_1$, observe that $\mathsf{ref}(\theta_1) \in \mathbf{L}$ implies $\theta_1 \in \mathbf{R}$.

□

8.6 When Integer References Suffice

Next we shall examine the conditions under which references of type $\mathsf{unit} \to \mathsf{unit}$ can also be eliminated, i.e. all uses of general references can be replaced with a single integer-valued memory cell. In technical terms, this requires us to characterize the prearenas where plays are guaranteed to satisfy visibility.

Definition 8.37 Let A be a prearena and $m_1, m_2 \in M_A$. We shall say that m_1 and m_2 are *computationally equireachable* if there are paths ms_1m_1 and ms_2m_2 in the graph $(M_A, \vdash_A)$ such that m is initial and, if s_1 and s_2 both start with an answer, say a_1 and a_2 respectively, then $a_1 = a_2$.

Definition 8.38 A prearena A is called ***visible*** if there are no computationally equireachable non-initial moves $m, m' \in M_A$ such that m is an O-question and m' enables a P-question.

Lemma 8.39 *Let A be a prearena such that each question enables an answer*[2]. *All plays of A satisfy the visibility condition if and only if A is visible.*

[2] All prearenas corresponding to typing judgments are of this kind.

Proof Let s be a play of A that violates the visibility condition. Suppose further that s ends in the P-move p_2, which breaks visibility for the first time and let o_1 be its justifier. Then, since s breaks visibility at p_2, it must look like:

$$m \cdots p_1 \cdots o_1 \cdots o_2 \cdots p_2$$

for some initial move m, where o_2 appears in the view before p_2 and where p_2 is a question. Observe also that, since p_2 violates visibility, its justifier o_1 cannot be initial. If o_2 is a question we are done: A is not visible because of $(m, m') = (o_2, o_1)$. So, suppose that o_2 is an answer. Then p_1 is a question and the move o_2' immediately following it in s is also a question (otherwise it would answer p_1). Moreover, o_2' is not initial (because it follows p_1). Consequently, A is not visible due to $(m, m') = (o_2', o_1)$.

Conversely, suppose that A is not visible and let the latter be witnessed by paths $m s_1 p_1 o_2$ and $m s_2 o_1 p_2$ in $(M_A, \vdash_A)$. We form a play s as follows.

- If s_2 does not start with an answer, we set

$$s \;=\; m\, s_1\, p_1\, o_2\, s_2\, o_1\, p_2\, o_2\, p_2\,.$$

- If $s_1 p_1, s_2$ both start with an answer, say $s_1 p_1 = a s_1'$ and $s_2 = a s_2'$, we set

$$s \;=\; m\, a\, s_1'\, s_2'\, o_1\, p_2\, o_2\, p_2$$

  where the leftmost pointer points to the last move of s_1'.
- If s_2 starts with an answer but s_1 does not, we set

$$s \;=\; m\, s_1\, p_1\, s_1'\, s_2\, o_1\, p_2\, o_2\, p_2$$

  where s_1' is a sequence of moves answering all open questions of $s_1 p_1$.

Now observe that, in each case, the play s breaks visibility at move p_2. □

As a next step we would like to understand what typing judgments give rise to visible prearenas. Our answer will be phrased in terms of syntactic shape. For simplicity, we shall now restrict our discussion to types generated from unit (Remark 8.41 explores the consequences of the results for the full type system). The following two lemmas capture scenarios relevant to verifying visibility for prearenas. We write Θ_1 for the collection of first-order types, generated by the grammar $\Theta_1 ::= \mathsf{unit} \mid \mathsf{unit} \to \Theta_1$. Similarly, $\Theta_1 \to \mathsf{unit}$ stands for $\{\theta_1 \to \mathsf{unit} \mid \theta_1 \in \Theta_1\}$.

Lemma 8.40 *Let* $A = [\![\theta_1, \ldots, \theta_k \vdash \theta]\!]$, *where* $\theta_1, \ldots, \theta_k, \theta$ *are generated from* unit.

- *All O-questions in A are initial iff* $\theta_i \in \Theta_1$ *for all* $1 \le i \le k$ *and* $\theta = \mathsf{unit}$.
- *A does not contain a P-question enabled by a non-initial O-move iff* $\theta_i \in \{\mathsf{unit}\} \cup (\Theta_1 \to \mathsf{unit})$ *for* $1 \le i \le k$ *and* $\theta \in \Theta_1$.

Consequently, A is visible if and only if one of the conditions above is satisfied.

Remark 8.41 To see whether any occurrences of ref-types generate visible prearenas, recall that $[\![\mathsf{ref}(\theta)]\!] = [\![\theta \to \mathsf{unit}]\!] \times [\![\mathsf{unit} \to \theta]\!]$. Consequently, for the purpose of determining visibility $\mathsf{ref}(\mathsf{unit})$ can be viewed as $\mathsf{unit} \to \mathsf{unit}$. Thus, $\mathsf{ref}(\mathsf{unit})$ can be used whenever $\mathsf{unit} \to \mathsf{unit}$ is allowed. Note also that it is immaterial whether we consider unit or int, because the notion of a visible prearena is based on paths and does not depend on the number of answers to questions. The observations yield the following typing constraints for visible prearenas: ($\theta_i ::= \beta \mid \mathsf{ref}(\beta) \mid \Theta_1 \to \beta$ and $\theta ::= \Theta_1$) or ($\theta_i ::= \Theta_1$ and $\theta ::= \beta$), where $\beta ::= \mathsf{unit} \mid \mathsf{int}$ and $\Theta_1 ::= \beta \mid \mathsf{ref}(\beta) \mid \beta \to \Theta_1$. Analogously, $\mathsf{ref}(\beta \to \beta)$ should be viewed as a combination of $(\beta \to \beta) \to \beta$ and $\beta \to \beta \to \beta$. The results above do not give us much room for using this type: it cannot occur on the right but, if $\theta \equiv \beta$ we can have $\theta_i \equiv \mathsf{ref}(\beta \to \beta)$.

Thanks to Theorems 8.15 and 8.16 we can derive:

Theorem 8.42 *Let $\Gamma \vdash \theta$ be such that $[\![\Gamma \vdash \theta]\!]$ is visible. For any $\Gamma \vdash M : \theta$, there exists $\Gamma, y : \mathsf{ref}(\mathsf{int}) \vdash M' : \theta$ such that the following conditions are satisfied.*

- $\Gamma \vdash M \cong \mathsf{let}\, y = \mathsf{ref}(0)\, \mathsf{in}\, M'$.
- *M' is ref-free.*
- *If $\Gamma \vdash \theta$ does not contain occurrences of ref, then M' is mkvar-free.*

Next we give several examples of terms in which uses of $\mathsf{ref}(\mathsf{unit} \to \mathsf{unit})$ are definitely not eliminable. This is because the terms generate plays that violate the visibility condition, to be contrasted with Proposition 8.11.

Example 8.43 The first example is simply $\vdash \mathsf{new}_{\mathsf{unit}\to\mathsf{unit}} : \mathsf{ref}(\mathsf{unit} \to \mathsf{unit})$. Its semantics contains the play

$$\star\ \circ\ \mathsf{write}\ \mathsf{ok}\ \mathsf{read}\ \star\ q_r\ \ q_w\ .$$

Below we list other examples featuring types that extend those from Lemma 8.40 in various ways.

$$\vdash \mathsf{let}\, x, y = \mathsf{new}_{\mathsf{unit}\to\mathsf{unit}}, \mathsf{new}_{\mathsf{int}}\, \mathsf{in}$$
$$\lambda f^{\mathsf{unit}\to\mathsf{unit}}.\ (\mathsf{if}\ (!y = 0)\ \mathsf{then}\ (y := 1; x := f)\ \mathsf{else}\ ()); (!x)() : (\mathsf{unit} \to \mathsf{unit}) \to \mathsf{unit}$$

$$g : ((\mathsf{unit} \to \mathsf{unit}) \to \mathsf{unit}) \to \mathsf{unit} \vdash \mathsf{let}\, x, y = \mathsf{new}_{\mathsf{unit}\to\mathsf{unit}}, \mathsf{new}_{\mathsf{int}}\, \mathsf{in}$$
$$g(\lambda f^{\mathsf{unit}\to\mathsf{unit}}.\ (\mathsf{if}\ (!y = 0)\ \mathsf{then}\ (y := 1; x := f)\ \mathsf{else}\ ()); (!x)()) : \mathsf{unit}$$

$$g : \mathsf{unit} \to \mathsf{unit} \to \mathsf{unit} \vdash \mathsf{let}\, x, y = \mathsf{new}_{\mathsf{unit}\to\mathsf{unit}}, \mathsf{new}_{\mathsf{int}}\, \mathsf{in}$$
$$\lambda u^{\mathsf{unit}}.(\mathsf{if}\ (!y = 0)\ \mathsf{then}\ (y := 1; x := g())\ \mathsf{else}\ ()); (!x)() : \mathsf{unit} \to \mathsf{unit}$$

$$g : (\mathsf{unit} \to \mathsf{unit}) \to \mathsf{unit} \to \mathsf{unit} \vdash$$
$$\mathsf{let}\, x = \mathsf{new}_{\mathsf{unit}\to\mathsf{unit}}\, \mathsf{in}\, (\, x := g(\lambda z^{\mathsf{unit}}.(!x)())\,);\ (!x)() : \mathsf{unit}$$

For yet another example, we refer the reader to Proposition 5.7 of Clairambault (2012).

8.7 When All References are Dispensible

Finally, at some types memory allocation turns out dispensible, i.e. there exist purely functional terms with equivalent observable behaviour. In game-semantic terms, these are types where all strategies are necessarily innocent Hyland & Ong (2000).

Definition 8.44 Let A be a prearena such that any question enables an answer. A is called ***innocent*** if all O-questions are initial.

Remark 8.45 Let us observe that $[\![\theta_1, \ldots, \theta_k \vdash \theta]\!]$ is an innocent prearena if and only if $\theta_i ::= \Theta_1$ and $\theta ::= \beta$.

Lemma 8.46 *Let A be a prearena such that any question enables an answer. A is innocent if and only if every strategy $\sigma : A$ is innocent.*

Proof Suppose A is not innocent, i.e. there exists a non-initial O-question q_O. Let s be the chain of enablers leading from some initial move to q_O and let a_P be an answer to q_O. Then the strategy on A consisting of even-length prefixes of sa_P is not innocent, because it will not contain $sa_P q_O a_P$. Thus, not all strategies in A are innocent.

Now assume that A is innocent. Consequently, all non-initial O-moves are answers. Thus, each odd-length play s in A must have the shape $q(qa)^*$. Consequently, $\mathit{view}(s) = s$ and each strategy on A is thus innocent. $\square$

The following result then follows from Theorem 8.16.

Theorem 8.47 *Suppose $\Gamma \vdash M : \theta$ is such that $[\![\Gamma \vdash M : \theta]\!]$ is innocent. Then there exists $\Gamma \vdash M' : \theta$ satisfying all the conditions below.*

- $\Gamma \vdash M \cong M'$.
- *M' is* ref*-free.*
- *If there are no occurrences of* ref*-types in $\Gamma \vdash \theta$, then M' is* mkvar*-free.*

Example 8.48 Here are two examples of terms not covered by Theorem 8.47, i.e. terms that do not have purely functional counterparts, because the corresponding strategies are not innocent. Note that they underlying types deviate from the shapes identified in Remark 8.45 in different ways.

$$\vdash \mathsf{let}\, y = \mathsf{new}_{\mathsf{int}}\, \mathsf{in}\, \lambda z^{\mathsf{unit}}.\mathsf{if}\, (!y = 0)\, \mathsf{then}\, y := 1\, \mathsf{else}\, \Omega : \mathsf{unit} \to \mathsf{unit}$$

$$g : (\mathsf{unit} \to \mathsf{unit}) \to \mathsf{unit} \vdash \mathsf{let}\, y = \mathsf{new}_{\mathsf{int}}\, \mathsf{in}$$
$$g(\lambda z^{\mathsf{unit}}.\mathsf{if}\, (!y = 0)\, \mathsf{then}\, y := 1\, \mathsf{else}\, \Omega) : \mathsf{unit}$$

8.8 Conclusion

We showed that general references in $\mathcal{L}$ can be simulated with two reference cells, of types $\mathsf{ref}(\mathsf{unit} \to \mathsf{unit})$ and $\mathsf{ref}(\mathsf{int})$ respectively. This was first demonstrated through a game-semantic argument and subsequently complemented by a syntactic recipe for program transformation. The latter was facilitated by the presence of the mkvar constructor. However, the results apply equally well to the mkvar-free framework, provided no reference types occur in the type of the term or those of its free identifiers (arbitrary internal uses are still allowed). Then the auxiliary occurrences of mkvar can actually be eliminated, so, in this context, mkvar can be viewed as a useful temporary addition to the language.

Theorems 8.18, 8.42 and 8.47 all exhibit certain effective program transformations whose existence was shown via semantic arguments based on game semantics. While for Theorem 8.18 we managed to find an alternative syntactic argument, it is not clear to us how to do the same in the other two cases.

In the future, we would like to conduct a similar study using the nominal game model of Murawski & Tzevelekos (2011). In the nominal setting, decomposition results such as Lemmata 8.23 and 8.24 cannot be expected to hold. Another surprising challenge is that the obvious adaptation of the visibility condition fails to be preserved by composition.

Acknowledgements We are grateful to the anonymous referees for constructive feedback.

References

Abramsky, S., Honda, K., & McCusker, G. (1998). Fully abstract game semantics for general references. In *Proceedings of LICS* (pp. 334–344). Computer Society Press.

Abramsky, S., Jagadeesan, R., & Malacaria, P. (2000). Full abstraction for PCF. *Information and Computation, 163,* 409–470.

Abramsky, S., & McCusker, G. (1997). Call-by-value games. In *Proceedings of CSL* (Vol. 1414, pp. 1–17). Lecture notes in computer science. Berlin: Springer.

Abramsky, S., & McCusker, G. (1997). Linearity, sharing and state: A fully abstract game semantics for Idealized Algol with active expressions. In P. W. O'Hearn, & R. D. Tennent (Eds.), *Algol-like languages* (pp. 297–329). Birkhaüser.

Bruce, K. B., Cardelli, L., & Pierce, B. C. (1999). Comparing object encodings. *Information and Computation, 155*(1–2), 108–133.

Clairambault, P. (2012). Isomorphisms of types in the presence of higher-order references (extended version). *Logical Methods in Computer Science, 8*(3).

Honda, K., & Yoshida, N. (1999). Game-theoretic analysis of call-by-value computation. *Theoretical Computer Science, 221*(1–2), 393–456.

Hyland, J. M. E., & Ong, C.-H. L. (2000). On full abstraction for PCF: I. Models, observables and the full abstraction problem, II. Dialogue games and innocent strategies, III. A fully abstract and universal game model. *Information and Computation, 163*(2), 285–408.

Murawski, A. S., & Tzevelekos, N. (2011). Game semantics for good general references. In *Proceedings of LICS* (pp. 75–84). IEEE Computer Society Press.

Murawski, A. S., & Tzevelekos, N. (2013). Deconstructing general references via game semantics. In *Proceedings of FoSSaCS* (Vol. 1414, pp. 241–256). Lecture notes in computer science. Berlin: Springer.

Reynolds, J. C. (1981). The essence of Algol. In J. W. de Bakker, & J. C. van Vliet (Eds.), Algorithmic languages (pp. 345–372). North Holland

Sanjabi, S. B., & C. -H. L. Ong. (2007). Fully abstract semantics of additive aspects by translation. In *Proceedings of AOSD* (pp. 135–148). ACM.

Chapter 9
The Mays and Musts of Concurrent Strategies

Simon Castellan, Pierre Clairambault, and Glynn Winskel

Abstract Concurrent strategies based on event structures are examined from the viewpoint of 'may' and 'must' testing in traditional process calculi. In their pure form concurrent strategies fail to expose the deadlocks and divergences that can arise in their composition. This motivates an extension of the bicategory of concurrent strategies to treat the 'may' and 'must' behaviour of strategies under testing. One extension adjoins neutral moves to strategies but in so doing loses identities w.r.t. composition. This in turn motivates another extension in which concurrent strategies are accompanied by stopping configurations; the ensuing stopping strategies inherit the structure of a bicategory from that of strategies. The technical developments converge in providing characterisations of the 'may' and 'must' equivalences and preorders on strategies.

Keywords Event structures · Concurrent games and strategies · 'may' and a must' equivalence on strategies · Testing equivalence · Stopping configurations

9.1 Introduction

This article relates to work on process calculi of the 1980s but from a modern perspective of processes as strategies, specifically as distributed/concurrent strategies based on event structures. It expands on two areas close to Samson Abramsky's heart, game semantics and concurrency: on a development of concurrent games based on

S. Castellan
Inria, University Rennes, IRISA, France
e-mail: simon.castellan@inria.fr

P. Clairambault
LIS, Aix Marseille Univ, Universitè de Toulon, CNRS, 13288 Marseille Cedex 09, France
e-mail: pierre.clairambault@cnrs.fr

G. Winskel (✉)
Computer and Information Sciences, University of Strathclyde, Glasgow, Scotland
e-mail: glynnwinskel@icloud.com

A. Palmigiano and M. Sadrzadeh (eds.), *Samson Abramsky on Logic and Structure in Computer Science and Beyond*, Outstanding Contributions to Logic 25,
https://doi.org/10.1007/978-3-031-24117-8_9

event structures which extends his early ideas with Paul-André Melliès of deterministic concurrent strategies as closure operators (Abramsky & Melliès, 1999); and equivalences on concurrent processes through testing (Abramsky, 1987).

Robin Milner and Tony Hoare's work of late seventies and early eighties drew attention to equivalences on processes; Milner's on forms of *bisimulation* (Milner, 1980) and Hoare's on *failures* equivalence (Brookes et al., 1984). Hoare had described failure equivalence informally as the minimum extension of trace equivalence that takes account of the possibility of failure due to deadlock. Matthew Hennessy and his PhD student Rocco de Nicola provided a rationale through an idea of testing processes (De Nicola & Hennessy, 1984). For them a test was a process with distinguished "success" states at which an action ✓ could occur. Putting a test in parallel composition with a process, *may* lead to success if some run does or *must* lead to success if all runs do. Processes can be regarded as equivalent if they have the same 'may' and 'must' behaviour w.r.t. tests. Modulo subtleties to do with the divergence of processes, Hennessy and de Nicola recovered failure equivalence as testing equivalence. What about Milner's central equivalence? Samson Abramsky investigated the extent to which bisimulation could be viewed as a testing equivalence (Abramsky, 1987): it could, but only at the cost of strengthening the power of tests considerably, by allowing testing to run and copy processes quite liberally.

Here we shall examine the 'may' and 'must' equivalence of concurrent strategies based on event structures (Rideau & Winskel, 2011; Castellan et al., 2017)—foreshadowed in the early definitions of concurrent strategy (Abramsky & Melliès, 1999; Ghica & Murawski, 2004; Melliès & Mimram, 2007; Faggian & Piccolo, 2009). Informally, a *strategy* for Player in a two-party game against Opponent, expresses a choice of Player moves, most often in reaction to moves made by Opponent, unpredictable for Player but for the constraints of the game. We shall implicitly regard a strategy as a strategy for Player. We regard Opponent as the environment uncontrollable by Player. We can express both the game—its moves and their constraints—and a strategy—its choice of Player moves subject to the moves of Opponent—as event structures. This chimes with our view of strategies and games as highly distributed. Player and Opponent are more accurately thought of as teams of players and opponents acting at possibly very different locations. Though we take the rather abstract view of location advocated by Petri in his concept of local state as a condition (or place): then locality reveals itself through the causal dependence and independence of events.

Event structures are the concurrent analogue of trees; just as transition systems unfold to trees, so Petri nets unfold to event structures. Whereas an unfolded behaviour of a transition system comprises sequences of actions/events, the unfolded behaviour of a Petri net, in which events make local changes to conditions, comprises partial orders of causal dependency between event occurrences (Nielsen et al., 1981). Event structures are a central model for concurrent computation, related to other models by adjunctions (Winskel & Nielsen, 1995). This plants concurrent strategies based on event structures firmly within theories of concurrency and interaction—anticipated in Abramsky's presentation of game semantics, with its emphasis on composition of strategies as given by their parallel interaction followed by hiding.

Perhaps more controversially, the view of processes as strategies suggests refinements to the assumptions usual in process calculi. In concurrent strategies, gone is the usual symmetry between a process and its environment; the conditions on a concurrent strategy take account of the unpredictability and uncontrollability of Opponent moves. This affects the appropriate equivalences to impose between concurrent strategies.

There is surely a long history behind the idea of composing strategies. Certainly the idea plays a key role in John Conway's "On Numbers and Games" (Conway, 2000), the categorical underpinnings of which were exposed by André Joyal (1997). For two-party games there is the obvious operation of reversing the roles of the two participants, Player and Opponent; this operation, forming the *dual* $G^\perp$ of a game G, played the role of *negation* for Conway. A useful convention is to regard a strategy in a game G as a strategy for Player; then a strategy for Opponent, or counter-strategy, is a strategy in the dual game $G^\perp$. If the games are broad enough, they often support a form of parallel composition, $G \| H$; for Conway it was the *sum* of games. A strategy *from* a game G *to* a game H is a strategy σ *in* the game $G^\perp \| H$. Given another strategy this time from the game H to the game K, *i.e.* a strategy τ in the game $H^\perp \| K$, we can let the strategies *interact* as $\tau \circledast \sigma$, essentially by playing them against each other over the common game H; there the strategies σ and τ adopt complementary roles—where one makes a move of Player in H the other sees a move of Opponent and *vice versa*.

The interaction $\tau \circledast \sigma$ involves moves in the parallel composition of all three games, $G^\perp \| H \| K$, though in writing the parallel composition in this way an imprecision has crept in: whereas the moves over $G^\perp$ and K described by $\tau \circledast \sigma$ are choices of moves for Player or moves open to choices of Opponent, those over H are either instantiations of Opponent moves of σ by Player moves of τ, or the converse, instantiations of Opponent moves of τ by Player moves of σ. As such the moves of $\tau \circledast \sigma$ over H behave like synchronisations between complementary moves of σ and τ, and as events internal to the interaction. Though internal, the events over H can affect the behaviour of the interaction by introducing deadlocks or divergence. In the *composition* of strategies it is usual to hide the internal events of interaction to obtain a strategy $\tau \odot \tau$ in the game $G^\perp \| K$, where the game H is elided to obtain a strategy from G to K. However when the original strategies σ or τ are nondeterministic significant behavioural distinctions can be lost in hiding internal events. In particular, the hidden events can affect the 'must' behaviour of the composition of strategies.

9.1.1 Contributions of the Paper

This brings us to the concerns of this paper. It motivates the definition of *bare* concurrent strategies in which internal events are exposed as neutral moves in the strategy. Through bare strategies we can examine the 'may' and 'must' testing of strategies. Although bare strategies compose their composition does not have identities, so they fail to form a bicategory. We have explored two ways to recover a bicategory while remaining faithful to the ‘must’ behaviour of strategies. One is through “essential events” in which one strips a bare strategy down to just those neutral moves critical to its behaviour (Castellan et al., 2018). The other, that we follow here, is through extending strategies with the extra structure of *stopping configurations* (Castellan et al., 2014). By distinguishing certain configurations as stopping we keep track of those visible configurations at which the strategy may appear to get stuck through the occurrence of hidden neutral moves.

Stopping configurations are the event-structure analogue of Russ Harmer and Guy McCusker’s “divergences” (Harmer & McCusker, 1999), though event structures add the refinement of locality and independence to the concept. In an interleaving model, in which behaviour is captured through sequences of actions, divergence anywhere has a global effect; generally, in the parallel composition of two processes if one can perform an infinite sequence of actions, these may block progress of the other process, unless additional fairness assumptions are enforced. This is not so in a model such as event structures where the independence/concurrency of actions is explicit. In a nondeterministic strategy a Player move is not blocked by the occurrence of moves with which it is independent. Whereas an interleaving model may require weak fairness assumptions these are generally built into the behaviour of strategies as event structures (Winskel, 1980). This makes for subtle differences in the nature of ‘must’ testing in concurrent games w.r.t. traditional games.

Perhaps surprisingly, in many situations a concurrent strategy may be replaced by its simpler “rigid image” in the game, despite this often forgetting nondeterministic branching—rigid-image strategies form a category rather than just a bicategory (Castellan et al., 2017)—and indeed this remains true for strategies with stopping configurations; none of the ‘may’ or ‘must’ behaviour is lost.

Our technical contribution concludes with characterisations of the ‘may’ and ‘must’ equivalences and preorders on strategies. The ‘may’ equivalence of strategies is captured through their inducing the same set of finite traces; a trace being understood as a sequence of moves in the game. This echoes the earlier results of Ghica and Murawski when showing their non-alternating games model is fully-abstract for Idealized Parallel Algol with respect to may-convergence (Ghica & Murawski, 2004). For ‘must’ equivalence, our result is to be compared with that of Harmer and McCusker for their sequential games model, based on Hyland-Ong games with explicit divergences (Harmer & McCusker, 1999). But whereas in a sequential setting only the first divergence matters, for us ‘must’ equivalence of strategies is equivalent to their sharing the same traces of *all* (possibly infinite) stopping configurations. See Example 9.16 and what follows for an in-depth discussion.

Because we restrict attention to linear bicategories of strategies, the tests here are linear too; they do not permit the tested strategy to be copied and rerun. From the point of view of distributed computation linearity is natural: it is often infeasible to copy a distributed system or strategy (Winskel, 2004). Then single-run ‘may’ and ‘must’ tests are appropriate.

On the other hand, most programming languages do allow some form of copying and nonlinearity. Through the addition of symmetry we can adjoin pseudo (co)monads—where the traditional laws hold up to symmetry, and model nonlinear features (Castellan et al., 2014, 2019). Through symmetry and pseudo comonads, we can realise a variety of nonlinear forms of testing, in which a test could dynamically copy and retest the strategy of interest. The nature of such broader testing on strategies, the equivalences and logics induced, are not well understood and deserve a systematic study, for which this paper forms a foundation. As shown by Samson Abramsky such broader tests are needed to realise equivalences such as bisimulation as testing equivalences (Abramsky, 1987).

9.2 Event Structures

An *event structure* comprises $(E, \leq, \mathrm{Con})$, consisting of a set E of *events* which are partially ordered by $\leq$, the *causal dependency relation*, and a nonempty *consistency* relation Con consisting of finite subsets of E. The relation $e' \leq e$ expresses that event e causally depends on the previous occurrence of event e'. That a finite subset of events is consistent conveys that its events can occur together by some stage in the evolution of the process. Together the relations satisfy several axioms. We insist that the partial order is *finitary*, *i.e.*

- $[e] =_{\mathrm{def}} \{e' \mid e' \leq e\}$ is finite for all $e \in E$,

and that consistency satisfies

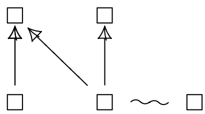
- $\{e\} \in \mathrm{Con}$ for all $e \in E$,
- $Y \subseteq X \in \mathrm{Con}$ implies $Y \in \mathrm{Con}$, and
- $X \in \mathrm{Con}$ & $e \leq e' \in X$ implies $X \cup \{e\} \in \mathrm{Con}$.

There is an accompanying notion of state, or history, those events that may occur up to some stage in the behaviour of the process described. A *configuration* is a, possibly infinite, set of events $x \subseteq E$ which is:

- *consistent,* $X \subseteq x$ and X is finite implies $X \in \mathrm{Con}$; and
- *down-closed,* $e' \leq e \in x$ implies $e' \in x$.

Two events e, e' are called *concurrent* if the set $\{e, e'\}$ is in Con and neither event is causally dependent on the other; then we write $e\ co\ e'$. In games the relation of *immediate* dependency $e \rightarrowtriangle e'$, meaning e and e' are distinct with $e \leq e'$ and no event in between, plays a very important role. We write $[X]$ for the down-closure of a subset of events X. Write $\mathcal{C}^\infty(E)$ for the configurations of E and $\mathcal{C}(E)$ for its finite configurations. (Sometimes we shall need to distinguish the precise event structure to which a relation is associated and write, for instance, $\leq_E$, $\rightarrowtriangle_E$ or co_E.)

Example 9.1 In examples it is often convenient to draw event structures. Often, though not always, consistency is determined in a binary fashion, in that a set of events is consistent if all of its pairs are. Then, it is economical to draw the binary relation of *conflict*, or inconsistency. For example, in the diagram



we illustrate the relations of immediate causal dependency $\rightarrowtriangle$ which yields the Hasse diagram of the partial order of causal dependency between events $\square$, and conflict by the wiggly line $\sim\!\sim\!\sim$. Neither the two events related by $\sim\!\sim\!\sim$ nor their dependants w.r.t. causal dependency can occur together in a configuration; there is no need draw all the conflicts that follow. $\square$

Let E and E' be event structures. A *map* of event structures $f : E \to E'$ is a partial function on events $f : E \rightharpoonup E'$ such that for all $x \in \mathcal{C}^\infty(E)$ its direct image $fx \in \mathcal{C}^\infty(E')$ and

$$\text{if } e_1, e_2 \in x \text{ and } f(e_1) = f(e_2) \text{ (with both defined), then } e_1 = e_2.$$

(Those maps defined is unaffected if we replace possibly infinite configurations $\mathcal{C}^\infty(E)$ by finite configurations $\mathcal{C}(E)$ above; this is because any configuration is the union of finite configurations and direct image preserves such unions.)

Maps of event structures compose as partial functions, with identity maps given by identity functions. Say a map is *total* if the function f is total. Notice that for a total map f the condition on maps now says it is *locally injective*, in the sense that w.r.t. any configuration x of the domain the restriction of f to a function from x is injective; the restriction of f to a function from x to fx is thus bijective. Say a total map of event structures is *rigid* when it preserves causal dependency.

Although a map $f : E \to E'$ of event structures does not generally preserve causal dependency, it does locally reflect causal dependency: whenever $e, e' \in x$, a configuration of E, and $f(e)$ and $f(e')$ are both defined with $f(e') \leq f(e)$, then $e' \leq e$. Consequently, f preserves the concurrency relation: if $e\ co\ e'$ in E and $f(e)$ and $f(e')$ are both defined then $f(e)\ co\ f(e')$.

9.3 Constructions

We provide the constructions which we use in the paper.

9.3.1 Partial-Total Factorisation

We shall realise an operation of hiding events via a factorisation property of maps of event structures.

Let $(E, \leq, \mathrm{Con})$ be an event structure. Let $V \subseteq E$ be a subset of 'visible' events. Define $E{\downarrow}V =_{\mathrm{def}} (V, \leq_V, \mathrm{Con}_V)$, where $v \leq_V v'$ iff $v \leq v'$ & $v, v' \in V$ and $X \in \mathrm{Con}_V$ iff $X \in \mathrm{Con}$ & $X \subseteq V$. The operation projects E to visible events V.

Consider a partial map of event structures $f : E \to E'$. Let

$$V =_{\mathrm{def}} \{e \in E \mid f(e) \text{ is defined}\}\,.$$

Then f clearly factors into the composition

$$E \xrightarrow{f_0} E{\downarrow}V \xrightarrow{f_1} E'$$

of f_0, a partial map of event structures taking $e \in E$ to itself if $e \in V$ and undefined otherwise, and f_1, a total map of event structures acting like f on V. We call f_1 the *defined part* of the partial map f. We say a map $f : E \to E'$ is a *projection* if its defined part is an isomorphism.

The *partial-total factorisation* is characterised to within isomorphism by the following universal property: for any factorisation

$$f : E \xrightarrow{g_0} E_1 \xrightarrow{g_1} E'$$

where g_0 is partial and g_1 is total there is a (necessarily total) unique map $h : E{\downarrow}V \to E_1$ such that

$$\begin{tikzcd} E \arrow[r, "f_0"] \arrow[dr, "g_0"'] & E{\downarrow}V \arrow[r, "f_1"] \arrow[d, dashed, "h"] & E' \\ & E_1 \arrow[ur, "g_1"'] & \end{tikzcd}$$

commutes.

9.3.2 Pullback

Event structures and their maps have pullbacks. For the composition of strategies we shall only need pullbacks of total maps. Consider a pullback

$$\begin{tikzcd} & P \arrow[dl, "\pi_1"'] \arrow[dr, "\pi_2"] \arrow[dd, phantom, "\lrcorner", very near start] & \\ A \arrow[dr, "f"'] & & B \arrow[dl, "g"] \\ & C & \end{tikzcd}$$

where f and g are total. Pullbacks are difficult to construct directly on the "prime" event structures we are using here, essentially because they associate each event with a unique minimum causal history. Such constructs are best first carried out in a broader model. Here we build the pullback of event structures out of the stable family of secured bijections.

Definition 9.2 A *secured bijection* comprises a composite bijection

$$\theta_{x,y} : x \cong fx = gy \cong y$$

between configurations $x \in \mathcal{C}^\infty(A)$ and $y \in \mathcal{C}^\infty(B)$ s.t. $fx = gy$, which is *secured* in the sense that the transitive relation generated on $\theta_{x,y}$ by taking

$$(a, b) \leq (a', b') \text{ if } a \leq_A a' \text{ or } b \leq_B b'$$

is a finitary partial order. Let $\mathcal{B}$ be the family of secured bijections. Say a subset $Z \subseteq \mathcal{B}$ is *compatible* iff $\exists \theta' \in \mathcal{B} \forall \theta \in Z.\ \theta \subseteq \theta'$, and *finitely compatible* iff every finite subset is compatible.

Proposition 9.3 *The family $\mathcal{B}$ is a* stable family,[1] *i.e. it is*

- Complete: $\forall Z \subseteq \mathcal{B}$. *$Z$ is finitely compatible* $\implies \bigcup Z \in \mathcal{B}$;
- Stable: $\forall Z \subseteq \mathcal{B}$. $Z \neq \emptyset$ & *Z is compatible* $\implies \bigcap Z \in \mathcal{B}$;
- Finitary: $\forall \theta \in \mathcal{B}, (a, b) \in \theta \exists \theta_0 \in \mathcal{B}.\ \theta_0$ *is finite* & $(a, b) \in \theta_0 \subseteq \theta$; *and*
- Coincidence-free: *For all* $\theta \in \mathcal{B}$, $(a, b), (a', b') \in \theta$ *with* $(a, b) \neq (a', b')$,

$$\exists \theta_0 \in \mathcal{B}.\ \theta_0 \subseteq \theta \ \& \ ((a, b) \in \theta_0 \iff (a', b') \notin \theta_0)\,.$$

We now apply a general construction $\mathrm{Pr}(\mathcal{B})$ for obtaining an event structure from the stable family $\mathcal{B}$. Suppose $(a, b) \in \theta$ where $\theta \in \mathcal{B}$ (Winskel, 1982, 1986). Because $\mathcal{B}$ is a stable family

$$[(a, b)]_\theta =_{\mathrm{def}} \bigcap \{\phi \in \mathcal{B} \mid \phi \subseteq \theta \ \& \ (a, b) \in \phi\} \in \mathcal{B}$$

and moreover is a finite set; it represents a minimal way in which (a, b) can occur. We build the pullback of event structures taking such minimal elements as events.

Proposition 9.4 *Defining* $\mathrm{Pr}(\mathcal{B}) = (P, \mathrm{Con}, \leq)$ *where:*

[1] Here it is useful to allow stable families to have infinite configurations, as originally (Winskel, 1982, 1986).

$$P = \{[(a,b)]_\theta \mid (a,b) \in \theta \;\&\; \theta \in \mathcal{B}\}\,,$$
$$Z \in \mathrm{Con} \textit{ iff } Z \subseteq P \;\&\; \bigcup Z \in \mathcal{B} \textit{ and}$$
$$p \leq p' \textit{ iff } p, p' \in P \;\&\; p \subseteq p'$$

yields an event structure. There is an order isomorphism

$$\beta : (\mathcal{C}(\mathrm{Pr}(\mathcal{B})), \subseteq) \cong (\mathcal{B}, \subseteq)$$

where $\beta(y) = \bigcup y$ *for* $y \in \mathcal{C}(\mathrm{Pr}(\mathcal{B}))$ *; its mutual inverse is* γ *where* $\gamma(\theta) = \{[(a,b)]_\theta \mid (a,b) \in \theta\}$ *for* $\theta \in \mathcal{B}$.

There are obvious maps $\pi_1 : \mathrm{Pr}(\mathcal{B}) \to A$ and $\pi_2 : \mathrm{Pr}(\mathcal{B}) \to B$ given by $\pi_1([(a,b)]_\theta) = a$ and $\pi_2([(a,b)]_\theta) = b$. These make the required pullback $\mathrm{Pr}(\mathcal{B})$, π_1, π_2 of event structures. Why? The family $\mathcal{B}$ is a pullback in the category of stable families (its maps are similar to those of event structures). There is a coreflection from the category of event structures to that of stable families. Its right adjoint is Pr which consequently preserves pullbacks, yielding the pullback of event structures when applied to $\mathcal{B}$ (Winskel, 1982, 2016).

Definition 9.5 We shall write $x \wedge y$ for the configuration $\gamma(\theta_{x,y})$ of $\mathrm{Pr}(\mathcal{B})$ which correponds to a secured bijection $\theta_{x,y} : x \cong fx = gy \cong y$ between $x \in \mathcal{C}^\infty(A)$ and $y \in \mathcal{C}^\infty(B)$. Note that any configuration of the pullback is of the form $x \wedge y$ for unique $x \in \mathcal{C}^\infty(A)$ and $y \in \mathcal{C}^\infty(B)$. Of course, given $x \in \mathcal{C}^\infty(A)$ and $y \in \mathcal{C}^\infty(B)$ we cannot be assured that they form a secured bijection even when $fx = gy$. We shall treat $\wedge$ as a partial operation with $x \wedge y$ only defined when $x \in \mathcal{C}^\infty(A)$ and $y \in \mathcal{C}^\infty(B)$ form a secured bijection.

9.4 Rigid Image

This section is only used late on in the paper when showing how 'may' and 'must' behaviour transfer to the rigid image of a strategy—Sect. 9.11.

There is an adjunction between $\mathcal{E}_r$, the category of event structures with rigid maps, to $\mathcal{E}_t$, the category of event structures with total maps. Its right adjoint's action on an event structure B is given as follows. For $x \in \mathcal{C}^\infty(B)$, an *augmentation* of x is a finitary partial order (x, α) where $\forall b, b' \in x.\, b \leq_B b' \implies b\,\alpha\,b'$. We can regard such augmentations as elementary event structures in which all subsets of events are consistent. Order all augmentations by taking $(x, \alpha) \hookrightarrow (x', \alpha')$ iff $x \subseteq x'$ and the inclusion $i : x \hookrightarrow x'$ is a rigid map $i : (x, \alpha) \to (x', \alpha')$. Augmentations under $\hookrightarrow$ form a prime algebraic domain (Nielsen et al., 1981; Winskel, 2009), so are isomorphic to the configurations of an event structure, $aug(B)$; its events are the complete primes, which are precisely the augmentations with a top element.

Proposition 9.6 (Winskel, 2007) *The inclusion functor $\mathcal{E}_r \hookrightarrow \mathcal{E}_t$ has a right adjoint aug. The category $\mathcal{E}_t$ is isomorphic to the Kleisli category of the monad induced on $\mathcal{E}_r$ by the adjunction.*

Rigid maps $f : A \to B$ have a useful image given by restricting the causal dependency of B to the set of events fA, the direct image of the events of A, and taking a finite set of events to be consistent if they are the image of a consistent set in A. More generally, a total map $f : A \to B$ has a *rigid image* given by the image of its corresponding Kleisli map, the rigid map $\bar{f} : A \to aug(B)$. Put more directly, a total map $f : A \to B$ has a *rigid image* comprising a factorisation $f = f_1 f_0$ where f_0 is rigid epi and f_1 is a total map,

$$\begin{tikzcd} A \arrow[r, two heads, "f_0"] \arrow[dr, "f"'] & B_0 \arrow[d, "f_1"] \\ & B \end{tikzcd},$$

with the following universal property: for any factorisation of $f = f'_1 f'_0$ where f'_0 is rigid epi, there is a unique map h such that the diagram

$$\begin{tikzcd} A \arrow[r, two heads, "f'_0"] \arrow[rr, bend left, "f_0"] \arrow[dr, "f"'] & B'_0 \arrow[r, dashed, "h"] \arrow[d, dashed, "f'_1"] & B_0 \arrow[dl, "f_1"] \\ & B & \end{tikzcd}$$

commutes; the map h is necessarily also rigid and epi.

From the universal property of rigid image we derive:

Proposition 9.7 *Let $f : A \to B$ and $g : B \to C$ be maps of event structures. Assume that f is rigid and epi. Then, g and $g \circ f$ have the same rigid image.*

9.5 Event Structures with Polarity

Both games and strategies will be represented by event structures with polarity. An *event structure with polarity* comprises (A, pol) where A is an event structure with a polarity function $pol_A : A \to \{+, -, 0\}$ ascribing a polarity $+$ (Player), $-$ (Opponent) or 0 (neutral) to its events. The events correspond to (occurrences of) moves. It will be technically useful to allow events of neutral polarity; they arise, for example, in the interaction between a strategy and a counterstrategy. We write A^0 for the event structure with polarity in which all the polarities are reassigned 0, so made neutral. A *game* shall be represented by an event structure with polarity in which no moves are neutral.

Notation 9.8 In an event structure with polarity (A, pol), with configurations x and y, write $x \subseteq^- y$ to mean inclusion in which all the intervening events are moves of Opponent, *i.e.* $pol(y \setminus x) \subseteq \{-\}$. Similarly, $x \subseteq^0 y$ signifies an inclusion in which all the intervening moves are neutral. However, we shall write $x \subseteq^+ y$ for inclusion in which the intervening events are either neutral or moves of Player. (The latter choice reflects the fact that neutral moves in a strategy behave as internal moves of Player.) We say a configuration $x \in \mathcal{C}^\infty(A)$ is *+-maximal* iff x is maximal in $\mathcal{C}^\infty(A)$ w.r.t. $\subseteq^+$, *i.e.* the only way that x extends to a larger configuration is through the occurrence of Opponent moves.

9.5.1 Operations on Games

We introduce two fundamental operations on games.

9.5.1.1 Dual

The *dual*, $A^\perp$, of a game A, comprises the same underlying event structure as A but with a reversal of polarities. As mentioned in the introduction, we shall implicitly adopt the view of Player and understand a strategy in a game A as strategy for Player. A counterstrategy in a game A is a strategy for Opponent in the game A, *i.e.* a strategy (for Player) in the game $A^\perp$.

9.5.1.2 Simple Parallel Composition

This operation simply juxtaposes two games, and more generally two event structures with polarity. Let $(A, \leq_A, \mathrm{Con}_A, pol_A)$ and $(B, \leq_B, \mathrm{Con}_B, pol_B)$ be event structures with polarity. The events of $A\|B$ are $(\{1\} \times A) \cup (\{2\} \times B)$, their polarities unchanged, with the only relations of causal dependency given by $(1, a) \leq (1, a')$ iff $a \leq_A a'$ and $(2, b) \leq (2, b')$ iff $b \leq_B b'$; a finite set X of events is consistent in $A\|B$ iff its components X_A in A and and X_B in B are individually consistent. The unit w.r.t. simple composition is the empty event structure with polarity, written $\emptyset$. We shall adopt the same operation for configurations of a game $A\|B$, regarding a configuration x of the parallel composition as $x_A\|x_B$.

If we are not a little careful we can run into distracting technical issues through $(A\|B)\|C$ not being strictly the same as $A\|(B\|C)$. For our purposes it will suffice to adopt the convention that when we write *e.g.* $A\|B\|C$ the simple parallel composition of three event structures with polarity we shall mean the event structure with events

$$\{1\} \times A \cup \{2\} \times B \cup \{3\} \times C\,,$$

with causal dependency and consistency copied from those of A, B and C. As in the binary case, we adopt the same notation for configurations and can describe a typical configuration x of $A\|B\|C$ as $x_A\|x_B\|x_C$.

9.5.2 Strategies Between Games

A strategy *from* a game A *to* a game B is a strategy in the compound game $A^\perp\|B$. Of course we shall have to define what it means to be a strategy in a game. Given another strategy τ from the game B to a game C, informally we obtain their composition $\tau\odot\sigma$ from A to C by playing the two strategies off against each other in the common game B and hiding the resulting interaction.

The composition of strategies can introduce hidden deadlocks, conflicts and divergences which affect its observable behaviour:

Example 9.9 Let B be the game consisting of two concurrent Player events b_1 and b_2, and C the game with a single Player event c. We illustrate the composition of two strategies σ_1 and σ_2 from the empty game $\emptyset$ to B, with τ from B to C. The strategy σ_1 in the game $\emptyset^\perp\|B$ nondeterministically plays b_1 or b_2. The strategy σ_2 also in the game $\emptyset^\perp\|B$ just plays b_2. The strategy τ in the game $B^\perp\|C$ does nothing if just b_1 is played and plays the single Player event c of C if b_2 is played. The composition $\tau\odot\sigma_1$ in the game $\emptyset^\perp\|C$ may play c or not according as σ_1 plays b_1 or b_2. The composition $\tau\odot\sigma_2$ also in the game $\emptyset^\perp\|C$ must play c. But the two compositions $\tau\odot\sigma_1$ and $\tau\odot\sigma_2$ are indistinguishable once the interaction over the common game B is hidden. $\square$

If we are to distinguish the two compositions of the example, we need to take some account of their internal moves of interaction.

9.6 Strategies with Neutral Moves

Thus motivated, we study *bare strategies* with neutral moves, in which we can see the events of interaction not visible in the game. Recall we assume that in games all events have +ve or −ve polarity.

Definition 9.10 A *bare strategy* from a game A to a game B comprises a total map $\sigma : S \to A^\perp\|N\|B$ of event structures with polarity (in which S may also have neutral events) where

(i) N is an event structure consisting solely of neutral events;
(ii) σ is *receptive*,
$\forall x \in \mathcal{C}(S), y \in \mathcal{C}(A^\perp\|N\|B).\ \sigma x \subseteq^- y \implies \exists! x' \in \mathcal{C}(S).\ x \subseteq x' \ \&\ \sigma x' = y$;

(iii) σ is innocent in that it is both +-innocent and −-innocent:
$+$-*innocent:* if $s \rightarrowtriangle s'$ & $pol(s) = +$ then $\sigma(s) \rightarrowtriangle \sigma(s')$;
$-$-*innocent:* if $s \rightarrowtriangle s'$ & $pol(s') = -$ then $\sigma(s) \rightarrowtriangle \sigma(s')$.

Note that s' in +-innocence and s in −-innocence may be neutral events.[2]

A *strategy* from a game A to a game B comprises a total map $\sigma : S \to A^\perp \| B$ of event structures with polarity for which the composite $\sigma : S \to A^\perp \| B \cong A^\perp \| \emptyset \| B$ is a bare strategy (Rideau & Winskel, 2011).

We shall often identify strategies with bare strategies with no neutral events, and (bare) strategies in a game with (bare) strategies from the empty game $\emptyset$.

Consider two bare strategies $\sigma : S \to A^\perp \| N \| B$ and $\sigma' : S' \to A^\perp \| N \| B$. A map between them, a 2-cell $f : \sigma \Rightarrow \sigma'$, comprises a map $f : S \to S'$ of event structures with polarity such that

$$\begin{tikzcd} S \arrow[r, "f"] \arrow[d, "\sigma"'] & S' \arrow[dl, "\sigma'"] \\ A^\perp \| N \| B & \end{tikzcd}$$

commutes. In this way bare strategies in $A^\perp \| N \| B$ form a category,

$$\mathbf{BStrat}(A, N, B).$$

We obtain the category $\mathbf{Strat}(A, B)$ of strategies from A to B from the special case when $N = \emptyset$.

9.6.1 Strategies from Bare Strategies

We obtain a strategy as the visible part of a bare strategy when we hide neutral events via a projection p:

Proposition 9.11 *Let $\sigma : S \to A^\perp \| N \| B$ be a bare strategy—so satisfying properties (i), (ii) and (iii) of Definition 9.10. Then, σ satisfies an additional property:*

(iv) in the partial-total factorisation of the composition of σ with the projection $A^\perp \| N \| B \to A^\perp \| B$,

[2] This definition of *linear innocence*, which applies in the presence of neutral events, appears in the work of Claudia Faggian and Mauro Piccolo (2009). It is not to be confused with the innocence of Martin Hyland and Luke Ong, to which it only relates indirectly; to disambiguate the two notions "courtesy" has been used for that here. An extension of Hyland-Ong innocence to concurrent games is given in Castellan et al. (2015). Bare strategies have also been called "partial" strategies (Castellan et al., 2014) and "uncovered" strategies (Castellan et al., 2018).

$$\begin{tikzcd}
S \arrow[r, "p"] \arrow[d, "\sigma"'] & S_{\downarrow} \arrow[d, "\sigma_{\downarrow}"] \\
A^{\perp} \| N \| B \arrow[r] & A^{\perp} \| B
\end{tikzcd}$$

the defined part $\sigma_{\downarrow}$ *is a strategy, which we call the* visible part *of* σ.

(Conversely, (iv) together with receptivity and no incidence of a +ve event immediately preceding a neutral event in S, suffice to establish that σ *is a bare strategy.)*

With the notation of the lemma above, write

$$x_{\downarrow} =_{\text{def}} px \in \mathcal{C}^{\infty}(S_{\downarrow})$$

for the visible image of a configuration $x \in \mathcal{C}^{\infty}(S)$. The hiding operation on strategies extends to a functor

$$(\_)_{\downarrow} : \mathbf{BStrat}(A, N, B) \to \mathbf{Strat}(A, B)\,;$$

a 2-cell $f : \sigma \Rightarrow \sigma'$ between bare strategies restricts to a 2-cell $f_{\downarrow} : \sigma_{\downarrow} \Rightarrow \sigma'_{\downarrow}$ between their visible parts. It acts so

$$f_{\downarrow} x_{\downarrow} = (fx)_{\downarrow}$$

for all $x \in \mathcal{C}^{\infty}(S)$.

9.6.2 Composition

We can compose two bare strategies

$$\sigma : S \to A^{\perp} \| M \| B \ \text{ and } \ \tau : T \to B^{\perp} \| N \| C$$

by pullback. Ignoring polarities temporarily, and padding with identity maps, we obtain $\tau \circledast \sigma$ via the pullback

$$\begin{tikzcd}
 & T \circledast S \arrow[dl] \arrow[dr] \arrow[dd, phantom, "\vee" very near start] & \\
S \| N \| C \arrow[dr, "\sigma \| N \| C"'] & & A \| M \| T \arrow[dl, "A \| M \| \tau"] \\
 & A \| M \| B \| N \| C &
\end{tikzcd}$$

as the ensuing map

$$\tau \circledast \sigma : T \circledast S \to A^{\perp} \| (M \| B^0 \| N) \| C$$

once we reinstate polarities and make the events of B neutral.

As a pullback the configurations of $T \circledast S$ are built from configurations of S and T. Let $x \in \mathcal{C}^{\infty}(S)$ and $y \in \mathcal{C}^{\infty}(T)$. Let $\sigma x = x_{A^{\perp}} \| x_0 \| x_B$ and $\tau y = y_{B^{\perp}} \| y_0 \| y_C$. Define

$$y \circledast x =_{\text{def}} (x \| y_0 \| y_C) \wedge (x_{A^{\perp}} \| x_0 \| y)$$

which will be defined and a configuration in $\mathcal{C}^{\infty}(T \circledast S)$ if $x_B = y_{B^{\perp}}$ and the corresponding bijection secured. The following property, useful later, is a consequence of the receptivity of σ and τ.

Lemma 9.12 *A configuration $y \circledast x \in \mathcal{C}^{\infty}(T \circledast S)$ is $+$-maximal in $\mathcal{C}^{\infty}(T \circledast S)$ iff x is $+$-maximal in $\mathcal{C}^{\infty}(S)$ and y is $+$-maximal in $\mathcal{C}^{\infty}(T)$.*

Given a 2-cell $f : \sigma \Rightarrow \sigma'$ between bare strategies in $A^{\perp} \| N \| B$ and $g : \tau \Rightarrow \tau'$ between bare strategies in $B^{\perp} \| M \| C$, from the universality of pullback we obtain the 2-cell

$$g \circledast f : \tau \circledast \sigma \Rightarrow \tau' \circledast \sigma'$$

between the two compositions in $A^{\perp} \| (N \| B^0 \| M) \| C$. It acts so

$$(g \circledast f)(y \circledast x) = (gy) \circledast (fx)$$

on a typical configuration $y \circledast x$. This extends composition of bare strategies to a functor

$$\circledast : \mathbf{BStrat}(B, N, C) \times \mathbf{BStrat}(A, M, B) \to \mathbf{BStrat}(A, M \| B^0 \| N, C)\,.$$

Composition of bare strategies restricts to a functor between rigid 2-cells.

We obtain the composition of strategies as the composite functor

$$\begin{aligned} \odot : \mathbf{Strat}(B, C) \times \mathbf{Strat}(A, B) \cong & \mathbf{BStrat}(B, \emptyset, C) \times \mathbf{BStrat}(A, \emptyset, B) \\ & \xrightarrow{\circledast} \mathbf{BStrat}(A, \emptyset \| B^0 \| \emptyset, C) \xrightarrow{(\_)_{\downarrow}} \mathbf{Strat}(A, C)\,. \end{aligned}$$

Though we generally elide the isomorphisms regarding strategies as bare strategies without neutral events, and write

$$\tau \odot \sigma =_{\text{def}} (\tau \circledast \sigma)_{\downarrow}$$

for the composition of strategies $\sigma \in \mathbf{Strat}(A, B)$ and $\tau \in \mathbf{Strat}(B, C)$. Describing the strategies $\sigma : S \to A^{\perp} \| B$ and $\tau : T \to B^{\perp} \| C$ as having composition

$$\tau\odot\sigma : T\odot S \to A^{\perp}\|C\,,$$

we can present a typical configuration of $T\odot S$ as

$$y\odot x =_{\text{def}} (y \circledast x)_{\downarrow}$$

for $x \in \mathcal{C}^{\infty}(S)$ and $y \in \mathcal{C}^{\infty}(T)$.

Composition is preserved in extracting the visible part from bare strategies:

Lemma 9.13 *Let $\sigma : S \to A^{\perp}\|M\|B$ and $\tau : T \to B^{\perp}\|N\|C$ be bare strategies. Then,*

$$(\tau \circledast \sigma)_{\downarrow} = \tau_{\downarrow}\odot\sigma_{\downarrow}\,.$$

For $x \in \mathcal{C}^{\infty}(S)$ and $y \in \mathcal{C}^{\infty}(T)$,

$$(y \circledast x)_{\downarrow} = y_{\downarrow}\odot x_{\downarrow}\,,$$

with one side defined if the other is.

9.6.3 The Copycat Strategy

The copycat strategy is the identity for composition of strategies. We present its construction and key property from Rideau and Winskel (2011).

Lemma 9.14 *Let A be an event structure with polarity. There is an event structure with polarity $\mathbb{C}_A$ having the same events and polarity as $A^{\perp}\|A$ but with causal dependency $\leq_{\mathbb{C}_A}$ given as the transitive closure of the relation*

$$\leq_{A^{\perp}\|A} \cup \{(\bar{c}, c) \mid c \in A^{\perp}\|A \;\&\; pol_{A^{\perp}\|A}(c) = +\}$$

and finite subsets of $\mathbb{C}_A$ consistent if their down-closure w.r.t. $\leq_{\mathbb{C}_A}$ are consistent in $A^{\perp}\|A$. (For $c \in A^{\perp}\|A$ we use $\bar{c}$ to mean the corresponding copy of c, of opposite polarity, in the alternative component, i.e. $\overline{(1, a)} = (2, a)$ and $\overline{(2, a)} = (1, a)$.)

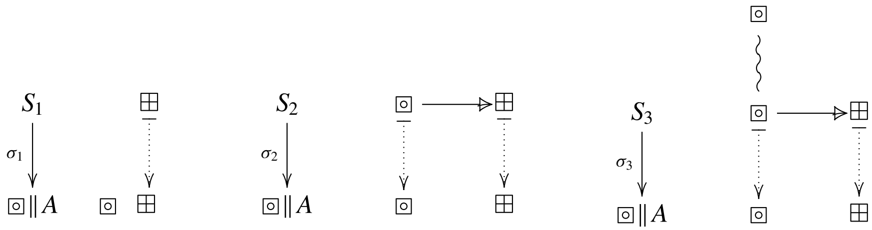
The configurations of $\mathbb{C}_A$ have the form $x\|y$ where $y \sqsubseteq_A x$, i.e. $y \supseteq^{-} x \cap y \subseteq^{+} x$, for $x, y \in \mathcal{C}^{\infty}(A)$. (The relation $\sqsubseteq_A$ is a partial order, called the Scott order *(Winskel, 2013).)*

The copycat *strategy for A is the map $\text{œ}_A : \mathbb{C}_A \to A^{\perp}\|A$ which acts as identity on events. We have $\sigma \cong \sigma\odot \text{œ}_A \cong \text{œ}_B\odot\sigma$, for any strategy $\sigma \in$* **Strat**(A, B).

The axioms on strategies are precisely those needed to ensure that copycat behaves as identity w.r.t. composition $\odot$ and thus obtain a bicategory **Strat** of games and strategies (Rideau & Winskel, 2011). Of course copycat is not the identity for the composition of bare strategies; that composition will generally have extra neutral events introduced through interactions.

9.7 'May' and 'Must' Tests

Consider the following three bare strategies in the game A comprising a single Player move $\boxplus$. Neutral events are drawn as $\boxdot$.



From the point of view of observing the move over the game A the first two bare strategies, σ_1 and σ_2, differ from the third, σ_3. In a maximal play both σ_1 and σ_2 must result in the observation of the single move of A. However, in σ_3 one maximal play is that in which the topmost neutral event of S_3 has occurred, in conflict with the only way of observing the single move of A.

We follow Hennessy and de Nicola in making these ideas precise (De Nicola & Hennessy, 1984).

Definition 9.15 Let σ be a bare strategy in a game A. Let $\tau : T \to A^{\perp} \| N \| \boxplus$ be a 'test' bare strategy from A to the game consisting of a single Player move $\boxplus$. Write $\checkmark =_{\text{def}} (3, \boxplus)$.

Say σ *may pass* τ iff there exists $y \circledast x \in \mathcal{C}^\infty(T \circledast S)$, where $x \in \mathcal{C}^\infty(S)$ and $y \in \mathcal{C}^\infty(T)$, with the image τy containing $\checkmark$. (Note that we may w.l.o.g. assume that the configuration $y \circledast x$ is finite.)

Say σ *must pass* τ iff for all $y \circledast x \in \mathcal{C}^\infty(T \circledast S)$, where $x \in \mathcal{C}^\infty(S)$ and $y \in \mathcal{C}^\infty(T)$ are $\subseteq^+$-maximal, the image τy contains $\checkmark$.

Say two bare strategies are *'may'* (respectively, *'must') equivalent* iff the tests they may (respectively, must) pass are the same.

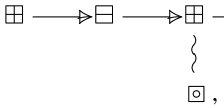
The definitions extend in the obvious fashion to bare strategies of type $A^{\perp} \| N \| B$.

A bare strategy is 'may' equivalent, but need not be 'must' equivalent, to the strategy which is its defined part; 'must' inequivalence is lost in moving from bare strategies to strategies.

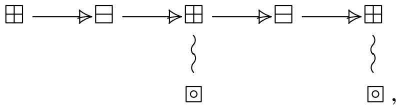
Example 9.16 As an illustration of the subtle nature of testing for 'must' equivalence, consider the following bare strategies in the game A, as drawn:

$$\begin{array}{ccccccccc} a & & b & & c & & d & & e \\ \boxplus & \rightarrowtriangle & \boxminus & \rightarrowtriangle & \boxplus & \rightarrowtriangle & \boxminus & \rightarrowtriangle & \boxplus \end{array}$$

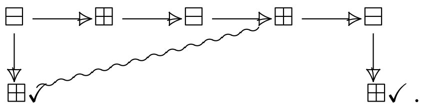
The game A consists of five events with the polarity and causal dependency shown. One bare strategy σ_1 is



with one neutral event, while the other σ_2 is



with two neutral events. Through the possible occurrence of neutral events the bare strategy σ_1 has a +-maximal configuration with just events $\{a, b\}$ visible from the game, while σ_2 has in addition a +-maximal configuration comprising visible moves $\{a, b, c, d\}$. The following test strategy distinguishes σ_1 and σ_2



in that σ_1 must pass the test while σ_2 need not. (The test is a strategy in $A^{\perp} \| \boxplus\checkmark$; thus the change in polarity of moves in A.) Composed with the test, σ_1 may fail to perform the leftmost $\boxplus\checkmark$ but if so must then perform the rightmost $\boxplus\checkmark$; whereas σ_2 may fail to perform both. $\square$

This example might be puzzling to readers familiar with Harmer and McCusker's fully abstract model for (sequential) finite non-determinism (Harmer & McCusker, 1999), in particular regarding their handling of 'must' equivalence through the addition of *divergences*. Indeed, there, if a trace has already potentially triggered a divergence, then any execution going past that divergence has already lost all hope for 'must' convergence. Accordingly, in Harmer & McCusker (1999), only the first divergence matters—further divergences are not recorded.

In contrast, Example 9.16 indicates how in concurrent strategies, all divergences matter. This is not an artificiality of concurrent strategies, but something inherent to the observational power of the tests they support: here the distinguishing tests manage to observe beyond the first divergence by running two threads in parallel. The first thread aims to directly register success; by +-maximality it will be run eventually if the observed strategy triggers the first divergence. The second thread follows the execution of the program, then immediately cancels the first thread if the program goes past the divergence; and then proceeds to test for the second divergence.

9.8 Strategies with Stopping Configurations

Bare strategies lack identities w.r.t. composition, so they do not form a bicategory. Fortunately, for 'may' and 'must' equivalence it is not necessary to use bare strategies; for 'may' equivalence strategies suffice; whereas for 'must' equivalence it is sufficient to carry with a strategy the extra structure of *stopping configurations*—to be thought of as images of $+$-maximal configurations in an underlying bare strategy. As we shall see, composition and copycat extend to composition and copycat on strategies with stopping configurations, while maintaining a bicategory. We tackle the simpler case in which games are assumed to be race-free. (The extension to games which are not race-free is outlined in Castellan et al. (2014).) We recall when an event structure with polarity is race-free and the allied notion of deterministic strategy:

Definition 9.17 Say A, an event structure with polarity, is *race-free* iff whenever $x \subseteq^+ y$ and $x \subseteq^- z$ for configurations x, y, z of A then $y \cup z$ is also a configuration.
Say S, an event structure with polarity, is *deterministic* iff whenever $x \subseteq^+ y$ and $x \subseteq z$ for configurations x, y, z of S then $y \cup z$ is also a configuration. Say a bare strategy $\sigma : S \to A^\perp \| N \| B$ is *deterministic* iff S is deterministic; with a strategy being deterministic iff it is so as a bare strategy.

Lemma 9.18 ((Rideau & Winskel, 2011; Winskel, 2012)) *Let A be a game. The copycat strategy α_A is deterministic iff the game A is race-free.*

Let $\sigma : S \to A^\perp \| N \| B$ be a bare strategy between race-free games A and B. Recall its associated partial-total factorisation

$$\begin{tikzcd} S \arrow[r, "p"] \arrow[d, "\sigma"'] & S_\downarrow \arrow[d, "\sigma_\downarrow"] \\ A^\perp \| N \| B \arrow[r] & A^\perp \| B \end{tikzcd}$$

where p is a projection sending configurations x of S to configurations $x_\downarrow$ of $S_\downarrow$. The visible part of σ is a strategy $\sigma_\downarrow$. Define the *stopping* configurations in $\mathcal{C}^\infty(S_\downarrow)$ to be

$$\mathrm{Stop}(\sigma) =_{\mathrm{def}} \{x_\downarrow \mid x \in \mathcal{C}^\infty(S) \text{ is } +\text{-maximal}\}\,.$$

So, in other words, the stopping configurations are the visible images of configurations which are maximal w.r.t. neutral or Player moves. Note that $\mathrm{Stop}(\sigma)$ will include all the +-maximal configurations of $S_\downarrow$: any +-maximal configuration y of $S_\downarrow$ is the image under p of its down-closure $[y]$ in S, and by Zorn's lemma this extends (necessarily by neutral events) to a maximal configuration x of S with image y under p; the configuration x is +-maximal by the +-maximality of y. If σ is deterministic, then $\mathrm{Stop}(\sigma)$ consists of precisely the +-maximal configurations of $S_\downarrow$. If σ is a strategy, *i.e.* it has no neutral events, then $\mathrm{Stop}(\sigma)$ is just the set consisting of all +-maximal configurations of S.

Definition 9.19 A *stopping strategy* in a game A comprises (σ, M_S), a strategy $\sigma : S \to A$ together with a subset $M_S \subseteq \mathcal{C}^\infty(S)$ called *stopping configurations*. As usual, a stopping strategy from a game A to game B is a stopping strategy in the game $A^\perp \| B$. We let $\mathrm{St} : \sigma \mapsto (\sigma_\downarrow, \mathrm{Stop}(\sigma))$ denote the operation motivated above from bare strategies σ to stopping strategies.

Remark. There is the issue of what axioms to adopt on stopping configurations. We do not insist that stopping configurations include all +-maximal configurations as this property will not be preserved in taking the rigid image of a stopping strategy—Example 9.38.

Given two stopping strategies $\sigma : S \to A^\perp \| B, M_S$ and $\tau : T \to B^\perp \| C, M_T$ we define their *interaction*,

$$(\tau, M_T) \circledast (\sigma, M_S) =_{\mathrm{def}} (\tau \circledast \sigma, M_T \circledast M_S)\,,$$

with the stopping configurations of the interaction $T \circledast S$ as

$$M_T \circledast M_S = \{y \circledast x \mid x \in M_S \;\&\; y \in M_T\}$$

—sensible because of Lemma 9.12. Define their *composition* by

$$(\tau, M_T) \odot (\sigma, M_S) =_{\mathrm{def}} (\tau \odot \sigma, M_T \odot M_S)\,,$$

where the stopping configurations of $T \odot S$ form the set

$$M_T \odot M_S = \{y \odot x \mid x \in M_S \;\&\; y \in M_T\}\,.$$

To make stopping strategies into a bicategory we must settle on an appropriate notion of 2-cell. The following choice of definition is useful for 'must' equivalence—see Lemma 9.28.

Definition 9.20 A 2-cell $f : (\sigma, M_S) \Rightarrow (\sigma', M_{S'})$ between stopping strategies is a 2-cell of strategies $f : \sigma \Rightarrow \sigma'$ such that $f M_S \subseteq M_{S'}$. We write $\mathbf{SStrat}(A, B)$ for the category of stopping strategies from game A to game B; its maps are 2-cells.

Composition extends to 2-cells between stopping strategies: their composition as 2-cells between strategies is easily shown to preserve stopping configurations.

Proposition 9.21 *For games A, B and C composition of stopping strategies is a functor* $\odot : \mathbf{SStrat}(B, C) \times \mathbf{SStrat}(A, B) \to \mathbf{SStrat}(A, C)$.

We should also extend copycat $\text{æ}_A : \mathbb{C}_A \to A^\perp \| A$ to a stopping strategy. Because we are assuming A is race-free, we do this by taking

$$M_{\mathbb{C}_A} =_{\mathrm{def}} \{(x \| x) \in \mathcal{C}^\infty(\mathbb{C}_A) \mid x \in \mathcal{C}^\infty(A)\}\,.$$

Because A is race-free, $M_{\mathbb{C}_A}$ comprises all the +-maximal configurations of $\mathbb{C}_A$. Then, $(\mathbb{c}_A, M_{\mathbb{C}_A})$ is an identity w.r.t. the extended composition.

With the operations and constructions above, stopping strategies inherit the structure of a bicategory **SStrat** from strategies; the objects are restricted to race-free games in order to have the above simple form of stopping configurations for copycat.

9.8.1 Bare Strategies and Stopping Strategies

We turn to relations between bare strategies and stopping strategies. Recall from Definition 9.19 the operation

$$\mathrm{St} : \sigma \mapsto (\sigma_\downarrow, \mathrm{Stop}(\sigma))$$

which takes a bare strategy σ to a stopping strategy. It preserves composition:

Lemma 9.22 *Let $\sigma : S \to A^\perp \| M \| B$ and $\tau : T \to B^\perp \| N \| C$ be bare strategies. Then,*

$$St(\tau \circledast \sigma) = St(\tau) \odot St(\sigma) .$$

Proof By Lemma 9.13, it suffices to show

$$\mathrm{Stop}(\tau \circledast \sigma) = \mathrm{Stop}(\tau) \odot \mathrm{Stop}(\sigma) .$$

Configurations of $\mathrm{Stop}(\tau \circledast \sigma)$ are of the form $(y \circledast x)_\downarrow$ where $y \circledast x$ is +-maximal for $x \in \mathcal{C}^\infty(S)$ and $y \in \mathcal{C}^\infty(T)$. By Lemma 9.12 these coincide with configurations $(y \circledast x)_\downarrow$ for which both x and y are +-maximal. Configurations of $\mathrm{Stop}(\tau) \odot \mathrm{Stop}(\sigma)$ take the form $y_\downarrow \odot x_\downarrow$ where $x \in \mathcal{C}^\infty(S)$ and $y \in \mathcal{C}^\infty(T)$ are +-maximal. But by Lemma 9.13, $(y \circledast x)_\downarrow = y_\downarrow \odot x_\downarrow$, ensuring the desired equality. $\square$

We have seen that there is a functor $(\_)_\downarrow : \mathbf{BStrat}(A, N, B) \to \mathbf{Strat}(A, B)$ from bare strategies to their visible part. However, it is not the case that St is a functor from bare strategies $\mathbf{BStrat}(A, N, B)$ to stopping strategies $\mathbf{SStrat}(A, B)$. Given an arbitrary 2-cell $f : \sigma \Rightarrow \sigma'$ between bare strategies $f_\downarrow$ can fail to preserve stopping configurations. However:

Proposition 9.23 *Let $\sigma : S \to A^\perp \| N \| B$ and $\sigma' : S' \to A^\perp \| N \| B$ be bare strategies. Say a 2-cell $f : \sigma \Rightarrow \sigma'$ is* +-reflecting *iff, for $x \in \mathcal{C}(S)$, $y \in \mathcal{C}(S')$,*

$$f x \subseteq^+ y \implies \exists x' \in \mathcal{C}(S).\ x \subseteq x' \ \&\ f x' = y .$$

Let $\mathbf{BStrat}^+(A, N, B)$ be the subcategory where 2-cells are +-reflecting. Then

$$St : \mathbf{BStrat}^+(A, N, B) \to \mathbf{SStrat}(A, B)$$

taking $f : \sigma \Rightarrow \sigma'$ *to* $f_\downarrow : St(\sigma) \Rightarrow St(\sigma')$ *is a functor.*

9.9 'May' and 'Must' Testing

We can rephrase 'may' and 'must' testing in terms of stopping strategies.

Definition 9.24 Let (σ, M_S) be a stopping strategy in a game A. Let $\tau : T \to A^\perp \| N \| \boxplus$ be a 'test' bare strategy from A to a the game consisting of a single Player move $\boxplus$. Write $St(\tau)$ as (τ_0, M_0) where $\tau_0 : T_0 \to A \| \boxplus$ is the visible part of τ and M_0 are its stopping configurations, obtained as images of the $+$-maximal configurations of T. Write $\checkmark =_{\text{def}} (2, \boxplus)$.

Say (σ, M_S) *may pass* τ iff there exists $y \circledast x \in \mathcal{C}^\infty(T_0 \circledast S)$, where $x \in \mathcal{C}^\infty(S)$ and $y \in \mathcal{C}^\infty(T_0)$, with the image $\tau_0 y$ containing $\checkmark$. (Note again, we may w.l.o.g. assume that the configurations x and y are finite.)

Say (σ, M_S) *must pass* τ iff for all $y \circledast x \in M_0 \circledast M_S$, where $x \in \mathcal{C}^\infty(S)$ and $y \in \mathcal{C}^\infty(T_0)$, the image $\tau_0 y$ contains $\checkmark$.

Say two stopping strategies are *'may'*, respectively *'must'*, *equivalent* iff the tests they may, respectively must, pass are the same.

Proposition 9.25 *With the notation above,*

(σ, M_S) may pass τ *iff there exists* $y \odot x \in \mathcal{C}^\infty(T_0 \odot S)$*, where* $x \in \mathcal{C}^\infty(S)$ *and* $y \in \mathcal{C}^\infty(T_0)$*, with the image* $\tau_0 y$ *containing* $\checkmark$ *—the configurations* x*,* y *may be assumed finite; and*

(σ, M_S) must pass τ *iff for all* $y \odot x \in M_0 \odot M_S$*, where* $x \in M_S$ *and* $y \in M_0$*, the image* $\tau_0\, y$ *contains* $\checkmark$*.*

Lemma 9.26 *Let* A *be a race-free game. Let* σ *be a bare strategy in* A*. Then,*

σ may pass a test τ *iff* $St(\sigma)$ may pass τ*;*

σ must pass a test τ *iff* $St(\sigma)$ must pass τ.

Proof Directly from the definitions, for the 'if' of the 'must' case, using Lemma 9.12. □

Example 9.27 It is tempting to think of neutral events as behaving like the internal "tau" events of CCS (Milner, 1980). However, in the context of concurrent strategies, because of their asynchronous nature, they behave rather differently. Consider three bare strategies, over a game comprising of just two concurrent +ve events, say a and b. The bare strategies have the following event structures in which we have named events by the moves they correspond to in the game:

$$S_1 \quad \begin{array}{c} a \\ \wr \\ b \end{array} \qquad\qquad S_2 \quad \begin{array}{l} \boxdot \rightarrowtriangle a \\ \wr \\ \boxdot \rightarrowtriangle b \end{array} \qquad\qquad S_3 \quad \begin{array}{l} \boxdot \rightarrowtriangle a \\ \wr \\ b \end{array}$$

No pair would be weakly bisimilar due to the presence of pre-emptive internal events (Milner, 1980). However, all three become isomorphic under St so are 'may' and 'must' equivalent to each other. □

2-cells between stopping strategies respect 'may' and 'must' behaviour in the sense of the following lemma.

Lemma 9.28 *Let $f : (\sigma, M_S) \Rightarrow (\sigma', M_{S'})$ be a 2-cell between stopping strategies. Then for any test τ,*

(σ, M_S) may pass τ implies $(\sigma', M_{S'})$ may pass τ; and

$(\sigma', M_{S'})$ must pass τ implies (σ, M_S) must pass τ.

Moreover, if f is a rigid epi and $f M_S = M_{S'}$, then (σ, M_S) and $(\sigma', M_{S'})$ are both 'may' and 'must' equivalent.

Proof In this proof, we shall identify a test with its image (τ, M_T) under St, as a strategy $\tau : T \to A^\perp \| \boxplus$ with stopping configurations M_T.

Let $f : (\sigma, M_S) \Rightarrow (\sigma', M_{S'})$ be a 2-cell. Assume $\sigma : S \to A$ and $\sigma' : S' \to A$.

Suppose (σ, M_S) may pass (τ, M_T). Then there is a (finite) configuration which we write $y \circledast x$ of $T \circledast S$, built as a secured bijection out of $y \in \mathcal{C}(T)$ and $x \in \mathcal{C}(S)$, whose image in the game contains $\checkmark$. The secured bijection built out of y and x induces a secured bijection built out of y and fx; this is because fx has no more causal dependency than x with which it is in bijection. This determines a configuration $y \circledast fx$, with image containing $\checkmark$.

Suppose $(\sigma', M_{S'})$ must pass (τ, M_T). Any $y \circledast x \in M_T \circledast M_S$ images under $\tau \circledast f$ to $y \circledast fx \in M_T \circledast M_{S'}$. As $(\sigma', M_{S'})$ must pass τ, the configuration $y \circledast fx$ has image containing $\checkmark$, ensuring that $y \circledast x$ does too.

Finally suppose that f is rigid epi and $f M_S = M_{S'}$. We have just shown that f preserves the passing of 'may' tests and reflects the passing of 'must' tests. Because f is rigid epi it also reflects the passing of 'may' tests. Because f is rigid and $f M_S = M_{S'}$ it preserves the passing of 'must' tests: any secured bijection $y \circledast fx$ in $M_T \circledast M_{S'}$ ensures by the rigidity of f a secured bijection $y \circledast x$ in $M_T \circledast M_S$; as (σ, M_S) must pass τ we have the image in the game of $y \circledast x$ contains $\checkmark$ ensuring the image of $y \circledast fx$ does too. □

Tests based on bare strategies are more discriminating than tests based on (pure) strategies:

Example 9.29 Let a game comprise a single Player move. Consider two stopping strategies:

σ_1, the empty strategy with the empty configuration $\emptyset$ as its single stopping configuration;

σ_2, the strategy performing the single Player move $\boxplus$ with stopping configurations $\emptyset$ and $\{\boxplus\}$. (We can easily realise this stopping strategy via St from a bare strategy with event structure $\boxplus \sim \boxdot$.)

By Lemma 9.28, we have $(\sigma_2, \{\emptyset, \{\boxplus\}\})$ must pass τ implies $(\sigma_1, \{\emptyset\})$ must pass τ, for any test τ. (The above would not hold if we had not included $\emptyset$ in the stopping configurations of σ_2.)

Using the fact that we need only consider rigid images of tests—shown later in Sect. 9.11, a little argument by cases establishes the converse implication too, provided we restrict just to tests which are strategies. The stopping strategies would be must equivalent w.r.t. tests based just on strategies.

However with tests based on bare strategies we can distinguish them. Consider the test τ comprising three events, one of them neutral, with only nontrivial causal dependency $\boxminus \rightarrowtriangle \boxdot$ and $\boxdot$ in conflict with the 'tick' event $\boxplus$. Then, it is not the case that $(\sigma_2, \{\emptyset, \{\boxplus\}\})$ must pass τ —the occurrence of the neutral event blocks success in a maximal execution—while $(\sigma_1, \{\emptyset\})$ must pass τ. Notice how the presence of the neutral event in the test turns the possibility that σ_2 can perform the Player move into a possibility of its failing the test. $\square$

9.10 'May' and 'Must' Behaviour Characterised

9.10.1 Preliminaries, Traces of a Strategy

Let S be an event structure. A possibly infinite sequence

$$s_1, s_2, \ldots, s_n, \ldots$$

in S constitutes a *serialisation* of a configuration $x \in \mathcal{C}^\infty(S)$ if $x = \{s_1, s_2, \ldots, s_n, \ldots\}$ and $\{s_1, \ldots, s_i\} \in \mathcal{C}(S)$ for all i at which the sequence is defined. We will often identify such a countable enumeration of a set with its associated total order. Note that in this way we can regard a serialisation as an elementary event structure in which causal dependency takes the form of a total order; a serialisation of a configuration is associated with a map to S whose image is the configuration.

Let $\sigma : S \to A$ be a strategy in a game A. A *trace* in σ is a possibly infinite sequence

$$\alpha = (\sigma(s_1), \sigma(s_2), \ldots, \sigma(s_n), \ldots)$$

of events in A obtained from a serialisation

$$s_1, s_2, \ldots, s_n, \ldots$$

of a configuration $x \in \mathcal{C}^\infty(S)$. Clearly α is a serialisation of $\sigma x \in \mathcal{C}^\infty(A)$. From the local injectivity of σ, the configuration x will be finite/infinite according as the trace is finite/infinite. We say that α is a trace of the configuration x in σ, or that x has trace α in σ.

Proposition 9.30 *Let $\sigma : S \to A$ be a strategy.*
(i) Any countable configuration of S has a trace.
(ii) Let $x \in \mathcal{C}^\infty(S)$ and α be an enumeration

$$a_1, a_2, \ldots, a_n, \ldots$$

of σx. Then, α is a trace of x in σ iff for all $s, s' \in x$ if $s \rightarrowtriangle s'$ then $\sigma(s)$ precedes $\sigma(s')$ in the enumeration α.

Proof (i) Let x be a countable configuration of S w.r.t. the strategy $\sigma : S \to A$. This follows because there is a serialisation $x = \{s_1, s_2, \ldots, s_n, \ldots\}$, in which $\{s_1, \ldots, s_i\}$ is down-closed in S at all i in the enumeration. To see this, from its countability we may assume a countable enumeration of x, which need not be a serialisation. Define $s_1 \in x$ to be the earliest event of the enumeration for which $[s_1) = \emptyset$ in S; such an s_1 is ensured to exist by the well-foundedness of causal dependency provided $x \neq \emptyset$. Inductively, define s_n to be the earliest event of the enumeration which is in $x \setminus \{s_1, \ldots, s_{n-1}\}$ and for which $[s_n) \subseteq \{s_1, \ldots, s_{n-1}\}$; again the well-foundedness of causal dependency ensures such an s_n exists provided $x \setminus \{s_1, \ldots, s_{n-1}\} \neq \emptyset$. It is elementary to check this provides a serialisation of x.

(ii) *"Only if":* Directly from the definition of trace of a configuration. *"If":* Via the local bijection between x and σx given by σ we obtain an enumeration

$$s_1, s_2, \ldots, s_n, \ldots$$

of x matching α in that $\sigma(s_i) = a_i$. The assumption that $s \rightarrowtriangle s'$ implies $\sigma(s)$ precedes $\sigma(s')$ in the enumeration α, entails $\{s_1, \ldots, s_i\} \in \mathcal{C}(S)$ for all i. Hence the enumeration of x is a serialisation making α a trace of x. □

Lemma 9.31 *Let $\sigma : S \to A$ be a strategy in a game A. Let $x \in \mathcal{C}^\infty(S)$. Let α be a serialisation of σx which is not a trace of $x \in \mathcal{C}^\infty(S)$. Then, there are $s, s' \in x$ with $pol(s) = -$ and $pol(s') = +$ and $s \rightarrowtriangle_S s'$ with (note the order reversal) $\sigma(s') \leq_\alpha \sigma(s)$ in α (regarded as a total order).*

Proof By assumption, any trace of x differs from α. We deduce there is $s \rightarrowtriangle s'$ in x with $\sigma(s) \not\leq \sigma(s')$ in the total order of α; otherwise we could serialise x to obtain the trace α —Proposition 9.30(ii). Now, $\sigma(s) \not\leq_A \sigma(s')$ in A as any serialisation must respect the order $\leq_A$. Hence, by the innocence of σ, we must have $pol(s) = -$ and $pol(s') = +$. Because α is totally ordered, $\sigma(s') \leq \sigma(s)$ in α. □

9.10.2 Characterisation of the 'May' Preorder

For stopping strategies (with games assumed race-free) we have:

Theorem 9.32 *Let (σ_1, M_1) and (σ_2, M_2) be stopping strategies in a common game. Then,*

(σ_1, M_1) may pass τ implies (σ_2, M_2) may pass τ, for all tests τ,
iff
all finite traces of σ_1 are traces of σ_2.

Proof Assume strategies $\sigma_1 : S_1 \to A$ and $\sigma_2 : S_2 \to A$. *"if":* Assume all finite traces of σ_1 are traces of σ_2. Suppose (σ_1, M_1) may pass test τ with event structure T. Then there is a successful configuration $w \circledast x_1 \in \mathcal{C}(T \circledast S_1)$, where $x_1 \in \mathcal{C}(S_1)$ and $w \in \mathcal{C}(T)$; it is successful in the sense that its image contains the success event $\checkmark$. Take a serialisation of $w \circledast x_1$; this induces a serialisation of x_1 to yield a trace. Then, by assumption, σ_2 has a configuration $x_2 \in \mathcal{C}(S_2)$ with the same trace, so a matching serialisation. Consequently the pairing $w \circledast x_2$ is defined with $w \circledast x_2 \in \mathcal{C}(T \circledast S_2)$; sharing the same image as $w \circledast x_1$ it is also successful.

"only if": We show the contraposition: assuming not all traces of σ_1 are traces of σ_2, we produce a test τ for which σ_1 may pass τ while it is not the case that σ_2 may pass τ.

Assume a trace α_1 of $x_1 \in \mathcal{C}(S_1)$ is not a trace of any $x_2 \in \mathcal{C}(S_2)$. Note that the trace α_1, and correspondingly x_1, must have at least one +ve event as otherwise, by receptivity, σ_2 could match the trace α_1. Any trace of x_2, with $\sigma_2 x_2 = \sigma_1 x_1$, differs from α_1. By Lemma 9.31, we deduce there are $s, s' \in x_2$ such that $s \rightarrowtriangle_2 s'$ with $pol(s) = -$ and $pol(s') = +$ and $\sigma_2(s') \leq_1 \sigma_2(s)$ in the total order α_1.

Thus for each $x_2 \in \mathcal{C}(S_2)$ with $\sigma_2 x_2 = \sigma_1 x_1$ we can choose $\theta(x_2) = (s, s')$ so that $s \rightarrowtriangle_2 s'$ in x_2 with $pol(s) = -$ and $pol(s') = +$ and $\sigma_2(s') \leq_1 \sigma_2(s)$ in α_1.

We now describe a test $\tau : T \to A^{\perp} \| \boxplus$ which will discriminate between σ_1 and σ_2. Let T_1' be the elementary event structure comprising events $T_1 =_{\text{def}} \sigma_1 x_1$ saturated with all accessible Opponent moves (note, in $A^{\perp}$), *i.e.* events

$$T_1' = \{a \in A \mid pol_{A^{\perp}}([a] \setminus T_1) \subseteq \{-\}\}$$

with order that of $A^{\perp}$ augmented with $\sigma_2(s') \leq_1 \sigma_2(s)$ for every choice $\theta(x_2) = (s, s')$ where $x_2 \in M_2$ and $\sigma_2 x_2 = \sigma_1 x_1$; the ensuing relation on T_1 is included in the total order α_1 so forms a partial order in which every element has only finitely many elements below it. (By design, T_1' "disagrees" with the causal dependency of each $x_2 \in \mathcal{C}(S_2)$ for which $\sigma_2 x_2 = \sigma_1 x_1$.) The polarities of events of T_1' are those of its events in $A^{\perp}$. On T_1' the map τ takes an event to its same event in $A^{\perp}$.

Let T be the event structure with polarity obtained from T_1' by adjoining a fresh 'success' event $\boxplus$ with additional causal dependency so $t_1 \leq_T \boxplus$ iff t_1 is $-$ve; as noted above there has to be at least one +ve event in x_1 and thus, by the reversal of polarity, at least one $t_1 \in T_1$ of $-$ve polarity. Then the obvious map $\tau : T \to A^{\perp} \| \boxplus$ is a strategy, and a suitable test for σ_1 and σ_2.

We have (i) σ_1 may pass τ, while (ii) it is not the case that σ_2 may pass τ.

To see (i), remark that the relation of causal dependency on T_1 is included in the the total order of the trace α_1 of x_1. Hence $\tau \circledast \sigma_1$ has a successful configuration $(T_1 \cup \{\boxplus\}) \circledast x_1$.

To show (ii), consider any finite configuration of $\tau \circledast \sigma_2$. It has the form $w \circledast x_2$ where $w \in \mathcal{C}(T)$ and $x_2 \in \mathcal{C}(S_2)$. The configuration $w \circledast x_2$ is unsuccessful because $\boxplus \notin w$, as we now show. By design, τ and σ_2 enforce opposing causal dependencies on a pair of synchronisations needed for $T_1 \circledast x_2$ to be defined whenever $x_2 \in \mathcal{C}(S_2)$ with $\sigma_2 x_2 = T_1$. At least two events of opposing polarity in T_1 are excluded from any pairing $w \circledast x_2$; one must be a $-$ve event of T_1 on which $\boxplus$ causally depends; hence $\boxplus \notin w$. $\square$

That the characterisation of the 'may' preorder above only depends on finite traces is not surprising, and familiar from previous work; the full-abstraction results of Dan Ghica and Andrzej Murawski w.r.t. 'may' behaviour rely only on finite traces (Ghica & Murawski, 2004).

Clearly the proof above does not rely on stopping configurations or tests being bare rather than pure strategies; the test used in the proof patently has no neutral events. The extra discriminating power of tests based on bare strategies, illustrated in Example 9.29, does play an essential role in the analgous result in the 'must' case, to be considered now.

9.10.3 Characterisation of the 'Must' Preorder

Recall an event structure $E = (E, \leq, \text{Con})$ is *consistent-countable* iff there is a function $\chi : E \to \omega$ from the events such that

$$\{e_1, e_2\} \in \text{Con} \;\&\; \chi(e_1) = \chi(e_2) \implies e_1 = e_2 \,.$$

Any configuration $x \in \mathcal{C}^{\infty}(E)$ of a consistent-countable event structure E is countable and so may be serialised as

$$x = \{e_1, e_2, \ldots, e_n, \ldots\}$$

so that $\{e_1, \ldots, e_n\} \in \mathcal{C}(E)$ for any finite subsequence. For the must case we assume that games are consistent-countable. It follows that strategies $\sigma : S \to A$ in consistent-countable games A have S consistent-countable. W.r.t. such a strategy σ, we have traces of all configurations.

Theorem 9.33 *Assume game A is consistent-countable. Let (σ_1, M_1) and (σ_2, M_2) be stopping strategies in A. Then,*

(σ_2, M_2) must pass τ implies (σ_1, M_1) must pass τ, for all tests τ,
iff
all traces of stopping configurations M_1 are traces of stopping configurations M_2.

Proof *"if":* Assume all traces of stopping configurations M_1 are traces of stopping configurations M_2. A stopping configuration of $\tau \circledast \sigma_1$ has the form $w \circledast x_1$ where w and x_1 are stopping configurations of τ and σ_1, respectively. A serialisation of $w \circledast x_1$ into a (possibly infinite) sequence induces a serialisation of $x_1 \in M_1$. By assumption, there is $x_2 \in M_2$ with the same trace in A as x_1. Consequently, $w \circledast x_2$ is a configuration of $\tau \circledast \sigma_2$ with the same image in $A \| \boxplus$. Moreover, $w \circledast x_2$ is a stopping configuration of $\tau \circledast \sigma_2$. Supposing (σ_2, M_2) must pass a test τ, the image of $w \circledast x_2$ contains $\checkmark$ whence the image of $w \circledast x_1$ contains $\checkmark$ ensuring (σ_1, M_1) must pass a test τ.

"only if": We show the contraposition: assuming not all traces of stopping configurations M_1 are traces of stopping configurations M_2, we produce a test τ for which (σ_2, M_2) must pass τ while it is not the case that (σ_1, M_1) must pass τ.

Assume a trace α_1 of $x_1 \in M_1$ is not a trace of any $x_2 \in M_2$.

In particular, consider any $x_2 \in M_2$ with $\sigma_2 x_2 = \sigma_1 x_1$. Then, any trace of x_2 differs from α_1. By Lemma 9.31, there are $s, s' \in x_2$ such that $s \rightarrowtriangle_2 s'$ with $pol(s) = -$ and $pol(s') = +$ and $\sigma_2(s') \leq_1 \sigma_2(s)$ in the total order α_1.

Thus for each $x_2 \in M_2$ with $\sigma_2 x_2 = \sigma_1 x_1$ we can choose $\theta(x_2) = (s, s')$ so that $s \rightarrowtriangle_2 s'$ in x_2 with $pol(s) = -$ and $pol(s') = +$ and $\sigma_2(s') \leq_1 \sigma_2(s)$ in α_1.

We build an event structure with polarity T and a test as bare strategy $\tau : T \to A^{\perp} \| N \| \boxplus$. We build the events of T as $T_1' \cup N \cup T_2$, a union of sets of events, assumed disjoint, described as follows.

- Let T_1' be the elementary event structure comprising events $T_1 =_{\mathrm{def}} \sigma_1 x_1$ saturated with all accessible Opponent moves, *i.e.* events

$$T_1' = \{a \in A \mid pol_{A^{\perp}}([a] \setminus T_1) \subseteq \{-\}\}$$

  with order that of A augmented with $\sigma_2(s') \leq_1 \sigma_2(s)$ for every choice $\theta(x_2) = (s, s')$ where $x_2 \in M_2$ and $\sigma_2 x_2 = \sigma_1 x_1$; the ensuing relation on T_1 is included in the total order α_1 so forms a partial order in which every element has only finitely many elements below it. (By design, T_1' "disagrees" with the causal dependency of each $x_2 \in M_2$ for which $\sigma_2 x_2 = \sigma_1 x_1$.) The polarities of events of T_1' are those of its events in $A^{\perp}$. On T_1' the map τ takes an event to its same event in $A^{\perp}$.
- N comprises a copy of the set of events of $-$ve polarity in T_1; all the events of N have neutral polarity; an event of N is sent by τ to its copy.
- T_2 comprises a copy of the set of events T_1; all the events of T_2 have +ve polarity; they are all sent by τ to $\checkmark =_{\mathrm{def}} (3, \boxplus)$.
- Causal dependency on T is that of T_1' augmented with dependencies from events of T_1 of $-$ve polarity to their corresponding copies in N.

- The consistency relation of T is that minimal relation which ensures that: any two distinct events of T_2 are in conflict; a +ve event of T_1 conflicts with its corresponding copy in T_2; and a neutral event in N conflicts with its corresponding copy in T_2. Formally,

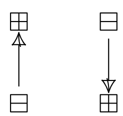
$$
\begin{aligned}
&X \in \mathrm{Con}_T \text{ iff } X \subseteq_{\mathrm{fin}} T_1 \cup N \cup T_2 \;\&\; |X \cap T_2| \leq 1 \;\&\\
&\quad (\forall t_1 \in X \cap T_1^+, t_2 \in X \cap T_2.\ t_1, t_2 \text{ are not copies of a common event}) \;\&\\
&\quad (\forall n \in X \cap N, t_2 \in X \cap T_2.\ n, t_2 \text{ are not copies of a common event}).
\end{aligned}
$$

Note that all the events over $\checkmark$, which together comprise the set T_2, can occur initially but can become blocked as moves are made in T_1. In particular, the set $T_1 \cup N$ is a $+$-maximal configuration of T with image in $A^\perp \| N \| \boxplus$ not containing any event over $\checkmark$. On the other hand any $+$-maximal configuration of T not including all the events T_1 will contain an event over $\checkmark$. Hence $\mathrm{St}(\tau)$ has an unsuccessful stopping configuration consisting of precisely all the events of T_1—it does not have an event over $\checkmark$—while all stopping configurations of $\mathrm{St}(\tau)$ which do not contain all the events of T_1 are successful—they contain an event over $\checkmark$.

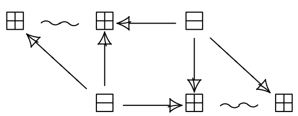
Consequently, (i) it is not the case that (σ_1, M_1) must τ, while (ii) (σ_2, M_2) must τ. To see (i), remark that the relation of causal dependency on T_1 is included in the total order of the trace α_1 of x_1. Hence $\mathrm{St}(\tau) \circledast \sigma_1$ has a stopping configuration $T_1 \circledast x_1$ which is unsuccessful and thus (σ_1, M_1) fails the must test τ. To show (ii), consider any stopping configuration of $\mathrm{St}(\tau) \circledast \sigma_2$. It comprises $w \circledast x_2$ where w is a stopping configuration of $\mathrm{St}(\tau)$ and $x_2 \in M_2$, a stopping configuration of σ_2. Now $w \not\supseteq T_1$, as by design τ and σ_2 enforce opposing causal dependencies on a pair of synchronisations needed for $T_1 \circledast x_2$ to be defined whenever $x_2 \in M_2$ with $\sigma_2 x_2 = T_1$. Thus w is successful in that it contains an event over $\checkmark$. Hence (σ_2, M_2) must pass τ. This completes the proof. □

Remark. By Example 9.29, the result above would not hold if tests were based solely on pure strategies.

Example 9.34 Let A be the game



Let σ_1 be the stopping strategy given by the identity map $\mathrm{id}_A : A \to A$ together with the +-maximal configurations of A. Let σ_2 be the stopping strategy derived from the event structure



in which there are additional occurrences of Player moves awaiting both moves of Opponent; the map to A is the obvious one and its stopping configurations the +-maximal ones. It can be checked that the two stopping strategies share the same traces of stopping configurations so are 'must' equivalent. $\square$

Infinite stopping configurations play an essential role in the 'must' behaviour of strategies:

Example 9.35 Let the game A consist of an infinite chain of alternating Player-Opponent moves:

$$\boxplus \longrightarrow \boxminus \longrightarrow \boxplus \longrightarrow \boxminus \longrightarrow \cdots \longrightarrow \boxplus \longrightarrow \boxminus \longrightarrow \cdots$$

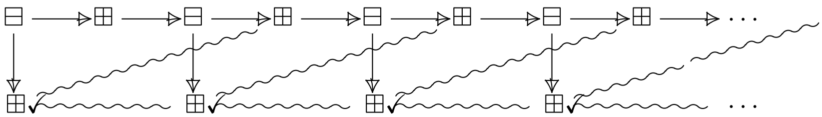
The strategy σ_1 has as consistent components a copy of A itself and copies of all its initial finite sequences of events ending with an Opponent move; events of different components are inconsistent with each other—we refrain from drawing the wiggly conflicts. The map σ_1 is obvious. (The construction is an instance of the sum of strategies.)

$$\begin{array}{l}
\boxplus \longrightarrow \boxminus \\
\boxplus \longrightarrow \boxminus \longrightarrow \boxplus \longrightarrow \boxminus \\
\boxplus \longrightarrow \boxminus \longrightarrow \boxplus \longrightarrow \boxminus \longrightarrow \boxplus \longrightarrow \boxminus \\
\qquad\qquad\qquad\quad \vdots \\
\boxplus \longrightarrow \boxminus \longrightarrow \boxplus \longrightarrow \boxminus \longrightarrow \cdots \longrightarrow \boxplus \longrightarrow \boxminus \\
\qquad\qquad\qquad\quad \vdots \\
\boxplus \longrightarrow \boxminus \longrightarrow \boxplus \longrightarrow \boxminus \longrightarrow \cdots \longrightarrow \boxplus \longrightarrow \boxminus \longrightarrow \boxplus \longrightarrow \cdots
\end{array}$$

The lowest component is a copy of A. The stopping configurations are all its +-maximal configurations, *i.e.* those sets consisting of a whole component or an initial subsequence of a component which ends in a Player move.

The strategy σ_2 is like σ_1 but without the extra infinite component of shape A. Its stopping configurations are all its +-maximal configurations, so are necessarily all finite.

The traces of σ_1's and σ_2's *finite* stopping configurations coincide. However the trace of the infinite stopping configuration of σ_1 cannot be a trace of σ_2. Accordingly, from Theorem 9.33, there is a test strategy which σ_2 must pass while σ_1 does not. A distinguishing test $\tau : T \to A^{\perp} \| \boxplus\checkmark$ has T comprising the infinite event structure shown below.



The strategy σ_2 must pass τ; all +-maximal configurations of $\tau \circledast \sigma_2$ are finite and contain a ✓-event. Whereas the strategy σ_1 can fail to enable a ✓-event through its extra infinite stopping configuration. □

9.11 The Rigid Image of a Stopping Strategy

In this section we rely on the material of Sect. 9.4, in particular that a strategy $\sigma : S \to A$ in a game A has a rigid image

$$\begin{tikzcd} S \arrow[r, two heads, "f"] \arrow[dr, "\sigma"'] & S_0 \arrow[d, "\sigma_1"] \\ & A \end{tikzcd},$$

where f is rigid epi and the rigid image σ_1 is a total map; the map σ_1 automatically inherits the properties required of a strategy. (The construction and key properties of rigid image are unaffected by the extra structure of polarity.) As has been remarked earlier (Winskel, 2016; Castellan et al., 2017), rigid-image strategies have the advantage of forming a category rather than a bicategory. Extended with stopping configurations they can support 'may' and 'must' behaviour.

Definition 9.36 Let (σ, M_S) be a stopping strategy. Let σ_1 be the rigid image of σ with accompanying 2-cell $f : \sigma \Rightarrow \sigma_1$ where f is rigid epi. We define the *rigid image* of (σ, M_S) to be $(\sigma_1, f M_S)$. A *rigid-image* stopping strategy is one which is its own rigid-image.

As a direct consequence of the last part of Lemma 9.28, we are assured the rigid image of a stopping strategy does not lose any 'may' and 'must' behaviour.

Proposition 9.37 *A stopping strategy is both 'may' and 'must' equivalent to its rigid image.*

As far as 'may' and 'must' behaviour is concerned it is sensible to regard two stopping strategies as equivalent if they share a common rigid image. Rigid-image equivalence transfers to an equivalence between bare strategies: two bare strategies are equivalent if under St we obtain equivalent stopping strategies. W.r.t. 'may' and 'must' behaviour we can choose to work in the category of rigid-image stopping strategies.

What axioms hold of stopping configurations? Such axioms should be preserved by the composition of stopping strategies and rigid image. They should also be complete in the sense that any stopping strategy which satisfied them is rigid-image equivalent to the stopping strategy of some bare strategy. We do not presently know a complete set of axioms for stopping configurations. Candidate axioms on a stopping strategy $\sigma : S \to A$ with stopping configurations M are

Axiom (i) $\forall x \in \mathcal{C}(S) \exists y \in M .\, x \subseteq y\,,$ and

Axiom (ii) $\forall y \in M, x \in \mathcal{C}^\infty(S) .\, x \subseteq y$ & x is +-maximal in $S \implies x \in M\,.$

The example below shows why we do not assume all +-maximal configurations are stopping. That property is not preserved by taking the rigid image.

Example 9.38 In forming the rigid image $\sigma_1 : S_1 \to A$ of a strategy $\sigma : S \to A$, related by rigid epi 2-cell $f : \sigma \Rightarrow \sigma_1$, it is possible to have an infinite configuration of S_1 which is not in the direct image under f of any configuration of S; in particular it is possible to have a +-maximal configuration of S_1 which is not a direct image of any +-maximal configuration S. For example, let A comprise an infinite chain of Player events. Take S to be the sum of all finite subchains. The rigid image of S is A itself which has +-maximal configuration comprising all the events in the infinite chain, not the image of any configuration of S_1. Thus, in forming the rigid image of a stopping strategy, we cannot assume that all the +-maximal configurations of the rigid image are stopping. □

9.12 Strategies as Concurrent Processes

The paper (Castellan et al., 2014) is a closely related study of concurrent strategies from the perspective of concurrent processes, considering how concurrent games and strategies are objects which we can program. Concurrent strategies are shown to support operations yielding an economic yet rich higher-order concurrent process language, which shares features both with process calculi and nondeterministic dataflow. There a slightly weakened definition of bare strategies plays a key role in providing an operational semantics. It would be satisfying to complete this story by providing inequational proof systems for 'may' and 'must' equivalence based on its syntax for strategies, drawing inspiration from the classic work of Hennessy and de Nicola (1984).

Process calculi often allow unrestricted recursion. Strategies, as presented here, form a model of linear logic which restricts the copying of parameters needed in recursive definitions. The treatment of unrestricted recursion requires a move to nonlinear strategies over games with symmetry (Castellan et al., 2014, 2019). The recursive definition of bare strategies and strategies can follow classical ideas; 2-cells include the rigid embeddings and inclusions of Winskel (1982). Less clear is how to carry out recursive definitions directly on stopping strategies; the 2-cells we have chosen would seem to be too restrictive; and the nature of stopping configurations, that they can be infinite without finite approximations, would push the development into non-continuous operations—nonstandard, if not in itself a bad thing.

A treatment of *winning* concurrent strategies has been presented (Clairambault et al., 2012). Informally a strategy is winning if it must end up in a winning configuration of the game regardless of the behaviour of Opponent. In idea this is very close to *controllability* in Alur et al. (2002). Because the semantics of composition of composition of strategies in Clairambault et al. (2012) is inattentive to the possibilities of deadlock and divergence, a strategy which is obtained as a composition may be deemed winning there and yet possibly deadlock or diverge before reaching a winning configuration (Castellan et al., 2014). Fortunately the treatment of winning strategies *ibid.* generalises straightforwardly to stopping strategies which keep track of deadlock and divergence, and thus repair this defect. The role of +-maximal configurations in Clairambault et al. (2012) is replaced by that of stopping configurations: a bare or stopping strategy is winning iff all its stopping configurations image to winning strategies in the game.

Forearmed with concurrent strategies and games with symmetry it would be interesting to revisit old ideas extending testing to other equivalences beyond those of 'may' and 'must' (Abramsky, 1987). Certainly one could wish for a better integration of games and strategies with the classical work on concurrency, process algebra and its equivalences included. The medium of concurrent games and strategies based on event structures also provides an inroad into the formalisation and analysis of probabilistic and quantum languages and processes (Winskel, 2013; Castellan et al., 2018; Clairambault et al., 2019).

Acknowledgements Thanks to Jonathan Hayman, Sacha Huriot, Martin Hyland, Marc Lasson, Conor McBride and Marc de Visme for encouragement and helpful discussions. Thanks to the anonymous referee. Support of Advanced Grant ECSYM (2011–17) of the European Research Council is acknowledged with gratitude. The second author gratefully acknowledges support by ANR project DyVerSe (ANR-19-CE48-0010-01) and Labex MiLyon (ANR-10-LABX-0070) of Université de Lyon, within the program "Investissements d'Avenir" (ANR-11-IDEX-0007), operated by the French National Research Agency (ANR).

References

Abramsky, S., & Melliès, P. A. (1999). Concurrent games and full completeness. In *LICS '99*. IEEE Computer Society.

Abramsky, S. (1987). Observation equivalence as a testing equivalence. *Theoretical Computer Science, 53,* 225–241.

Alur, R., Henzinger, T. A., & Kupferman, O. (2002). Alternating-time temporal logic. *Journal of the ACM, 49*(5), 672–713.

Brookes, S., Hoare, C. A. R., & Roscoe, A. W. (1984). A theory of communicating sequential processes. *Journal of the ACM, 31,* 560–599.

Castellan, S., Clairambault, P., & Winskel, G. (2014). Symmetry in concurrent games. In *Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS '14, Vienna, Austria, 14–18 July 2014*. ACM.

Castellan, S., Clairambault, P., & Winskel, G. (2015). The parallel intensionally fully abstract games model of PCF. In *30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015, Kyoto, Japan, 6–10 July 2015* (pp. 232–243). IEEE Computer Society.

Castellan, S., Clairambault, P., & Winskel, G. (2019). Thin games with symmetry and concurrent hyland-ong games. *Logical Methods in Computer Science, 15*(1).

Castellan, S., Clairambault, P., & Winskel, G.: Distributed strategies made easy. In K. G. Larsen, H. L., Bodlaender, J. Raskin, (Eds.), *42nd International Symposium on Mathematical Foundations of Computer Science, MFCS 2017, 21–25 Aug 2017—Aalborg, Denmark* (Vol. 83, pp. 81:1–81:13). LIPIcs., Schloss Dagstuhl—Leibniz-Zentrum für Informatik.

Castellan, S., Clairambault, P., Hayman, J., & Winskel, G. (2018). Non-angelic concurrent game semantics. In *Proceedings of FOSSACS 2018* (Vol. 10803, pp. 3–19). Lecture notes in computer science. Berlin: Springer.

Castellan, S., Clairambault, P., Paquet, H., & Winskel, G. (2018). The concurrent game semantics of probabilistic PCF. In A. Dawar, & E. Grädel, (Eds.), *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, Oxford, UK, 09–12 July 2018* (pp. 215–224). ACM.

Castellan, S., Clairambault, P., Rideau, S., & Winskel, G. (2017). Games and strategies as event structures. *Logical Methods in Computer Science, 13*(3).

Castellan, S., Hayman, J., Lasson, M., & Winskel, G. (2014). Strategies as concurrent processes. *ENTCS, 308*.

Clairambault, P., de Visme, M., & Winskel, G. (2019). Game semantics for quantum programming. *Proceedings of the ACM on Programming Languages, 3*(POPL), 32:1–32:29.

Clairambault, P., Gutierrez, J., & Winskel, G. (2012). The winning ways of concurrent games. In *LICS 2012* (pp. 235–244).

Conway, J. (2000). *On numbers and games*. Wellesley, MA: A K Peters.

De Nicola, R., & Hennessy, M. (1984). Testing equivalences for processes. *Theoretical Computer Science, 34,* 83–133.

Faggian, C., & Piccolo, M. (2009). Partial orders, event structures and linear strategies. In *TLCA '09* (Vol. 5608). LNCS. Berlin: Springer.

Ghica, D. R., & Murawski, A. S. (2004). Angelic semantics of fine-grained concurrency. In *FOSSACS'04, LNCS 2987*. Berlin: Springer.

Harmer, R., & McCusker, G. (1999). A fully abstract game semantics for finite nondeterminism. In *14th Annual IEEE Symposium on Logic in Computer Science, Trento, Italy, 2–5 July 1999* (pp. 422–430).

Joyal, A. (1997). Remarques sur la théorie des jeux à deux personnes. *Gazette des Sciences Mathématiques du Québec, 1*(4)

Melliès, P. A., & Mimram, S. (2007). Asynchronous games: Innocence without alternation. In *CONCUR '07* (Vol. 4703). LNCS. Berlin: Springer.

Milner, R. (1980). *A calculus of communicating systems* (Vol. 92). Lecture notes in computer science. Berlin: Springer.

Nielsen, M., Plotkin, G., & Winskel, G. (1981). Petri nets, event structures and domains. *TCS, 13,* 85–108.

Rideau, S., & Winskel, G. (2011). Concurrent strategies. In *LICS 2011*.

Winskel, G. (1980). Events in computation. Ph.D. thesis, Edinburgh.

Winskel, G. (1982). Event structure semantics for CCS and related languages. In *ICALP'82* (Vol. 140). LNCS. Berlin: Springer. A full version is available from Winskel's homepage.

Winskel, G. (1986). Event structures. In *Advances in petri nets* (Vol. 255, pp. 325–392). LNCS. Berlin: Springer.

Winskel, G. (2004). Linearity and nonlinearity in distributed computation. In *Linear logic in computer science* (pp. 151–188). London Mathematical Society Lecture Note Series (CUP).
Winskel, G. (2013). Strategies as profunctors. In *FOSSACS 2013*. Lecture notes in computer science. Berlin: Springer.
Winskel, G. (2016). Event structures, stable families and concurrent games. http://www.cl.cam.ac.uk/~gw104/ecsym-notes.pdfd.
Winskel, G., & Nielsen, M. (1995). Models for concurrency. In S. Abramsky, D. Gabbay (Eds.), *Semantics and logics of computation*. OUP.
Winskel, G. (2007). Event structures with symmetry. *Electronic Notes in Theoretical Computer Science, 172,* 611–652.
Winskel, G. (2009). Prime algebraicity. *Theoretical Computer Science, 410*(41), 4160–4168.
Winskel, G. (2012). Deterministic concurrent strategies. *Formal Aspects of Computing, 24*(4–6), 647–660.
Winskel, G. (2013). Distributed probabilistic and quantum strategies. *Electronic Notes in Theoretical Computer Science, 298,* 403–425.

Chapter 10
A Tale of Additives and Concurrency in Game Semantics

Pierre Clairambault

Abstract Twenty years ago, Abramsky and Melliès published their famous paper, *Concurrent Games and Full Completeness*. In that paper, they advocated the switch to a *truly concurrent* canvas to address the issue known as the *Blass problem*, the non-associativity of composition in Blass' games model for Linear Logic, diagnosed as an excess of sequentiality. Their model, *concurrent games*, was the first of a family of *positional* or *causal* game semantics which has since then shown merits far beyond the full completeness problem for Linear Logic. In this paper, we tell and revisit the story of models of MALL in game semantics, in the modern clothes of concurrent games on event structures, from Blass games to Melliès' approach to fully complete models of Linear Logic.

Keywords Game semantics · Concurrent games · Linear logic

10.1 Introduction

Game semantics in its modern form arose in the early 90s, driven by the problem of *full abstraction for PCF* (Milner, 1977). The idea to represent formulas as games and validity as the existence of a winning strategy was not new, going back to at least the work of Lorenzen and Lorenz in the 60s. But in the 80s and early 90s, the scientific landscape was rich in developments hinting at a dynamic semantics for programs and proofs. In 1982, Berry and Curien introduced *sequential algorithms* (Berry & Curien, 1982), attempting to capture higher-order sequentiality by presenting programs as functions along with a specific order, or "algorithms" to compute them, prefiguring strategies.[1] In 1989, Girard introduced the *Geometry of Interaction (GoI)* (Girard,

[1]It appeared later that sequential algorithms are *indeed* a game semantics, as they admit a linear decomposition into the category of simple games via the Curien-Lamarche exponential (Amadio et al., 1998).

P. Clairambault (✉)
LIS, Aix Marseille Univ, Université de Toulon, CNRS, 13288 Marseille Cedex 09, France
e-mail: pierre.clairambault@cnrs.fr

A. Palmigiano and M. Sadrzadeh (eds.), *Samson Abramsky on Logic and Structure in Computer Science and Beyond*, Outstanding Contributions to Logic 25,
https://doi.org/10.1007/978-3-031-24117-8_10

1989), a model of Linear Logic (Girard, 1987) representing proofs as operators on Hilbert spaces with an interactive form of composition.[2] In 1991, Coquand gave a game semantics of classical arithmetic, interpreting proofs as strategies with the ability to *backtrack*[3] (Coquand, 1995). In 1992, Blass gave a games model for full propositional Linear Logic (Blass, 1992).

Perhaps the first paper on *game semantics*, taken with a modern understanding of the term, is Abramsky and Jagadeesan's 1992 paper on *full completeness* for Multiplicative Linear Logic (MLL) with the MIX rule (Abramsky & Jagadeesan, 1992). More than just novel techniques, the paper introduces a change in perspective. Earlier games models were interested in capturing *logical validity*: a formula A is interpreted as a two-player game $[\![A]\!]$, and A is valid iff the first player (hereby called *Player*, with a capital 'P') has a winning strategy against the second player (hereby called *Opponent*)— a standard *completeness* property. In contrast, in Abramsky and Jagadeesan's view, the winning strategy is not merely a witness for truth; it is the representation of a *proof*.[4] Accordingly, the interpretation of proofs as strategies should respect the natural equivalences between proofs, such as commuting conversions and cut elimination rules. Furthermore, *completeness* may be refined into *full completeness*, which asks that every winning strategy should be the representation of a proof. A fully complete games model allows one to view winning strategies as intrinsic, canonical representations of proofs, with all bureaucracy factored out.

That the natural equivalences between proofs should be preserved structures the interpretation. The CUT rule, in particular, corresponds to *composition* of strategies— and as two proofs differing only with respect to associativity of CUT should be the same, games and strategies should form a category.[5] In contrast, Blass games, while interpreting full propositional Linear Logic, do not form a category: composition of strategies fails associativity, a phenomenon now known as the *Blass problem*,[6] reviewed in Sect. 10.2.2.

Following this early history, the first decade of game semantics was intertwinned with Linear Logic. Hyland and Ong extended the Abramsky-Jagadeesan model to get rid of MIX (Hyland & Ong, 1993). Further fragments of Linear Logic were addressed: e.g. classical Linear Logic by Baillot et al. (1997), the intuitionistic fragment was modeled by Lamarche (1992), McCusker (1998) and Abramsky et al. (2000). Despite

[2] It appeared later that these operators represent a "history-free skeleton" in terms of Abramsky-Jagadeesan games (Abramsky & Jagadeesan, 1992) or Abramsky-Jagadeesan-Malacaria (AJM) games (Abramsky et al., 2000), informing links between GoI and game semantics (Baillot, 1999).

[3] Backtracking prefigures the *pointers* of Hyland-Ong games, with a composition mechanism prefiguring innocent interaction.

[4] To our knowledge the first paper examining what should be a model for *proofs* is Girard's (1991) This change of focus is also in line with a wealth of developments at the same time on the *Curry-Howard correspondence*, moving the focus from mere provability to proofs, their identity and computational content.

[5] It seems that Joyal should be attributed the very first category of games and strategies [of Conway games (Conway, 2001)], in a 1977 paper in French in the *Gazette mathématique du Québec* (Joyal, 1977).

[6] Note however that Blass never claimed composition to be associative in his model.

this, a proper treatment (the established "gold standard" now being *full completeness*) of additives in classical Linear Logic remained long elusive. This finally came in the 1999 paper by Abramsky and Melliès, "*Concurrent games and full completeness*" (Abramsky & Melliès, 1999).

To construct a game semantics of Multiplicative Additive Linear Logic (MALL), it seems reasonable to start with Blass games, and attempt to understand and sidestep the Blass problem. In Abramsky (2003), Abramsky diagnoses the non-associativity as caused by an excessive sequentiality, in a sense that will be reviewed in the course of the paper. In that view, and although it is not immediately clear in what sense MALL is intrinsically concurrent, it makes sense to move to a concurrent framework for games. Abramsky and Melliès introduce *concurrent games* (Abramsky & Melliès, 1999) to that end. In this paper we shall however argue—following intuitions by Melliès ultimately leading to his fully complete model of full propositional linear logic (Melliès, 2005)—that the key aspect allowing concurrent games to achieve full completeness is not quite that they are concurrent. Instead, it is due to their *causal*, or *positional* nature. This will be discussed at length in this paper.

This *positional/causal* aspect (we shall see that those come hand in hand) is far from anecdotal. In that respect, Abramsky and Melliès' model is the first of a growing family of game semantics questioning the premise that strategies should simply be the aggregation of totally ordered, chronological execution traces. This family includes Melliès and Mimram's *asynchronous games* (Melliès & Mimram, 2007), Faggian and Piccolo's strategies as partial orders (Faggian & Piccolo, 2009), Rideau and Winskel's non-deterministic extension to *concurrent games on event structures* (Rideau & Winskel, 2011), Sakayori and Tsukada's framework (Sakayori & Tsukada, 2017) using DAG-like structures as plays. While those games are typically not the best fit for full abstraction results (as the causal information they record is unobservable), they form a valuable alternative to traditional models. Their *causal* nature allows to extend conservatively traditional notions such as *innocence* to parallel evaluation of programs (Castellan et al., 2015); and more generally may be leveraged for causal analysis of programs or proofs (Alcolei et al., 2018, 2019). Their *positional* nature sheds light on the relationship between static and dynamic denotational models, including in the presence of quantitative effects (Castellan et al., 2018; Clairambault & Marc de Visme, 2020). These cited works put at play the same *causality* and *positionality* that—as is our view—permitted Abramsky and Melliès' full completeness result for MALL.

In this paper, the phrase *concurrent games* will refer to this entire family of games. We will adopt more specific phrases to refer to precise technical settings. The purpose of this paper is to give a modern account of fully complete games models for MALL, putting the historical approach in perspective with recent developments in this family of concurrent games. The paper has a few original contributions: most notably, the account given of the link between concurrent games via event structures and via closure operators is new. But mostly, the paper assembles and presents in a uniform technical setting results appearing in various earlier papers, notably by Abramsky and Melliès (1999), Melliès (2005), Melliès and Mimram (2007), Melliès and Tabareau (2010)—we nonetheless hope that it will be helpful in making more accessible a nice line of research on which few researchers have a complete view.

The paper is organized as follows. In Sect. 10.2 we present the model of Abramsky and Melliès (1999) and its historical context. We describe MALL, Blass' model and the Blass problem, and introduce concurrent games via closure operators. In Sect. 10.3 we introduce concurrent games on event structures, at first as an alternative way to formulate Abramsky and Melliès' interpretation. We detail the connection between the two concurrent games frameworks. After a discussion on the quotient involved in Abramsky and Melliès' construction, in Sect. 10.4 we give a fully complete model for (a fragment of) MALL, following Melliès' methodology for his fully complete model of full propositional linear logic (Melliès, 2005).

10.2 The Blass Problem and Concurrent Games

In this section we introduce MALL, then review the Blass problem and concurrent games.

10.2.1 Multiplicative Additive Linear Logic

We consider here the multiplicative additive fragment of Linear Logic with units, but *without* propositional atoms. Atoms are not a fundamental obstacle: all the results presented here could be extended for instance as in Abramsky and Melliès (1999) by representing formulas with atoms as functors of mixed variance and proofs as dinatural transformations—Cuvillier has recently proposed an alternative relying on nominal sets (Cuvillier, 2023). We omit them for simplicity.

Formulas of MALL are generated by the following grammar.

$$A, B ::= 1 \mid \bot \mid 0 \mid \top \mid A \otimes B \mid A \parr B \mid A \oplus B \mid A \,\&\, B$$

We call $1, \bot, \otimes, \parr$ the **multiplicative** connectives, and $0, \top, \oplus$ and $\&$ the **additive** connectives. Each formula A has a **dual** $A^\perp$, defined by De Morgan duality between 1 and $\bot$, 0 and $\top$, $\otimes$ and $\parr$ and $\oplus$ and $\&$. We consider one-sided **sequents**, of the form $\vdash \Gamma$ where $\Gamma = A_1, \ldots, A_n$ is a list of formulas. We give the rules in Fig. 10.1. In addition to these, we consider that there is an explicit exchange rule allowing us to reorder formulas in a sequent, coping with the fact that sequents are lists rather than multisets. We will however, keep applications of this rule silent throughout this paper.

The fragment with only multiplicative connectives is known as Multiplicative Linear Logic (MLL). It is well-known that a (categorical) model of MLL is a $\star$-

$$\frac{}{\vdash A^{\perp}, A}\,\text{Ax} \qquad \frac{}{\vdash 1}\,1 \qquad \frac{\vdash \Gamma}{\vdash \Gamma, \perp}\,\perp \qquad \frac{}{\vdash \Gamma, \top}\,\top \qquad \frac{\vdash \Gamma, A \quad \vdash B, \Delta}{\vdash \Gamma, A \otimes B, \Delta}\,\otimes \qquad \frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \parr B}\,\parr$$

$$\frac{\vdash \Gamma, A \quad \vdash \Gamma, B}{\vdash \Gamma, A \,\&\, B}\,\& \qquad \frac{\vdash \Gamma, A}{\vdash \Gamma, A \oplus B}\,\oplus_l \qquad \frac{\vdash \Gamma, B}{\vdash \Gamma, A \oplus B}\,\oplus_r \qquad \frac{\vdash \Gamma, A \quad \vdash A^{\perp}, \Delta}{\vdash \Gamma, \Delta}\,\textsc{Cut}$$

Fig. 10.1 Rules of MALL

autonomous category, i.e. a symmetric monoidal closed category $\mathcal{C}$ with a *dualizing object* $\perp$, such that for all object A, the canonical map $A \to (A \multimap \perp) \multimap \perp$ is an isomorphism. It follows from this structure that $\mathcal{C}$ is self-dual: the *negation* $(-) \multimap \perp : \mathcal{C}^{\mathrm{op}} \to \mathcal{C}$ is an equivalence; so in particular $\mathcal{C}$ has products if and only if it has coproducts. A model of MALL is a $\star$-autonomous category which is additionally cartesian—hence also cocartesian.

10.2.2 The Blass Problem

Constructing a games-based self-dual category with products and coproducts, is really difficult. It is well-known among game semanticists that the behaviour of additive connectives strongly depends on the *polarity* of the games considered. All these notions will be made precise later on, but say—for the moment informally—that a game is *positive* if Player always starts, and that a game is *negative* if Opponent always starts. It is part of the folklore of game semantics that categories of negative games support products (and naturally apply to model Call-By-Name languages), while categories of positive games support coproducts (and naturally apply to model Call-By-Value languages).[7] This reading matches the natural game-theoretic reading of the additive connectives of Linear Logic: in $A \oplus B$ we have A or B but we, the proof, choose—which is positive; while in $A \& B$ we have A or B but the environment chooses—which is negative.

But this follows the implicit premise that formulas should be interpreted into a single model with fixed polarity, positive or negative. It would make sense instead to have some formulas give positive games, and some others give negative games; and this is indeed how Blass games proceed. We now recall Blass games and the Blass problem, following closely the presentation of Abramsky in (2003). Formally, Blass games are trees

$$A ::= \prod_{i \in I} A_i \,\Big|\, \coprod_{j \in J} A_j$$

[7] It is possible to add the missing connective formally, for instance one can add coproducts freely via the *Fam construction* (Abramsky & McCusker, 1997) in a category of negative games so as to have both products and coproducts, but this does not provide a model of MALL as we still lack self-duality.

where I, J are finite sets. A game $\coprod_{j\in J} A_j$ is positive, and a game $\prod_{i\in I} A_i$ is negative. A strategy (for Player) on $\prod_{i\in I} A_i$ is the data of a strategy (for Player) on A_i, for all $i \in I$. Likewise, a strategy (for Player) on $\coprod_{j\in J} A_j$ is the data of some $j_0 \in J$, and a strategy (for Player) on A_{j_0}. For now, the only assumption the definition of Blass games makes is that games should be sequential: at each point, it is one of the players' turn to play. There is no general assumption as to which player starts the game, and the two might not alternate.

Setting up the interpretation of MALL formulas into Blass games, it is clear from the discussion that we should have $[\![A\&B]\!] = [\![A]\!] \sqcap [\![B]\!]$ and $[\![A \oplus B]\!] = [\![A]\!] \sqcup [\![B]\!]$. But if we are to form a category of Blass games and strategies, then it follows that $\otimes$, as a left adjoint, should preserve coproducts, hence a $\otimes$ involving at least one positive game should be positive. By duality, a $⅋$ involving at least one negative game should be negative. The only case left is the $\otimes$ of two negative games, which is defined as

$$A \otimes B = \prod_{i\in I}(A_i \otimes B) \sqcap \prod_{j\in J}(A \otimes B_j)$$

for $A = \prod_{i\in I} A_i$ and $B = \prod_{j\in J} B_j$, saying that if A and B are negative, Opponent first picks a component of the tensor and makes a move in that component. So a tensor of negative games is negative—in fact in exposing the Blass problem we will not refer to this specific definition but only to the fact that the tensor of negative games is negative, which is hard to avoid (as otherwise the tensor would only ever yield positive games). The definition for the $⅋$ of two positive games is dual.

At this point it looks like there is no obstacle to form a category Blass, with Blass games as objects and, as morphisms from A to B, strategies on $A^\perp ⅋ B$. But here comes the "Blass problem". Assume we want to compose (with games annotated with their polarity):

$$\sigma : (A_-)^\perp ⅋ B_+ \qquad \tau : (B_+)^\perp ⅋ C_- \qquad \delta : (C_-)^\perp ⅋ D_+$$

where σ wants to play immediately on the left, and δ wants to play immediately on the right. Both strategies want to perform immediately some visible action, so in principle there is no reason to make them wait. Indeed $(A_-)^\perp ⅋ D_+$ is positive: it is Player's time to play on either or A or D, and the moves offered by both σ and δ may apply; but since Blass games are sequential, only one of them will be able to play immediately. The situation being symmetric, it is clear that something is going to unfold differently between the two associations. And indeed, imagine we first form $\tau \circ \sigma : (A_-)^\perp ⅋ C_-$, and consider

$$\tau \circ \sigma : (A_-)^\perp ⅋ C_- \qquad \delta : (C_-)^\perp ⅋ D_+.$$

The game $(A_-)^\perp ⅋ C_-$ is now negative, so σ is not able to play on the left, leaving δ to win the race playing on D. Symmetrically, in $(\delta \circ \tau) \circ \sigma$, σ starts playing on the left.

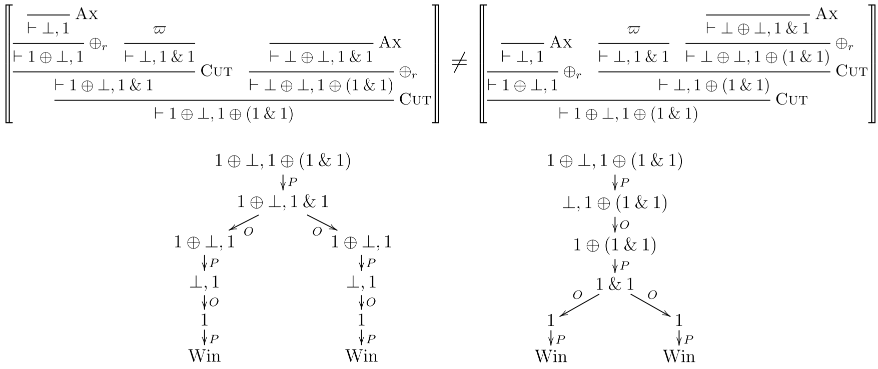


Fig. 10.2 The concrete impact on the Blass problem on the interpretation

This happens very concretely in the interpretation of proofs: in Fig. 10.2 we show two proofs differing only with the order of cuts, and the corresponding strategies, which differ because of the Blass problem—the strategies perform the same actions, but not in the same order. Here ϖ is the proof of $\vdash \perp, 1 \,\&\, 1$ obtained with a $\&$ rule followed by axioms.

The Blass problem is sometimes mistakenly quoted as expressing that composition is not associative in a non-polarized setting, i.e. unless one fixes the ambient polarity of games to be positive or negative. The author has heard some people explicitly avoiding non-polarized settings, for "fear of the Blass problem". But these people should rest in peace: associativity of composition is actually quite robust and does not need at all polarization. In fact the very first category of games and strategies, Joyal's category of Conway games, assumes no general polarization hypothesis—in a Conway game both players can have available actions in the same state, and we will see further examples later on in this paper. Instead, the Blass problem is a consequence of the very specific way in which we have set up the traffic lights so that to always give priority to coproducts on tensors and products on pars, which in turn was required to get the right behaviour for $\oplus$ and $\&$.

Abramsky analyses the Blass problem as an excess of sequentiality (Abramsky, 2003). And indeed, if above we authorized both σ and δ to play *concurrently*, the non-associativity would be resolved. We next review Abramsky and Melliès' *concurrent games via closure operators* (1999), and observe how they resolve the non-associativity phenomenon.

10.2.3 Concurrent Games

Concurrent games via closure operators (Abramsky & Melliès, 1999) were motivated as a way around the Blass problem. They depart from the sequential substrate of earlier games models. In particular, concurrent strategies may play several moves simultaneously.

Firstly, *games* are replaced by *domains* with elements thought of as *positions*. More specifically, we consider games to be **dI-domains** (Berry, 1979), i.e. directed-complete, bounded-complete partial orders satisfying two further axioms *"d"* and *"I"* that we shall not need to repeat here. If D is a dI-domain, $D^\top$ denotes its extension with a top element $\top$—it then follows that $D^\top$ is a complete lattice.

Secondly, strategies are *continuous, stable closure operators*[8]—$f : D \to D'$ between dI-domains is **stable** (Berry, 1979) iff for $x, y \in D$, if x, y are bounded then $f(x \wedge y) = f(x) \wedge f(y)$.

Definition 1 A **closure-strategy** on dI-domain D, written $\sigma : D$, is a **continuous stable closure operator** on $D^\top$, i.e. a monotone and continuous function $\sigma : D^\top \to D^\top$ which is (i) **extensive** (for all $x \in D^\top$, $x \leq \sigma(x)$), (ii) **idempotent** (for all $x \in D^\top$, $\sigma(\sigma(x)) = \sigma(x)$), and (iii) **stable**[9] (there is a stable function $f : D \to D$ such that for all $x \in D$ such that $\sigma(x) \neq \top$, $\sigma(x) = x \vee f(x)$).

Intuitively, given a position $x \in D$, $\sigma(x)$ is the new position obtained by adding all moves that σ is prepared to play in position x. The first axiom, $x \leq \sigma(x)$, means intuitively that σ may only add new moves to those already present. The second axiom, $\sigma(\sigma(x)) = \sigma(x)$, formalizes the idea that as applying σ saturates the current position with all moves available with the current knowledge, any $\sigma(x)$ must be a fixpoint. The $\top$ element is meant to capture positions on which σ is undefined: $\sigma(x) = \top$ means that σ has no well-defined behaviour on x. It has to be a top element by monotonicity.

One may wonder in what sense it is legitimate to call this a game semantics. There are no polarities in the definition, no Player, no Opponent. In fact there are no moves, only positions. Moves can be captured indirectly as pairs $x, y \in D$ such that $x < y$ with no position in between (for which we write $x \mathrel{-\!\!\subset} y$), but even then such a move has no well-defined notion of polarity. Nevertheless, any Blass game A induces a dI-domain D_A by starting with the partial order of finite branches which is then completed, adding the missing infinite branches. Any (sequential) strategy on A yields a closure-strategy which, for any finite branch, extends it with the moves it is prepared to play.

Furthermore, domains and concurrent strategies may be organized as a category. If D_1, D_2 are domains, then their *tensor* is defined simply as

$$D_1 \otimes D_2 = D_1 \times D_2$$

the cartesian product. A concurrent strategy *from* D_1 to D_2 is $\sigma : D_1 \otimes D_2$, written $\sigma : D_1 \mathrel{\rightarrow\!\!\!\!\!+\!\!\!>} D_2$. To define composition, we first define *closed interaction*. If $\sigma : D$ and $\tau : D$ are two closure operators on the same domain, we may define, following (Abramsky & Melliès, 1999):

[8] Some additional conditions appear in the course of Abramsky and Melliès (1999), omitted here as they play no role.

[9] Condition (iii) implies condition (i), however we state conditions (i) and (ii) because together they define a *closure operator*, a standard notion independently of stability.

$$\langle\sigma\big|\tau\rangle = Y\,(\sigma \circ \tau) = Y\,(\tau \circ \sigma) \in D$$

obtained by playing alternatively σ and τ until reaching a fixpoint. Given $\sigma : D_1 \twoheadrightarrow D_2$, $\tau : D_2 \twoheadrightarrow D_3$, and $(x, z) \in D_1 \otimes D_3$ we first compute $y \in D_2$ that they agree to reach with

$$y = \langle\pi_2 \circ \sigma(x, -)\big|\pi_1 \circ \tau(-, z)\rangle \in D_2$$

and define $(\tau \odot \sigma)(x, z) = (\pi_1 \circ \sigma(x, y), \pi_2 \circ \tau(y, z)) \in D_1 \otimes D_3$; this defines a concurrent strategy. Composition is associative, and for any domain D, there is a strategy $\text{æ}_D : D \twoheadrightarrow D$ defined as $\text{æ}_D(x, y) = (x \vee y, x \vee y)$ serving as identity. Moreover:

Proposition 1 *There is a compact closed category* Clos *having as objects dI-domains and as morphisms from* D_1 *to* D_2 *the closure-strategies* $\sigma : D_1 \twoheadrightarrow D_2$.

Recall that a compact closed category is a degenerate model of MLL where $\otimes = \parr$ (Cockett & Seely, 1997). Here, we furthermore have a trivial duality $D^* = D$, similarly to the relational model. Nevertheless, this lets us interpret the multiplicative connectives of MLL. We may extend this to the additives as well by setting

$$[\![A \& B]\!] = [\![A \oplus B]\!] = ([\![A]\!] + [\![B]\!])_\perp$$

the lifted sum of $[\![A]\!]$ and $[\![B]\!]$. There are associated constructions on strategies for the introduction rules, omitted for now. Altogether this gives an interpretation of MALL in Clos, which does "solve the Blass problem" in the sense that composition is associative: so, for instance, the two proofs of Fig. 10.2 have the same interpretation, a closure-strategy which intuitively starts playing the two competing Player moves of Fig. 10.2 *in parallel*.

We postpone for now the concrete description of this interpretation and of its properties. Indeed, before we do that, we will see that the interpretation of MALL in Clos factors through the more concrete games formalism of *concurrent games on event structures* (where, for instance, the two proofs of Fig. 10.2 will be both interpreted by the one parallel strategy of Fig. 10.3). In the next section we give an introduction to concurrent games on event structures and a formal link with closure-strategies. Then we revisit the interpretation above, and discuss its properties.

10.3 Games on Event Structures and Closure Operators

Concurrent games via closure operators are inherently *positional*: points in the dI-domain interpreting a formula correspond to *positions* in the corresponding game. In contrast, concurrent games on event structures are more fine-grained: games focus on *individual observable events* rather than positions.

10.3.1 Games and Domains

Games and constructions. To start our concrete reconstruction of the interpretation of MALL of the previous section, we will first aim to represent the dI-domains interpreting formulas as explicit domains of positions, i.e. to explicitly have points of the domains be sets of moves. For that it is natural to start with the definition of *event structures*, in light of the fact that their domains of configurations are exactly dI-domains (Winskel, 2009). *Event structures* come in many variants—in this paper we use *prime event structures* as in Winskel (1986), except that the general *consistency relation* is restricted to a binary conflict.

Definition 2 An **event structure** is a tuple $E = \langle |E|, \leq_E, \#_E \rangle$ where $|E|$ is a set of **events**, $\leq_E$ is a partial order called **causality**, and $\#_E$ is an irreflexive symmetric binary relation called **conflict**. These data must moreover satisfy the following additional axioms.

finite causes: for all $e \in |E|$, the set $[e]_E = \{e' \in E \big| e' \leq_E e\}$ is finite,
conflict inheritance: if $e_1 \#_E e_2$ and $e_2 \leq_E e_2'$, then $e_1 \#_E e_2'$.

In English: any event has a finite number of causes, and if two events are in conflict, then the events they cause are also in conflict.

If $e_1, e_2 \in |E|$, we say that e_1 **immediately causes** e_2, written $e_1 \rightarrowtriangle_E e_2$, iff $e_1 <_E e_2$ and if $e_1 \leq_E e \leq_E e_2$, then $e_1 = e$ or $e_2 = e$. Events e_1 and e_2 are in **minimal conflict**, written $e_1 \rightsquigarrow_E e_2$, if $e_1 \#_E e_2$, for all $e_1' <_E e_1$ we have $\neg(e_1' \#_E e_2)$, and symmetrically. The **configurations** of E are those $x \subseteq |E|$ that are down-closed for $\leq_E$, and pairwise compatible, i.e. for all $e_1, e_2 \in x$ we have $\neg(e_1 \#_E e_2)$. We write $\mathscr{C}(E)$ for the set of finite configurations of an event structure E, and $\mathscr{C}^\infty(E)$ for possibly infinite configurations.

Proposition 2 *(Winskel, 2009) For any event structure E, $(\mathscr{C}^\infty(E), \subseteq)$, forms a dI-domain.*

Hence, in our attempt to recover concurrent games in the sense of the previous section as explicit *positions*, i.e. sets of events/moves, it is sensible to define games simply as event structures. But we also want moves to be explicitly Player or Opponent moves, so, following (Rideau & Winskel, 2011), we define games as event structures with an additional *polarity* annotation. From now on, in this paper, by *game* we will mean the following.

Definition 3 A **game** is a tuple $\langle |A|, \leq_A, \#_A, \mathrm{pol}_A \rangle$ where $\langle |A|, \leq_A, \#_A \rangle$ is an event structure, and $\mathrm{pol}_A : |A| \to \{-, +\}$ provides, for each event $a \in |A|$, a polarity indicating whether it is a Player move ($\mathrm{pol}_A(a) = +$), or an Opponent move ($\mathrm{pol}_A(a) = -$).

We additionally require that games are **race-free**, i.e. that for all $a_1, a_2 \in |A|$, if $a_1 \rightsquigarrow_A a_2$ then $\mathrm{pol}_A(a_1) = \mathrm{pol}_A(a_2)$.

Games support a number of constructions. The **empty game** $\emptyset$ has no events. If A is a game, its **dual** $A^\perp$ is A with polarities reversed. If A and B are games, their **simple parallel composition** $A \parallel B$ is the game with events the tagged disjoint union $(\{1\} \times |A|) \cup (\{2\} \times |B|)$, with causal order, conflict, and polarities simply inherited from A and B. Their **sum** $A + B$ has same components as $A \parallel B$, with additional conflicts all $(1, a) \#_{A+B} (2, b)$ for $a \in |A|, b \in |B|$. If A is a game, its **down-shift** $\downarrow A$ has events $|A| \uplus \{\updownarrow\}$ (where by $\uplus$ we mean $|A| \cup \{\updownarrow\}$ with the implicit assumption that $\updownarrow \notin |A|$), causal order that of A plus $\updownarrow \leq_{\downarrow A} a$ for all $a \in |\downarrow A|$, conflict the same as in A, and polarities those of A plus $\mathrm{pol}_{\downarrow A}(\updownarrow) = +$. The **up-shift** $\uparrow A$ is defined in the same way, with $\mathrm{pol}_{\uparrow A}(\updownarrow) = -$.

We introduce now some notations for configurations of these compound games. Configurations of $A \parallel B$ have the form $(\{1\} \times x_A) \cup (\{2\} \times x_B)$ where $x_A \in \mathscr{C}(A)$ and $x_B \in \mathscr{C}(B)$, also written $x_A \parallel x_B$. Configurations of $\downarrow A$ are either empty, or $\{\updownarrow\} \cup x_A$ with $x_A \in \mathscr{C}(A)$, also written $\updownarrow x_A$. Configurations of A and $A^\perp$ are the same: A and $A^\perp$ have the same underlying set of events. In particular, the polarity of an event is not an intrinsic property of that event, but depends of the ambient game within which that polarity is taken. Finally, all of the above applies to both finite and possibly infinite configurations.

Interpretation of MALL formulas. All units (multiplicative and additive) are interpreted by the empty game $\emptyset$. For other constructors:

$$\begin{array}{ll} [\![A \otimes B]\!] = [\![A]\!] \parallel [\![B]\!] \qquad & [\![A \oplus B]\!] = \downarrow[\![A]\!] + \downarrow[\![B]\!] \\ [\![A \parr B]\!] = [\![A]\!] \parallel [\![B]\!] \qquad & [\![A \,\&\, B]\!] = \uparrow[\![A]\!] + \uparrow[\![B]\!]. \end{array}$$

Some expected laws from linear logic obviously do not hold under this interpretation. For instance we have $[\![A \oplus 0]\!] \neq [\![A]\!]$—associativity of $\oplus$ and & also fails. This shows clearly already at this point that some additional work will have to be done in order to get full completeness. Notice that this is already true of the interpretation of the previous section, of which this is a direct refinement, in the following sense:

Proposition 3 *For any* MALL *formula* A*, we have* $[\![A]\!]_{\mathsf{Clos}} \cong \mathscr{C}^\infty([\![A]\!]_{\mathsf{Games}})$*, where* $[\![-]\!]_{\mathsf{Clos}}$ *is the interpretation of the previous section, while* $[\![-]\!]_{\mathsf{Games}}$ *is the one introduced just above.*

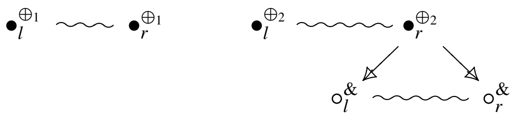
Proof For units, $\mathscr{C}^\infty(\emptyset)$ is the singleton domain, which matches the interpretation of units in Abramsky and Melliès (1999). It is direct from the definition that for the other constructors we have

$$\begin{aligned} \mathscr{C}^\infty(A \parallel B) &\cong \mathscr{C}^\infty(A) \times \mathscr{C}^\infty(B) \\ \mathscr{C}^\infty(\downarrow A + \downarrow B) = \mathscr{C}^\infty(\uparrow A + \uparrow B) &\cong (\mathscr{C}^\infty(A) + \mathscr{C}^\infty(B))_\perp \end{aligned}$$

from which the property announced follows by induction on A. $\square$

The interpretation of multiplicatives remains degenerate. The interpretation of additives $\oplus$ and & yields events of distinct polarity, a distinction that is forgotten when considering the associated domain of configurations.

As an example, we show below the interpretation as a game of the sequent $\vdash 1 \oplus_1 \perp, 1 \oplus_2 (1 \,\&\, 1)$ of Fig. 10.2, where the two occurrences of $\oplus$ have been labeled for disambiguation. In this diagram and others to come, we take the convention that we label with $\circ$ moves of negative polarity, and with $\bullet$ moves with positive polarity.



Each event corresponds to selecting one component of an additive connective. Occurrences of $\oplus$ correspond to positive/Player events, occurrences of & to negative/Opponent events, and causal dependency corresponds to the nesting of additive connectives. Multiplicative connectives, being interpreted as juxtaposition, do not contribute events. By Proposition 3, the domain of configurations of the sequent matches its interpretation in Clos. The closure operator interpreting either of the proofs of Fig. 10.2, when applied to any configuration containing $\bullet_l^{\oplus_1}$ or $\bullet_l^{\oplus_2}$, returns $\top$. Applied to any other configuration, it adds both $\bullet_r^{\oplus_1}$ and $\bullet_r^{\oplus_2}$, effectively playing them simultaneously, illustrating the resolution of the Blass problem.

We shall now give a corresponding notion of *strategy*.

10.3.2 Deterministic Concurrent Strategies

In the past two decades, besides closure-strategies, multiple alternative ways to set up concurrent strategies have appeared. Melliès and Mimram (2007) define concurrent strategies as certain sets of plays subject to stability conditions. Faggian and Piccolo (2009) define them as partial orders enriching the causality of the game. Rideau and Winskel (2011) define them as event structures labeled by the game. Castellan and Clairambault (2016) define them as *rigid families*, i.e. prefix-closed sets of partial orders (Rensink, 1992). Finally, Castellan et al. (2017) define them simply as certain sets of configurations of the game. These settings differ in expressivity, but for causally deterministic strategies such as those obtained by interpreting MALL, those are, strikingly, all equivalent.

In this paper we will exploit these last two presentations of concurrent strategies.

10.3.2.1 Strategies as Rigid Families

Whereas closure-strategies are inherently *positional*, strategies as rigid families offer a *causal* presentation of concurrent strategies. Following Castellan and Clairambault (2016), we first recall:

Definition 4 If A is a game, a **courteous augmentation** on A is a finite event structure $\mathrm{q} = \langle |\mathrm{q}|, \leq_{\mathrm{q}}, \emptyset \rangle$, with no conflict, such that $\mathscr{C}(\mathrm{q}) \subseteq \mathscr{C}(A)$, and satisfying **courtesy**: if $a_1 \rightarrowtriangle_{\mathrm{q}} a_2$, then either $\mathrm{pol}_A(a_1) = -$ and $\mathrm{pol}_A(a_2) = +$, or $a_1 \rightarrowtriangle_A a_2$.
We write $\mathrm{aug}(A)$ the set of *courteous augmentations* on A.

The terminology "courtesy" is due to Melliès and Mimram (2007). However, the concept was introduced to game semantics earlier, by Laird in his pioneering model of Idealized CSP (Laird, 2001) [also used later by Ghica and Murawski (2008)], formulated technically very differently as a closure property of strategies under certain move exchanges.

Following courtesy, a strategy may only condition the appearance of Player moves on the prior appearance of certain Opponent moves. So a strategy cannot delay an Opponent move until after a Player move if that is not already forced by the game, but neither can it force a causal dependency between its own moves. This may be understood as an *asynchrony* property: a program sending two packets on the network has no guarantee that they will arrive in the same order if that is not controlled by the protocol, so it makes no sense to impose that order in the first place. Courtesy is necessary in order for strategies to behave well with respect to the asynchronous copycat (Rideau & Winskel, 2011).

Courteous augmentations on A will be *states* for our strategies, carrying the *causal history* behind actions. There is a natural ordering on courteous augmentations. We say that $\mathrm{q} \in \mathrm{aug}(A)$ **rigidly embeds** into $\mathrm{q}' \in \mathrm{aug}(A)$, or is a **prefix** of q', if $\mathscr{C}(\mathrm{q}) \subseteq \mathscr{C}(\mathrm{q}')$, and the inclusion preserves causality: if $a_1 \leq_{\mathrm{q}} a_2$, then $a_1 \leq_{\mathrm{q}'} a_2$ as well. We write $\mathrm{q} \hookrightarrow \mathrm{q}'$. It follows that for $a_1, a_2 \in |\mathrm{q}|$, we have $a_1 \leq_{\mathrm{q}} a_2$ iff $a_1 \leq_{\mathrm{q}'} a_2$. A **courteous rigid family** on A is a non-empty subset of $\mathrm{aug}(A)$ closed under prefix. We can finally define:

Definition 5 A **strategy** $\sigma : A$ is a *courteous rigid family* on A which is additionally:

(i) *Receptive*: if $\mathrm{q} \in \sigma$ and $a^- \notin |\mathrm{q}|$ such that $|\mathrm{q}| \cup \{a^-\} \in \mathscr{C}(A)$, then there is a (necessarily unique) $\mathrm{q} \hookrightarrow \mathrm{q}'$ such that $|\mathrm{q}'| = |\mathrm{q}| \cup \{a^-\}$.
(ii) *Deterministic*: if $X \subseteq \sigma$ is a finite set of augmentations such that $\cup\{|\mathrm{q}|_- | \mathrm{q} \in X\}$ is compatible, then X has a supremum $\vee X$ in σ with respect to $\hookrightarrow$.

Here, $|\mathrm{q}|_-$ comprises the negative events of q. Without determinism, this is exactly the notion of strategy from Castellan and Clairambault (2016). For finite games, strategies are entirely characterized by their maximal augmentations. For instance, Fig. 10.3 displays the two maximal augmentations of the strategy arising as the interpretation of the two proofs of Fig. 10.2—they are augmentations of the game for $\vdash 1 \oplus \perp, 1 \oplus (1 \,\&\, 1)$ presented in the previous section.

Let us pause and disambiguate our convention for such diagrams. Formally, all moves in games arising from the interpretation of formulas are built from $\{\updownarrow\}$ along with a sequence of *tags* in $\{1, 2\}$ originating from tagged disjoint unions, and pointing to the adequate component. For instance, the moves of the left hand side diagram of Fig. 10.3, from top to bottom and left to right, are $(1, (2, \updownarrow))$, $(2, (2, \updownarrow))$ and $(2, (2, (1, \updownarrow)))$. Displaying this is cumbersome, so we adopt the convention, common in game semantics, to represent these tags implicitly by drawing events below the corresponding component of the formula.

Observe that the two augmentations both admit as prefix the two Player moves that both proofs of Fig. 10.2 are prepared to make unconditionally. They only differ with respect to the two incompatible resolutions of the & that Opponent may make. There are only two Player moves, whereas there are three in Fig. 10.2; this is because unlike in Blass games, we have interpreted units as the empty game, following Abramsky and Melliès (1999).

Let us see how to *compose* strategies. Composition relies on a (partial) composition of *courteous augmentations*. If $\mathrm{q} \in \mathrm{aug}(A^\perp \parallel B)$ and $\mathrm{q}' \in \mathrm{aug}(B^\perp \parallel C)$, they are **causally compatible** if $|\mathrm{q}| = x_A \parallel x_B$, $|\mathrm{q}'| = x_B \parallel x_C$, and q, q′ induce no causal loop, i.e.

$$(\leq_{\mathrm{q}} \parallel \leq_C) \cup (\leq_A \parallel \leq_{\mathrm{q}'}) \text{ is acyclic,}$$

where $\leq_{\mathrm{q}} \parallel \leq_C$ and $\leq_A \parallel \leq_{\mathrm{q}'}$ denote partial orders on $x_A \parallel x_B \parallel x_C$ in the obvious way.[10]

If two augmentations do induce a causal loop, that means that their interaction *deadlocks*: they impose incompatible constraints as to the order following which moves should be played. In contrast, if q and q′ are *causally compatible*, then the transitive closure

$$\leq_{\mathrm{q}' \circledast \mathrm{q}} = ((\leq_{\mathrm{q}} \parallel \leq_C) \cup (\leq_A \parallel \leq_{\mathrm{q}'}))^*$$

is a partial order, and $\mathrm{q}' \circledast \mathrm{q} = \langle x_A \parallel x_B \parallel x_C, \leq_{\mathrm{q}' \circledast \mathrm{q}} \rangle$ is the **interaction** of q and q′. Their **composition** is then $\mathrm{q}' \odot \mathrm{q} = \langle x_A \parallel x_C, \leq_{\mathrm{q}' \odot \mathrm{q}} \rangle$ where $\leq_{\mathrm{q}' \odot \mathrm{q}}$ is $\leq_{\mathrm{q}' \circledast \mathrm{q}}$ restricted to $x_A \parallel x_C$; then we have $\mathrm{q}' \odot \mathrm{q} \in \mathrm{aug}(A^\perp \parallel C)$. From this, we can compose strategies via

$$\tau \odot \sigma = \{\mathrm{q}' \odot \mathrm{q} \,|\, \mathrm{q} \in \sigma,\ \mathrm{q}' \in \tau \text{ are causally compatible}\}$$

Fig. 10.3 One strategy on $[\![\vdash 1 \oplus \perp, 1 \oplus (1 \,\&\, 1)]\!]$ for the two proofs of Fig. 10.2

[10] Here $x_A \parallel x_B \parallel x_C$ is any ternary disjoint union—$\parallel$ is only associative up to isomorphism. The precise tags used matter very little as long as the associated operations apply adequate coercion; for definiteness say $x_A \parallel x_B \parallel x_C = \{1\} \times x_A \cup \{1\frac{1}{2}\} \times x_B \cup \{2\} \times x_C$ so that $(x_A \parallel x_C) \subseteq (x_A \parallel x_B \parallel x_C)$.

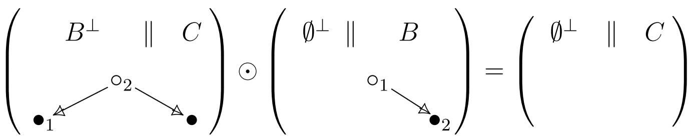


Fig. 10.4 Composition of deadlocking strategies

for $\sigma : A^\perp \parallel B$ and $\tau : B^\perp \parallel C$—it is a strategy on $A^\perp \parallel C$ as required.

It is worth emphasizing and illustrating the *causal compatibility* condition in the definition of $\mathrm{q}' \circledast \mathrm{q}$. Consider a game B with two incomparable but compatible events $\circ_1$ and $\bullet_2$, with $\mathrm{pol}(\circ_1) = -$ and $\mathrm{pol}(\bullet_2) = +$; and a game C with unique event $\bullet$, of positive polarity. Consider two strategies $\sigma : \emptyset^\perp \parallel B$ and $\tau : B^\perp \parallel C$, generated each by one maximal augmentation, represented in Fig. 10.4. Although these two maximal augmentations play the same events on B, they are not causally compatible: the order on B induced by their union is *cyclic*, with $\circ_1 \leq \circ_2 \leq \circ_1$. In fact, the only compatible augmentation of σ and τ is empty, which entails that, as in Fig. 10.4, their composition will be restricted to the empty augmentation on $A^\perp \parallel C$. Their composition *deadlocks*, as they impose incompatible constraints on the order of events. This deadlocking mechanism is a fundamental aspect present, implicitly or explicitly, in almost all games models.

With respect to this notion of composition, we have:

Proposition 4 *There is a compact closed category* Games *with games as objects, and as morphisms from A to B the strategies $\sigma : A^\perp \parallel B$.*

To prove this proposition we must define a number of other constructions on strategies, including e.g. copycat strategies and the functorial action of $\parallel$. Those may be defined directly on strategies as rigid families, as is done e.g. in Castellan and Clairambault (2016). Instead we will describe them in an alternative description of strategies as sets of configurations in the next subsection.

10.3.2.2 Strategies as Sets of Configurations

We now give a different, *positional*, presentation of the *same* deterministic concurrent strategies. We are aware that it is a lot to ask to the reader to digest not only *one*, but *two* definitions for a games model. But it is a distinctive feature of deterministic concurrent strategies that they may be described in these different ways. Each representation has distinct advantages: the *causal* description above relates to the inductive structure of terms. It highlights the causal flavour of traditional game semantic notions such as *P-views* (Hyland & Ong, 2000), and accordingly supports a simple notion of innocence (see Sect. 10.4.3.1). In contrast, the *positional* presentation of strategies as sets of configurations that we are about to present emphasises their proximity with relational-like models. This connection will be extensively used in the remainder of the paper.

If X is a set of sets of events of a game A, $x, y \in X$, and $a \in |A|$, we write $x \overset{a}{\longrightarrow\!\!\subset}$ if $y = x \cup \{a\}$, or simply $x \longrightarrow\!\!\subset y$. In that case we say that x *extends to* y *within* X. We write $X \uparrow^-$ if X is *negatively compatible*, meaning that $\{a \in \cup X \mid \mathrm{pol}(a) = -\}$ is compatible. Finally, if $\sigma : A$ is a strategy on A, we write $\mathscr{C}(\sigma) = \{\mathscr{C}(\mathrm{q}) \mid \mathrm{q} \in \sigma\}$ for its **configurations**.

Proposition 5 *For any game A, there is a* 1*-to-*1 *correspondence between strategies* $\sigma : A$ *and sets of finite configurations* $\mathcal{S} \subseteq \mathscr{C}(A)$ *satisfying:*

(i) $for\ any\ X \subseteq \mathcal{S}, if\ X\uparrow^-\ then\ \cup X \in \mathcal{S}\ and\ (for\ X\ non-empty) \cap X \in \mathcal{S},$
(ii) *if* $a_1, a_2 \in x \in \mathcal{S}$, *there exists* $y \subseteq x$ *such that* $y \in \mathcal{S}$ *and* $a_1 \in y \Leftrightarrow \neg(a_2 \in y)$,
(iii) *if* $x \overset{a_1}{\longrightarrow\!\!\subset} \overset{a_2}{\longrightarrow\!\!\subset}$ *in* $\mathcal{S}$ *with* $x \overset{a_2}{\longrightarrow\!\!\subset}$ *in* $\mathscr{C}(A)$ *but not in* $\mathcal{S}, \mathrm{pol}(a_1) = -$ *and* $\mathrm{pol}(a_2) = +$,
(iv) $if x \in \mathcal{S}\ and\ x \overset{a^-}{\longrightarrow\!\!\subset} in \mathscr{C}(A),\ then\ x \overset{a^-}{\longrightarrow\!\!\subset} in\ \mathcal{S}.$

Proof Let $\sigma : A$ be a strategy. First, conditions (i)–(iv) may be directly verified on $\mathscr{C}(\sigma)$.

Reciprocally, for each $x \in \mathcal{S}$ we construct a partial order $\mathrm{q}_x \in \mathrm{aug}(A)$ as $(x, \leq_x)$ where $a_1 \leq_x a_2$ iff for all $y \subseteq x$ in $\mathcal{S}$, if $a_2 \in y$ then $a_1 \in y$ also. We refer the reader to Winskel (1986) for properties of this partial order, which is used to link *prime event structures* and *stable families*. In particular, we have $\mathscr{C}(\mathrm{q}_x) \subseteq \mathcal{S}$, and if $x \subseteq y$, then $\mathrm{q}_x \hookrightarrow \mathrm{q}_y$. Moreover, by (iii) it follows that q_x is courteous, and by (iv) it follows that $\{\mathrm{q}_x \mid x \in \mathcal{S}\}$ is receptive.

It is direct to verify that these constructions are inverses of each other. □

In particular, strategies are determined by their configurations: if $\sigma, \tau : A$ are such that $\mathscr{C}(\sigma) = \mathscr{C}(\tau)$, then $\sigma = \tau$. But this also lets us *define* deterministic strategies simply via sets of configurations. Indeed, for instance, we may define the *copycat strategy* via

$$\mathscr{C}(\text{\ae}_A) = \{x_A \parallel y_A \in \mathscr{C}(A^\perp \parallel A) \mid y_A \supseteq^- x_A \cap y_A \subseteq^+ x_A\}$$

where $\subseteq^+$, $\subseteq^-$ mean inclusion where the elements added have the polarity indicated, and polarity is always taken to be in A (not $A^\perp$). In other words, those are identity pairs $x_A \parallel x_A \in \mathscr{C}(A^\perp \parallel A)$ closed under receptivity on both sides. Likewise, it is convenient to define the functorial action of $\parallel$ on strategies via its action on sets of configurations, as

$$\mathscr{C}(\sigma_1 \parallel \sigma_2) = \{(x_{A_1} \parallel x_{A_2}) \parallel (x_{B_1} \parallel x_{B_2}) \mid x_{A_1} \parallel x_{B_1} \in \mathscr{C}(\sigma_1) \wedge x_{A_2} \parallel x_{B_2} \in \mathscr{C}(\sigma_2)\}$$

for $\sigma_1 : A_1^\perp \parallel B_1$ and $\sigma_2 : A_2^\perp \parallel B_2$. Other structural components of the compact closed structure of Games may be defined similarly. This definition of the tensor in Games is strikingly similar to the corresponding structure in the *relational model*. Likewise, composition of strategies viewed as sets of configurations is fairly close to relational composition:

Proposition 6 *Let $\sigma : A^\perp \parallel B$ and $\tau : B^\perp \parallel C$ be two strategies. Then, $\mathscr{C}(\tau \odot \sigma)$ comprises exactly the pairs $x_A \parallel x_C \in \mathscr{C}(A^\perp \parallel C)$ such that there exists $x_A \parallel x_B \parallel x_C$ such that $x_A \parallel x_B \in \mathscr{C}(\sigma)$ and $x_B \parallel x_C \in \mathscr{C}(\tau)$ which is additionally* reachable*, i.e. there is*

$$x_A^0 \parallel x_B^0 \parallel x_C^0 \;\multimap\!\!\subset\; x_A^1 \parallel x_B^1 \parallel x_C^1 \;\multimap\!\!\subset\; \dots \;\multimap\!\!\subset\; x_A^n \parallel x_B^n \parallel x_C^n$$

such that x_A^0, x_B^0, x_C^0 are empty, $x_A^n = x_A$, $x_B^n = x_B$ and $x_C^n = x_C$, and for all $0 \leq i \leq n$, we have $x_A^i \parallel x_B^i \in \mathscr{C}(\sigma)$ and $x_B^i \parallel x_C^i \in \mathscr{C}(\tau)$.

Proof If $x \in \mathscr{C}(\tau \odot \sigma)$, then there are $\mathrm{q} \in \sigma$, $\mathrm{q}' \in \tau$ such that $x = |\mathrm{q}' \odot \mathrm{q}|$. By definition, $\mathrm{q}' \odot \mathrm{q}$ is the restriction on A, C of $\mathrm{q}' \circledast \mathrm{q}$ with $|\mathrm{q}' \circledast \mathrm{q}| = x_A \parallel x_B \parallel x_C$. But then, any linearization of the partial order $\leq_{\mathrm{q}' \circledast \mathrm{q}}$ on $x_A \parallel x_B \parallel x_C$ yields a chain as required.

Reciprocally, if $x_A \parallel x_B \parallel x_C$ is such that $x_A \parallel x_B \in \mathscr{C}(\sigma)$, $x_B \parallel x_C \in \mathscr{C}(\tau)$ with a chain as above, then that chain is a linearization of the transitive closure of

$$(\leq_{\mathrm{q}_{x_A \parallel x_B}} \parallel \leq_C) \cup (\leq_A \parallel \leq_{\mathrm{q}_{x_B \parallel x_C}})$$

which is therefore acyclic, making $\mathrm{q}_{x_A \parallel x_B} \in \sigma$ and $\mathrm{q}_{x_B \parallel x_C} \in \tau$ causally compatible. We then have $|\mathrm{q}_{x_B \parallel x_C} \odot \mathrm{q}_{x_A \parallel x_B}| = x_A \parallel x_C$ by construction. $\square$

The reachability condition corresponds to the *causal compatibility* requirement in the definition of interaction of augmentations. Coming back to Fig. 10.4, we have $\mathscr{C}(\sigma) = \{\emptyset, \{\circ_1\}, \{\circ_1, \bullet_2\}\}$ and $\mathscr{C}(\tau) = \{\emptyset, \{\circ_2\}, \{\circ_2, \bullet_1\}, \{\circ_2, \bullet\}, \{\circ_2, \bullet_1, \bullet\}\}$. Then, the set

$$\emptyset \parallel \{\circ_1, \circ_2\} \parallel \{\circ\} \in \mathscr{C}(\emptyset \parallel B \parallel C)$$

is a candidate to be a configuration of the interaction as $\emptyset \parallel \{\circ_1, \bullet_2\} \in \mathscr{C}(\sigma)$ and $\{\bullet_1, \circ_2\} \parallel \{\bullet\} \in \mathscr{C}(\tau)$. However, it is rejected by the *reachability* condition, although it would be present in a purely relational composition of the strategies.

This presentation of composition highlights the proximity of deterministic concurrent strategies with relational semantics, but also the fundamental difference between the two models: namely, *reachability*, and the ability of the composition of strategies to deadlock.

10.3.3 Strategies and Closure Operators

Now, we link deterministic concurrent strategies and strategies as closure operators.

10.3.3.1 From Strategies to Closure Operators

For $\sigma : A$, the developments in the previous section yield a set of finite configurations $\mathscr{C}(\sigma)$. From this, we may define the *possibly infinite configurations* $\mathscr{C}^\infty(\sigma)$ of σ as the unions of directed sets of finite configurations. Possibly infinite configurations are partially ordered by inclusion, and we write $x \subseteq^+ y$ or $x \subseteq^- y$ as for finite configurations.

In defining a closure operator, we will use that any compatible set of negative events enables a unique $+$-maximal possibly infinite configuration in $\mathscr{C}^\infty(\sigma)$. In the sequel, for $x \in \mathscr{C}^\infty(A)$ we write x_- for its set of negative events and x_+ for its set of positive events.

Lemma 1 *Let $\sigma : A$ be a strategy, and $x \in \mathscr{C}^\infty(A)$. Then, defining the set*

$$\overline{x}^\sigma = \cup\{y \in \mathscr{C}(\sigma) \big| y_- \subseteq x_-\},$$

we have $\overline{x}^\sigma \in \mathscr{C}^\infty(\sigma)$.

Proof All finite subsets of $Y = \{y \in \mathscr{C}(\sigma) \big| y_- \subseteq x_-\}$ are negatively compatible, hence by (1) of Proposition 5, have a union in $\mathscr{C}(\sigma)$. Therefore, $\cup Y \in \mathscr{C}^\infty(\sigma)$; we set $\overline{x}^\sigma = \cup Y$. $\square$

If $x \in \mathscr{C}^\infty(A)$, $\overline{x}^\sigma$ is obtained by playing all moves that σ is prepared to play with the *negative moves* already present in x. It *is not necessarily the case* that $x \subseteq \overline{x}^\sigma$; indeed x might contain positive moves that σ is not prepared to play with the negative moves in x. In fact, it is not necessarily the case that $x \cup \overline{x}^\sigma \in \mathscr{C}^\infty(A)$.

Proposition 7 *Let $\sigma : A$ be a strategy. Then, the function*

$$\begin{aligned} \mathsf{C}(\sigma) : \mathscr{C}^\infty(A) &\to \mathscr{C}^\infty(A)^\top \\ x &\mapsto \begin{cases} x \cup \overline{x}^\sigma & \textit{if } x \cup \overline{x}^\sigma \textit{ is compatible} \\ \top & \textit{otherwise,} \end{cases} \end{aligned}$$

extended to $\mathsf{C}(\sigma) : \mathscr{C}^\infty(A)^\top \to \mathscr{C}^\infty(A)^\top$ with $\mathsf{C}(\sigma)(\top) = \top$, is a closure-strategy.

Proof By construction, $\mathsf{C}(\sigma) : \mathscr{C}^\infty(A)^\top \to \mathscr{C}^\infty(A)^\top$ is extensive, monotone, and idempotent. Continuity is longer but essentially direct, exploiting the axiom of finite causes for A. Stability is simply stability of $\overline{(-)}^\sigma$, which is obvious from the definition. $\square$

For a strategy $\sigma : A$, the closure operator $\mathsf{C}(\sigma)$ will take any $x \in \mathscr{C}^\infty(A)$ (to $\top$ or) to a "closed" configuration $\mathsf{C}(\sigma)(x)$ obtained by adding to x all the positive moves whose causal dependencies in σ appear in x. This is done regardless of the fact that there may be moves in x that σ will never play, but if the effect of adding these events yields an incompatible set, then the result is $\top$ instead. This differs from the two other transformations from concurrent strategies to closure-strategies appearing in the literature that we are aware of: in writing this paper we observed that they both suffer from some pathologies (for instance they both fail continuity), see Appendix 1.

10.3.3.2 On the Functoriality of the Transformation

We now investigate whether the transformation from concurrent strategies to closure-strategies is functorial. We first make a key observation on closure-strategies: their composition may be presented relationally. Although this fact does not appear in Abramsky and Melliès (1999), it was known by Melliès in the 00s when working on *asynchronous games*. To our knowledge its only appearance in a published source is in Mimram's Ph.D. thesis (Mimram, 2008).

If $\sigma : D$ is a closure-strategy, write $\mathsf{fix}(\sigma)$ for its set of **fixpoints**, i.e. those $x \in D$ such that $\sigma(x) = x$. Closure operators on complete lattices are determined by their fixpoints; in particular from $X = \mathsf{fix}(\sigma)$ one can recover σ as $\sigma(x) = \wedge\{y \in X \big| x \leq_D y\}$ for $x \in D$. Perhaps surprisingly in view of the interactive definition of composition, we have:

Proposition 8 *Let $\sigma : D_1 \twoheadrightarrow D_2$ and $\tau : D_2 \twoheadrightarrow D_3$ be closure-strategies. Then:*

$$\mathsf{fix}(\tau \odot \sigma) = \mathsf{fix}(\tau) \circ \mathsf{fix}(\sigma)$$

where $\circ$ is relational composition.

Proof $\subseteq$. If $(x, z) \in \mathsf{fix}(\tau \odot \sigma)$ then by definition there is $y \in D_2$ such that $y = \pi_2(\sigma(x, y))$ and $y = \pi_1(\tau(y, z))$, and with $x = \pi_1(\sigma(x, y))$ and $z = \pi_2(\tau(y, z))$. In particular, $(x, y) \in \mathsf{fix}(\sigma)$ and $(y, z) \in \mathsf{fix}(\tau)$, so $(x, z) \in \mathsf{fix}(\tau) \circ \mathsf{fix}(\sigma)$.

$\supseteq$. If $(x, y) \in \mathsf{fix}(\sigma)$ and $(y, z) \in \mathsf{fix}(\tau)$, then compute

$$y' = \langle \pi_2 \circ \sigma(x, -) \big| \pi_1 \circ \tau(-, z) \rangle = \bigvee_{n \in \mathbb{N}} ((\pi_2 \circ \sigma(x, -)) \circ (\pi_1 \circ \tau(-, z)))^n(\bot) \in D_2.$$

Since $\sigma(x, y) = (x, y)$ and $\tau(y, z) = (y, z)$ with both monotone, it follows by induction on n that for all $n \in \mathbb{N}$ the n-th approximation $y_n \in D_2$ is below y; hence $y' \leq y$. But then we have $\pi_1(\sigma(x, y')) = x$ since σ is monotone and increasing, and likewise $\pi_2(\tau(y', z)) = z$; therefore $(\tau \odot \sigma)(x, z) = (x, z)$ as required. □

Hence composition of closure-strategies can be presented purely relationally—one can observe notably the absence of a *reachability* condition as in Proposition 6. This seems in stark contrast with the original interactive flavour of composition of closure-strategies. In light of our previous discussion on reachability, this gives the impression that composition of closure-strategies fails to take *deadlocks* into account and eliminate causal loops.

To put some light on this issue, it is informative to look at the deadlocking composition of Fig. 10.4 through the lens of closure-strategies. We first fix some notations. If $\sigma : A^\perp \parallel B$ is a strategy from A to B, $\mathsf{C}(\sigma)$ is a closure-strategy on $\mathscr{C}^\infty(A^\perp \parallel B) \cong \mathscr{C}^\infty(A) \times \mathscr{C}^\infty(B)$—we still write $\mathsf{C}(\sigma)$ for the corresponding closure-strategy on the latter domain.

Considering the closure-strategies coming from the strategies of Fig. 10.4, we have $\mathsf{C}(\tau)(\emptyset \parallel \{\bullet\}) = (\emptyset \parallel \{\bullet\})$ so $\emptyset \parallel \{\bullet\}$ is a fixpoint of $\mathsf{C}(\tau)$ even though τ will never play $\bullet$ on its own. Likewise, $\emptyset \parallel \emptyset \in \mathsf{fix}(\mathsf{C}(\sigma))$; hence $\emptyset \parallel \{\bullet\} \in \mathsf{fix}(\mathsf{C}(\tau) \odot \mathsf{C}(\sigma))$. Although this seems to vindicate the view that deadlocks are not satisfactorily taken into account by composition, this is misleading. Instead, we argue that it is inaccurate to think of fixpoints as stopping states of a strategy: not because they are not stopping, but because they might not be *states*, in the sense that they may not be reachable through a normal interactive computation. Indeed, in this example we also have $\emptyset \parallel \emptyset \in \mathsf{fix}(\mathsf{C}(\tau) \odot \mathsf{C}(\sigma))$—in particular applying $\mathsf{C}(\tau) \odot \mathsf{C}(\sigma)$ on $\emptyset \parallel \emptyset$ does *not* add $\bullet$; so the deadlock *is* accurately represented. The configuration $\emptyset \parallel \{\bullet\}$ is a fixpoint for $\mathsf{C}(\tau) \odot \mathsf{C}(\sigma)$, but not a *reachable* one.

In fact, composition of closure-strategies *does* agree with composition of strategies:

Proposition 9 *For any two strategies $\sigma : A^\perp \parallel B$ and $\tau : B^\perp \parallel C$, we have*

$$\mathsf{C}(\tau \odot \sigma) = \mathsf{C}(\tau) \odot \mathsf{C}(\sigma).$$

Proof To save space we only detail the right-to-left inclusion, which is the most surprising in light of the relational nature of composition of closure-strategies. Take $(x, z) \in \mathsf{fix}(\mathsf{C}(\tau) \odot \mathsf{C}(\sigma))$. By Proposition 8, there are $(x, y) \in \mathsf{fix}(\mathsf{C}(\sigma))$ and $(y, z) \in \mathsf{fix}(\mathsf{C}(\tau))$. Take $x' \parallel z' \in \mathscr{C}(\tau \odot \sigma)$ such that $(x' \parallel z')_- \subseteq x \parallel z$. There is some $y' \in \mathscr{C}(B)$ such that $x' \parallel y' \in \mathscr{C}(\sigma)$ and $y' \parallel z' \in \mathscr{C}(\tau)$, and which is reachable in the sense that there is a covering chain

$$x'_0 \parallel y'_0 \parallel z'_0 \longrightarrow\!\!\!\!\subset \ldots \longrightarrow\!\!\!\!\subset x'_n \parallel y'_n \parallel z'_n$$

such that $x'_0 \parallel y'_0 \parallel z'_0 = \emptyset$, $x'_n \parallel y'_n \parallel z'_n = x' \parallel y' \parallel z'$, and for all $0 \leq i \leq n$ we have $x'_i \parallel y'_i \in \mathscr{C}(\sigma)$ and $y'_i \parallel z'_i \in \mathscr{C}(\tau)$. By induction on i, using $x \parallel y \in \mathsf{fix}(\mathsf{C}(\sigma))$ and $y \parallel z \in \mathsf{fix}(\mathsf{C}(\tau))$, we have $y'_i \subseteq y$, hence $y' \subseteq y$. Hence, $(x' \parallel y')_- \subseteq x \parallel y$, so $x' \parallel y' \subseteq x \parallel y$ since $x \parallel y \in \mathsf{fix}(\mathsf{C}(\sigma))$. Likewise, $y' \parallel z' \subseteq y \parallel z$. Therefore, $x' \parallel z' \subseteq x \parallel z$ and $(x, z) \in \mathsf{fix}(\mathsf{C}(\tau \odot \sigma))$ as required. $\square$

However, for a game A it almost never holds that $\mathsf{C}(\text{æ}_A) = \text{æ}_{\mathscr{C}^\infty(A)}$. For instance, consider the game A having only one positive move $\bullet$. Then $A^\perp \parallel A$ has two moves, one negative move written $\circ$ and one positive still written $\bullet$. Then, $\text{æ}_{\mathscr{C}^\infty(A)}(\emptyset, \{\bullet\}) = (\{\circ\}, \{\bullet\})$, whereas $\mathsf{C}(\text{æ}_A)(\emptyset, \{\bullet\}) = (\emptyset, \{\bullet\})$. The identity in Clos adds missing negative dependencies (as it must, because it must be defined on arbitrary domains, with therefore no access to polarity information). In contrast, applying closure operators imported from strategies only adds positive moves. For this reason, it is tempting, instead of the transformation from strategies to closure-strategies presented above, to adopt one adding the missing negative dependencies to reachable positive events, as does the identity in Clos. But as presented in Appendix 1.1 this leads to issues, notably non-stability and non-continuity of the corresponding closure operators—besides, then, $\mathsf{C}(-)$ would not give a functor either: identities would be preserved, but not composition.

Instead, we moderate this mismatch on identities by remarking that although $\mathsf{C}(\mathbb{c}_A)$ and $\mathbb{c}_{\mathscr{C}^\infty(A)}$ do not coincide, *they have the same reachable configurations*. The set of **reachable** fixpoints of a closure-strategy $\sigma : \mathscr{C}^\infty(A)$ is the smallest subset of $\mathscr{C}^\infty(A)$ containing $\sigma(\emptyset)$, and such that if $x \in \mathsf{fix}(\sigma)$ is reachable and $x \subseteq^- y$, then $\sigma(y)$ is reachable. Write $\mathsf{reach}(\sigma)$ for the set of reachable fixpoints of σ. Then, $\sigma, \sigma' : \mathscr{C}^\infty(A)$ are **reachable-equivalent** if $\mathsf{reach}(\sigma) = \mathsf{reach}(\sigma')$, written $\sigma \approx \sigma'$. Then, it is direct to prove that $\mathsf{C}(\mathbb{c}_A) \approx \mathbb{c}_{\mathscr{C}^\infty(A)}$.

Computing reachable fixpoints of compositions only uses reachable fixpoints:

Lemma 2 *Let $\sigma : \mathscr{C}^\infty(A^\perp \parallel B)$ and $\tau : \mathscr{C}^\infty(B^\perp \parallel C)$ be two closure-strategies. Using silently the order-isomorphism $\mathscr{C}^\infty(A^\perp \parallel B) \cong \mathscr{C}^\infty(A) \times \mathscr{C}^\infty(B)$, we regard them as morphisms from A to B and from B to C respectively.*

Then, for any $x_A \parallel x_C \in \mathsf{fix}(\tau \odot \sigma)$, we have $x_A \parallel x_C \in \mathsf{reach}(\tau \odot \sigma)$ iff there is a chain

$$x_A^0 \parallel x_B^0 \parallel x_C^0 \subseteq \dots \subseteq x_A^n \parallel x_B^n \parallel x_C^n$$

where $x_A^0 \parallel x_B^0 \parallel x_C^0 = \emptyset$, $x_A^n \parallel x_C^n = x_A \parallel x_C$, and where each $x_A^i \parallel x_B^i \parallel x_C^i \longrightarrow\!\!\!\subset x_A^{i+1} \parallel x_B^{i+1} \parallel x_C^{i+1}$ is obtained by (1) $x_A^i \parallel x_B^i \in \mathsf{reach}(\sigma)$, $x_B^i \parallel x_C^i \in \mathsf{reach}(\tau)$, $x_B^i = x_B^{i+1}$ and $x_A^i \parallel x_C^i \subseteq^- x_A^{i+1} \parallel x_C^{i+1}$; or (2) $x_C^i = x_C^{i+1}$ and $x_A^{i+1} \parallel x_B^{i+1} = \sigma(x_A^i \parallel x_B^i)$; or (3) $x_A^i = x_A^{i+1}$ and $x_B^{i+1} \parallel x_C^{i+1} = \tau(x_B^i \parallel x_C^i)$. It follows that $x_A \parallel x_B \in \mathsf{reach}(\sigma)$ and $x_B \parallel x_C \in \mathsf{reach}(\tau)$.

Proof Direct verification. □

A direct consequence is that $\mathsf{reach}(\tau \odot \sigma) \subseteq \mathsf{reach}(\tau) \circ \mathsf{reach}(\sigma)$ where $\circ$ is relational composition. But unlike the case for all fixpoints in Proposition 8, the converse does not hold for *reachable* fixpoints. Focusing on reachable fixpoints, we recover the reachability side-condition of Proposition 6 and composition is no longer purely relational.

Finally, we deduce:

Proposition 10 *Consider $\sigma, \sigma' : \mathscr{C}^\infty(A^\perp \parallel B)$ and $\tau, \tau' : \mathscr{C}^\infty(B^\perp \parallel C)$ satisfying $\sigma \approx \sigma'$ and $\tau \approx \tau'$, regarded as morphisms from A to B and from B to C in Clos. Then, we have*

$$\tau \odot \sigma \approx \tau' \odot \sigma'.$$

Proof Straightforward from Lemma 2 as reachable fixpoints of $\tau \odot \sigma$ and $\tau' \odot \sigma'$ are reduced to chains formed from reachable fixpoints of σ/σ' and τ/τ', which are the same. □

From all the developments above, we may conclude:

Theorem 1 *There is a faithful, strong compact closed functor*

$$\mathsf{C}(-) : \mathsf{Games} \to \mathsf{Clos}/{\approx}$$

where $\mathsf{Clos}/{\approx}$ *has as morphisms closure operators up to* $\approx$*.*

It might seem that the quotient $\approx$ may create a mismatch with respect to the interpretation of MALL in Clos, but it is in fact much milder than the extensional collapse used in Abramsky and Melliès (1999) to obtain full completeness—we will introduce it in the next subsection.

10.3.4 Extensional Collapse

The interpretation of MALL formulas as games was given in Sect. 10.3.1. A context $\Gamma = A_1, \ldots, A_n$ is interpreted as a tensor $[\![\Gamma]\!] = \otimes_{1 \leq i \leq n} [\![A_i]\!]$ and a proof of $\vdash \Gamma$ as a strategy on $[\![\Gamma]\!]$. The interpretation of MLL rules proceeds as is standard in a compact closed category. A proof starting with an introduction rule for $\oplus$ will have the corresponding positive move as minimal, otherwise playing as the sub-proof—for $\sigma : A$, we write $\mathrm{in}_l(\sigma) : A \oplus B$. A proof starting with an introduction rule for & will delay positive moves until it receives one of the Opponent moves coming from the &; it then proceeds as the corresponding sub-proof—for $\sigma_A : A$ and $\sigma_B : B$, we write $\langle \sigma_A, \sigma_B \rangle : A \,\&\, B$. We have not been able to formally verify that this interpretation is compatible [through $\mathsf{C}(-)$] with that of (Abramsky & Melliès, 1999) up to $\approx$, as the details of the interpretation do not appear in Abramsky and Melliès (1999). Nevertheless we believe this to be the case, and our interpretation seems compatible with informal descriptions in Abramsky and Melliès (1999). Thus it is informative to look at the interpretation of some proofs in our model as representations of their interpretation with closure-strategies.

In Fig. 10.5, we display two proofs, along with one typical maximal augmentation in their respective interpretations in Games. In the proofs, we omit the $\vdash$ symbol and we color the units to track the specific rules used. For each proof we only display one maximal augmentation, corresponding to the one complete branch of the proof where the left component of & is always selected. The two proofs are convertible using standard commuting conversions. Despite this, they are distinguished by the semantics in Games and Clos. The maximal augmentations pictured show the phenomenon: if Opponent always selects the left component of & then the two proofs perform the same actions, but *not in the same order*. This phenomenon was of course noticed in Abramsky and Melliès (1999), where the authors say:

> To motivate the passage to the extensional category, note that Clos only has weak products and coproducts. Indeed, the lifted sum which we used to model the additives is non-associative, and we need to quotient out the behaviour at the partial elements in order to obtain the required structure.

Indeed, they quotient the model using partial equivalence relations (*pers*, satisfying transitivity and symmetry but not reflexivity), a standard methodology to construct models of linear logic (Hyland & Schalk, 2003). Concretely, for every formula A they build (by induction on A) a per $\sim_A$ on strategies on A. It has two effects: firstly, it identifies strategies with the same extensional behaviour, even though they might be intensionally distinct. With respect to the interpretation of MALL above, this quotients out the intensional behaviour caused by the lifts in the interpretation of additive connectives. In particular, the two proofs of Fig. 10.5 are identified. Secondly, it cuts out those strategies that can taste intensional information: the new model restricts to strategies that are self-equivalent, which for morphisms from A to B essentially amounts to sending $\sim_A$-equivalent strategies to $\sim_B$-equivalent strategies.

Sometimes, a miracle occurs after this "cutting out" process: only definable elements remain and the new model is fully complete—and indeed Abramsky and Melliès show that this is the case for closure-strategies. This is by all means not a general fact: for instance, the same construction applied to the relational model does not yield full completeness.[11] Performing this construction on an intensional canvas such as game semantics helps, in that morphisms in the new model are equivalence classes of concrete strategies. Representatives have intensional behaviour that can be tracked down to reconstruct a proof.

However, something remains puzzling. There seems to be a tension between definability, which is facilitated by more *intensionality*; and validating all required equations, which is facilitated by more *extensionality*. The solution of Abramsky and Melliès (1999) is to first build an interpretation failing some equations but with a tractable intensional description of proofs, and then quotient and cut it down by extensional collapse. But then, why did we need concurrent games to do that? After all, there are plenty of simpler intensional not-quite-models around, the obvious one

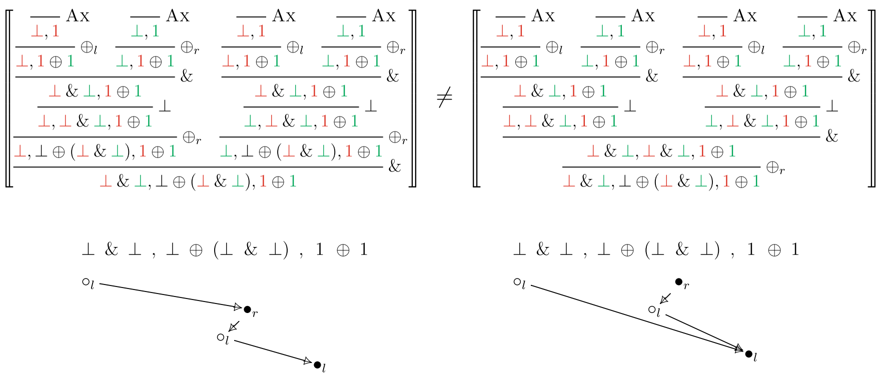


Fig. 10.5 Two distinct strategies for two equivalent proofs

[11] A similar construction applied on *hypercoherences*, which build on the relational model, does yield full completeness (Blute et al., 2005)—note that there are links between hypercoherences and game semantics (Melliès, 2005).

$$
\begin{array}{cccccc}
(\mathbb{B} \multimap & \mathbb{B} \multimap & \mathbb{B}) \multimap & \mathbb{B} \multimap & \mathbb{B} \multimap & \mathbb{B} \\
 & & & & & q \\
 & & q & & & \\
q & & & & & \\
\mathtt{tt} & & & & & \\
 & q & & & & \\
 & \mathtt{tt} & & & & \\
 & & b_1 & & & \\
 & & & q & & \\
 & & & b_2 & & \\
 & & & & q & \\
 & & & & b_3 & \\
 & & & & & f(b_1, b_2, b_3)
\end{array}
\qquad
\begin{array}{cccccc}
(\mathbb{B} \multimap & \mathbb{B} \multimap & \mathbb{B}) \multimap & \mathbb{B} \multimap & \mathbb{B} \multimap & \mathbb{B} \\
 & & & & & q \\
 & & q & & & \\
 & q & & & & \\
 & \mathtt{tt} & & & & \\
q & & & & & \\
\mathtt{tt} & & & & & \\
 & & b_1 & & & \\
 & & & & q & \\
 & & & & b_3 & \\
 & & & q & & \\
 & & & b_2 & & \\
 & & & & & f(b_1, b_2, b_3)
\end{array}
$$

Fig. 10.6 Example of extensionally correct yet undefinable behaviour

being Blass games. Composition is not associative in Blass games, which was the original motivation for concurrent games. But that cannot be the end of the story: non-associativity means that the two equivalent proofs of Fig. 10.2 are interpreted by two strategies "doing the same actions but not in the same order". Moving to Clos and Games solved the Blass problem and made those two equivalent, but with the cost that the two proofs of Fig. 10.5 now give rise to two strategies "doing the same actions but not in the same order"—but those two were interpreted with the same strategy in Blass games! So have we just moved the problem around?

As it turns out, the conceptual advance offered by concurrent games is much greater than merely solving the Blass problem. While the extensional collapse may be applied to a sequential games substrate (such as Blass games) in order to get a model of MALL, this will usually land us far from full completeness. Figure 10.6 illustrates this in a basic game semantics setting [e.g. simple games (Hyland, 1997)], though we do not see why the phenomenon would not occur as well in Blass games. The strategy pictured with its two maximal plays acts like $\lambda gxy.\, f(g\ \mathtt{tt}\ \mathtt{tt}, x, y)$, except it calls x and y in the same order than the argument g used to call its arguments. This strategy is undefinable (it is not *innocent*), yet it survives the extensional collapse. Such a behaviour cannot be expressed with closure-strategies, because it is not *positional*: after the first 7 moves, the strategy acts differently in the two plays, although the set of moves that have been played is the same. So it appears that the ability of concurrent games to give a fully complete model to MALL is not due to concurrency per se.[12] Instead, and as investigated in depth by Melliès in *asynchronous games*, the key conceptual advance of concurrent games is that they are *positional/causal*.

10.4 Full Completeness via MALLP

We have seen that Abramsky and Melliès' full completeness result rests on two ingredients: (1) an unsound intensional model, whose dynamics can be tracked down to

[12] Indeed the Blass problem could easily be solved by a non-polarized version of Ghica and Murawski's concurrent games (Ghica & Murawski, 2008), but there is no reason why it would not suffer from phenomena as in Fig. 10.6.

guide definability; and (2) a quotient which restores the necessary equations between proofs. The definability process of Abramsky and Melliès (1999) is challenging, as strategies are far from sequential. Rather than a sequent proof, the argument reconstructs from the action of the strategy an MALL proof structure in the sense of Girard (2017), which is proved correct.

In this paper we do not review this argument. Instead we adopt a different route, following later work by Melliès (2005). Since it seems a quotient is required anyway, why not add much more intensional information, with enough dynamic content as to make definability straightforward, yielding directly a sequent proof by induction? This will mechanically break more of the expected MALL laws, but those will be reinstated by quotient anyway. Likewise, this added sequentiality should not prevent us from quotienting, provided the model is phrased in a *positional* setting.

In the remainder of this paper we build a fully complete model of MALL following that route. Rather than directly giving a sequential interpretation of MALL, we first interpret a *polarized* variant. It is obtained by annotating formulas with new constructors marking additional observable computation steps, and constraining proofs making their dynamic sequential (MALL formulas will later on be interpreted by first *polarizing* them, then interpreting the obtained formula). We will first build a fully complete model of polarized MALL, then perform the quotient and deduce full completeness for MALL.

The first fully complete games model for a polarized version of MALL was by Girard, in the framework of Ludics (Girard, 2001). In his thesis, Laurent introduced a more symmetric presentation of Girard's system, called MALLP (Laurent, 2002), which we adopt in this paper. The developments in this section and the next are strongly inspired from Melliès' in *asynchronous games*, rephrased in the game semantics language presented in Sect. 10.3.

10.4.1 Polarized Multiplicative Additive Linear Logic

Linear Logic is inherently *non-polarized*: $(1 \oplus 1) \otimes (1 \,\&\, 1)$ is interpreted (following the previous section) by the game $\bullet_1 \sim \bullet_2 \;\; \circ_1 \sim \circ_2$ with both players having available events, reflecting the fact that the formula does not carry explicit information as to which side of the tensor is to be resolved first, if any. *Polarized Multiplicative Additive Linear Logic (*MALLP*)*, introduced by Laurent (2002), starts with the same connectives, but restricts formulas so as to follow a strict *polarity* discipline ensuring among other things that execution is sequential. New unary connectives, *shifts*, are used to transport between polarities.

The **formulas** of MALLP are as follows

$$P, Q ::= 0 \,\big|\, 1 \,\big|\, P \otimes Q \,\big|\, P \oplus Q \,\big|\, {\downarrow} M$$
$$M, N ::= \top \big| \bot \big| M \parr N \big| M \,\&\, N \big| {\uparrow} P$$

$$\frac{\vdash \Gamma, P \qquad \vdash \Delta, Q}{\vdash \Gamma, \Delta, P \otimes Q}\otimes \qquad \frac{\vdash \Gamma, M, N, [P]}{\vdash \Gamma, M \parr N, [P]}\parr \qquad \frac{}{\vdash 1}1 \qquad \frac{\vdash \Gamma, [P]}{\vdash \Gamma, \bot, [P]}\bot \qquad \frac{\vdash \Gamma, M}{\vdash \Gamma, {\downarrow} M}\downarrow$$

$$\frac{\vdash \Gamma, P}{\vdash \Gamma, {\uparrow} P}\uparrow \qquad \frac{\vdash \Gamma, P}{\vdash \Gamma, P \oplus Q}\oplus_l \qquad \frac{\vdash \Gamma, Q}{\vdash \Gamma, P \oplus Q}\oplus_r \qquad \frac{\vdash \Gamma, M, [P] \qquad \vdash \Gamma, N, [P]}{\vdash \Gamma, M \,\&\, N, [P]}\&$$

$$\frac{}{\vdash \Gamma, \top, [P]}\top \qquad \frac{}{\vdash P^\perp, P}\textsc{Ax} \qquad \frac{\vdash \Gamma, P \qquad \vdash \Delta, P^\perp, [Q]}{\vdash \Gamma, \Delta, [Q]}\textsc{Cut}$$

Fig. 10.7 Rules of MALLP

where P, Q are called **positive** and M, N are called **negative**. There is a clear duality between the two, defined as $0^\perp = \top$, $1^\perp = \bot$, $({\downarrow}M)^\perp = {\uparrow}M^\perp$, $(P \otimes Q)^\perp = P^\perp \parr Q^\perp$, and $(P \oplus Q)^\perp = P^\perp \,\&\, Q^\perp$; and vice versa. There are two kinds of sequents: those of the form $\vdash \Gamma$ and those of the form $\vdash \Gamma, P$, where in both cases all formulas in Γ are assumed negative. Following Melliès and Tabareau (2010) we write $\vdash \Gamma, [P]$ any of the two cases. We show the rules in Fig. 10.7. As before we consider exchange rules present, though not written explicitly.

MALLP is a refinement of MALL, in the sense that given a MALLP proof, erasing the shifts and the corresponding deduction rules yields an MALL proof. In fact MALL proofs obtained from MALLP are *focused*[13] proofs: indeed, in a focused proof, at any given time in proof construction we focus on at most one positive formula. The only positive rules used must apply to this positive formula, until we reach a negative formula. This process is faithfully reflected by the presence of at most one positive formula in a MALLP sequent. In fact, the focusing property of Linear Logic, first noticed by Andreoli (1992), can be proved through a translation of MALL in MALLP.

10.4.2 Interpretation of MALLP

The polarity of formulas is directly reflected in the accompanying games. A game A is **positive** (resp. **negative**) if all its minimal events are positive (resp. negative). In general games may be neither negative nor positive, as in the example above interpreting $(1 \oplus 1) \otimes (1 \,\&\, 1)$. In contrast, games interpreting MALLP formulas will always have a clear polarity. As a matter of fact, their shape will be even more restricted.

[13] More precisely, *weakly focused* in the sense of Laurent (2023).

Definition 6 A finite game A is an **arena** if: (1) all its minimal events share the same polarity and conflict with each other; (2) causal dependency is *alternating* (if $a_1 \rightarrowtriangle_A a_2$ then $\mathrm{pol}_A(a_1) \neq \mathrm{pol}_A(a_2)$) and tree-shaped (if $a_1, a_1' \leq_A a_2$ then either $a_1 \leq_A a_1'$ or $a_1' \leq_A a_1$); and (3) conflict is *local* in the sense that if $a_1 \sim\!\sim a_2$, then either they are both minimal or they share the same (necessarily unique) immediate predecessor.

By definition, an arena is either negative or positive. We denote negative arenas by M, N and positive arenas by P, Q. Every positive arena may be written as

$$P = \sum_{i \in I} \downarrow N_i$$

with I finite, and where each N_i is a negative game (which might not be an arena). This lets us define the **tensor** of positive arenas as

$$P \otimes Q = \sum_{(i,j) \in I \times J} \downarrow (N_i \parallel M_j)$$

where $P = \sum_{i \in I} \downarrow N_i$ and $Q = \sum_{j \in J} \downarrow M_j$. We also define their **sum** $P \oplus Q$ simply as $P + Q$ (note that this use of the notation $P \oplus Q$ is incompatible with that in Sect. 10.3.1—from now on, all uses of $\oplus$ refer to the present definition). The arena 1 consists of only one move, which is positive; and 0 is the empty arena. Negative arenas and their constructions are defined dually. Altogether, this gives us an interpretation $[\![-]\!]$ of formulas as arenas.

To interpret proofs, we flesh out the categorical structure relative to these constructions. We preface this with a few remarks. Firstly, MALLP may be presented as a 2-sided sequent calculus with at most one formula on the right: $\vdash N_1, \ldots, N_n, [P]$ is represented simply as $N_1^\perp, \ldots, N_n^\perp \vdash P$. All formulas involved are then positive. Positive MALLP formulas are uniquely written with the positive connectives 0, 1, $\otimes$ and $\oplus$; along with *negation* $\neg$ defined as $\neg P = \downarrow(P^\perp)$. The resulting system is known as Multiplicative Additive Tensorial Logic (Melliès & Tabareau, 2010). To interpret MALLP we construct a *dialogue category with coproducts* (Melliès & Tabareau, 2010) which matches the Tensorial Logic presentation of MALLP, but the two are completely equivalent.

We start with the category Arenas, whose objects are positive arenas, and morphisms from P to Q are strategies $\sigma : P^\perp \parallel Q$ which are **negative**, in the sense that each $x \in \mathscr{C}(\sigma)$ must contain at least one negative event, and **thunkable**, in the sense that for each $x \in \mathscr{C}(\sigma)$, if x contains one positive event, then it contains one in Q. So morphisms in Arenas first wait for an Opponent input on the left; then immediately play on the right. Let us write $\sigma : P \overset{\mathrm{Ar}}{\rightarrow\!\!\!\!+\!\!\!>} Q$ to denote that σ is a morphism from P to Q in Arenas.

The construction $\oplus$ yields a coproduct in Arenas, let us write $\mathrm{in}_l : P \overset{\mathrm{Ar}}{\rightarrow\!\!\!\!+\!\!\!>} P \oplus Q$ and $\mathrm{in}_r : Q \overset{\mathrm{Ar}}{\rightarrow\!\!\!\!+\!\!\!>} P \oplus Q$ for the two injections, and $[\sigma_1, \sigma_2] : P_1 \oplus P_2 \overset{\mathrm{Ar}}{\rightarrow\!\!\!\!+\!\!\!>} Q$ for the co-pairing of $\sigma_1 : P_1 \overset{\mathrm{Ar}}{\rightarrow\!\!\!\!+\!\!\!>} Q$ and $\sigma_2 : P_2 \overset{\mathrm{Ar}}{\rightarrow\!\!\!\!+\!\!\!>} Q$. We use similar notations for the corresponding n-ary construction. This lets us decompose any $\sigma : P \overset{\mathrm{Ar}}{\twoheadrightarrow} Q$ as

$$\sigma = [\mathrm{in}_{f_\sigma(i)} \odot \downarrow(\sigma_i) \big| i \in I] : \sum_{i \in I} \downarrow M_i \overset{\mathrm{Ar}}{\twoheadrightarrow} \sum_{j \in J} \downarrow N_j$$

where $f_\sigma : I \to J$ is a function, $\sigma_i : M_i \overset{\mathrm{Ga}}{\twoheadrightarrow} N_{f_\sigma(i)}$ is a (necessarily) negative strategy in Games; and $\downarrow(-)$ is the functorial action of $\downarrow$ in Games defined in the obvious way. This also lets us define the functorial action of $\otimes$: if $\sigma : P \overset{\mathrm{Ar}}{\twoheadrightarrow} Q$ and $\sigma' : P' \overset{\mathrm{Ar}}{\twoheadrightarrow} Q'$,

$$\sigma \otimes \sigma' : [\mathrm{in}_{(f_\sigma(i), f_{\sigma'}(i'))} \odot \downarrow(\sigma_i \parallel \sigma'_{i'}) \big| (i, i') \in I \times I'] : P \otimes P' \overset{\mathrm{Ar}}{\twoheadrightarrow} Q \otimes Q'$$

making Arenas a symmetric monoidal category with coproducts, where $\otimes$ distributes over coproducts in the sense that the canonical morphisms $(P \otimes Q_1) \oplus (P \otimes Q_2) \overset{\mathrm{Ar}}{\twoheadrightarrow} P \otimes (Q_1 \oplus Q_2)$ and $0 \overset{\mathrm{Ar}}{\twoheadrightarrow} P \otimes 0$ are isomorphisms—Arenas may be regarded as the free coproduct completion of a category of negative games and strategies.

Finally, Arenas has a **tensorial negation**, i.e. a (necessarily self-adjoint) functor $\neg : \mathsf{Arenas} \to \mathsf{Arenas}^{\mathrm{op}}$ together with a family of bijections

$$\varphi_{P,Q,R} : \mathsf{Arenas}[P \otimes Q, \neg R] \cong \mathsf{Arenas}[P, \neg(Q \otimes R)]$$

natural in P, Q and R and subject to a coherence condition. On arenas, we define $\neg P = \downarrow P^\perp$, extended to strategies in the obvious way with the functorial action of $\downarrow$ and the compact closed structure of Games. Altogether we get a **dialogue category** with coproducts (Melliès & Tabareau, 2010) hence a model of Multiplicative Additive Tensorial Logic, or equivalently MALLP. A proof ϖ of $\vdash N_1, \ldots N_n, P$ with a positive formula is interpreted as a morphism

$$[\![\varpi]\!] : \bigotimes_{1 \leq i \leq n} [\![N_i]\!]^\perp \overset{\mathrm{Ar}}{\twoheadrightarrow} [\![P]\!]$$

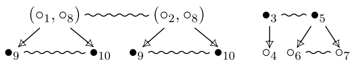
while a proof ϖ of a sequent $\vdash N_1, \ldots, N_n$ is interpreted as $[\![\varpi]\!] : \bigotimes_{1 \leq i \leq n} [\![N_i]\!]^\perp \overset{\mathrm{Ar}}{\twoheadrightarrow} \neg 1$.

The categorical structure should make it plain how the rules are interpreted; we only comment the introduction rules for shifts: the introduction of $\downarrow$ directly matches the natural isomorphism $\varphi_{\Gamma^\perp, M^\perp, 1}$; while the introduction rule for $\uparrow$ first composes $[\![\varpi]\!] : [\![\Gamma]\!]^\perp \overset{\mathrm{Ar}}{\twoheadrightarrow} P$ with the unit of the continuation monad $P \overset{\mathrm{Ar}}{\twoheadrightarrow} \neg\neg P$, before applying $\varphi^{-1}_{\Gamma, \neg P, 1}$.

We display in Fig. 10.8 a branch of a proof and its corresponding interpretation. As in Fig. 10.5, we omit the $\vdash$ symbol and color units to disambiguate the rules. On the right hand side, we first show the game interpreting the sequent, and the maximal augmentation of the strategy $[\![\varpi]\!]$ corresponding to the branch of the proof displayed on the left hand side. We observe that this augmentation is completely linear—in

$$\varpi = \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{1}\,1}{1 \oplus 1}\,\oplus_l}{\uparrow(1 \oplus 1)}\,\uparrow}{\bot, \uparrow(1 \oplus 1)}\,\bot \quad \dfrac{\dots}{\bot, \uparrow(1 \oplus 1)}}{\bot \,\&\, \bot, \uparrow(1 \oplus 1)}\,\&}{\downarrow(\bot \,\&\, \bot), \uparrow(1 \oplus 1)}\,\downarrow}{\downarrow\bot \oplus \downarrow(\bot \,\&\, \bot), \uparrow(1 \oplus 1)}\,\oplus_r}{\bot, \downarrow\bot \oplus \downarrow(\bot \,\&\, \bot), \uparrow(1 \oplus 1)}\,\bot \quad \dfrac{\dots}{\bot, \downarrow\bot \oplus \downarrow(\bot \,\&\, \bot), \uparrow(1 \oplus 1)}}{\bot \,\&\, \bot, \downarrow\bot \oplus \downarrow(\bot \,\&\, \bot), \uparrow(1 \oplus 1)}\,\&$$

$$[\![\bot_1 \,\&\, \bot_2, \downarrow_3\bot_4 \oplus \downarrow_5(\bot_6 \,\&\, \bot_7), \uparrow_8(1_9 \oplus 1_{10})]\!] =$$



$$(\circ_1, \circ_8) \to \bullet_5 \to \circ_6 \to \bullet_9 \quad \in \quad [\![\varpi]\!]$$

Fig. 10.8 A proof in MALLP and its interpretation

fact, we will see that all strategies obtained as the interpretation of proofs in tensorial logic have a forest-like causal structure, a property that in the next subsection we will call *sequential innocence*.

10.4.3 Full Completeness for MALLP$^\flat$

Now, we refine the interpretation in order to obtain full completeness. From now on and for the remainder of this paper, we will restrict to the fragments MALL$^\flat$ and MALLP$^\flat$, respectively of MALL and MALLP, without the additive units 0 and $\top$. While the methodology we present here does extend in their presence, they come with technical complications that are a significant obstacle to our objective of keeping this paper as simple as possible and focused on the conceptual ideas. The reader will find in Appendix 2 a generalization of the constructions for full MALL and MALLP.

Strategies coming from proofs satisfy constraints of two different natures. The first two conditions, *totality* and *sequential innocence*, are *causal*: they capture the causal patterns of strategies arising from MALLP proofs. The third condition, *exhaustivity*, is *positional* and expresses that complete positions of strategies should validate the linearity constraints by exhausting all resources in their complete positions.

10.4.3.1 Totality and Sequential Innocence

Our first two conditions, *totality* and *sequential innocence*, are intrinsic to strategies, meaning that they restrict their causal shapes without enriching the interpretation of types.

Totality. In game semantics, *proofs* (as opposed to *programs*) are traditionally interpreted as strategies that are *total*, in the sense that they always have a response to any move by the environment. Game semantics for proofs makes formal a debate between two players, arguing about the validity of a formula: Player aims to establish the truth of the formula, while Opponent attempts to falsify it. In this view, a proof should yield a strategy that never gives up, and has a valid counter-argument to any attack by Opponent.

In our games, totality may be formulated as follows.

Definition 7 A strategy $\sigma : A$ is **total** if for any $\mathrm{q} \in \sigma$ maximal in σ (for rigid embedding), the maximal events (for $\leq_{\mathrm{q}}$) of q have positive polarity.

Regarding strategies as descriptions of normal forms, totality is a normalization property—any exploration of the normal form by Opponent will uncover new parts of the term and will not trigger divergence. Just as terms with a normal form are not usually closed under composition (considering e.g. $\delta\,\delta$ in the pure λ-calculus), total strategies are not in general stable under composition as two total strategies may enter in a *livelock*, never producing an observable result. Getting total strategies to compose often requires some technology (Clairambault & Harmer, 2010); but here as our games are finite, compositionality of totality is easy.

Sequential Innocence. In Hyland-Ong games, *innocent strategies* are those whose behaviour only depends on a partial sub-history of the play called the *P-view*. In causal game semantics, the P-views appear simply as the underlying causal structure. In traditional game semantics this causal structure is usually derived: strategies are typically defined and composed as sets of general plays, and appear a posteriori to be representable as sets of P-views. Our direct handling of the causal structure makes innocence appear very differently from its traditional presentation: we must simply restrict the causal shapes to those that follow the tree-like inductive structure of proofs.

Definition 8 A strategy $\sigma : A$ is **sequential innocent** iff any $\mathrm{q} \in \sigma$ is forest-shaped, and O-branching: if $a \rightarrowtriangle_{\mathrm{q}} a_1$ and $a \rightarrowtriangle_{\mathrm{q}} a_2$ with $a_1, a_2 \in |\mathrm{q}|$ distinct, $\mathrm{pol}(a_1) = \mathrm{pol}(a_2) = -$.

If A is an arena, then this means that for any $\mathrm{q} \in \sigma$ and $a \in |\mathrm{q}|$, the causal history of a in q is a linearly ordered causal chain, which is alternating:

$$a_0^- \rightarrowtriangle_{\mathrm{q}} a_1^+ \rightarrowtriangle_{\mathrm{q}} a_2^- \rightarrowtriangle_{\mathrm{q}} a_3^+ \rightarrowtriangle_{\mathrm{q}} \dots \rightarrowtriangle_{\mathrm{q}} a.$$

Note that because arenas are forest-shaped, each move that is not minimal in A has a unique antecedent in A. From the conditions imposed on augmentations, if a_i appears in a causal chain as above, then its antecedent must also appear before. Let us call the antecedent of a_i its **justifier**. Then by courtesy, in a chain as above the justifier of a_{2n+2}^- must be a_{2n+1}^+; and the justifier of a_{2n+1}^+ must be one of the earlier negative events. So, this is exactly a *P-view*; making more concrete the intuitions suggested above. Globally, σ may then be regarded as a prefix-closed set of linearly ordered causal chains (P-views) as above branching only at Opponent moves. Two branching chains (P-views) may be either *compatible* (if they are both prefixes of a common augmentation), or *conflicting* (if not).

This link with more traditional structures of innocence in game semantics is a strength of the presentation of strategies as sets of augmentations rather than as sets of configurations, sets of plays (Melliès & Mimram, 2007) or closure operators (Abramsky & Melliès, 1999). As in Hyland-Ong games, this also means that strategies have a simple inductive structure, aiding definability.

In traditional game semantics, proving that innocence is stable under composition is tricky. In contrast here, stability of sequential innocence under composition is very easy:

Proposition 11 *If* $\sigma : A \overset{\mathrm{Ar}}{\twoheadrightarrow} B$ *and* $\tau : B \overset{\mathrm{Ar}}{\twoheadrightarrow} C$ *are sequential innocent, then so is* $\tau \odot \sigma$.

Proof It suffices to show that if $\mathrm{q} \in \sigma$ and $\mathrm{p} \in \tau$ are causally compatible, then every event e in $\mathrm{p} \circledast \mathrm{q}$ has at most one immediate antecedent. Looking for a contradiction, assume that

$$e_1 \rightarrowtriangle_{\mathrm{p}\circledast\mathrm{q}} e \qquad\qquad e_2 \rightarrowtriangle_{\mathrm{p}\circledast\mathrm{q}} e.$$

If e is an external Opponent move, by courtesy $e_1, e_2 \rightarrowtriangle_{A^\perp \| C} e$, so $e_1 = e_2$ since arenas are forest-shaped. Otherwise e is positive for either σ or τ, say *w.l.o.g.* σ. Then, by an analysis of immediate causality in an interaction [essentially Lemma 2.10 of Castellan et al. (2019)] along with courtesy of τ, we have $e_1, e_2 \rightarrowtriangle_\sigma e$, so $e_1 = e_2$ since σ is sequential innocent. $\square$

In other words, no *causal join* can emerge in an interaction between strategies that do not perform causal joins. Structural morphisms are sequential innocent, and all other constructions on strategies are easily shown to preserve sequential innocence. One can wonder why stability of innocence is so easy here, compared to traditional games. It seems that in traditional games, the complexity comes from the back and forth between P-views (the *causal* structure) and plays, which we completely avoid here.

Finally, an observation on the name *sequential innocence*: in concurrent games, the notion above appears as a sequential specialization of a more general notion of *parallel innocence* (Castellan et al., 2015). Parallel innocent strategies have no side-effect but may perform computations in parallel—they include strategies for *parallel-or* (Castellan, 2017), or strategies arising from the parallel evaluation of purely functional programming languages (Castellan et al., 2015).

10.4.3.2 Exhaustive Strategies

For the simply-typed λ-calculus, totality and innocence suffice to obtain definability; but not here: indeed, totality allows *affine behaviour*[14] whereas we want strict linearity.

[14] In fact, the proof of Theorem 2 shows that without exhaustivity, the model is fully complete for Polarized Multiplicative Additive Affine Logic (MAALP) (Laurent, 2002).

Several existing mechanisms could be used here to ensure strict linearity. Our choice of name, *exhaustive strategies*, reminds one of the *exhausting strategies* of Murawski and Ong (2003), in which one asserts that all moves of the game should be somehow reachable by strategies. Unlike what our name suggests, we opt instead for a simple and elegant construction due to Melliès (2005) and then refined by Melliès and Tabareau (2010). The construction works by enriching arenas with a notion of *payoff*.

Definition 9 An **arena with payoff** is an arena A with $\kappa_A : \mathscr{C}(A) \to \{-1, 0, +1\}$ such that for A non-empty, A is positive (*resp.* negative) iff $\kappa_A(\emptyset) = -1$ (*resp.* $\kappa_A(\emptyset) = +1$).

Configurations $x \in \mathscr{C}(A)$ such that $\kappa_A(x) = 0$ are called **exhaustive**—for games coming from $\mathsf{MALLP}^\flat$, we will see that those are exactly the maximal configurations. For non-exhaustive configurations, κ assigns the *responsibility* of non-exhaustivity, i.e. points out which of the two players is stalling. Configurations $x \in \mathscr{C}(A)$ such that $\kappa_A(x) = +1$ are called **winning**: the responsibility of non-exhaustivity is assigned to Opponent. Dually, configurations $x \in \mathscr{C}(A)$ such that $\kappa_A(x) = -1$ are called **losing**.

Here, we make three observations. Firstly, the reader can observe the proximity with the winning conditions of Clairambault et al. (2012): the main difference is the existence of *neutral* positions, or *draws*, with null payoff. Secondly, unlike e.g. Winskel (2012), the objective of strategies will be to at least ensure a draw, i.e. either reach an exhaustive configuration, or a state where the responsibility of non-exhaustivity may be assigned to Opponent.

Constructions. Let us show how the constructions on arenas extend in the presence of payoff functions. For units, $\kappa_1(\emptyset) = -1$ and $\kappa_1(\{\updownarrow\}) = 0$—the payoff on $\perp$ is defined dually, with $\kappa_{A^\perp} = -\kappa_A$. For lifts, we set $\kappa_{\downarrow N}(\emptyset) = -1$, and $\kappa_{\downarrow N}(\{\bullet\} \cup x_N) = \kappa_N(x_N)$.

For positive $P = \sum_{i \in I} \downarrow N_i$ and $Q = \sum_{j \in J} \downarrow M_j$, we set $\kappa_{P \oplus Q}(x_P) = \kappa_P(x_P)$ if $x_P \in \mathscr{C}(P)$ and symmetrically for Q. For the tensor, we first set $\kappa_{P \otimes Q}(\emptyset) = -1$. Non-empty configurations of $P \otimes Q$ necessarily have the form $\{\updownarrow_{(i,j)}\} \cup (x_{N_i} \parallel x_{M_j})$, written $x_P \otimes x_Q$ where $x_P = \{\updownarrow_i\} \cup x_{N_i}$ and $x_Q = \{\updownarrow_j\} \cup x_{M_j}$. We then set

$$\kappa_{P \otimes Q}(x_P \otimes x_Q) = \kappa_P(x_P) \otimes \kappa_Q(x_Q)\,,$$

where, for $\alpha, \beta \in \{-1, 0, 1\}$, we set $\alpha \otimes \beta = 0$ iff $\alpha = \beta = 0$, $\alpha \otimes \beta = -1$ if $\alpha = -1$ or $\beta = -1$, and $\alpha \otimes \beta = 1$ otherwise. Finally, $\kappa_{N \parr M}$ is defined dually. A non-exhaustive configuration of $P \otimes Q$ is winning if it is winning or exhaustive on both sides, whereas a non-exhaustive configuration of $M \parr N$ must be winning on at least one side.

For each $\mathsf{MALLP}^\flat$ formula A we may build by induction on A, following the definitions above, an arena with payoff $[\![A]\!]$. We mention in passing the following straightforward lemma, where we say that $x \in \mathscr{C}(A)$ **is $+$-maximal in** $\mathscr{C}(A)$ iff for any $y \in \mathscr{C}(\sigma)$ such that $x \subseteq^+ y$ we have $x = y$; and symmetrically for $-$-maximal configurations.

Lemma 3 *For any* $\mathsf{MALLP}^\flat$ *formula* A*, (1) if* x *is* $+$*-maximal in* $\mathscr{C}(\llbracket A \rrbracket)$*, then* $\kappa_{\llbracket A \rrbracket}(x) \geq 0$*; (2) if* x *is* $-$*-maximal in* $\mathscr{C}(\llbracket A \rrbracket)$*, then* $\kappa_{\llbracket A \rrbracket}(x) \leq 0$*; and (3)* x *is maximal iff* $\kappa_{\llbracket A \rrbracket}(x) = 0$*.*

If we were to interpret all units, 0 would yield the empty arena with $\kappa_0(\emptyset) = -1$, failing the lemma above—this is the reason why the proof of definability we present here does not directly apply to additive units, which require more elaborate constructions.

Exhaustive strategies. We may now define *exhaustive strategies*. If $\sigma : A$ is a strategy and $x \in \mathscr{C}(\sigma)$, we say that it is **+-maximal** if σ has no further move to play, i.e. for any $y \in \mathscr{C}(\sigma)$ such that $x \subseteq^+ y$ we have $x = y$.

Definition 10 Let P, Q be positive arenas with payoff. A strategy $\sigma : P \overset{\mathrm{Ar}}{\twoheadrightarrow} Q$ is **exhaustive** if for any $x_P \parallel x_Q \in \mathscr{C}(\sigma)$ $+$*-maximal* we have $\kappa_{P^\perp}(x_P) \parr \kappa_Q(x_Q) \geq 0$.

The proof that exhaustive strategies are stable under composition is exactly as in the proof of stability of winning strategies in Clairambault et al. (2012); other constructions on strategies are direct. From now on, we consider all arenas to be equipped with a payoff function, and all strategies to be total, sequential innocent and exhaustive. Altogether we get a dialogue category with coproducts, that we will keep referring to as Arenas.

The reader may wonder why we call those strategies *exhaustive* rather than *winning*. For us, the use of *winning* in game semantics usually conveys the idea that winning strategies witness logical validity. But here, exhaustivity does not ensure logical validity. It is perfectly conceivable, for instance, to have a programming language with recursion and divergence but with a strict linearity discipline that will ensure exhaustivity but where two definable exhaustive $\sigma : A$ and $\tau : A^\perp$ simultaneously exist. Then, the exhaustivity mechanism only ensures that their interaction yields an exhaustive configuration of A.

10.4.4 Full Completeness

To obtain full completeness for $\mathsf{MALLP}^\flat$ it remains to prove definability.

To any sequent $\vdash N_1, \ldots, N_n, [P]$ of $\mathsf{MALLP}^\flat$ and strategy

$$\sigma : \bigotimes_{1 \leq i \leq n} \llbracket N_i \rrbracket^\perp \overset{\mathrm{Ar}}{\twoheadrightarrow} \llbracket [P] \rrbracket,$$

where $\llbracket [P] \rrbracket$ means $\neg 1$ if there is no P, we associate a proof of $\vdash N_1, \ldots, N_n, [P]$ whose interpretation yields σ. This is done, as expected, by induction on the number of events in σ and the size (number of symbols) in the sequent.

We start by taking care of a few easy cases. If one of the N_i is $\perp$ or starts with $\parr$, then we apply directly the corresponding rule, not changing the game and strategy up to iso. If one of the N_i—say N_n is a product $N_n^1 \& N_n^2$, then the interpretation of the context is, up to isomorphism, a product, so that σ can be regarded as inhabiting:

$$\sigma : ([\![N_1]\!]^\perp \otimes \ldots \otimes [\![N_{n-1}]\!]^\perp \otimes [\![N_n^1]\!]^\perp) \oplus ([\![N_1]\!]^\perp \otimes \ldots \otimes [\![N_{n-1}]\!]^\perp \otimes [\![N_n^2]\!]^\perp) \overset{\mathrm{Ar}}{\twoheadrightarrow} [\![[P]]\!]$$

hence σ is a co-pair $[\sigma_1, \sigma_2]$. By induction hypothesis, each σ_i is defined with a proof, and hence σ may be defined via the introduction rule for &.

Hence, we can assume that all arenas for N_i have the form $\uparrow P_i$, so that the game for

$$\sigma : \bigotimes_{1 \le i \le n} \downarrow [\![P_i]\!]^\perp \overset{\mathrm{Ar}}{\twoheadrightarrow} [\![[P]]\!]$$

has a unique negative minimal move corresponding to the shifts on the left hand side (this also holds in the case where the tensor is empty, as its unit 1 has exactly one event). We now distinguish several cases, depending on the shape of $[P]$. Of these, the crucial case—by far the most subtle—is that where there is one positive formula, of the form $Q_1 \otimes Q_2$.

Lemma 4 *Let $(P_k)_{1 \le k \le n}$, Q_1, Q_2 be arenas, and consider a strategy*

$$\sigma : \bigotimes_{1 \le i \le n} \downarrow P_k^\perp \overset{\mathrm{Ar}}{\twoheadrightarrow} Q_1 \otimes Q_2.$$

Then, up to reordering of the context there are strategies

$$\sigma_1 : \bigotimes_{1 \le k \le p} \downarrow P_k^\perp \overset{\mathrm{Ar}}{\twoheadrightarrow} Q_1 \qquad \sigma_2 : \bigotimes_{p+1 \le k \le n} \downarrow P_k^\perp \overset{\mathrm{Ar}}{\twoheadrightarrow} Q_2$$

such that $\sigma = \sigma_1 \otimes \sigma_2$.

Proof *W.l.o.g.* we can assume that neither Q_1 nor Q_2 is 1. Then, the game has the shape

$$\sigma : \downarrow(\|_{1 \le k \le n} \sum_{i \in I_k} \uparrow M_{k,i}^\perp) \overset{\mathrm{Ar}}{\twoheadrightarrow} \sum_{(l_1,l_2) \in L_1 \times L_2} \downarrow(N_{l_1} \parallel N_{l_2}).$$

where $Q_i = \sum_{l \in L_i} \downarrow N_{l_i}$ and $P_k^\perp = M_k = \|_{1 \le k \le n} \sum_{i \in I_k} \uparrow M_{k,i}^\perp$. After the unique minimal negative move, σ starts by playing some (l_1, l_2), and then resumes as a negative strategy

$$\sigma' : \|_{1 \le k \le n} \sum_{i \in I_k} \uparrow M_{k,i}^\perp \overset{\mathrm{Ga}}{\twoheadrightarrow} N_{l_1} \parallel N_{l_2}.$$

But then, there is a partition of the components of the parallel composition on the left hand side into those that may be accessed through N_{l_1}, through N_{l_2}, and those (in principle) that will *not* be accessed. Indeed, recall that augmentations in σ' are forest-shaped (because σ is sequential innocent). Two augmentations q_1 and q_2 visiting one component $M_k = \sum_{i \in I_k} \uparrow M_{k,i}^{\perp}$ cannot be compatible, so they contain respectively conflicting Opponent events. But since conflict is local in arenas, this is only possible if q_1 and q_2 either share the same minimal event, or if their minimal events are conflicting. In both cases, they start in the same component, N_{l_1} or N_{l_2}. So for each $1 \leq k \leq n$, M_k may be accessed only via N_{l_1}, or via N_{l_2}. Reordering the context we can rewrite the game for σ' as

$$\sigma' \ : \ \Gamma_1 \parallel \Gamma_2 \parallel \Gamma_3 \ \overset{\mathrm{Ga}}{\twoheadrightarrow} \ N_{l_1} \parallel N_{l_2}.$$

where all components of $\Gamma_1 = \|_{1 \leq k \leq p} \sum_{i \in I_k} \uparrow M_{k,i}^{\perp}$ (*resp.* $\Gamma_2 = \|_{p \leq n} \sum_{i \in I_k} \uparrow M_{k,i}^{\perp}$) are accessed through N_{l_1} (*resp.* N_{l_2}) and only, and components of $\Gamma_3 = \|_{n_2 \leq k \leq n} \uparrow M_{k,i}$ are not accessed. But then Γ_3 must be empty. Indeed if $x \in \mathscr{C}(\sigma)$ is maximal, then it is $-$-maximal in the game so $\kappa(x) \leq 0$ by Lemma 3 and $\kappa(x) \geq 0$ since σ is exhaustive, so $\kappa(x) = 0$. But then it follows that x is maximal in the game, so Γ_3 must indeed be accessed if non-empty. Then, σ' decomposes as $\sigma_1' : \Gamma_1 \overset{\mathrm{Ga}}{\twoheadrightarrow} N_{l_1}$ and $\sigma_2' : \Gamma_2 \overset{\mathrm{Ga}}{\twoheadrightarrow} N_{l_2}$, yielding

$$\sigma_1 = \mathrm{in}_{l_1} \odot (\downarrow\sigma_1') : \Delta_1 \overset{\mathrm{Ar}}{\twoheadrightarrow} Q_1 \qquad \sigma_2 = \mathrm{in}_{l_2} \odot (\downarrow\sigma_2') : \Delta_2 \overset{\mathrm{Ar}}{\twoheadrightarrow} Q_2$$

(where $\Delta_1 = \otimes_{1 \leq k \leq p} \downarrow M_k = \downarrow\Gamma_1$ and $\Delta_2 = \otimes_{p \leq k \leq n} \downarrow M_k = \downarrow\Gamma_2$) such that $\sigma = \sigma_1 \otimes \sigma_2$. From the fact that σ is exhaustive, along with Lemma 3 and the fact that any configuration of σ may be extended to a $-$-maximal one, it follows that σ_1 and σ_2 are exhaustive. □

With that, we can finally wrap up and conclude:

Theorem 2 Arenas *is fully complete for* $\mathsf{MALLP}^\flat$.

Proof We resume the proof where it was before Lemma 4, i.e. we must define

$$\sigma : \bigotimes_{1 \leq i \leq n} \downarrow [\![P_i]\!]^{\perp} \ \overset{\mathrm{Ar}}{\twoheadrightarrow} \ [\![[P]]\!].$$

If $P = 1$ then a $+$-maximal configuration must be neutral on both sides, which is only possible if $\kappa(\{\updownarrow\}) = 0$ on the left. Because the context contains no $\perp$, this in turn is only possible if $n = 0$, but then $\vdash 1$ is provable. If $P = Q_1 \oplus Q_2$ is a coproduct; then after the initial move on the left, σ must either play on Q_1 or on Q_2 (say e.g. on Q_1) hence $\sigma = \mathrm{in}_l \circ \sigma'$ with $\sigma' : \bigotimes_{1 \leq i \leq n} \downarrow [\![P_i]\!]^{\perp} \ \overset{\mathrm{Ar}}{\twoheadrightarrow} \ [\![Q_1]\!]$. By induction hypothesis σ' may be defined, and we define σ using the introduction rule for $\oplus$. If $P = Q_1 \otimes Q_2$, we apply Lemma 4.

There are two cases left, which have to do with shifts. First, we consider the case where the positive formula is a down-shift, so that we have

$$\sigma : \bigotimes_{1\leq i\leq n} \downarrow M_i \xrightarrow{\mathrm{Ar}} \downarrow N.$$

Then, σ is obtained via $\varphi_{\Gamma^\perp, N^\perp, 1}$ from $\sigma' : (\bigotimes_{1\leq i\leq n} \downarrow M_i) \otimes N^\perp \xrightarrow{\mathrm{Ar}} \neg 1$, which is definable by induction hypothesis. Finally, the last remaining case is that where

$$\sigma : \bigotimes_{1\leq i\leq n} \downarrow M_i \xrightarrow{\mathrm{Ar}} \neg 1.$$

Necessarily, after the initial negative move on the left σ immediately plays on the right, and after the subsequent move on 1, by totality it plays on the left again, say *w.l.o.g.* on M_n. Then, removing from (all augmentations in) σ the two moves in $\neg 1$, we get

$$\sigma' : \bigotimes_{1\leq i\leq n-1} \downarrow M_i \xrightarrow{\mathrm{Ar}} \downarrow M_n$$

which is definable by induction hypothesis. It follows then that σ is definable as well, obtained via the introduction rule for $\uparrow$. At each step of the definability procedure, the number of connectives in the sequent decreases, ensuring termination. □

Behind the details of this definability procedure lies a very direct geometric correspondence between derivations in $\mathsf{MALLP}^\flat$ (up to natural commutations between rules) and sequential innocent total strategies, akin to the usual correspondence between Böhm trees and innocent strategies in the traditional sense for the simply-typed λ-calculus. This full completeness result makes it appropriate to think of strategies as normal forms for proofs modulo cut elimination and bureaucratic commutations between proof rules. This is also related to Melliès' result that innocent strategies (for a different but related notion of innocence) in asynchronous games form the free dialogue category (Melliès, 2012).

The reader will find in Appendix 2 an extension of this result to MALLP with all units.

10.4.5 Relational Collapse and Full Completeness for $\mathsf{MALL}^\flat$

Finally we show how to interpret unpolarized MALL in Arenas, describe the relational collapse, and deduce full completeness. Additive units cause no further difficulty here, so we formulate our constructions in their presence even though we will only be able to conclude full completeness for $\mathsf{MALL}^\flat$.

10.4.5.1 Interpretation of MALL and Polarized Translation

We first give an interpretation of MALL which, as in Sect. 10.3.1, will not be quite sound since it will fail some required equations between proofs.

Remember that the **positive connectives** of MALL are defined as 0, 1, $\otimes$ and $\oplus$; while the **negative connectives** are the others. The **polarity** of an MALL formula is defined as the polarity of its outermost connective. As for MALLP, below we denote positive formulas of MALL as P, Q and negative formulas as M, N. To any formula A of MALL we associate two arenas, $(\!|A|\!)_-$ negative and $(\!|A|\!)_+$ positive, mutually inductively with

$$\begin{array}{rclrcl} (\!|1|\!)_+ &=& 1 & (\!|A \otimes B|\!)_+ &=& (\!|A|\!)_+ \otimes (\!|B|\!)_+ \\ (\!|0|\!)_+ &=& 0 & (\!|A \oplus B|\!)_+ &=& (\!|A|\!)_+ \oplus (\!|B|\!)_+ \\ (\!|\bot|\!)_- &=& \bot & (\!|A \parr B|\!)_- &=& (\!|A|\!)_- \parr (\!|B|\!)_- \\ (\!|\top|\!)_- &=& \top & (\!|A \& B|\!)_- &=& (\!|A|\!)_- \& (\!|B|\!)_- \end{array}$$

along with $(\!|P|\!)_- = \uparrow(\!|P|\!)_+$ and $(\!|N|\!)_+ = \downarrow(\!|N|\!)_-$ to insert the shifts when polarities do not match. This interpretation corresponds to translations $(-)^-$ of MALL formulas as *negative* MALLP formulas, and $(-)^+$ of MALL formulas as *positive* MALLP formulas, followed by the interpretation of MALLP formulas as arenas $[\![-]\!]$ defined in the previous section.

This interpretation can easily be extended to MALL proofs: any proof ϖ of a sequent $\vdash A_1, \ldots, A_n$ is interpreted as a negative, sequential innocent, exhaustive and total strategy:

$$(\!|\varpi|\!) : (\!|A_1|\!)_-^\perp \otimes \ldots \otimes (\!|A_n|\!)_-^\perp \overset{\mathrm{Ar}}{\twoheadrightarrow} \neg 1.$$

It is straightforward to extend this interpretation to all rules of MALL. Altogether, this exactly amounts to the translation of MALL proofs into MALLP proofs described in Melliès and Tabareau (2010) (along with other Linear Logic connectives). Overall this gives an interpretation of MALL which, however, will not validate all the expected equations between MALL proofs.

10.4.5.2 Relational Collapse

To restore these missing equations, the final step is to quotient out from the interpretation all the additional behaviour corresponding to the shifts. For that purpose, Melliès' idea in Melliès (2005) was to quotient the strategies coinciding on certain *stopping positions*, ignoring that they might have taken different routes to reach those positions. The same idea may also be simply presented as a *functorial collapse* to the relational model.

The relational model. The category Rel has as objects *sets*, and as morphisms from A to B *relations* $R \subseteq A \times B$ from A to B. The cartesian product of sets extends to a symmetric monoidal closed structure on Rel. Furthermore Rel is compact closed, with duality being the identity. It has biproducts, given by the disjoint union of sets. Altogether this yields an interpretation of MALL into Rel, defined on formulas with

$$\begin{array}{rcl} [\![0]\!]_{\mathsf{Rel}} = & [\![\top]\!]_{\mathsf{Rel}} & = \emptyset \\ [\![1]\!]_{\mathsf{Rel}} = & [\![\bot]\!]_{\mathsf{Rel}} & = \{\star\} \\ [\![A \otimes B]\!]_{\mathsf{Rel}} = & [\![A \parr B]\!]_{\mathsf{Rel}} & = [\![A]\!]_{\mathsf{Rel}} \times [\![B]\!]_{\mathsf{Rel}} \\ [\![A \oplus B]\!]_{\mathsf{Rel}} = & [\![A \& B]\!]_{\mathsf{Rel}} & = [\![A]\!]_{\mathsf{Rel}} + [\![B]\!]_{\mathsf{Rel}} \end{array}$$

and extended to proofs following the categorical structure. See e.g. Thomas (2012) for details.

The collapse of games. Now, we have argued earlier in the paper that games being positional meant that they have a clean connection with relational semantics. Intuitively, the relational semantics of a proof records positions reached by completed executions. In contrast, concurrent games record all positions reached by a proof, including intermediary ones matching partial executions.[15] Thus, in principle, it would seem that the correspondence between concurrent games and relational semantics should be rather straightforward: simply forget the intermediary steps, and keep only complete positions.

Following this intuition, to any arena A we associate the set

$$\textstyle\int A = \{x \in \mathscr{C}(A) \,\big|\, \kappa_A(x) = 0\}$$

of *exhaustive* configurations. Crucially, this operation is compatible with all constructions used to interpret formulas in Arenas and Rel. For instance, for two positive arenas P, Q, we have seen that configurations with null payoff are exactly those of the form $x_P \otimes x_Q$ with x_P and x_Q of null payoff in P and Q respectively; so $\int(P \otimes Q)$ is isomorphic to the cartesian product $(\int P) \times (\int Q)$—observe also that *shifts* leave the set of exhaustive configurations invariant, up to isomorphism. Going through all formula constructors, we establish:

Lemma 5 *For any* MALL *formula A, there is an isomorphism* $\int (\!|A|\!) \cong [\![A]\!]_{\mathsf{Rel}}$.

It remains then to extend this collapse operation to strategies.

The collapse of strategies. For $\sigma : A \overset{\mathrm{Ar}}{\twoheadrightarrow} B$, the appropriate definition seems obvious:

[15] This is in contrast with traditional game semantics, that record all *paths* rather than positions.

$$\int \sigma = \{(x_A, x_B) \in \int A \times \int B \;\big|\; x_A \parallel x_B \text{ is } +\text{-maximal in } \mathscr{C}(\sigma)\} \in \mathsf{Rel}(\int A, \int B).$$

It is immediate that $\int(-)$ preserves identities, and almost all constructions on strategies used in the interpretation up to the isomorphism of Lemma 5.

One central property, however, requires some care: *functoriality*. Indeed, to show that $\int(-)$ preserves composition there is a significant obstacle, at least conceptually: composition in Arenas is not pure relational composition. As made explicit in Proposition 6, composition of strategies is relational composition augmented with an additional *reachability* assumption, eliminating synchronized states resulting in *deadlocks*. So we have $\int(\tau \odot \sigma) \subseteq \int \tau \circ \int \sigma$, but it is not clear that the converse equality also holds.

In fact, for general strategies this functoriality property fails—it is easy to construct a situation like that of Fig. 10.4 on arenas arising from the interpretation of formulas of MALL or MALLP. But on that respect, (sequential) innocent strategies are special in that their composition causes *no deadlocks*—this phenomenon, which seems to have been noticed independently by Boudes (2004) and Melliès (2005), entails the following:

Lemma 6 *For $\sigma : P \overset{\mathrm{Ar}}{\twoheadrightarrow} Q$, $\tau : Q \overset{\mathrm{Ar}}{\twoheadrightarrow} R$ sequential innocent exhaustive strategies,*

$$\int(\tau \odot \sigma) = \int \tau \circ \int \sigma$$

Using the causal presentation of games, this may be established by analysing cycles arising when computing interactions between sequential innocent strategies, as in Sect. 10.3.2.1. By iteratively simplifying such hypothetical cycles, one may prove that they do not exist. The proof does not actually depend on sequential innocence, but on the much weaker property we call *visibility* (Castellan et al., 2015). This development is too lengthy to appear here, but the interested reader may find a detailed statement and proof as Lemma 5.32 in Castellan (2017).

Finally, we conclude:

Theorem 3 *There is a functor $\int(-) : \mathsf{Arenas} \to \mathsf{Rel}$ preserving interpretation up to iso.*

10.4.5.3 Full Completeness for $\mathsf{MALL}^\flat$

The functor $\int(-)$ induces an equivalence relation on strategies in Arenas, defined as $\sigma \equiv \sigma'$ iff $\int \sigma = \int \sigma'$. Because $\int(-)$ preserves the structure used in the interpretation, it follows that $\equiv$ is a congruence, so we may quotient homsets in Arenas. It remains then to conclude:

Theorem 4 *The interpretation $(\!|-|\!) : \mathsf{MALL}^\flat \to \mathsf{Arenas}/\equiv$ is fully complete.*

Proof Although the interpretation $(\!|-|\!)$ into Arenas fails soundness in general, the interpretation in $\mathsf{Arenas}/\equiv$ is sound. Moreover, if $\vdash A_1, \ldots, A_n$ is an $\mathsf{MALL}^\flat$ sequent and

$$\sigma : (\!|A_1|\!)^{\perp}_{-} \otimes \ldots \otimes (\!|A_n|\!)^{\perp}_{-} \overset{\mathrm{Ar}}{\twoheadrightarrow} \neg 1,$$

then by Theorem 2, there is a proof ϖ in $\mathsf{MALLP}^{\flat}$ of the sequent $\vdash A_1^-, \ldots, A_n^-$ such that $[\![\varpi]\!] = \sigma$. Removing shifts in ϖ yields a proof ϖ' of $\vdash A_1, \ldots, A_n$ in $\mathsf{MALL}^{\flat}$. Finally, the interpretation of $\mathsf{MALL}^{\flat}$ into Arenas preserves the relational interpretation, i.e. $\int (\!|\varpi'|\!) = [\![\varpi']\!]_{\mathsf{Rel}}$, hence $[\![\varpi']\!]_{\mathsf{Rel}} \equiv \sigma$ as required. □

It is worth noting that as the $\mathsf{MALL}^{\flat}$ proofs coming from definability are obtained by erasing the shifts from $\mathsf{MALLP}^{\flat}$ proofs, they are *focused* proofs.

10.5 Conclusion

We hope that this paper will help in making more accessible the work on games models of MALL, starting with Abramsky and Melliès' seminal paper on concurrent games via closure operators. In writing this paper we have attempted to make it as pedagogical and self-contained as possible so that besides telling the story of concurrency and additives, it may also be used as an introduction to concurrent games.

Positionality and causality. As a take-home message, we emphasize once more the *causal* and *positional* nature of deterministic concurrent strategies under their various forms. The *positional* presentation reveals a clear understanding of the similarities—and differences—between game and relational semantics. The *causal* presentation endows strategies with a concrete nature that may be leveraged to capture innocence.

In this paper, we have used the word *causal* to describe the model construction in Sect. 10.3.2.1 and *positional* to describe that in Sect. 10.3.2.2. It is in our opinion a fundamental, deep property of deterministic concurrent strategies that they enjoy such sharply different presentations. But as in this paper the qualifiers *causal* and *positional* accompany the same model, the reader may wonder to what extent these two are intrinsically related. One element of answer if that beyond the deterministic case, concurrent strategies are defined causally but the purely positional presentation given here does not survive: for a non-deterministic strategy σ on A, the behaviour of σ in a configuration $x \in \mathscr{C}(A)$ may depend on the *path* used to reach x, as this path might constrain the current augmentation (i.e. the non-deterministic slice) more than the configuration does.

Nevertheless, one can push the causal presentation way further than the deterministic case. Some constructions of this paper survive, in particular we have recent generalizations of the relational collapse to the *probabilistic* (Castellan et al., 2018) and *quantum* (Clairambault & Marc de Visme, 2020) cases.

On sequentiality. In the end, it appears that the ability to express concurrency is not per se what allows us to get full completeness: in fact, the interpretation of MALLP that leads to full completeness for MALL is completely sequential. We hope to have convinced the reader that beyond concurrency, the true conceptual advance offered by Abramsky and Melliès' closure-strategies was *positionality*, and that despite their names, the family of *concurrent games* have a lot to offer to the study of *sequential* languages.

Nevertheless, even to study proof systems it is compelling to explore the use of the concurrency offered by the model. For instance, in light of *multifocusing* (Chaudhuri et al., 2008) it would seem natural to seek canonical representations of MALL proofs exploiting the parallelism inherent to concurrent games. The recent work of Castellan and Yoshida (2019) goes in that direction, representing dependencies between logical rules in MALL as a disjunctive deterministic strategy (Castellan, 2017), but precise connections remain to be explored.

Acknowledgements The author would like to thank Olivier Laurent and Simon Castellan for numerous discussions on the contents of this paper, and comments on earlier versions. Thanks also to the anonymous reviewer, whose insightful comments have been precious in preparing the final version. This work was supported by ANR project DyVerSe (ANR-19-CE48-0010-01); and Labex MiLyon (ANR-10-LABX-0070) of Université de Lyon, within the program "Investissements d'Avenir" (ANR-11-IDEX-0007), operated by the French ANR.

Appendix 1: Other Closure Operators from Strategies

In Sect. 10.3.3, we have studied the transformation of deterministic concurrent strategies into closure-strategies. As pointed out in the introduction, this transformation is new. In this first appendix we review some tempting alternative definitions, that have been considered in the literature (Melliès & Mimram, 2007; Rideau & Winskel, ch10DBLP:confspslicsspsRideauW11).

1.1 Intersection of +-Maximal Configurations

Firstly, recall from Sect. 10.2.3 that for a domain D, the identity $\text{æ}_D : D \multimap\!\!\!\!\!\!\rightarrow D$ is defined as $\text{æ}_D(x, y) = (x \vee y, x \vee y)$. As already pointed out in Sect. 10.3.3.2, if D is the domain of configurations of a game, this has the puzzling consequence that æ_D may actually play Opponent moves and not just Player moves. For instance, if A is the game with just one Player move $\bullet$, then as observed in Sect. 10.3.3.2, we have $\text{æ}_{\mathscr{C}^\infty(A)}(\emptyset, \{\bullet\}) = (\{\circ\}, \{\bullet\})$—in other words, applying $\text{æ}_{\mathscr{C}^\infty(A)}$ has the effect of adding the missing Opponent dependency to an already present Player move $\bullet$. This invites the following definition (Melliès & Mimram, 2007)[16]:

Definition 11 If $\sigma : A$ is a strategy and $x \in \mathscr{C}^\infty(A)$, then we set

[16] The formal setting differs superficially and a detailed proof of equivalence is out of scope of this paper, but we have checked that the problem described here also occurs in Melliès and Mimram (2007).

$$\mathsf{C}'(\sigma)(x) = \bigwedge\{y \in \mathscr{C}^\infty(\sigma) \big| y \text{ is } +\text{-maximal } \wedge\ x \subseteq y\} \in \mathscr{C}^\infty(A)^\top$$

Given $x \in \mathscr{C}^\infty(A)$, if there is no $+$-maximal $y \in \mathscr{C}^\infty(\sigma)$ such that $x \subseteq y$, then this definition yields $\mathsf{C}'(\sigma)(x) = \top$, otherwise it is their intersection. In other words, applied to a (possibly infinite) configuration $x \in \mathscr{C}^\infty(A)$, $\mathsf{C}'(\sigma)$ adds all moves that are known to appear in all $+$-maximal configurations containing x. This includes of course the Player events enabled by Opponent events in x, but also all Opponent events that are necessary requirements for the Player events already present in x.

This definition is tempting, because it is analogous to the copycat closure-strategy: for any game A, it is apparent that $\mathsf{C}'(\text{æ}_A) = \text{æ}_{\mathscr{C}^\infty(A)}$. However, unfortunately it does not in general give a closure-strategy. We give below counter-examples to stability and continuity.

Example 1 (Non-stability). Consider the game

$$\begin{tikzcd} A & = & \ominus_1 \arrow[r, no head, squiggly, bend left] \arrow[rr, no head, squiggly, bend right] & \ominus_2 \arrow[r, no head, squiggly, bend left] & \ominus_3 \arrow[r, no head, squiggly, bend left] & \ominus_4 & \oplus_1 & \oplus_2 & \oplus_3 \end{tikzcd}$$

and the deterministic concurrent strategy $\sigma : A$ having maximal augmentations

$$\begin{tikzcd} \ominus_1 \arrow[d] \arrow[dr] & \ominus_2 \arrow[d] & \ominus_2 \arrow[d] & \ominus_3 \arrow[dl] \arrow[d] & \ominus_2 & \ominus_4 \arrow[d] \\ \oplus_1 & \oplus_2 & \oplus_2 & \oplus_3 & & \oplus_2 \end{tikzcd}$$

yielding, by Definition 11,

$$\begin{aligned} \mathsf{C}'(\sigma)(\{\oplus_1, \oplus_2\}) &= \{\oplus_1, \oplus_2\} \cup \{\ominus_1, \ominus_2\} \\ \mathsf{C}'(\sigma)(\{\oplus_2, \oplus_3\}) &= \{\oplus_2, \oplus_3\} \cup \{\ominus_2, \ominus_3\}. \end{aligned}$$

But $\{\oplus_1, \oplus_2\}$ and $\{\oplus_2, \oplus_3\}$ are compatible, since $\{\oplus_1, \oplus_2, \oplus_3\} \in \mathscr{C}(A)$. Therefore, the stability condition of closure-strategies entails that we should have

$$\mathsf{C}'(\sigma)(\{\oplus_2\}) = \{\oplus_2\} \cup \{\ominus_2\}.$$

However, this is false: instead we have $\mathsf{C}'(\sigma)(\{\oplus_2\}) = \{\oplus_2\}$ since there is a $+$-maximal configuration of σ, namely $\{\ominus_4, \oplus_2\}$ which does not contain $\ominus_2$.

Note that Melliès and Mimram (2007) did not claim stability. We now also give a counter-example to continuity.

Example 2 (Non-continuity). Consider the game A having as events

$$\{\ominus_i \big| i \geq 0\} \cup \{\ominus'_j \geq j \geq 1\} \cup \{\oplus\}.$$

Causality is trivial, comprising only reflexive pairs. Minimal conflicts are described by $\ominus_i \rightsquigarrow \ominus_j$ for all $i \neq j$, and $\ominus_i \rightsquigarrow \ominus'_i$ for all $i \geq 1$. We then consider $\sigma : A$ the deterministic concurrent strategy defined with maximal augmentations of the form

$$\begin{tikzcd} \ominus_i \arrow[d] & \ominus'_j \\ \oplus & \end{tikzcd}$$

for $i \neq j$. Then, for all $n \geq 1$, we have

$$\mathsf{C}'(\sigma)(\{\oplus, \ominus'_1, \ldots, \ominus'_n\}) = \{\oplus, \ominus'_1, \ldots, \ominus'_n\} :$$

F no new event is added, because there are still many mutually inconsistent possible causal histories for $\oplus$. By continuity, we should therefore also have $\mathsf{C}'(\sigma)(\{\oplus\} \cup \{\ominus'_j \,|\, j \geq 1\}) = \{\oplus\} \cup \{\ominus'_j \,|\, j \geq 1\}$. However, instead we have

$$\mathsf{C}'(\sigma)(\{\oplus\} \cup \{\ominus'_j \,|\, j \geq 1\}) = \{\oplus\} \cup \{\ominus'_j \,|\, j \geq 1\} \cup \{\ominus_0\}.$$

Indeed, any $+$-maximal configuration of σ which includes $\{\ominus'_j \,|\, j \geq 1\}$ must also contain $\ominus_0$: it is the only possible cause left for $\oplus$, and is therefore included by the definition.

Both of these pathologies boil down to the fact that the configurations of a deterministic concurrent strategy $\sigma : A$ are not in general closed under intersection; unless we assume that there is no conflict between Opponent events in the game. It is noteworthy that despite these, the reachable fixpoints of $\mathsf{C}'(\sigma)$ are always the same as those of $\mathsf{C}(\sigma)$.

1.2 Least $+$-Maximal Configuration

Finally, we mention a variation of Definition 11 that also appears in the literature Rideau and Winskel (2011).

Definition 12 If $\sigma : A$ is a deterministic concurrent strategy and $x \in \mathscr{C}^\infty(A)$, we set $\mathsf{C}''(\sigma)(x)$ as the *least* $+$-maximal $y \in \mathscr{C}^\infty(\sigma)$ *s.t.* $x \subseteq y$, if such exists, and $\top$ otherwise.

The difference with respect to Definition 11 is that we consider the least $+$-maximal configuration containing x rather than their intersection. This may be tempting, because it ensures that for all $x \in \mathscr{C}^\infty(\sigma)$, we always have $\mathsf{C}''(\sigma)(x) \in \mathscr{C}^\infty(\sigma)$ unless $\mathsf{C}''(\sigma)(x) = \top$—this natural property is satisfied by neither in the definition of Proposition 7 not in Definition 11. However, this definition unfortunately fails monotonicity.

Example 3 (Non-monotonicity). Consider the game $A = \ominus_1 \sim \ominus_2 \quad \oplus$ and $\sigma : A$ with maximal augmentations $\ominus_1 \rightarrowtriangle \oplus$ and $\ominus_2 \rightarrowtriangle \oplus$. Then,

$$\mathsf{C}''(\sigma)(\{\oplus\}) = \top$$
$$\mathsf{C}''(\sigma)(\{\oplus, \ominus_1\}) = \{\oplus, \ominus_1\},$$

failing monotonicity of $\mathsf{C}''(\sigma)$. Indeed, there are two incomparable $+$-maximal $y_1, y_2 \in \mathscr{C}^\infty(\sigma)$ such that $\{\oplus\} \subseteq y_1, y_2$, so there is no least one, leading to $\top$.

Although it does not give a closure-strategy in general, it is noteworthy that $\mathsf{C}''(\sigma)$ also has the same *reachable* fixpoints as $\mathsf{C}(\sigma)$ and $\mathsf{C}'(\sigma)$.

Appendix 2: Full Completeness for MALLP

In this final section, we show how to refine the fully complete model for $\mathsf{MALLP}^\flat$ of Sect. 10.4.3 into a fully complete model for MALLP. This comes with significant technical complications in order to deal adequately with the additive units.

First, payoff is extended to additive units by setting $\kappa_0(\emptyset) = -1$, and dually, $\kappa_\top(\emptyset) = 1$. As pointed out in the text, this breaks Lemma 3 which was useful to prove definability for $\mathsf{MALLP}^\flat$, but the interpretation itself still works out, yielding for every proof a total, sequential innocent, and exhaustive strategy. We will shortly see, however, that those conditions are not enough for definability in the presence of additive units.

2.1 Locally Winning Strategies

Unlike multiplicative units, additive units allow a proof to leave parts of the arena unexplored. Indeed, any sequent $\vdash \Gamma, \top$ is provable by the $\top$ rule, yielding a strategy that will never visit Γ—*garbage-collects* it. This is captured by the notion of exhaustivity in the presence of additive units: an exhaustive strategy σ may garbage-collect part of the context, provided σ is able to uncover a unit $\top$ ensuring that the global payoff is 1.

However, for definability we must ensure that the uncovered $\top$ belongs to a component that will "stay with" the garbage-collected context during the inductive definability process. Unfortunately, this is not ensured by exhaustivity: we show in Fig. 10.9 a total, sequential innocent and exhaustive strategy failing definability. The figure displays a strategy (call it σ) playing on the game for $\uparrow 1, \downarrow(\perp \,\&\, \uparrow 1) \otimes \downarrow \top$—indices are added to emphasize the correspondence between moves and formula components. The strategy satisfies all of our conditions, even though the sequent is not provable in MALLP. Attempting to apply the definability process, one must decompose σ as a tensor of two strategies. The only way forward is defining σ'

with the same moves as σ, but on sequent $\uparrow 1, \downarrow(\perp \,\&\, \uparrow 1)$. But σ' is not exhaustive anymore—the configuration $\{\circ_1, \bullet_3, \circ_5, \bullet_6\}$, which had payoff 1 in σ thanks to the presence of the $\top$ allowing us to leave part of the context unexplored, is now losing.

Definability arguments in game semantics require “good” (i.e. satisfying all the imposed conditions) strategies to be stable under *decomposition*, in the sense that strategies obtained by decomposing good strategies should be good. This property, which is usually for free, fails here due to the non-local behaviour of additive units. It is precisely to deal with this issue that Melliès considers in Melliès (2005) a payoff for *walks* on strategies rather than simply positions. Rather than reproduce Melliès' construction, we give a variant of the mechanism, which we believe to be more explicit. Our condition, called *local exhaustivity*, expresses that “σ is exhaustive on all sub-games”. To express it, we first need to enrich payoffs so that they also assign valuations to configurations on sub-games.

Definition 13 If A is an arena and $x \in \mathscr{C}(A)$, a **sub-arena** of A is a subset $X \subseteq A$ which is up-closed for $\leq_A$, and such that there is $x \in \mathscr{C}(A)$ such that all minimal events of X are enabled in x. A **local payoff** on A consists in functions

$$\kappa_A^X : \mathscr{C}(X) \to \{-1, 0, 1\}$$

for any sub-arena X, where X inherits from A the components of an event structure. Furthermore, those satisfy the additional properties that (1) if $X = \emptyset$, then $\kappa_A^{\emptyset}(\emptyset) = 0$; and (2) if $X \neq \emptyset$ and its minimal events are minimal in A, then $\kappa_A^X(y) = \kappa_A(y)$.

For instance, in Fig. 10.9, the set $\{\bullet_2, \circ_4, \circ_5, \bullet_6\}$ is a sub-arena.

Besides being exhaustive globally, strategies must also be exhaustive locally in the sub-arenas they reach. If $\sigma : A$ is a strategy, $x \in \mathscr{C}(\sigma)$ is $+$-maximal, and we have $x \subseteq x_1, \ldots, x_n \in \mathscr{C}(\sigma)$ distinct configurations that are also $+$-maximal, then we write

$$\lceil x_1, \ldots, x_n \rceil_x = \{a \in A \,\big|\, \exists a' \in x_1 \cup \cdots \cup x_n,\ a' \leq_A a\} \setminus x$$

for the up-closure of $x_1, \ldots, x_n$ with x removed. By construction, it is a sub-arena of A. We then ask that each x_i is then exhaustive, localized to this sub-arena.

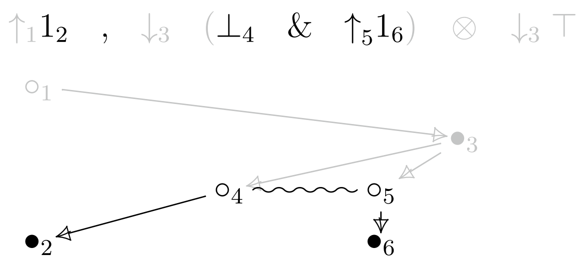


Fig. 10.9 A winning, non-decomposable strategy

Definition 14 A strategy $\sigma : A$ is **locally exhaustive** iff for all $x, x_1, \ldots, x_n \in \mathscr{C}(\sigma)$ which are all $+$-maximal and such that $x \subseteq x_1, \ldots, x_n$, for all $1 \leq i \leq n$, we have

$$\kappa_A^{\lceil x_1,\ldots,x_n \rceil_x}(x_i \setminus x) \geq 0.$$

We extend arena constructions with local payoff. For units, the local payoff is forced by the conditions. For $A \oplus B$, if a sub-arena X is entirely included in A and $y \in \mathscr{C}(X)$, we set $\kappa^X_{A\oplus B}(y) = \kappa^X_A(y)$ and likewise for B. If X has components in A and B, then its minimal events are necessarily minimal in A and B. We then set $\kappa^X_{A\oplus B}(y) = \kappa_{A\oplus B}(y)$. For $A \otimes B$, if X is empty then $\kappa^X_{A\otimes B}$ is forced by condition (1). If its minimal events are minimal in $A \otimes B$ then $\kappa^X_{A\otimes B}$ is forced by condition (2). Otherwise, X decomposes into X_A a sub-arena of A and X_B a sub-arena of B. Likewise, if $y \in \mathscr{C}(X)$, it decomposes into $y_A \in \mathscr{C}(X_A)$ and $y_B \in \mathscr{C}(X_B)$. We then set $\kappa^X_{A\otimes B}(y) = \kappa^{X_A}_A(y_A) \otimes \kappa^{X_B}_B(y_B)$. For $\downarrow N$, either $X = \downarrow N$ in which case $\kappa^X_{\downarrow N} = \kappa_{\downarrow N}$ by condition (2), or X is a sub-arena of N, and we set $\kappa^X_{\downarrow N}(y) = \kappa^X_N(y)$. Other cases follow by duality, with $\kappa^X_{A^\perp}(y) = -\kappa^X_A(y)$.

Example 4 The strategy of Fig. 10.9 is not locally exhaustive: we have $+$-maximal

$$\{\circ_1, \bullet_3\} \subseteq \{\circ_1, \bullet_3, \circ_4, \bullet_2\}, \{\circ_1, \bullet_3, \circ_5, \bullet_6\}$$

inducing the reachable sub-game $\{\circ_4, \bullet_2, \circ_5, \bullet_6\}$, which corresponds to the part of Fig. 10.9 which is not grayed out. But then, the configuration $\{\circ_5, \bullet_6\}$ fails to be exhaustive:

$$\begin{aligned}
\kappa^{\{\circ_4,\bullet_2,\circ_5,\bullet_6\}}_{(\uparrow_1 1_2) \parr (\downarrow_3(\perp_4 \,\&\, \uparrow_5 1_6)\otimes(\downarrow_3 \top))}(\{\circ_5, \bullet_6\}) &= \kappa^{\{\bullet_2\}}_{\uparrow_1 1_2}(\emptyset) \parr (\kappa^{\{\circ_4,\circ_5,\bullet_6\}}_{\perp_4 \,\&\, \uparrow_5 1_6}(\{\circ_5, \bullet_6\}) \otimes \kappa^{\emptyset}_{\downarrow_3 \top}(\emptyset))\\
&= \kappa^{\{\bullet_2\}}_{1_2}(\emptyset) \parr (\kappa_{\uparrow_5 1_6}(\{\circ_5, \bullet_6\}) \otimes \kappa^{\emptyset}_{\downarrow_3 \top}(\emptyset))\\
&= \kappa_{1_2}(\emptyset) \parr (\kappa_{\uparrow_5 1_6}(\{\circ_5, \bullet_6\}) \otimes \kappa^{\emptyset}_{\downarrow_3 \top}(\emptyset))\\
&= -1 \parr (0 \otimes 0)\\
&= -1.
\end{aligned}$$

There is a category $\mathsf{LocExAr}$ having as objects arenas with local payoff, and as morphisms total, sequential innocent strategies that are both exhaustive and locally exhaustive. Furthermore, $\mathsf{LocExAr}$ inherits from Arenas the structure of a dialogue category with coproducts, supporting the interpretation of MALLP.

2.2 Definability and Full Completeness

We now prove full completeness. From now on, *strategies* are always assumed to satisfy sequential innocence, totality, exhaustivity and locally exhaustivity.

With respect to the proof of definability of Sect. 10.4.4, the only difference is the decomposition of a tensor, which requires local exhaustivity in the presence of additive units.

Lemma 7 *Let* $(P_k)_{1\leq k\leq n}$, Q_1, Q_2 *be arenas, and consider*

$$\sigma : \bigotimes_{1\leq i\leq n} \downarrow P_k^\perp \overset{\mathrm{Ar}}{\twoheadrightarrow} Q_1 \otimes Q_2$$

a morphism in LocExAr. *Then, up to reordering of the context there are strategies*

$$\sigma_1 : \bigotimes_{1\leq k\leq p} \downarrow P_k^\perp \overset{\mathrm{Ar}}{\twoheadrightarrow} Q_1 \qquad \sigma_2 : \bigotimes_{p+1\leq k\leq n} \downarrow P_k^\perp \overset{\mathrm{Ar}}{\twoheadrightarrow} Q_2$$

such that $\sigma = \sigma_1 \otimes \sigma_2$.

Proof At first the proof proceeds as in Lemma 4. We first extract

$$\sigma_1 : \Delta_1 \overset{\mathrm{Ar}}{\twoheadrightarrow} Q_1 \qquad \sigma_2 : \Delta_2 \overset{\mathrm{Ar}}{\twoheadrightarrow} Q_2$$

as in the proof of Lemma 4, using the same notations. It follows easily that σ_1 and σ_2 are locally exhaustive from the fact that σ is locally exhaustive. We first consider the case where Γ_3 is empty in the construction of σ_1 and σ_2, i.e. $\Delta_1 \otimes \Delta_2 = \Delta = \bigotimes_{1\leq i\leq n} \downarrow P_k^\perp$.

The main novelty is that exploiting local exhaustivity, we may show that σ_1 and σ_2 are exhaustive as well. Indeed, take $x \in \mathscr{C}(\sigma_1)$ $+$-maximal, and consider X the set of non-empty $+$-maximal configurations of σ_1—necessarily, $x \in X$. Leaving the renaming implicit, we regard X as a set of $+$-maximal configurations of σ. Moreover, for all $y \in X$ we have $x_0 = \{\circ, \bullet_{(l_1,l_2)}\} \subseteq y$ where $\circ$ and $\bullet_{(l_1,l_2)}$ are the initial two moves of σ. We may then use that σ is locally exhaustive, and obtain

$$\kappa^{\lceil X \rceil_{x_0}}_{(\Delta_1\otimes\Delta_2)^\perp \parr (Q_1\otimes Q_2)}(x \setminus x_0) \geq 0$$

where $\lceil X \rceil_{x_0}$ is a sub-arena of $(\Delta_1 \otimes \Delta_2)^\perp \parr (Q_1 \otimes Q_2)$. But by construction of X, this sub-arena contains no move in Δ_2 and Q_2, so it is a sub-arena $X_l \parallel X_r$ of $\Delta_1^\perp \parr Q_1$, where X_l is a sub-arena of Δ_1 and X_r is a sub-arena of Q_1. We compute:

$$\begin{aligned}
\kappa^{\lceil X \rceil_{x_0}}_{(\Delta_1\otimes\Delta_2)^\perp \parr (Q_1\otimes Q_2)}(x \setminus x_0) &= \kappa^{X_l\parallel X_r}_{(\Delta_1\otimes\Delta_2)^\perp \parr (Q_1\otimes Q_2)}(x_l \parallel x_r)\\
&= \kappa^{X_l}_{(\Delta_1\otimes\Delta_2)^\perp}(x_l) \parr \kappa^{X_r}_{(Q_1\otimes Q_2)}(x_r)\\
&= (\kappa^{X_l}_{\Delta_1^\perp}(x_l) \parr 0) \parr (\kappa^{X_r}_{Q_1}(x_r) \otimes 0)\\
&= \kappa^{X_l}_{\Delta_1^\perp}(x_l) \parr \kappa^{X_r}_{Q_1}(x_r)
\end{aligned}$$

where we have used that X_l is entirely in Δ_1 and X_r in Q_1, so the local payoffs in Δ_2 and Q_2 are null by condition (1) of Definition 13. But now, recall that:

$$\Delta_1 = \bigotimes_{1 \leq k \leq p} \downarrow M_k \qquad\qquad Q_1 = \sum_{l_1 \in L_1} \downarrow N_{l_1}.$$

Since σ is total, after $\circ$, $\bullet_{(l_1,l_2)}$ it has a response to any of the minimal events of N_{l_1}. So, each minimal event of N_{l_1} appears in at least one $+$-maximal configuration of σ_1, thus $X_r = N_{l_1}$. Likewise, recall that for each $1 \leq k \leq p$, we have $M_k = \sum_{i \in I_k} \uparrow M_{k,i}^\perp$. Recall that Δ_1 was constructed by selecting those components M_k that were accessed by an augmentation with minimal negative event (after $\circ$, $\bullet_{(l_1,l_2)}$) in N_{l_1}. Therefore, X_l comprises at least one of the $\uparrow M_{k,i}^\perp$ for each $1 \leq k \leq p$. From these two observations, it follows directly by induction on Δ_1 and Q_1 and condition (2) of Definition 13 that $\kappa_{\Delta_1^\perp}^{X_l}(x_l) = \kappa_{\Delta_1^\perp}(\{\circ\} \cup x_l)$ and $\kappa_{Q_1}^{X_r}(x_r) = \kappa_{Q_1}(\{\bullet_{l_1}\} \cup x_r)$ so

$$\kappa^{\lceil X \rceil_{x_0}}_{(\Delta_1 \otimes \Delta_2)^\perp \parr (Q_1 \otimes Q_2)}(x \setminus x_0) = \kappa_{\Delta_1^\perp \parr Q_1}(x)$$

which is therefore positive, as required. Likewise, σ_2 is exhaustive as well.

In the proof of Lemma 4, we proved that Γ_3 must always be empty. In the presence of additive units, that is of course no longer true. We prove that in this case as well, σ_1 and σ_2 are still exhaustive. Consider $x \in \mathscr{C}(\sigma)$ non-empty and $+$-maximal, and write $x = \{\circ\} \cup x_l \parallel \{\bullet_{(l_1,l_2)}\} \cup (x_{l_1} \parallel x_{l_2})$. Because Γ_3 is not explored, we have $\kappa_{\Delta^\perp}(\{\circ\} \cup x_l) = -1$, so we must have $\kappa_{Q_1 \otimes Q_2}(\{\bullet_{(l_1,l_2)}\} \cup (x_{l_1} \parallel x_{l_2})) = 1$ to compensate, so that $\kappa_{Q_1}(\{\bullet_{l_1}\} \cup x_{l_1}) = 1$ or $\kappa_{Q_2}(\{\bullet_{l_2}\} \cup x_{l_2}) = 1$. But in fact there must be a side, Q_1 or Q_2, that always has payoff 1 independently of x. Indeed say we have $\kappa_{Q_1}(x_{l_1}) = 0$ and $\kappa_{Q_2}(x'_{l_2}) = 0$ where

$$\{\circ\} \cup x_l \parallel \{\bullet_{(l_1,l_2)}\} \cup (x_{l_1} \parallel x_{l_2}) \in \mathscr{C}(\sigma) \qquad \{\circ\} \cup x'_l \parallel \{\bullet_{(l_1,l_2)}\} \cup (x'_{l_1} \parallel x'_{l_2}) \in \mathscr{C}(\sigma)$$

These configurations are images of a unique augmentation, so each move in x_l depends either on x_{l_1} or on x_{l_2}, and likewise for x'. So the two configurations above admit as subsets $+$-maximal configurations

$$\{\circ\} \cup y_l^1 \parallel \{\bullet_{(l_1,l_2)}\} \cup (x_{l_1} \parallel \emptyset) \in \mathscr{C}(\sigma) \qquad \{\circ\} \cup y_l^2 \parallel \{\bullet_{(l_1,l_2)}\} \cup (\emptyset \parallel x_{l_2}) \in \mathscr{C}(\sigma).$$

By determinism, we may now take their union

$$\{\circ\} \cup (y_l^1 \cup y_l^2) \parallel \{\bullet_{(l_1,l_2)}\} \cup (x_{l_1} \parallel x'_{l_2}) \in \mathscr{C}(\sigma)$$

which by construction has payoff -1. So, there is $i \in \{1, 2\}$ so that for all $x \in \mathscr{C}(\sigma)$ non-empty $+$-maximal, we have $\kappa_{Q_i}(x_{l_i}) = 1$. Say *w.l.o.g.* that it is $i = 2$. Then, we form

$$\sigma_1 : \Delta_1^{\perp} \overset{\mathrm{Ar}}{\twoheadrightarrow} Q_1 \qquad\qquad \sigma_2 : (\Delta_2 \otimes \Delta_3)^{\perp} \overset{\mathrm{Ar}}{\twoheadrightarrow} Q_2$$

as previously, but with a larger domain for σ_2. By construction we have $\sigma = \sigma_1 \otimes \sigma_2$, and σ_2 is exhaustive by construction. The proof that σ_1 and σ_2 satisfy all the required conditions is as in the case above with Γ_3 empty. □

With that, we can now complete the proof of:

Theorem 5 LocExAr *is fully complete for* MALLP.

Proof The rest of the proof is as in Theorem 2. □

From there, the exact same construction as in Sect. 10.4.5 can be applied in order to get a fully complete model for MALL with all units. We omit the details, which are unchanged.

References

Abramsky, S. (2003). Sequentiality vs. concurrency in games and logic. *Mathematical Structures in Computer Science, 13*(4), 531–565.

Abramsky, S., & Jagadeesan, R. (1992). Games and full completeness for multiplicative linear logic (extended abstract). In *Foundations of Software Technology and Theoretical Computer Science, 12th Conference*, New Delhi, India, December 18–20, 1992 (pp. 291–301).

Abramsky, S., Jagadeesan, R., & Malacaria, P. (2000). Full abstraction for PCF. *Information and Computation, 163*(2), 409–470.

Abramsky, S., & McCusker, G. (1997). Call-by-value games. In *Computer Science Logic, 11th International Workshop, CSL '97, Annual Conference of the EACSL*, Aarhus, Denmark, August 23–29, 1997, Selected Papers (pp. 1–17).

Abramsky, S., & Melliès, P. (1999). Concurrent games and full completeness. In *14th Annual IEEE Symposium on Logic in Computer Science*, Trento, Italy, July 2–5, 1999 (pp. 431–442).

Alcolei, A., Clairambault, P., Hyland, M., & Winskel, G. (2018). The true concurrency of Herbrand's theorem. In *27th EACSL Annual Conference on Computer Science Logic, CSL 2018*, September 4–7, 2018, Birmingham, UK (pp. 5:1–5:22).

Alcolei, A., Clairambault, P., & Laurent, O. (2019). Resource-tracking concurrent games. In *Foundations of Software Science and Computation Structures—22nd International Conference, FOSSACS 2019, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2019*, Prague, Czech Republic, April 6–11, 2019 (pp. 27–44).

Amadio, R. M., Curien, P.-L., et al. (1998). *Domains and lambda-calculi* (No. 46). Cambridge University Press.

Andreoli, J.-M. (1992). Logic programming with focusing proofs in linear logic. *Journal of Logic and Computation, 2*(3), 297–347.

Baillot, P. (1999). *Approches dynamiques de la logique linéaire: jeux et géométrie de l'interaction* (Ph.D. thesis), Aix-Marseille 2 University.

Baillot, P., Danos, V., Ehrhard, T., & Regnier, L. (1997). Believe it or not, AJM's games model is a model of classical linear logic. In *Proceedings, 12th Annual IEEE Symposium on Logic in Computer Science*, Warsaw, Poland, June 29–July 2, 1997 (pp. 68–75).

Berry, G. (1979). *Modèles complètement adéquats et stables des λ-calculs typés* (Ph.D. thesis), Paris 7.

Berry, G., & Curien, P.-L. (1982). Sequential algorithms on concrete data structures. *Theoretical Computer Science, 20*, 265–321.

Blass, A. (1992). A game semantics for linear logic. *Annals of Pure and Applied Logic, 56*(1–3), 183–220.

Blute, R., Hamano, M., & Scott, P. (2005). Softness of hypercoherences and MALL full completeness. *Annals of Pure and Applied Logic, 131*(1–3), 1–63.

Boudes, P. (2004). Projecting games on hypercoherences. In *Automata, Languages and Programming: 31st International Colloquium, ICALP 2004*, Turku, Finland, July 12–16, 2004 (pp. 257–268).

Castellan, S. (2017). *Concurrent structures in game semantics (Structures concurrentes en sémantique des jeux)* (Ph.D. thesis), University of Lyon, France.

Castellan, S., & Clairambault, P. (2016). Causality vs. interleavings in concurrent game semantics. In *27th International Conference on Concurrency Theory, CONCUR 2016*, August 23–26, 2016, Québec City, Canada (pp. 32:1–32:14).

Castellan, S., Clairambault, P., Paquet, H., & Winskel, G. (2018). The concurrent game semantics of probabilistic PCF. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018*, Oxford, UK, July 09–12, 2018 (pp. 215–224).

Castellan, S., Clairambault, P., Rideau, S., & Winskel, G. (2017). Games and strategies as event structures. *Logical Methods in Computer Science, 13*(3).

Castellan, S., Clairambault, P., & Winskel, G. (2015). The parallel intensionally fully abstract games model of PCF. In *30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015*, Kyoto, Japan, July 6–10, 2015 (pp. 232–243).

Castellan, S., Clairambault, P., & Winskel, G. (2017). Observably deterministic concurrent strategies and intensional full abstraction for parallel-or. In *2nd International Conference on Formal Structures for Computation and Deduction, FSCD 2017*, September 3–9, 2017, Oxford, UK (pp. 12:1–12:16).

Castellan, S., & Yoshida, N. (2019). Causality in linear logic-full completeness and injectivity (unit-free multiplicative-additive fragment). In *Foundations of Software Science and Computation Structures—22nd International Conference, FOSSACS 2019, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2019*, Prague, Czech Republic, April 6–11, 2019 (pp. 150–168).

Chaudhuri, K., Miller, D., & Saurin, A. (2008). Canonical sequent proofs via multi-focusing. In *Fifth IFIP International Conference on Theoretical Computer Science—TCS 2008, IFIP 20th World Computer Congress, TC 1*, Foundations of Computer Science, September 7–10, 2008, Milano, Italy (pp. 383–396).

Clairambault, P., & de Visme, M. (2020). Full abstraction for the quantum λ-calculus. In *PACMPL* (Vol. 4, POPL).

Clairambault, P., Gutierrez, J., & Winskel, G. (2012). The winning ways of concurrent games. In *Proceedings of the 27th Annual IEEE Symposium on Logic in Computer Science, LICS 2012*, Dubrovnik, Croatia, June 25–28, 2012 (pp. 235–244).

Clairambault, P., & Harmer, R. (2010). Totality in arena games. *Annals of Pure and Applied Logic, 161*(5), 673–689.

Cockett, J. R. B., & Seely, R. A. G. (1997). Weakly distributive categories. *Journal of Pure and Applied Algebra, 114*(2), 133–173.

Conway, J. H. (2001). *On numbers and games* (2nd ed.). A K Peters.

Coquand, T. (1995). A semantics of evidence for classical arithmetic. *Journal of Symbolic Logic, 60*(1), 325–337.

Cuvillier, T. D. (2023). *Nominal models of linear logic* (Ph.D. thesis), Queen Mary University of London.

Ehrhard, T. (2012). The Scott model of linear logic is the extensional collapse of its relational model. *Theoretical Computer Science, 424*, 20–45.

Faggian, C., & Piccolo, M. (2009). Partial orders, event structures and linear strategies. In *Typed Lambda Calculi and Applications, 9th International Conference, TLCA 2009*, Brasilia, Brazil, July 1–3, 2009 (pp. 95–111).

Ghica, D. R., & Murawski, A. S. (2008). Angelic semantics of fine-grained concurrency. *Annals of Pure and Applied Logic, 151*(2–3), 89–114.

Girard, J.-Y. (1987). Linear logic. *Theoretical Computer Science, 50*, 1–102.

Girard, J.-Y. (1989). Geometry of interaction 1: Interpretation of system F. In *Studies in logic and the foundations of mathematics* (Vol. 127, pp. 221–260). Elsevier.

Girard, J.-Y. (1991). A new constructive logic: Classical logic. *Mathematical Structures in Computer Science, 1*(3), 255–296.

Girard, J.-Y. (2001). Locus solum: From the rules of logic to the logic of rules. *Mathematical Structures in Computer Science, 11*(3), 301–506.

Girard, J.-Y. (2017). Proof-nets: The parallel syntax for proof-theory. In *Logic and algebra* (pp. 97–124). Routledge.

Hyland, J. M. E., & Ong, C.-H.L. (2000). On full abstraction for PCF: I, II, and III. *Information and Computation, 163*(2), 285–408.

Hyland, J. M. E., & Ong, C.-H. L. (1993). Fair games and full completeness for multiplicative linear logic without the mix-rule.

Hyland, M. (1997). Game semantics. *Semantics and Logics of Computation, 14*, 131.

Hyland, M., & Schalk, A. (2003). Glueing and orthogonality for models of linear logic. *Theoretical Computer Science, 294*(1/2), 183–231.

James, L. (2001). A game semantics of idealized CSP. In S. D. Brookes & M. W. Mislove (Eds.), *Seventeenth Conference on the Mathematical Foundations of Programming Semantics, MFPS 2001*, Aarhus, Denmark, May 23–26, 2001. *Electronic Notes in Theoretical Computer Science* (Vol. 45, pp. 232–257) . Elsevier.

Joyal, A. (1977). Remarques sur la théorie des jeux à deux personnes. *Gazette des sciences mathématiques du Québec, 1*(4), 46–52.

Lamarche, F. (1992). Sequentiality, games and linear logic (Unpublished).

Laurent, O. A proof of the focusing property of linear logic. Available at https://perso.ens-lyon.fr/olivier.laurent/llfoc2.pdf

Laurent, O. (2002). *Étude de la polarisation en logique, Université Aix-Marseille II* (Ph.D. thesis), Thèse de Doctorat.

McCusker, G. (1998). *Games and full abstraction for a functional metalanguage with recursive types (CPHC/BCS distinguished dissertations)*. Springer.

Melliès, P.-A. (2005a). Asynchronous games 4: A fully complete model of propositional linear logic. In *20th IEEE Symposium on Logic in Computer Science (LICS 2005)*, June 26–29, 2005, Chicago, IL, USA (pp. 386–395).

Melliès, P.-A. (2005). Sequential algorithms and strongly stable functions. *Theoretical Computer Science, 343*(1–2), 237–281.

Melliès, P.-A. (2012). Game semantics in string diagrams. In *Proceedings of the 27th Annual IEEE Symposium on Logic in Computer Science, LICS 2012*, Dubrovnik, Croatia, June 25–28, 2012 (pp. 481–490).

Melliès, P.-A., & Mimram, S. (2007). Asynchronous games: Innocence without alternation. In *CONCUR 2007—Concurrency Theory, 18th International Conference, CONCUR 2007*, Lisbon, Portugal, September 3–8, 2007 (pp. 395–411).

Melliès, P.-A., & Tabareau, N. (2010). Resource modalities in tensor logic. *Annals of Pure and Applied Logic, 161*(5), 632–653.

Milner, R. (1977). Fully abstract models of typed λ-calculi. *Theoretical Computer Science, 4*(1), 1–22.

Mimram, S. (2008). *Sémantique des jeux asynchrones et réécriture 2-dimensionnelle (Asynchronous game semantics and 2-dimensional rewriting systems)* (Ph.D. thesis), Paris Diderot University, France.

Murawski, A. S., & Ong, C.-H.L. (2003). Exhausting strategies, joker games and full completeness for IMLL with unit. *Theoretical Computer Science, 294*(1/2), 269–305.

Rensink, A. (1992). Posets for configurations! In *CONCUR '92, Third International Conference on Concurrency Theory*, Stony Brook, NY, USA, August 24–27, 1992 (pp. 269–285).

Rideau, S., & Winskel, G. (2011). Concurrent strategies. In *Proceedings of the 26th Annual IEEE Symposium on Logic in Computer Science, LICS 2011*, June 21–24, 2011, Toronto, Ontario, Canada (pp. 409–418).

Sakayori, K., & Tsukada, T. (2017). A truly concurrent game model of the asynchronous π-calculus. In *Foundations of Software Science and Computation Structures—20th International Conference, FOSSACS 2017, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2017*, Uppsala, Sweden, April 22–29, 2017 (pp. 389–406).

Winskel, G. (1986). Event structures. In *Petri Nets: Central Models and Their Properties, Advances in Petri Nets 1986, Part II, Proceedings of an Advanced Course*, Bad Honnef, Germany, September 8–19, 1986 (pp. 325–392).

Winskel, G. (2009). Prime algebraicity. *Theoretical Computer Science, 410*(41), 4160–4168.

Winskel, G. (2012). Winning, losing and drawing in concurrent games with perfect or imperfect information. In *Logic and Program Semantics—Essays Dedicated to Dexter Kozen on the Occasion of His 60th Birthday* (pp. 298–317).

Chapter 11
The Game Semantics of Game Theory

Jules Hedges

Abstract We use a reformulation of compositional game theory to reunite game theory with game semantics, by viewing an open game as the System and its choice of contexts as the Environment. Specifically, the system is jointly controlled by $n \geq 0$ noncooperative players, each independently optimising a real-valued payoff. The goal of the system is to play a Nash equilibrium, and the goal of the environment is to prevent it. The key to this is the realisation that lenses (from functional programming) form a dialectica category, which have an existing game-semantic interpretation. In the second half of this paper, we apply these ideas to build a compact closed category of 'computable open games' by replacing the underlying dialectica category with a wave-style geometry of interaction category, specifically the Int-construction applied to the traced cartesian category of directed-complete partial orders.

This paper is dedicated to Samson Abramsky. I will leave it to others to tell the story of Samson's influence on science as a whole, and restrict myself to his influence on my own small field of *applied category theory*. His work with Bob Coecke in the early 2000s that founded categorical quantum mechanics (Abramsky & Coecke, 2004) is arguably the beginning of applied category theory in the sense that the term has been used recently. From there a direct line can be *traced* back via (Abramsky & Coecke, 2002) to Samson's earlier work on the geometry of interaction (Abramsky & Jagadeesan, 1994b; Abramsky, 1996), which itself connects directly to his work on game semantics and concurrency.

Going forwards, Samson and Bob's research group in Oxford slowly transformed from an exclusively quantum computation group, via the expansion into linguistics (Coecke et al., 2010), into a place where anybody interested in applications of category theory could make a home: I spent three years there are a postdoc, and Brendan Fong's influential D.Phil thesis on categorical systems theory (Fong, 2016) was written there for example. Samson himself seems particularly interested in applications of category theory to microeconomics, for example social choice theory (Abram-

J. Hedges (✉)
Department of Computer & Information Sciences, Livingstone Tower, 26 Richmond Street, Glasgow G1 1XH, UK
e-mail: jules.hedges@strath.ac.uk

A. Palmigiano and M. Sadrzadeh (eds.), *Samson Abramsky on Logic and Structure in Computer Science and Beyond*, Outstanding Contributions to Logic 25,
https://doi.org/10.1007/978-3-031-24117-8_11

sky, 2015) and game theory (Abramsky & Winschel, 2017). Samson's current work on contextuality began with his collaboration with Adam Brandenburger, another microeconomist (Abramsky & Brandenburger, 2011).

Given this, an attempt to connect game semantics with game theory is an obvious choice for this volume. After writing it I discovered that I was not the first to have the same thought: no fewer than two chapters in Samson's previous festschrift (Hankin & Malacaria, 2013; van Benthem, 2013) also concern interactions between game semantics and game theory. I interpret this to mean it's a good idea.

11.1 Introduction

Although the mathematics of games shares a common ancestor in Zermelo's work on backward induction, it split early on into two subjects that are essentially disjoint: game theory and game semantics. Game theory is the applied study of modelling real-world interacting agents, for example in economics or artificial intelligence. Game semantics, by contrast, uses agents to model—this time in the sense of *semantics* rather than *mathematical modelling*—situations in which a system interacts with an environment, but neither would usually be thought of as agents in a philosophical sense. On a technical level, game semantics not only restricts to the two-player zero-sum case, but moreover promotes one of the players to be *the* Player, and demotes the other to mere Opponent. This induces a deep logical duality that pervades game semantics, apparently destroying any hope of bridging the gap to game theory, which typically involves n players treated symmetrically.

Compositional game theory (Hedges, 2016; Ghani et al., 2018), as its name suggests, is an attempt to introduce the principle of compositionality into game theory, motivated by practical concerns about modelling large (for example economic) systems. It is loosely inspired by game semantics, as well as categorical quantum mechanics (Abramsky & Coecke, 2004; Coecke & Kissinger, 2017) and much recent work in applied category theory (e.g. Fong, 2016). On a technical level, game semantics involves (typically monoidal) categories in which games are the objects and strategies (with various conditions) are the morphisms, whereas open games form the *morphisms* of a monoidal category. This means that open games can be denoted by string diagrams, which is invaluable for working with them in practice. As with other categories of open systems, ordinary "closed" games are recovered as *scalars*, or endomorphisms of the monoidal unit (see Abramsky, 2005), and depicted as string diagrams with trivial boundary.

Central to understanding open games is the concept of a *context*, which is a compressed representation of a game-theoretic situation in which an open game can be played. Whereas an ordinary game has a set of strategy profiles and a subset of those which are Nash equilibria, in an open game the equilibria depend on the context. This is the key to reuniting game theory and game semantics: we ignore the linguistic coincidence of the term *player*, and instead view an open game as the System and the choice of contexts as the Environment.

The essence of this idea is already contained in the following quote from the introduction of (Abramsky, 1997): “If Tom, Tim and Tony converse in a room, then from Tom’s point of view, he is the System, and Tim and Tony form the Environment; while from Tim’s point of view, he is the System, and Tom and Tony form the Environment.” The view of open games presented in this paper makes this precise when Tom, Tim and Tony are players in a noncooperative game.

In Hedges (2018) open games were reformulated in terms of *lenses* from functional programming (Pickering et al., 2017). This was extremely useful as a technical trick, but lenses are usually used as destructive update operators on data structures and it is unclear what they have to do with game theory, if anything. The key was a comment by Dusko Pavlovic to the author that the category of lenses $\mathfrak{L}$ is a dialectica category (de Paiva, 1991); combined with a game-*semantic* view of dialectica categories (Blass, 1991) we can see open games in their true form: as an interleaving of game theory and game semantics.

Specifically we find that an open game is a dialogue of a particular sort played between a system and its environment. The system is jointly controlled by $n \geq 0$ noncooperative players, each independently optimising a real-valued payoff. The winning condition turns out to be Nash equilibrium: the goal of the system is to play an equilibrium, and the goal of the environment is to prevent it. Specifically, an open game consists of three pieces of data: a set Σ of strategy profiles, a labelling function $\Sigma \to \{P\text{-strategies}\}$, and a winning (for P) relation $\mathbf{E} \subseteq \Sigma \times \{O\text{-strategies}\}$.

Taking a step back, this is a rare example of a cross-link in the family tree of the mathematics of games. From the common ancestor in Zermelo’s theorem (Schwalbe & Walker, 2001) there was an almost immediate split, with little contact or commonality between the branches. One branch led to game theory via (von Neumann & Morgenstern, 1944; Nash, 1951), and eventually found its home as a central tool in microeconomics (Osbourne & Rubinstein, 1994), as well as applications in biology and computer science. The other branch concerned applications in logic and focussed on two-player zero-sum games, including dialogical semantics (Lorenzen & Lorenz, 1978), Borel games (Martin, 1975) and eventually game semantics in its modern sense (Abramsky & Jagadeesan, 1994a; Hyland & Ong, 2000; Abramsky et al., 2000; Abramsky & McCusker, 1999).

Perhaps the only systematic attempt to bridge the two branches is the work of van Benthem and collaborators on game logics (van Benthem, 2014). Other examples of more ad-hoc bridges can be found for example in Hankin and Malacaria (2013), Le Roux (2014), Gutierrez and Wooldridge (2014). The work of Pavlovic (2009), which is not specifically about game semantics, is perhaps the most closely related to this paper.

In the second half of this paper, we apply these ideas to build a compact closed category of ‘computable open games’ by replacing the underlying dialectica category with a wave-style geometry of interaction category, specifically the Int-construction applied to the traced cartesian category of directed-complete partial orders. (The category of directed-complete partial orders and Scott-continuous maps is a standard setting for the semantics of possibly-nonterminating recursive computations.) Ultimately we rely on the following transport of structure result:

Proposition 1 *Let $\mathcal{C}$ be a compact closed category, $\mathcal{D}$ a symmetric monoidal category and $F : \mathcal{C} \to \mathcal{D}$ a strict symmetric monoidal functor that is bijective on objects. Then $\mathcal{D}$ can be given a compact closed structure, with duals given by $F(X)^* = F(X^*)$, units by $\eta_{F(X)} = F(\eta_X)$ and counits by $\varepsilon_{F(X)} = F(\varepsilon_X)$.*

Proof The assumption that F is bijective on objects means that every object of $\mathcal{D}$ is uniquely assigned a dual, unit and counit. It is simple to check the yanking equations (Kelly & Laplaza, 1980):

$$\begin{aligned}
&\rho_{F(X)} \circ (\mathrm{id}_{F(X)} \otimes \varepsilon_{F(X)}) \circ a_{F(X),F(X)^*,F(X)} \circ (\eta_{F(X)} \otimes \mathrm{id}_{F(X)}) \circ \lambda^{-1}_{F(X)} \\
&= F(\rho_X) \circ (F(\mathrm{id}_X) \otimes F(\varepsilon_X)) \circ F(a_{X,X^*,X}) \circ (F(\eta_X) \otimes F(\mathrm{id}_X)) \circ F(\lambda^{-1}_X) \\
&= F(\rho_X \circ (\mathrm{id}_X \otimes \varepsilon_X) \circ a_{X,X^*,X} \circ (\eta_X \otimes \mathrm{id}_X) \circ \lambda^{-1}_X) \\
&= F(\mathrm{id}_X) = \mathrm{id}_{F(X)}
\end{aligned}$$

and similarly for the other equation. □

The hypotheses of this theorem are already satisfied by a particular functor $\mathfrak{L} \to$ **OG** that identifies $\mathfrak{L}$ with the subcategory of *zero-player open games*. Thus it suffices to replace the source category with one that is compact closed, while preserving the hypotheses (and the game-theoretic interpretation).

We end with a worked example, a 'paradoxical' variant of matching pennies where both players have the ability and incentive to play a strategy that is contingent on the other's move - something that appears causally absurd, and can result in the play deadlocking while each player waits for the other to move first.

11.2 Dialogues

While the name 'dialectica' should bring to mind dialogues in the tradition of philosophical logic (for example via Hegel's dialectics), this is apparently a coincidence. The dialectica interpretation is named after the journal Dialectica, who published Gödel's paper in their Paul Bernays festschrift (Gödel, 1958). But the dialectica interpretation does have a very dialectical feeling to it.

The game semantic viewpoint on Gödel's dialectica interpretation (Avigad & Feferman, 1998) and de Paiva's dialectica categories (de Paiva, 1991) was described in Blass' paper that first introduced game semantics (Blass, 1991). In this section we recall this viewpoint in detail.

We first introduce a category $\mathfrak{L}$ of dialogues and strategies, which is the dialectica category over an inconsistent (1-valued) logic.

An object of $\mathfrak{L}$ is a 2-stage dialogue $X^+; S^-$ in which first the System chooses $x : X$, and then the Environment chooses $s : S$, where X and S are any sets. This breaks a common requirement in game semantics that the Environment moves first. We denote the dialogue $X^+; S^-$ by $\binom{X}{S}$.

Notice that the set of P-strategies for $\binom{X}{S}$ is X, and the set of O-strategies is S^X, the set of functions $X \to S$.

We introduce a monoidal product operator given by synchronous parallel play. Specifically, the parallel play of $\binom{X}{S}$ and $\binom{Y}{R}$ is the 4-stage dialogue $X^+; Y^+; R^-; S^-$. This peculiar ordering of moves, with the right-hand dialogue being played in the middle of the left-hand dialogue, is characteristic of dialectica. This 4-stage dialogue is strategically equivalent to the 2-stage dialogue $(X \times Y)^+; (R \times S)^-$, so we set $\binom{X}{S} \otimes \binom{Y}{R} = \binom{X \times Y}{R \times S}$.

Next, given objects $\binom{X}{S}$ and $\binom{Y}{R}$, we consider the same 4-stage dialogue but with the players interchanged in the former. That is, we consider the dialogue $X^-; Y^+; R^-; S^+$. We consider this to be $\binom{Y}{R}$ played relative to $\binom{X}{S}$, and denote it by $\binom{X}{S} \to \binom{Y}{R}$.

The set of P-strategies for $\binom{X}{S} \to \binom{Y}{R}$ is $Y^X \times S^{X \times R}$, or isomorphically $(Y \times S^R)^X$. The set of O-strategies is $X \times R^Y$. We denote the set of P-strategies for $\binom{X}{S} \to \binom{Y}{R}$ by $\mathfrak{L}\left(\binom{X}{S}, \binom{Y}{R}\right)$. As the notation suggests, these are the morphisms of $\mathfrak{L}$.

Given an object $\binom{X}{S}$, there is a *copycat* P-strategy for $\binom{X}{S} \to \binom{X}{S} = X^-; X^+; S^-; S^+$. As an element of $X^X \times S^{X \times S}$ it is the pair consisting of the identity and the projection. This is the identity morphism for $\binom{X}{S}$. Following (Abramsky, 1997) we denote this strategy by a string diagram:

$$\begin{tikzcd}
X^- \arrow[r, bend left] & X^+ & S^- \arrow[r, bend left] & S^+ \\
\uparrow & \downarrow & \uparrow & \downarrow
\end{tikzcd}$$

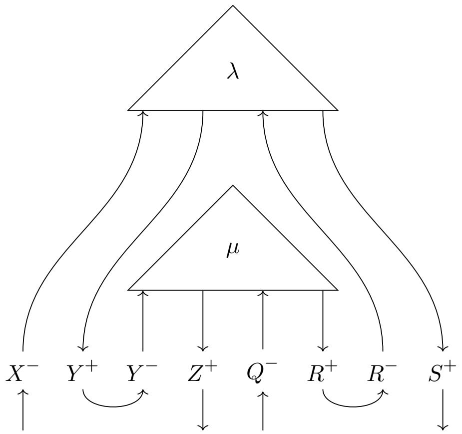
A major theme of this paper is that we take this notation seriously, pushing it far beyond what was originally intended. While it is common for papers to contain a caveat that string diagrams are 'officially' informal pending a coherence theorem, in this case they are far more informal than usual: it is completely unclear what category they live in, or exactly which topological moves they are invariant under. While there is an immediate surface similarity to grammatical reductions in pregroups (Preller & Lambek, 2007; Coecke et al., 2010), there appears to be a much deeper connection to string diagrams in the bicategory of finite product categories, Tambara modules (profunctors compatible with the cartesian product) and natural transformations (Boisseau, 2020) (see also Pastro and Street 2008), something we leave for later work.

Now suppose we are given P-strategies λ for $\binom{X}{S} \to \binom{Y}{R} = X^-; Y^+; R^-; S^+$ and μ for $\binom{Y}{R} \to \binom{Z}{Q} = Y^-; Z^+; Q^-; R^+$. There is a way to combine them to produce a P-strategy $\mu \circ \lambda$ for $\binom{X}{S} \to \binom{Z}{Q} = X^-; Z^+; Q^-; S^+$. Namely, P *simulates* playing the two together with O playing a copycat strategy for the middle moves. That is, she simulates the 8-stage dialogue

$$X^-; Y^+; Y^-; Z^+; Q^-; R^+; R^-; S^+$$

with the assumption that O uses a copycat strategy for the moves Y^- and R^-. By then hiding the Y and R moves we get a P-strategy for the required 4-stage dialogue.

We denote this P-strategy by the following string diagram:



Whereas the cap denotes a copycat P-strategy, the cup denotes a copycat O-strategy.

A little calculation shows that if λ is given by $v_\lambda : X \to Y$ and $u_\lambda : X \times R \to S$, and μ is given by $v_\mu : Y \to Z$ and $u_\mu : Y \times Q \to R$, then the composite is given by

$$v_{\mu\circ\lambda}(x) = v_\mu(v_\lambda(x))$$

and

$$u_{\mu\circ\lambda}(x, q) = u_\lambda(x, u_\mu(v_\lambda(x), q))$$

It is routine to check that this is associative, with identities given by copycat. Thus $\mathfrak{L}$ is indeed a category. These equations are commonly known in functional programming as composition of lenses (Foster et al., 2007; Gibbons & Stevens, 2016).

Given this category structure we can also make $\otimes$ into a genuine symmetric monoidal product. Given P-strategies $\lambda : \binom{X_1}{S_1} \to \binom{Y_1}{R_1}$ and $\mu : \binom{X_2}{S_2} \to \binom{Y_2}{R_2}$, we can combine them to produce a P-strategy $\lambda \otimes \mu : \binom{X_1 \times X_2}{S_2 \times S_1} \to \binom{Y_1 \times Y_2}{R_2 \times R_1}$.

Finally, we notice that all of the above can be generalised to any base category $\mathcal{C}$ with finite products, replacing sets and functions, yielding a category $\mathfrak{L}(\mathcal{C})$ whose morphisms are strategies internal to $\mathcal{C}$. Specifically, we set

$$\mathfrak{L}(\mathcal{C})\left(\begin{pmatrix} X \\ S \end{pmatrix}, \begin{pmatrix} Y \\ R \end{pmatrix}\right) = \mathcal{C}(X, Y) \times \mathcal{C}(X \times R, S)$$

(By writing it this way, we do not need to assume that $\mathcal{C}$ is cartesian closed.) The category we have been considering so far is $\mathfrak{L} = \mathfrak{L}(\mathbf{Set})$.

Proposition 2 *For any category $\mathcal{C}$ with finite products, $\mathfrak{L}(\mathcal{C})$ is a symmetric monoidal category.*

There is a much less obvious generalisation of $\mathfrak{L}(\mathcal{C})$ when $\mathcal{C}$ is only a monoidal category (Riley, 2018), but we will not need it in this paper.

11.3 Negation and O-strategies

To talk about open games, we need to talk explicitly about O-strategies in a dialogue. However, the categorical structure of $\mathfrak{L}$ is built on P-strategies. In turns out, however, that we can use P-strategies to talk about O-strategies, in a way that respects composition.

The monoidal unit of $\mathfrak{L}$ is the trivial game $I = \binom{1}{1} = 1^+; 1^-$. The dialogue $I \to \binom{X}{S}$ is $1^-; X^+; S^-; 1^+$, which is strategically equivalent to $\binom{X}{S}$. Thus the set of P-strategies for $I \to \binom{X}{S}$ is X.

If we fix a P-strategy $h : X$ for $\binom{X}{S}$ and another P-strategy $\lambda : \binom{X}{S} \to \binom{Y}{R}$, we can compose them to yield a P-strategy $\lambda \circ h$ for $\binom{Y}{R}$, by

$$I \xrightarrow{h} \begin{pmatrix} X \\ S \end{pmatrix} \xrightarrow{\lambda} \begin{pmatrix} Y \\ R \end{pmatrix}$$

Succinctly, there is a functor $\mathbb{V} : \mathfrak{L} \to \mathbf{Set}$ taking every object to its set of P-strategies, namely the covariant functor represented by I. Explicitly, $\mathbb{V}\binom{X}{S} = X$ and $\mathbb{V}(\lambda) = v_\lambda$.

On the other hand, the dialogue $\binom{X}{S} \to I$ is $X^-; 1^+; 1^-; S^+$, which is equivalent to $X^-; S^+$. This is not an object, but is $\binom{X}{S}$ with players interchanged. Thus the set of P-strategies for $\binom{X}{S} \to I$ is equal to the set of O-strategies for $\binom{X}{S}$, namely S^X.

Given an O-strategy k for $\binom{Y}{R}$ and a P-strategy $\lambda : \binom{X}{S} \to \binom{Y}{R}$, we obtain an O-strategy $k \circ \lambda$ for $\binom{X}{S}$ by

$$\begin{pmatrix} X \\ S \end{pmatrix} \xrightarrow{\lambda} \begin{pmatrix} Y \\ R \end{pmatrix} \xrightarrow{k} I$$

In this, O 'hijacks' P's strategy to produce an element of S, since λ is a P-strategy for a dialogue in which P plays the role of O in $\binom{X}{S}$.

Succinctly, there is a functor $\mathbb{K} : \mathfrak{L}^{\mathrm{op}} \to \mathbf{Set}$ taking every object to its set of O-strategies, namely the contravariant functor represented by I. In the terminology of categorical quantum mechanics, P-strategies are *states* and O-strategies are *effects*.

Since an O-strategy for $\binom{X}{S} \to \binom{Y}{R}$ is precisely an element of $X \times Y^R$, it can be equivalently seen as a P-strategy for $\binom{X}{S}$ and an O-strategy for $\binom{Y}{R}$. This defines a functor $\overline{\mathfrak{L}} : \mathfrak{L} \times \mathfrak{L}^{\mathrm{op}} \to \mathbf{Set}$, namely

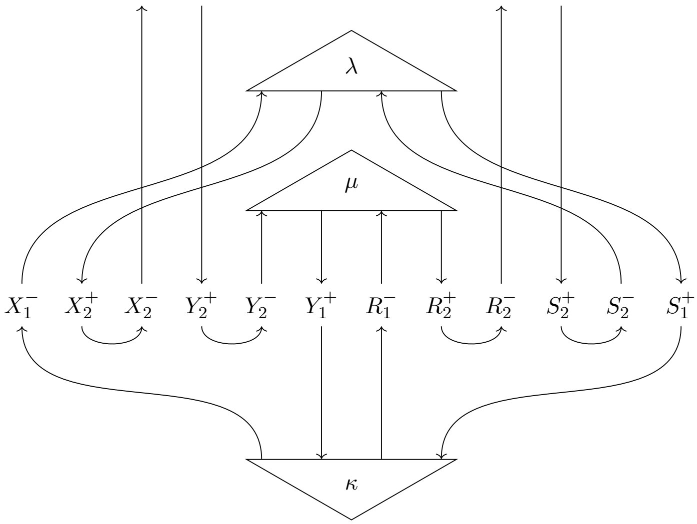
$$\mathfrak{L} \times \mathfrak{L}^{\mathrm{op}} \xrightarrow{\mathbb{V}\times\mathbb{K}} \mathbf{Set} \times \mathbf{Set} \xrightarrow{\times} \mathbf{Set}$$

On objects, it is concretely given by $\overline{\mathfrak{L}}\left(\binom{X}{S}, \binom{Y}{R}\right) = X \times R^Y$, or more generally over a category $\mathcal{C}$ with finite products, $\overline{\mathfrak{L}(\mathcal{C})}\left(\binom{X}{S}, \binom{Y}{R}\right) = \mathcal{C}(1, X) \times \mathcal{C}(Y, R)$.

Given an O-strategy $\kappa = (h, k)$ for $\binom{X_1}{S_1} \to \binom{Y_1}{R_1}$, a P-strategy $\lambda : \binom{X_1}{S_1} \to \binom{X_2}{S_2}$ and a P-strategy $\mu : \binom{Y_2}{R_2} \to \binom{Y_1}{R_1}$, we obtain an O-strategy $\overline{\mathfrak{L}}(\lambda, \mu)(h, k) = (\lambda \circ h, k \circ \mu)$ for $\binom{X_2}{S_2} \to \binom{Y_2}{R_2}$. This is the O-strategy for the dialogue

$$X_1^-; X_2^+; X_2^-; Y_2^+; Y_2^-; Y_1^+; R_1^-; R_2^+; R_2^-; S_2^+; S_2^-; S_1^+$$

with appropriately hidden copycat moves, as given by the string diagram



11.4 Open Games

We can now give an equivalent definition of open games (Hedges, 2016; Ghani et al., 2018) in terms of dialogues. The treatment in this section and the next will be conceptual, with examples deferred until the end of Sect. 11.6 after building up some theory.

An open game $\binom{X}{S} \to \binom{Y}{R}$ is in one dimension a dialogue played between a System and an Environment, and in another dimension it is a non-cooperative game in the sense of economics, in which several players jointly control the System while independently optimising payoffs.

An open game $\mathcal{G} : \binom{X}{S} \to \binom{Y}{R}$ is defined by three pieces of data:

- A set $\Sigma_\mathcal{G}$ of strategy profiles
- A labelling function $\mathcal{G}_- : \Sigma_\mathcal{G} \to \mathfrak{L}\left(\binom{X}{S}, \binom{Y}{R}\right)$, by which every element $\sigma : \Sigma_\mathcal{G}$ labels a P-strategy $\mathcal{G}_\sigma$ for the 4-stage dialogue $\binom{X}{S} \to \binom{Y}{R}$
- A winning condition, which is a relation between $\Sigma_\mathcal{G}$ and the set of O-strategies of $\binom{X}{S} \to \binom{Y}{R}$, namely $|\mathcal{G}| \subseteq \Sigma_\mathcal{G} \times \overline{\mathfrak{L}}\left(\binom{X}{S}, \binom{Y}{R}\right)$.

We write $|\mathcal{G}|^\sigma_\kappa$ for $(\sigma, \kappa) \in |\mathcal{G}|$. We say that σ is a *winning strategy profile* if $|\mathcal{G}|^\sigma_\kappa$ for all O-strategies $\kappa : \overline{\mathfrak{L}}\left(\binom{X}{S}, \binom{Y}{R}\right)$.

We interpret $|\mathcal{G}|$ as an equilibrium condition. That is, from the dialogue perspective the goal of the System is to reach equilibrium and the goal of the Environment is to prevent equilibrium. In real examples there is rarely a winning strategy profile, and so we focus on $|\mathcal{G}|$ as a binary relation, or ask about winning strategy profiles for the System against a fixed O-strategy.

From the dialogue perspective, the order of play in an open game $\binom{X}{S} \to \binom{Y}{R}$ is:

1. The Environment chooses an initial state of the game from X
2. The System chooses the final state of the game from Y
3. The Environment chooses payoffs for the System from R
4. The System chooses payoffs for the Environment from S

An O-strategy is a pair $\kappa = (h, k)$ where $h : X$ and $k : Y \to R$. The *history* h determines the *initial state* of the game. The *continuation* k determines the payoffs for System given the final state. The pair (h, k) completely determines the strategic context in which the players that make up System make their choices, reducing the open game to an ordinary normal-form game. For this reason, we also call an O-strategy a *context* for the open game.

We only need two families of examples of open games to generate a large family of examples, corresponding roughly to extensive-form games, using the sequential and parallel play operators we will define in the next section. These two generating families are the *zero-player open games* and the *decisions*, which could loosely be called *one-player open games*.

The zero-player open games $\binom{X}{S} \to \binom{Y}{R}$ are in bijection with the P-strategies $\lambda : \binom{X}{S} \to \binom{Y}{R}$, and correspond to the situation in which the System has no strategic choices but always follows the strategy λ like an automaton. Specifically, the zero-player open game λ is defined by:

- The set of strategy profiles is the singleton $\Sigma_\lambda = \{*\}$, where $*$ is a token representing the P-strategy λ
- The labelling function is $\lambda_* = \lambda$
- $*$ is a winning strategy profile, that is, $|\lambda|^*_\kappa$ for all O-strategies κ

Perhaps the only surprising part of this definition is that $*$ is a winning strategy profile. The reason for this ultimately comes down to agreeing with Nash equilibrium on real examples. Nash equilibrium is a *negative* definition: a strategy profile should fail to be a Nash equilibrium if some particular player has positive incentive to deviate from it. Since there are no players in λ, $*$ is declared a Nash equilibrium by default.

The second family of examples are the decisions. There is one such open game $\mathcal{D} = \mathcal{D}_{Y|X} : \binom{X}{1} \to \binom{Y}{\mathbb{R}}$ for every nonempty set X and Y, representing a single agent's choice from Y given an observation from X. In this game:

1. The Environment chooses an initial state from X
2. The (now unique) Player chooses a final state from Y
3. The Environment chooses a payoff from $\mathbb{R}$

The winning condition of this game is *intensional* by being a property of the *strategies* of both Player and Environment, and cannot be written in terms of the play alone. This is because optimality in game theory is a counterfactual: *if* the System had made a different choice then the resulting payoff *would have* been lower.

Observe that a P-strategy for this game is a function $\sigma : X \to Y$, and we choose the set of strategy profiles $\Sigma_{\mathcal{D}_{Y|X}}$ to be precisely the set of P-strategies. An O-strategy is a pair (h, k) where $h : X$ and $k : Y \to \mathbb{R}$. By definition, the Player wins this game iff $\sigma(h) \in \arg\max(k)$, that is to say, if $k(\sigma(h)) \geq k(y)$ for all $y : Y$.

This is a small shift in perspective that is quite natural from the perspective of game semantics. In game theory there is no concept of *winning*, only optimality and equilibrium. Declaring a player to have *won* if they make an optimal choice may not be meaningful as game theory, but it is appropriate terminology when combining game theory with game semantics.

Writing this out:

- The set of strategy profiles is $\Sigma_{\mathcal{D}} = Y^X$
- The labelling function takes $\sigma : X \to Y$ to itself considered as a P-strategy $\mathcal{D}_\sigma : \binom{X}{1} \to \binom{Y}{\mathbb{R}}$, via the bijection $\mathfrak{L}\left(\binom{X}{1}, \binom{Y}{\mathbb{R}}\right) \cong Y^X$
- The winning condition is $|\mathcal{D}|^\sigma_{h,k}$ iff $\sigma(h) \in \arg\max(k)$

11.5 Composing Open Games

We can make open games into the morphisms of a symmetric monoidal category. The two composition operators, categorical composition and tensor product, correspond to *sequential play* and *simultaneous play*.

Suppose we are given open games $\mathcal{G} : \binom{X}{S} \to \binom{Y}{R}$ and $\mathcal{H} : \binom{Y}{R} \to \binom{Z}{Q}$. The sequential composition $\mathcal{H} \circ \mathcal{G} : \binom{X}{S} \to \binom{Z}{Q}$ has set of strategy profiles $\Sigma_{\mathcal{H}\circ\mathcal{G}} = \Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{H}}$. Informally, the idea is that $\mathcal{G}$ and $\mathcal{H}$ are each associated with sets G, H of decisions. Each decision $g \in G, h \in H$ has an associated set Σ_g, Σ_h of strategies, and the set of strategy profiles in each case should be thought of as the set of tuples of strategies, one for each decision: $\Sigma_{\mathcal{G}} = \prod_{g\in G} \Sigma_g$ and $\Sigma_{\mathcal{H}} = \prod_{h\in H} \Sigma_h$. The set of decisions made in a composite game is the disjoint union of the decisions made in the components, and so $\Sigma_{\mathcal{H}\circ\mathcal{G}} = \prod_{g\in G+H} \Sigma_g = \prod_{g\in G} \Sigma_g \times \prod_{h\in H} \Sigma_h = \Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{H}}$.

The labelling function for a sequential composition can be defined using the underlying composition in $\mathfrak{L}$: $(\mathcal{H} \circ \mathcal{G})_{\sigma,\tau} = \mathcal{H}_\tau \circ \mathcal{G}_\sigma$.

In order to define the winning condition of $\mathcal{H} \circ \mathcal{G}$, we must modify a context for $\mathcal{H} \circ \mathcal{G}$ into contexts for $\mathcal{G}$ and $\mathcal{H}$. We can do this using the fact that $\overline{\mathfrak{L}}$ is a functor, together with the fact that we have strategy profiles for $\mathcal{G}$ and $\mathcal{H}$ available. A strategy profile (σ, τ) for $\mathcal{H} \circ \mathcal{G}$ is winning (that is to say, a Nash equilibrium) against the O-strategy κ iff σ is winning in $\mathcal{G}$ against the O-strategy $\overline{\mathfrak{L}}(\mathrm{id}, \mathcal{H}_\tau)(\kappa)$, and τ is winning in $\mathcal{H}$ against the O-strategy $\overline{\mathfrak{L}}(\mathcal{G}_\sigma, \mathrm{id})(\kappa)$. That is to say,

$$|\mathcal{H} \circ \mathcal{G}|_\kappa^{\sigma,\tau} \iff |\mathcal{G}|^\sigma_{\overline{\mathfrak{L}}(\mathrm{id},\mathcal{H}_\tau)(\kappa)} \wedge |\mathcal{H}|^\tau_{\overline{\mathfrak{L}}(\mathcal{G}_\sigma,\mathrm{id})(\kappa)}$$

This makes open games into the morphisms of a category (or, more properly, the 1-cells of a bicategory).

Next we consider simultaneous play. Given open games $\mathcal{G} : \binom{X_1}{S_1} \to \binom{Y_1}{R_1}$ and $\mathcal{H} : \binom{X_2}{S_2} \to \binom{Y_2}{R_2}$, we combine them to form an open game

$$\mathcal{G} \otimes \mathcal{H} : \begin{pmatrix} X_1 \times X_2 \\ S_2 \times S_1 \end{pmatrix} \to \begin{pmatrix} Y_1 \times Y_2 \\ R_2 \times R_1 \end{pmatrix}$$

As before the strategy profiles of $\mathcal{G} \otimes \mathcal{H}$ are pairs, $\Sigma_{\mathcal{G}\otimes\mathcal{H}} = \Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{H}}$, for the same reason as before: we take the disjoint union of the set of decisions. The strategy profile (σ, τ) labels the synchronous parallel play of $\mathcal{G}_\sigma$ and $\mathcal{H}_\tau$, that is, $(\mathcal{G} \otimes \mathcal{H})_{\sigma,\tau} = \mathcal{G}_\sigma \otimes \mathcal{H}_\tau$.

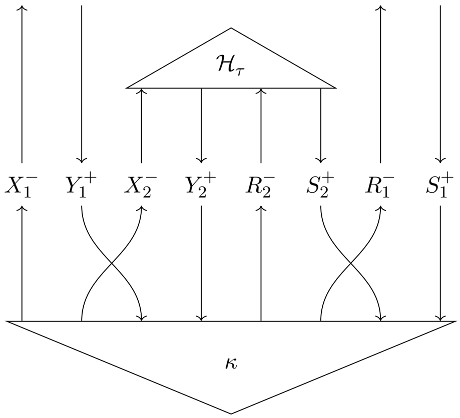
In order to define the winning condition for $\mathcal{G} \otimes \mathcal{H}$ we need to do some more work.

Given strategy profiles $\sigma : \Sigma_{\mathcal{G}}$ and $\tau : \Sigma_{\mathcal{H}}$, and an O-strategy κ for $\binom{X_1\times X_2}{S_2\times S_1} \to \binom{Y_1\times Y_2}{R_2\times R_1}$, we need to 'project' κ to $\mathcal{G}$ and $\mathcal{H}$'s view of it, as O-strategies for $\binom{X_1}{S_1} \to \binom{Y_1}{R_1}$ and $\binom{X_2}{S_2} \to \binom{Y_2}{R_2}$.

We can indeed do this. To produce an O-strategy for $\binom{X_1}{S_1} \to \binom{Y_1}{R_1}$, consider the dialogue

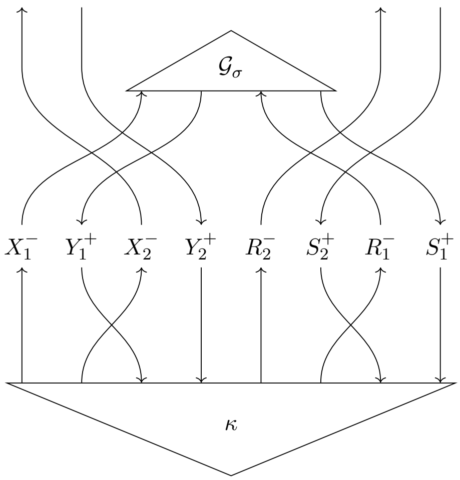
$$X_1^-;\, Y_1^+;\, X_2^-;\, Y_2^+;\, R_2^-;\, S_2^+;\, R_1^-;\, S_1^+;$$

with the strategy



We call this O-strategy $\mathcal{H}_\tau/\kappa$. When $\kappa = ((h_1, h_2), k)$ for $k : Y_1 \times Y_2 \to R_2 \times R_1$, we write $\mathcal{H}_\tau/\kappa = (h_1, k_1^{h_2,\mathcal{H}_\tau})$. Concretely, the new continuation is $k_1^{h_2,\mathcal{H}_\tau}(y_1) = k(y_1, v_{\mathcal{H}_\tau}(h_2))_2$.

Similarly, we can produce an O-strategy $\mathcal{G}_\sigma \setminus \kappa$ for $\binom{X_2}{S_2} \to \binom{Y_2}{R_2}$ by considering the same dialogue with the strategy



When $\kappa = ((h_1, h_2), k)$ we write $\mathcal{G}_\sigma \setminus \kappa = (h_2, k_2^{h_1,\mathcal{G}_\sigma})$, where $k_2^{h_1,\mathcal{G}_\sigma}(y_2) = k(v_{\mathcal{G}_\sigma}(h_1), y_2)_1$.

With this, we can finally define the winning condition for $\mathcal{G} \otimes \mathcal{H}$: The strategy profile (σ, τ) is winning against κ in $\mathcal{G} \otimes \mathcal{H}$ iff σ is winning against $\mathcal{H}_\tau/\kappa$ in $\mathcal{G}$ and τ is winning against $\mathcal{G}_\sigma \setminus \kappa$ in $\mathcal{H}$, that is to say,

$$|\mathcal{G} \otimes \mathcal{H}|_{\kappa}^{\sigma,\tau} \iff |\mathcal{G}|^{\sigma}_{\mathcal{H}_{\tau}/\kappa} \wedge |\mathcal{H}|^{\tau}_{\mathcal{G}_{\sigma}\backslash\kappa}$$

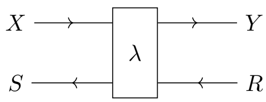
Proposition 3 *There is a symmetric monoidal (bi)category* **OG** *whose objects are pairs of sets and morphisms are open games.*

Although **OG** should properly be thought of as a bicategory with 2-cells given by appropriately compatible functions between sets of strategy profiles, this is an uninteresting technicality and we will instead quotient out these 2-cells, treating open games as defined only up to compatible bijections of strategy profiles. The details of this can be found in Hedges (2018).

11.6 Picturing Open Games

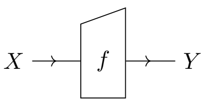
Since open games are the morphisms of a monoidal category, we can depict them by string diagrams, and in fact this turns out to be invaluable for working with them in practice. As a special case of this we also obtain string diagrams for the monoidal category of P-strategies, which are equivalently the wide subcategory of zero-player open games. These diagrams should not be confused with the (less well understood) diagrams for dialogues that have appeared so far in this paper, which are very different, although to some extent it is possible to translate between them. This section contains nothing new, but is included from (Ghani et al., 2018) for completeness.

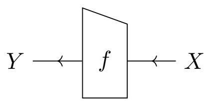
A P-strategy $\lambda : \binom{X}{S} \to \binom{Y}{R}$, viewed as a zero-player open game, is depicted as a string diagram



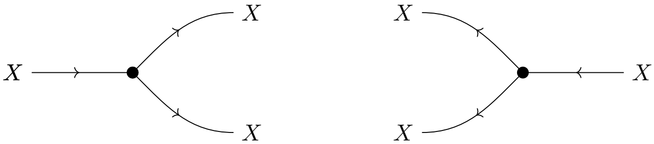
We regard the forwards-oriented strings labelled X and Y as respectively representing the objects $\binom{X}{1}$ and $\binom{Y}{1}$, and the backwards-oriented strings labelled R and S are respectively representing the objects $\binom{1}{R}$ and $\binom{1}{S}$. Thus we are implicitly using the natural isomorphisms $\binom{X}{1} \otimes \binom{1}{S} = \binom{X\times 1}{S\times 1} \cong \binom{X}{S}$ and $\binom{Y}{1} \otimes \binom{1}{R} = \binom{Y\times 1}{R\times 1} \cong \binom{Y}{R}$.

As special cases of this, a function $f : X \to Y$ can be regarded as a P-strategy and as a zero-player open game either covariantly as $f : \binom{X}{1} \to \binom{Y}{1}$, or contravariantly as $f^* : \binom{1}{Y} \to \binom{1}{X}$. We depict these respectively with the diagrams

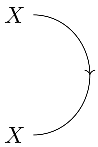




As a further special case, the liftings $\Delta_X : \binom{X}{1} \to \binom{X \times X}{1}$ and $\Delta_X^* : \binom{1}{X \times X} \to \binom{1}{X}$ of the copy functions are given the special syntax



For any set X there is a copycat P-strategy $\varepsilon_X : \binom{X}{X} \to I$, arising from the copycat O-strategy for $X^+; X^-$ via the representation $\mathbb{K} \cong \mathfrak{L}(-, I)$. We depict this P-strategy and the corresponding zero player open game by a cap

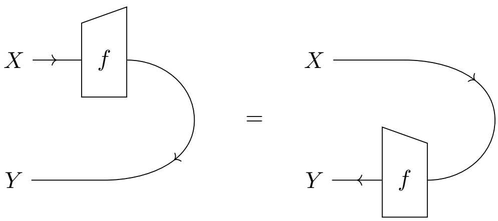


However, there is no corresponding family of cups $\eta_X : I \to \binom{X}{X}$, so we do not allow wires to bend the other way in this class of diagrams.

The P-strategies $\varepsilon_X : \binom{X}{X} \to I$ are dinatural in X, which means that for any function $f : X \to Y$ the diagram

$$\begin{tikzcd} \binom{X}{Y} \arrow[r, "f \otimes \mathrm{id}_{\binom{1}{Y}}"] \arrow[d, "\mathrm{id}_{\binom{X}{1}} \otimes f^*"'] & \binom{Y}{Y} \arrow[d, "\varepsilon_Y"] \\ \binom{X}{X} \arrow[r, "\varepsilon_X"] & I \end{tikzcd}$$

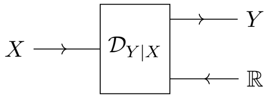
in $\mathfrak{L}$ commutes. In string diagrams, this equation is depicted

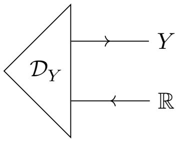


The reader should visualise f flipping over rather than rotating within the plane. This comes from the convention that $\otimes$ reverses the contravariant part of an object, and corresponds to the choice of algebraic rather than diagrammatic transpose in Coecke and Kissinger (2017, Sect. 4.2.2).

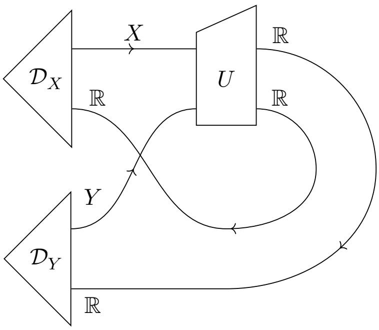
This can be seen as a sort of partial duality, which is defined on all objects by $\binom{X}{S}^* = \binom{S}{X}$ (which is interchange of players in a dialogue) and on P-strategies of the form f and f^*, but on no other open games besides these. In the last section of this paper we will extend this to a fully-fledged duality in the sense of compact closure.

A decision $\mathcal{D}_{Y|X} : \binom{X}{1} \to \binom{Y}{\mathbb{R}}$ and its special case $\mathcal{D}_Y = \mathcal{D}_{Y|1} : I \to \binom{Y}{\mathbb{R}}$ are respectively depicted



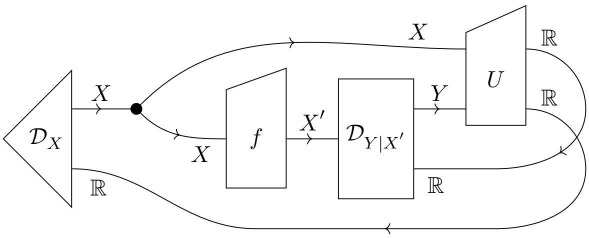


The string diagrams built from these diagram elements correspond to the open games generated from zero-player open games and decisions by sequential and parallel composition. Given a pair of payoff matrices $U : X \times Y \to \mathbb{R} \times \mathbb{R}$, the resulting bimatrix game corresponds to the diagram



in the sense that the scalar open game $\mathcal{G} : I \to I$ defined by the diagram has as strategy profiles $\Sigma_{\mathcal{G}} = X \times Y$ the pure strategy profiles of the bimatrix game, and as equilibria the pure strategy Nash equilibria of the bimatrix game: $|\mathcal{G}|^{x,y}$ holds iff $x \in \arg\max_{x'} U_1(x', y)$ and $y \in \arg\max_{y'} U_2(x, y')$. This directly generalises to normal-form games with any finite number of players.

Similarly, the diagram



describes a 2-player sequential game in which the first player chooses x and then the second player chooses y after observing $f(x)$ for some function $f : X \to X'$, which is equivalently an extensive form with player 2's information sets given by the equivalence relation on X' induced by f. As special cases, if f is the identity function then the f node can be drawn as a plain wire and we obtain a game of perfect information, and if $f : X \to 1$ is the delete function then f cancels with the copy function and the diagram can be deformed into the previous one to obtain a bimatrix game. The scalar game $\mathcal{G} : I \to I$ defined by the diagram has $\Sigma_{\mathcal{G}} = X \times Y^{X'}$ given by the pure strategy profiles, and $|\mathcal{G}|^{x,f}$ holds iff $x \in \arg\max_{x'} U_1(x', f(x'))$ and $f(x) \in \arg\max_{y'} U_2(x, y')$. Notice that these are the Nash equilibria of the extensive form game, rather than the subgame perfect equilibria. Again, this generalises to extensive form games with any finite number of players.

11.7 Dialogues and Wave-Style Geometry of Interaction

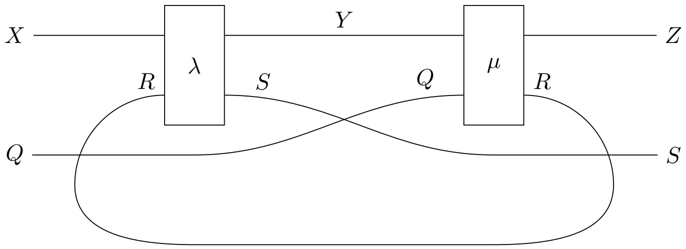
In order to obtain a connection between the dialectica and **Int** constructions, we need to apply the **Int** construction to categories that are traced cartesian monoidal. This is *wave-style geometry of interaction*, so-called because every point in our string diagrams is consistently assigned a value (Abramsky, 1996). (It is contrasted with *particle-style GoI*, which applies to monoidal categories built on a coproduct and in which we imagine a token moving around the diagram.)

Game-semantic interpretations of wave-style GoI have not been widely considered. In this section we suggest such an interpretation that will be suitable for our purposes.

The **Int**-construction can be defined over any traced monoidal category $\mathcal{C}$, but we restrict to traced cartesian categories. These are equivalent to *Conway cartesian categories*, or cartesian categories with a natural family of fixpoint operators (Hasegawa, 1999) (see also Ponto and Shulman 2014). A canonical example is the category **DCPO** of directed-complete partial orders and Scott-continuous maps.

By definition, an object of the category $\mathbf{Int}(\mathcal{C})$ is a pair $\binom{X}{S}$ of objects of $\mathcal{C}$, and a morphism $\binom{X}{S} \to \binom{Y}{R}$ in $\mathbf{Int}(\mathcal{C})$ is a morphism $X \times R \to Y \times S$ in $\mathcal{C}$. Since $\mathcal{C}$ is cartesian monoidal, a morphism $\binom{X}{S} \to \binom{Y}{R}$ is equivalently a pair of morphisms $X \times R \to Y$ and $X \times R \to S$.

The identity on $\binom{X}{S}$ in $\mathbf{Int}(\mathcal{C})$ is the identity on $X \times S$ in $\mathcal{C}$. The composition of $\lambda : \binom{X}{S} \to \binom{Y}{R}$ and $\mu : \binom{Y}{R} \to \binom{Z}{Q}$ in $\mathbf{Int}(\mathcal{C})$ is given by



in $\mathcal{C}$, using the string diagram language for traced monoidal categories (Selinger, 2011, Sect. 5.7).

The monoidal product of $\mathbf{Int}(\mathcal{C})$ is defined on objects by $\binom{X_1}{S_1} \otimes \binom{X_2}{S_2} = \binom{X_1 \otimes X_2}{S_2 \otimes S_1}$, with the obvious definition on morphisms. As is well known, $\mathbf{Int}(\mathcal{C})$ can be equipped with the structure of a compact closed category, which satisfies the universal property of being the free compact closed category on the traced monoidal category $\mathcal{C}$. Note that there are two different conventions in use: we follow (Joyal, 1996), which defines $\otimes$ with a twist in the contravariant place, rather than Abramsky (1996) which does not.

The idea of interpreting objects and morphisms of $\mathbf{Int}(\mathcal{C})$ as dialogues is to view them as repeated play of the corresponding dialogues for $\mathfrak{L}(\mathcal{C})$, starting from $\perp$ and converging to a fixpoint, after which the play terminates and all moves except the final ones are hidden.

We view the object $\binom{X}{S}$ as a dialogue

$$X^+; S^-; X^+; S^-; \cdots$$

We do not allow arbitrary strategies, but restrict the allowed P-strategies to $\mathcal{C}$-morphisms $S \to X$, and the allowed O-strategies to the $\mathcal{C}$-morphisms $X \to S$. Given such a pair of strategies (h, k), the play that results is by definition

$$\perp_X; \perp_S; h(\perp_S); k(\perp_X); h(k(\perp_X)); k(h(\perp_S)); \cdots$$

When $\mathcal{C}$ is **DCPO** or another suitable category, this play stabilises after finitely many stages to the (x, s) that is the least fixpoint of the recursion $x = h(s)$, $s = k(x)$. By hiding the approximating moves, we consider the play resulting from (h, k) to be (x, s).

Given objects $\binom{X}{S}$ and $\binom{Y}{R}$, the dialogue $\binom{X}{S} \to \binom{Y}{R}$ is

$$X^-; Y^+; R^-; S^+; X^-; Y^+; R^-; S^+; \cdots$$

We restrict the allowed P-strategies to $\mathcal{C}$-morphisms $X \times R \to Y \times S$ and the allowed O-strategies to $\mathcal{C}$-morphisms $Y \times S \to X \times R$. Given a P-strategy $\lambda =< v, u >$ and an O-strategy $\kappa =< h, k >$, the resulting play is

$$\begin{array}{llll} x_0 = \bot_X; & y_0 = \bot_Y; & r_0 = \bot_R; & s_0 = u(x_0, r_0); \\ x_{n+1} = h(y_n, s_n); & y_{n+1} = v(x_{n+1}, r_n); & r_{n+1} = k(y_{n+1}, s_n); & s_{n+1} = u(x_{n+1}, r_{n+1}) \end{array}$$

This stabilises after finitely many stages to (x, y, r, s) which is the least fixpoint of $(x, r) = \kappa(y, s)$, $(y, s) = \lambda(x, r)$. Again we hide the approximating moves so that (x, y, r, s) is the visible play.

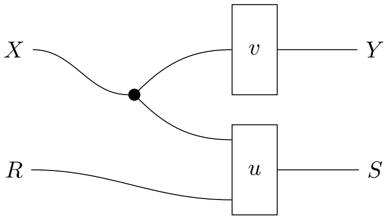
11.8 From Dialectica to Geometry of Interaction

The previous section suggests that every P-strategy in $\mathfrak{L}(\mathcal{C})\left(\binom{X}{S}, \binom{Y}{R}\right)$ can also be viewed as a P-strategy in $\mathbf{Int}(\mathcal{C})\left(\binom{X}{S}, \binom{Y}{R}\right)$. We could also discover this fact simply by inspecting the definitions, without thinking in terms of dialogues.

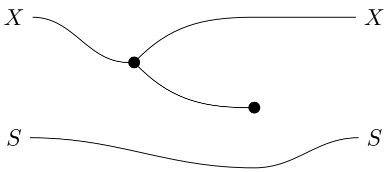
Incidentally, Hasuo and Hoshino (2017) refers to the Int-construction as "bidirectional computation", a technical term that usually refers to lenses and related constructions (e.g. Gibbons and Stevens, 2016).

In this section we use string diagrams in the underlying category $\mathcal{C}$. This is the language of traced symmetric monoidal categories (Selinger, 2011, Sect. 5.7) which are cartesian monoidal (Selinger, 2011, Sect. 6.1). Implicitly, string diagrams for cartesian monoidal categories use the fact that a monoidal product is cartesian iff every object can be compatibly equipped with a commutative comonoid structure making every morphism into a comonoid homomorphism (Fox, 1976).

Proposition 4 *Let $\mathcal{C}$ be a traced cartesian category. Then there is a strict monoidal functor $-^* : \mathfrak{L}(\mathcal{C}) \to \mathbf{Int}(\mathcal{C})$, which is identity on objects and takes the strategy (v, u) to*

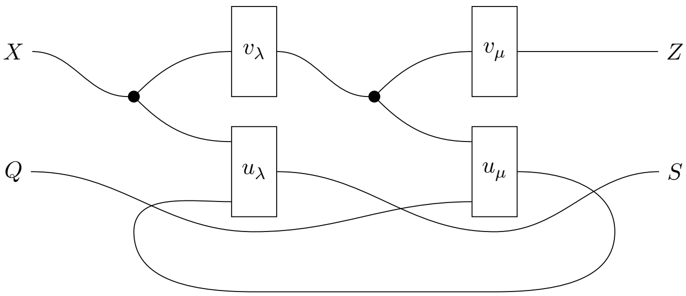


Proof The identity morphism $\binom{X}{S} \to \binom{X}{S}$ of $\mathfrak{L}(\mathcal{C})$ is sent to

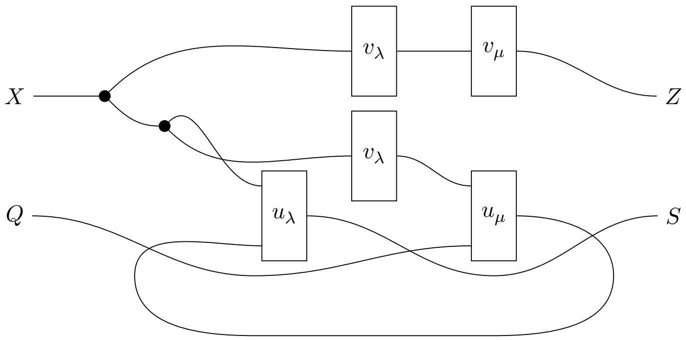


which is equal to the identity on $X \times S$ since the black structure is a comonoid. This is the identity morphism $\binom{X}{S} \to \binom{X}{S}$ of $\mathbf{Int}(\mathcal{C})$.

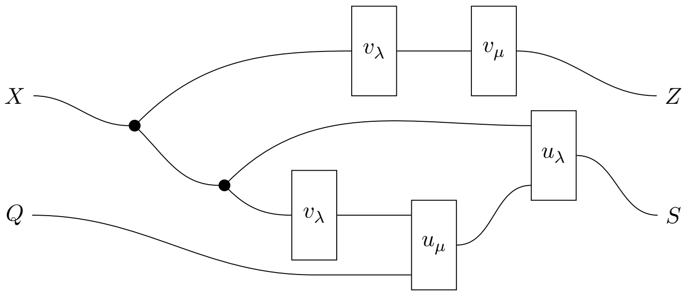
Next, consider morphisms $\lambda : \binom{X}{S} \to \binom{Y}{R}$ and $\mu : \binom{Y}{R} \to \binom{Z}{Q}$ in $\mathfrak{L}(\mathcal{C})$. If we compose them in $\mathbf{Int}(\mathcal{C})$ we obtain the morphism $\mu^* \circ \lambda^*$ with string diagram



Using the fact that v_λ is a comonoid homomorphism, followed by coassociativity and symmetry of the black structure, we can transform this to



On the other hand, if we compose in $\mathfrak{L}(\mathcal{C})$, we obtain $(\mu \circ \lambda)^*$ with string diagram



By inspection, we see that these string diagrams are equivalent. Equality of the two morphisms then follows from the coherence theorem for traced symmetric monoidal categories (Selinger, 2011, Theorem 5.22).

Finally, it can be seen by inspection that the functor is strict monoidal, since $\mathfrak{L}(\mathcal{C})$ and $\mathbf{Int}(\mathcal{C})$ have the same objects and the monoidal product is defined in the same way. $\square$

The previous result is still true when $\mathcal{C}$ is an arbitrary traced monoidal category, where $\mathfrak{L}(\mathcal{C})$ is replaced with the more general category of optics (Riley, 2018). This was proved by Elena Di Lavore and Mario Román (private communication).

We also note that the functor $-^*$ takes the P-strategy $\varepsilon_X : \binom{X}{X} \to I$ to the morphism $\varepsilon_X : \binom{X}{X} \to I$ that is the counit of the compact closed structure of $\mathbf{Int}(\mathcal{C})$.

11.9 Abstracting Open Games

Inspecting the definition of open games, it appears that we can define open games replacing $\mathfrak{L}$ with any symmetric monoidal category $\mathcal{C}$ with a chosen functor $\overline{\mathcal{C}} : \mathcal{C} \times \mathcal{C}^{\mathrm{op}} \to \mathbf{Set}$. This does indeed give us a *category* of open games, but defining a monoidal product of open games requires an additional piece of structure, namely the ability to project individual P-strategies out of a O-strategy for a composite. This is axiomatised by the following definition.

Definition 1 A *context* for a symmetric monoidal category $\mathcal{C}$ is a symmetric lax monoidal functor $\overline{\mathcal{C}} : \mathcal{C} \times \mathcal{C}^{\mathrm{op}} \to \mathbf{Set}$ together with a natural family of functions

$$/ : \mathcal{C}(X_2, Y_2) \to \left(\overline{\mathcal{C}}(X_1 \otimes X_2, Y_1 \otimes Y_2) \to \overline{\mathcal{C}}(X_1, Y_1)\right)$$

The naturality condition required is that for all morphisms $W_1 \xrightarrow{\lambda_1} X_1$, $Y_2 \xrightarrow{\nu_1} Z_1$ and $W_2 \xrightarrow{\lambda_2} X_2 \xrightarrow{\mu_2} Y_2 \xrightarrow{\nu_2} Z_2$, the diagram

$$\begin{tikzcd}
\overline{\mathcal{C}}(Z_1 \otimes Z_2, W_1 \otimes W_2) \arrow[r, "(\nu_2 \circ \mu_2 \circ \lambda_2)/-"] \arrow[d, "\overline{\mathcal{C}}(\nu_1 \otimes \nu_2, \lambda_1 \otimes \lambda_2)"] & \overline{\mathcal{C}}(Z_1, W_1) \arrow[d, "\overline{\mathcal{C}}(\nu_1, \lambda_1)"] \\
\overline{\mathcal{C}}(Y_1 \otimes Y_2, X_1 \otimes X_2) \arrow[r, "\mu_2/-"] & \overline{\mathcal{C}}(Y_1, X_1)
\end{tikzcd}$$

in $\mathbf{Set}$ commutes.

Using the symmetry, we can derive from this a natural family of functions

$$\backslash : \mathcal{C}(X_1, Y_1) \to \left(\overline{\mathcal{C}}(X_1 \otimes X_2, Y_1 \otimes Y_2) \to \overline{\mathcal{C}}(X_2, Y_2)\right)$$

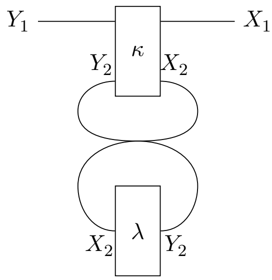
and vice versa.

The structures we defined earlier do indeed give a context on $\mathfrak{L}(\mathcal{C})$, namely

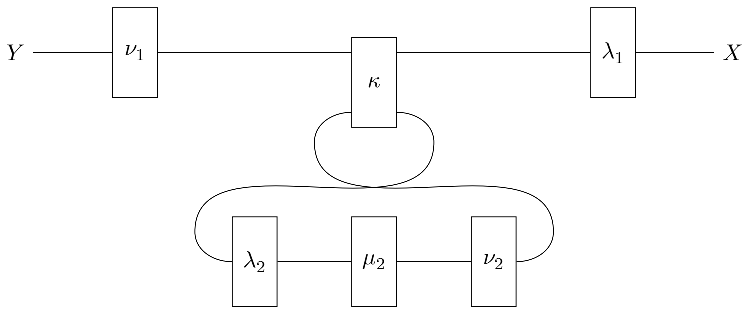
$$\overline{\mathfrak{L}(\mathcal{C})}\left(\begin{pmatrix}X\\S\end{pmatrix}, \begin{pmatrix}Y\\R\end{pmatrix}\right) = \mathcal{C}(1, X) \times \mathcal{C}(Y, R)$$

There are trivial examples of contexts that carry no game-theoretic information, which we will ignore. For example, we can always take $\overline{\mathcal{C}}$ to be a constant functor. We give a second family of nontrivial examples, which we will use later.

Proposition 5 *Every traced symmetric monoidal category $\mathcal{C}$ can be equipped with the context $\overline{\mathcal{C}}(X, Y) = \mathcal{C}(Y, X)$, with λ/κ defined by*

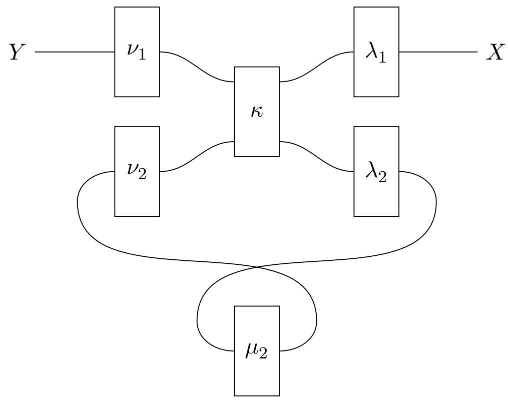


Proof Let $\kappa : \mathcal{C}(Z \otimes Z', W \otimes W')$, $\lambda_1 : W_1 \to X_1$, $\nu_1 : Y_2 \to Z_1$ and $W_2 \xrightarrow{\lambda_2} X_2 \xrightarrow{\mu_2} Y_2 \xrightarrow{\nu_2} Z_2$. We chase[1] the context κ around the commuting diagram in Definition 1. By the upper route we obtain



and by the lower route we obtain

[1] The author has named this proof technique 'string diagram chasing', i.e. chasing an element around a commuting diagram whose nodes are all formed from homsets in a monoidal category.



By the coherence theorem for traced monoidal categories, these denote equal morphisms. □

Definition 2 Let $\mathcal{C}$ be a symmetric monoidal category with a context $\overline{\mathcal{C}}$, and let X, Y be objects of $\mathcal{C}$. An *open game* $\mathcal{G} : X \to Y$ over $\mathcal{C}$ consists of

- A set $\Sigma_{\mathcal{G}}$ of strategy profiles
- A labelling function $\mathcal{G}_{-} : \Sigma_{\mathcal{G}} \to \mathcal{C}(X, Y)$
- A winning condition $|\mathcal{G}| \subseteq \Sigma_{\mathcal{G}} \times \overline{\mathcal{C}}(X, Y)$

Given open games $\mathcal{G} : X \to Y$ and $\mathcal{H} : Y \to Z$ over $\mathcal{C}$, their sequential composition $\mathcal{H} \circ \mathcal{G} : X \to Z$ is defined by $\Sigma_{\mathcal{H}\circ\mathcal{G}} = \Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{H}}$, $(\mathcal{H} \circ \mathcal{G})_{\sigma,\tau} = \mathcal{H}_\tau \circ \mathcal{G}_\sigma$ and

$$|\mathcal{H} \circ \mathcal{G}|^{\sigma,\tau}_{\kappa} \iff |\mathcal{G}|^{\sigma}_{\overline{\mathcal{C}}(\mathrm{id}_X, \mathcal{H}_\tau)(\kappa)} \wedge |\mathcal{H}|^{\tau}_{\overline{\mathcal{C}}(\mathcal{G}_\sigma, \mathrm{id}_Z)(\kappa)}$$

Given open games $\mathcal{G} : X_1 \to Y_1$ and $\mathcal{H} : X_2 \to Y_2$ over $\mathcal{C}$, their simultaneous composition $\mathcal{G} \otimes \mathcal{H} : X_1 \otimes X_2 \to Y_1 \otimes Y_2$ is defined by $\Sigma_{\mathcal{G}\otimes\mathcal{H}} = \Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{H}}$, $(\mathcal{G} \otimes \mathcal{H})_{\sigma,\tau} = \mathcal{G}_\sigma \otimes \mathcal{H}_\tau$ and

$$|\mathcal{G} \otimes \mathcal{H}|^{\sigma,\tau}_{\kappa} \iff |\mathcal{G}|^{\sigma}_{\mathcal{H}_\tau/\kappa} \wedge |\mathcal{H}|^{\tau}_{\mathcal{G}_\sigma\backslash\kappa}$$

Proposition 6 *For any symmetric monoidal category $\mathcal{C}$ with a context, there is a symmetric monoidal category* $\mathbf{OG}(\mathcal{C})$ *of open games over $\mathcal{C}$. When $\mathcal{C} = \mathfrak{L}(\mathbf{Set})$ with the usual context, we obtain the original category of open games.*

The proof of this proposition formally follows the proof that **OG** is a symmetric monoidal category. (The clearest presentation is in Hedges (2018, Sect. 5 & Appendix).) The definition of a context contains precisely the conditions needed for this proof to work.

Given a morphism $\lambda : X \to Y$ of $\mathcal{C}$, we define an open game $\lambda : X \to Y$ over $\mathcal{C}$ by $\Sigma_\lambda = \{*\}$, $\lambda_* = \lambda$ and $|\lambda|^*_\kappa$ holding for all κ.

Proposition 7 *This defines a faithful identity-on-objects symmetric strong monoidal functor* $\mathcal{C} \to \mathbf{OG}(\mathcal{C})$.

11.10 Morphisms of Contexts

Given a pair of categories with contexts $\mathcal{C}$, $\mathcal{D}$, it is possible to relate open games in $\mathbf{OG}(\mathcal{C})$ to open games in $\mathbf{OG}(\mathcal{D})$ if we have a strict monoidal functor $\mathcal{C} \to \mathcal{D}$ that is compatible with the context functors (despite the fact that $\mathbf{OG}(-)$ is not functorial due to mixed variance). In this section we will prove (mostly for completeness of the presentation) that when $\mathcal{C}$ is traced cartesian, the functor $-^* : \mathfrak{L}(\mathcal{C}) \to \mathbf{Int}(\mathcal{C})$ satisfies the required properties. This allows us to compare open games over $\mathfrak{L}(\mathbf{DCPO})$ and $\mathbf{Int}(\mathbf{DCPO})$ for example.

The reason we do not develop this idea fully is that it does not seem possible to obtain a strict (or even strong) monoidal functor $\mathfrak{L}(\mathbf{Set}) \to \mathfrak{L}(\mathbf{DCPO})$ that would allow us to understand 'computable game theory' as far as possible in terms of classical game theory. Ultimately this stems from the lack of a suitable product-preserving functor $\mathbf{Set} \to \mathbf{DCPO}$.

It does seem possible to overcome this using machinery that is known. One possibility is to consider $\mathbf{DCPO}$ with the smash product, which is a monoidal product that is not the categorical product, and then consider optics over this. Open games over a particular category of optics (over the monoidal category of conditional probability distributions) were considered in the context of Bayesian games (Bolt et al., 2019), but they are more subtle and less intuitive so we leave this for future work.

Definition 3 Let $\mathcal{C}$ and $\mathcal{D}$ be symmetric monoidal categories with contexts $\overline{\mathcal{C}}$ and $\overline{\mathcal{D}}$. A strict morphism of contexts is a strict symmetric monoidal functor $F : \mathcal{C} \to \mathcal{D}$ together with a monoidally natural family of functions

$$\overline{F}(X, Y) : \overline{\mathcal{C}}(X, Y) \to \overline{\mathcal{D}}(F(X), F(Y))$$

such that

$$\begin{tikzcd}
\overline{\mathcal{C}}(X_1 \otimes X_2, Y_1 \otimes Y_2) \arrow[r, "f/-"] \arrow[d, "\overline{F}(X_1 \otimes X_2, Y_1 \otimes Y_2)"] & \overline{\mathcal{C}}(X_1, Y_1) \arrow[d, "\overline{F}(X_1, Y_1)"] \\
\overline{\mathcal{D}}(F(X_1 \otimes X_2), F(Y_1 \otimes Y_2)) \arrow[r, "F(f)/-"] & \overline{\mathcal{D}}(F(X_1), F(Y_1))
\end{tikzcd}$$

commutes for all $f : X_2 \to Y_2$.

Defining non-strict morphisms of contexts takes a bit more care, but is not necessary for our purposes.

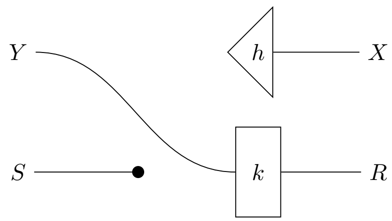
Given a traced monoidal category $\mathcal{C}$, the category $\mathbf{Int}(\mathcal{C})$ is compact closed, and hence in particular traced monoidal. We consider it to have the context defined for traced monoidal categories. That is,

$$\overline{\mathbf{Int}(\mathcal{C})}\left(\binom{X}{S}, \binom{Y}{R}\right) = \mathbf{Int}(\mathcal{C})\left(\binom{Y}{R}, \binom{X}{S}\right) = \mathcal{C}(Y \otimes S, X \otimes R)$$

Proposition 8 *Let $\mathcal{C}$ be a traced cartesian category. Then $-^* : \mathfrak{L}(\mathcal{C}) \to \mathbf{Int}(\mathcal{C})$ can be made into a strict morphism of contexts, by defining*

$$\overline{\ast} : \mathcal{C}(1, X) \times \mathcal{C}(Y, R) \to \mathcal{C}(Y \times S, X \times R)$$

to take (h, k) to



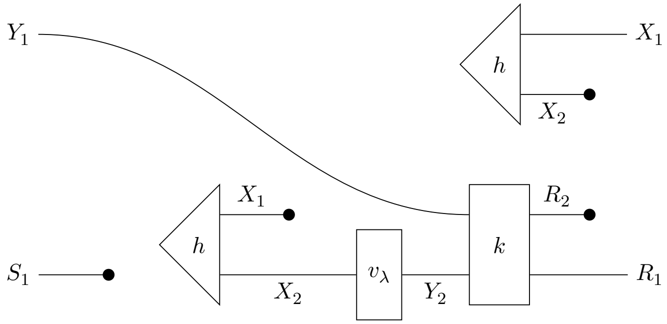
Proof We already checked that $-^*$ is strict symmetric monoidal. We get naturality for free by noting that $\overline{\ast}$ can be equivalently defined by

$$\begin{aligned}\mathcal{C}(I, X) \times \mathcal{C}(Y, R) &\xrightarrow{\cong} \mathfrak{L}(\mathcal{C})\left(I, \binom{X}{S}\right) \times \mathfrak{L}(\mathcal{C})\left(\binom{Y}{R}, I\right)\\ &\xrightarrow{\circ} \mathfrak{L}(\mathcal{C})\left(\binom{Y}{R}, \binom{X}{S}\right)\\ &\xrightarrow{-^*} \mathbf{Int}(\mathcal{C})\left(\binom{Y}{R}, \binom{X}{S}\right)\end{aligned}$$

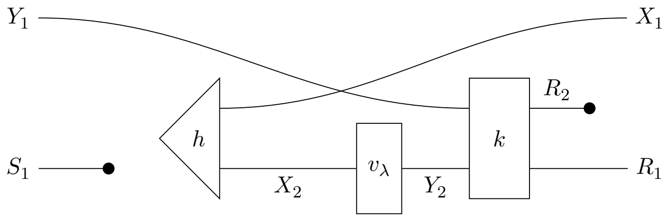
Suppose we have a P-strategy $\lambda : \binom{X_2}{S_2} \to \binom{Y_2}{R_2}$. We must verify that the square

$$\begin{tikzcd}
\mathcal{C}(1, X_1 \times X_2) \times \mathcal{C}(Y_1 \times Y_2, R_2 \times R_1) \arrow[r, "\lambda/-"] \arrow[d] & \mathcal{C}(1, X_1) \times \mathcal{C}(Y_1, R_1) \arrow[d] \\
\mathcal{C}(Y_1 \times Y_2 \times S_2 \times S_1, X_1 \times X_2 \times R_2 \times R_1) \arrow[r, "\lambda^*/-"] & \mathcal{C}(Y_1 \times S_1, X_1 \times R_1)
\end{tikzcd}$$

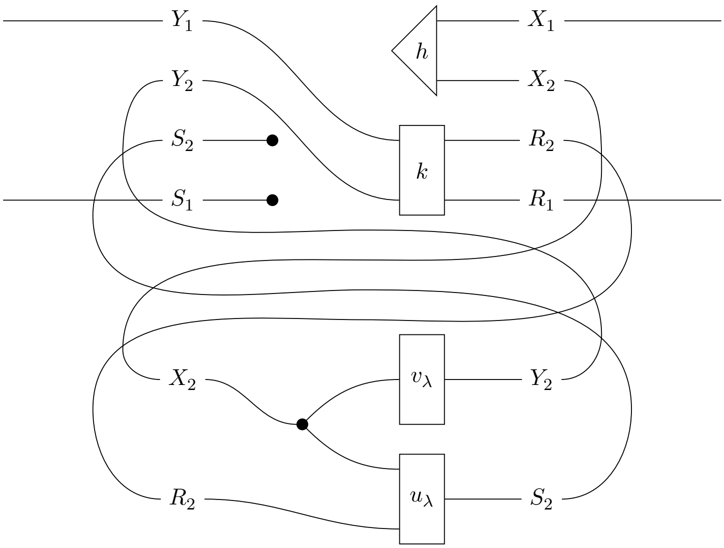
commutes. Chasing a context (h, k) around the top yields



As a useful intermediate point, since h is a comonoid homomorphism this is equivalent to



On the other hand, chasing (h, k) around the bottom yields



To see the equivalence of this diagram to the previous one modulo traced cartesian categories, trace the deletion on S_2 backwards. □

11.11 Compositional Computable Game Theory

We can finally put all the pieces together, by considering the category $\mathbf{OG}(\mathbf{Int}(\mathbf{DCPO}))$. Concretely, for DCPOs X, S, Y, R, such an open game $\mathcal{G} : \binom{X}{S} \to \binom{Y}{R}$ consists of:

1. A set Σ of *strategy profiles*
2. A family of continuous *play functions* $\mathbf{P}_{\mathcal{G}}(\sigma) : X \times R \to Y$
3. A family of continuous *coplay functions* $\mathbf{C}_{\mathcal{G}}(\sigma) : X \times R \to S$
4. An equilibrium set $\mathbf{E}_{\mathcal{G}}(h, k) \subseteq \Sigma$ for each continuous *history* $h : Y \times S \to X$ and *continuation* $k : Y \times S \to R$

This definition is written in the style of the original concrete definition of open games in Ghani et al. (2018) for ease of comparison. Specifically, besides changing the base category from **Set** to **DCPO** this definition differs by making $\mathbf{C}_{\mathcal{G}}$ additionally a function of R, h a function of Y and S, and k a function of S.

The category $\mathbf{OG}(\mathbf{Int}(\mathbf{DCPO}))$ is compact closed, as a result of applying Proposition 1 to the zero-player functor $\mathbf{Int}(\mathbf{DCPO}) \to \mathbf{OG}(\mathbf{Int}(\mathbf{DCPO}))$.

As an exercise, we work out the transpose $\mathcal{G}^* : \binom{R}{Y} \to \binom{S}{X}$ of a general open game $\mathcal{G} : \binom{X}{S} \to \binom{Y}{R}$ over $\mathbf{Int}(\mathcal{C})$. The set of strategy profiles stays the same up to isomorphism, $\Sigma_{\mathcal{G}^*} \cong \Sigma_{\mathcal{G}}$, because the transpose is defined by composition with various open games whose set of strategy profiles is 1. The play function is modified by taking the transpose in $\mathbf{Int}(\mathcal{C})$, which in the end simply exchanges the play and coplay functions $X \times R \to Y$, $X \times R \to S$ and swaps their inputs. A context (h, k), for $h : S \times Y \to R$ and $k : S \times Y \to X$ is again swapped to give a context $h' = k : Y \times S \to X$ and $k' = h : Y \times S \to R$ for $\mathcal{G}$, so equilibrium is defined by $|\mathcal{G}^*|^{\sigma}_{h,k} \iff |\mathcal{G}|^{\sigma}_{k,h}$.

Given a continuous function $f : X \to Y$, the covariant and contravariant liftings $f : \binom{X}{1} \to \binom{Y}{1}$ and $f^* : \binom{1}{Y} \to \binom{1}{X}$ are now transposes of each other. Thus we are conservatively extending the notion of duality that already exists in categories of open games.

Recall that for a decision $\mathcal{D}_{Y|X} : \binom{X}{1} \to \binom{Y}{\mathbb{R}}$ over $\mathfrak{L}(\mathbf{Set})$, a context is a pair $h : X$ and $k : Y \to \mathbb{R}$, and the equilibrium condition for a strategy $\sigma : X \to Y$ is that $\sigma(h) \in \arg\max(k)$. We can make a similar definition over $\mathfrak{L}(\mathbf{DCPO})$ given a suitable domain of reals $\mathbb{R}$ and a suitable arg max operator defined on continuous functions $Y \to \mathbb{R}$. There are several options for defining these, and we remain largely agnostic between them. (We do not assume that arg max is internalised in **DCPO** as a function $\mathbb{R}^Y \to \mathcal{P}(Y)$ for some powerdomain $\mathcal{P}$, since a naive definition would not be continuous.)

A simple example of a domain of reals that can serve as a mental model is the domain of closed intervals $[x, y]$ with the reverse inclusion order, together with $\perp_{\mathbb{R}} = (-\infty, +\infty)$. Here $[x, y]$ represents an approximation of some $z \in [x, y]$, and a standard real number z is represented by the degenerate interval $[z, z]$. Note that the arg max operator is still defined for the standard order on reals (which must be extended to all elements of the domain), which is not related to the inclusion order. As a minimal requirement in order to work out an example later, we suppose that $\perp_{\mathbb{R}}$

is below every standard real in the extended standard order. This corresponds to the assumption that players always prefer a terminating payoff, no matter how small, to a nonterminating one.

Over **Int**(**DCPO**), the context for a decision $\mathcal{D}_{X,Y}$ has the form

$$\kappa = (h, k) : \overline{\mathbf{Int}(\mathbf{DCPO})}\left(\begin{pmatrix} X \\ 1 \end{pmatrix}, \begin{pmatrix} Y \\ \mathbb{R} \end{pmatrix}\right) \cong \mathbf{DCPO}(Y, X) \times \mathbf{DCPO}(Y, \mathbb{R})$$

Notice that since the coutility type $S = 1$ is the terminal DCPO, the only difference from a context over $\mathfrak{L}(\mathcal{C})$ is that the history h may depend on the move from Y. That is, the future action may affect the past observation.

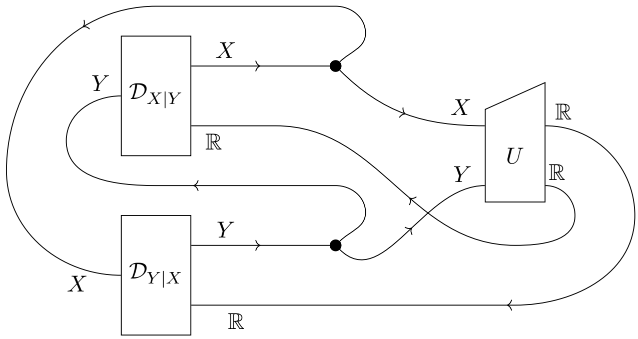
We define the decision $\mathcal{D} = \mathcal{D}_{Y|X} : \binom{X}{1} \to \binom{Y}{\mathbb{R}}$ over **Int**(**DCPO**) to have $\Sigma_{\mathcal{D}} =$ **DCPO**(X, Y), and the play function

$$\mathbf{DCPO}(X, Y) \to \mathbf{Int}(\mathbf{DCPO})\left(\begin{pmatrix} X \\ 1 \end{pmatrix}, \begin{pmatrix} Y \\ \mathbb{R} \end{pmatrix}\right) \cong \mathbf{DCPO}(X \times \mathbb{R}, Y)$$

given by composition with the projection. A natural definition for equilibrium is that the least fixpoint y of $y = \sigma(h(y))$ is in $\arg\max(k)$, which we write

$$|\mathcal{D}|^{\sigma}_{h,k} \iff \mu y.\sigma(h(y)) \in \arg\max(k)$$

As a worked example, we can build a 2-player game in which each player's strategy may be contingent on the choice of the other, something that is causally absurd. Let the game $\mathcal{G} : I \to I$ be defined by the string diagram



We suppose X and Y to be finite flat domains, say $X = Y = \{\bot, a, b\}$. We also suppose that U is zero-player and encodes some function $U : X \times Y \to \mathbb{R} \times \mathbb{R}$, where $\mathbb{R}$ in the latter is the set of standard real numbers. That is to say, every $U(x, y)$ is some pair of total real numbers for $x, y \neq \bot$. As a specific example, let U be the payoff matrix of matching pennies, extended as follows:

$$U(x, y) = \begin{cases} (1,0) & \text{if } x = y \neq \perp \\ (0,1) & \text{if } x \neq y, x \neq \perp, y \neq \perp \\ (\perp_{\mathbb{R}}, \perp_{\mathbb{R}}) & \text{if } x = \perp \text{ or } y = \perp \end{cases}$$

Matching pennies is an interesting example here because it exhibits second-move advantage: either player would benefit from the ability to play contingently on the other's move.

The set of strategy profiles of this game is $\Sigma = \mathbf{DCPO}(X, Y) \times \mathbf{DCPO}(Y, X)$. The equilibrium condition $|\mathcal{G}|^{\sigma,\tau}$ holds iff

$$\mu x.\sigma(\tau(x)) \in \arg\max_x U_1(x, \tau(x))$$

$$\mu y.\tau(\sigma(y)) \in \arg\max_y U_2(\sigma(y), y)$$

We can now check these conditions on some specific examples. First suppose that σ and τ are both constant functions, say $\sigma(y) = a$ and $\tau(x) = b$, including for $y = \perp$ and $x = \perp$. (These are 'lazy' functions: they terminate with a total value even when their input does not.) We can then directly calculate that $\mu x.\sigma(\tau(x)) = a$ and $\mu y.\tau(\sigma(y)) = b$. The first player has incentive to deviate because $a \notin \arg\max_x U_1(x, b) = \{b\}$, although the second player is satisfied since $b \in \arg\max_y U_2(a, y) = \{b\}$. Thus (σ, τ) is not an equilibrium of this game. By this reasoning $\mathcal{G}$ has no 'lazy' equilibria of this form, since matching pennies has no pure strategy Nash equilibria.

Next consider the strategies $\sigma(y) = y$ and $\tau(x) = a$, in which player 1 seizes the second-move advantage by playing the optimal response to player 2's move, namely copying it. Then $\mu x.\sigma(\tau(x)) = \mu y.\tau(\sigma(y)) = a$, and (σ, τ) is an equilibrium since $\arg\max_x U_1(x, a) = \{a\}$ and $\arg\max_y U_2(y, y) = \{a, b\}$. There is another equilibrium given by $\sigma(y) = y$ and $\tau(x) = b$. Notice that player 2 could play $\perp$ and deadlock the play, but we assume that she prefers the 'losing' total payoff of 0. Similarly there are two more equilibria in which it is player 2 who takes the second-move advantage with $\tau(x) = \overline{x}$, given by $\overline{a} = b$ and $\overline{b} = a$ (and in consequence, $\overline{\perp} = \perp$).

Finally, suppose that both players attempt to move second, with the strategy profile $\sigma(y) = y$ and $\tau(x) = \overline{x}$. Then $\mu x.\sigma(\tau(x)) = \mu y.\tau(\sigma(y)) = \perp$: the play deadlocks as each player waits for the other to move first. However $\arg\max_x U_1(x, \overline{x}) = \arg\max_y U_2(y, y) = \{a, b\}$, so both players have incentive to deviate and (σ, τ) is not an equilibrium. Given our assumptions, either player would prefer to move first and take the losing total payoff, rather than deadlocking the play.

References

Abramsky, S., & Brandenburger, A. (2011). The sheaf-theoretic structure of non-locality and contextuality. *New Journal of Physics, 13*.

Abramsky, S. (1996). Retracing some paths in process algebra. In *CONCUR '96: Concurrency Theory*, volume 1119 of Lecture Notes in Computer Science (pp. 1–17). Springer.

Abramsky, S. (1997). Semantics and logics of computation. In *Semantics of interaction*. Cambridge University Press.

Abramsky, S. (2005). Abstract scalars, loops, and free traced and strongly compact closed categories. In *Proceedings of CALCO'05*, volume 3629 of Lectures notes in computer science (pp. 1–29).

Abramsky, S. (2015). Arrow's theorem by arrow theory. In *Logic without borders: Essays on set theory, model theory, philosophical logic and philosophy of mathematics*, volume 5 of Ontos mathematical logic (pp. 15–30). De Gruyter.

Abramsky, S., & Coecke, B. (2002). Physical traces: Quantum versus classical information processing. In *Category theory and comptuer science*, volume 69 of Electronic notes in theoretical computer science (pp. 1–22).

Abramsky, S., & Coecke, B. (2004). A categorical semantics of quantum protocols. In *Proeedings of Logic in Comptuer Science (LiCS)* 2004.

Avigad, J., & Feferman, S. (1998). Gödel's functional ("Dialectica") interpretation. In *Handbook of proof theory*, volume 137 of Studies in logic and the foundations of mathematics (pp. 337–405). Elsevier.

Abramsky, S., & Jagadeesan, R. (1994). Games and full completeness for multiplicative linear logic. *Journal of Symbolic Logic, 59*(2), 543–574.

Abramsky, S., & Jagadeesan, R. (1994). New foundations for the geometry of interaction. *Information and Computation, 111*(1), 53–119.

Abramsky, S., Jagadeesan, R., & Malacaria, P. (2000). Full abstraction for PCF. *Information and Computation, 163*(2), 409–470.

Abramsky, S., & McCusker, G. (1999). Game semantics. In *Computational logic*, volume 165 of NATO ASI Series. Springer.

Abramsky, S., & Winschel, V. (2017). Coalgebraic analysis of subgame-perfect equilibria in infinite games without discounting. *Mathematical Structures in Computer Science, 27*(5), 751–761.

Bolt, J., Hedges, J., & Zahn, P. (2019). *Bayesian open games*. arXiv:1910.03656.

Blass, A. (1991). A game semantics for linear logic. *Annals of Pure and Applied Logic, 56,* 183–220.

Boisseau, G. (2020). *String diagrams for optics*. arXiv:2002.11480.

Coecke, B., & Kissinger, A. *Picturing quantum processes*. Cambridge University Press.

Coecke, B., Sadrzadeh, M., & Clark, S. (2010). Mathematical foundations for a compositional distributional model of meaning. *Linguistic Analysis, 36*(1), 345–384.

de Paiva, V. (1991). *The dialectica categories*, Technical report, University of Cambridge.

Foster, N., Greenwald, M., Moore, J., Pierce, B., & Schmitt, A. (2007). Combinators for bidirectional tree transformations: A linguistic approach to the view update problem. *ACM Transactions on Programming Languages and Systems, 29*(3).

Fong, B. (2016). *The algebra of open and interconnected systems*. Ph.D. thesis, University of Oxford.

Fox, T. (1976). Coalgebras and cartesian categories. *Communications in Algebra, 4*(7), 665–667.

Ghani, N., Hedges, J., Winschel, V., & Zahn, P. (2018). Compositional game theory. In *Proceedings of Logic in Computer Science (LiCS) 2018* (pp. 472–481). ACM.

Gödel, K. (1958). Über eine bisher noch nicht benütze Erweiterung des finiten Standpunktes. *Dialectica, 12,* 280–287.

Gibbons, J., & Stevens, P. (Eds.). (2016). *Bidirectional transformations*, volume 9715 of Lecture Notes in Computer Science. Springer.

Gutierrez, J., & Wooldridge, M. (2014). Equilibria of concurrent games on event structures. In *Proceedings of Logic in Computer Science (LiCS) 2014*. ACM.

Hasegawa, M. (1999). Recursion from cyclic sharing. In *Models of sharing graphs, Distinguished Disseratations* (pp. 83–101). Springer.

Hedges, J. (2016). *Towards compositional game theory*. Ph.D. thesis, Queen Mary University of London.

Hedges, J. (2018). Morphisms of open games. In *Proceedings of MFPS* 2018, volume 341 of Electronic notes in theoretical computer science (pp. 151–177).

Hasuo, I., & Hoshino, N. (2017). Semantics of higher-order quantum computation via geometry of interaction. *Annals of Pure and Applied Logic, 168*(2), 404–469.

Hankin, C., & Malacaria, P. (2013). Payoffs, intensionality and abstraction in games. In B. Coecke, L. Ong & P. Panangaden (Eds.), *Computation, logic, games and quantum foundations: The many facets of Samson Abramsky*, volume 7860 of Lecture Notes in Computer Science (pp. 69–82). Springer.

Hyland, M., & Ong, L. (2000). On full abstraction for PCF: I, II and III. *Information and Computation, 163*(2), 285–408.

Joyal, A., Street, R., & Verity, D. (1996). Traced monoidal categories. *Mathematical Proceedings of the Cambridge Philosophical Society, 119,* 447–468.

Kelly, G. M., & Laplaza, M. L. (1980). Coherence for compact closed categories. *Journal of Pure and Applied Algebra, 19,* 193–213.

Lorenzen, P., & Lorenz, K. (1978). *Dialogische Logik. Wissenschaftliche Buchgesellschaft*.

Le Roux, S. (2014). From winning strategy to Nash equilibrium. *Mathematical Logic Quarterly, 60*(4–5), 354–371.

Martin, D. (1975). Borel determinacy. *Annals of Mathematics. Second Series, 101*(2), 363–371.

Nash, J. (1951). Non-cooperative games. *Annals of Mathematics, 54*(2), 286–295.

Osbourne, M., & Rubinstein, A. (1994). *A course in game theory*. MIT Press.

Pavlovic, D. (2009). A semantical approach to equilibria and rationality. In *Algebra and coalgebra in computer science*, volume 5728 of Lectures notes in computer science (pp. 317–334). Springer.

Pickering, M., Gibbons, J., & Wu, N. (2017). Profunctor optics: Modular data accessors. *The Art, Science and Engineering of Programming, 1*(2).

Preller, A., & Lambek, J. (2007). Free compact 2-categories. *Mathematical Structures in Computer Science, 17*(2), 309–340.

Pastro, C., & Street, R. (2008). Doubles for monoidal categories. *Theory and Applications of Categories, 21*(4), 61–75.

Ponto, K., & Shulman, M. (2014). Traces in symmetric monoidal categories. *Expositiones Mathematicae, 32*(3), 248–273.

Riley, M. (2018). *Categories of optics*. arXiv:1809.00738.

Selinger, P. (2011). A survey of graphical languages for monoidal categories. In B. Coecke (Ed.), *New structures for physics* (pp. 289–355). Springer.

Schwalbe, U., & Walker, P. (2001). Zermelo and the early history of game theory. *Games and Economic Behaviour, 34,* 123–137.

van Benthem, J. (2013). Reasoning about strategies. In B. Coecke, L. Ong & P. Panangaden (Eds.), *Computation, logic, games and quantum foundations: The many facets of Samson Abramsky*, volume 7860 of Lecture Notes in Computer Science (pp. 336–347). Springer.

van Benthem, J. (2014). *Logic in games*. MIT Press.

von Neumann, J., & Morgenstern, O. (1944). *Theory of games and economic behaviour*. Princeton University Press.

Part III
Contextuality and Quantum Computation

Chapter 12
Describing and Animating Quantum Protocols

Richard Bornat and Rajagopal Nagarajan

Abstract We introduce a quantum protocol simulator, the Qtpi simulator, which allows description of quantum protocols with multiple agents. It has protection against cloning and sharing of qbits within a simulation. Its symbolic calculator is adequate for the protocols we have examined, and can be pushed to simulate some non-protocol algorithms.

Keywords Concurrency · Security · Protocols · Simulation · Quantum

12.1 Prologue

One of us (Richard) was Samson's Ph.D. supervisor, and the other (Raja) was his Ph.D. student. We therefore claim the right to salute him.

Back in the 1980s Richard and Samson collaborated for a while on programming language stuff, on distributed computing, concurrent programming, and teaching programming; and they had a great time. They devised, with others, the concurrent-programming language Pascal-m (Abramsky & Bornat, 1983), intended as a language for distributed computing. They helped to implement a demonstration operating system in it, and the language might have been a success if Richard hadn't insisted on a gross mistake in its design, allowing a process to make a guarded choice with both read and write partners (a 'mixed choice' in the terminology of the pi calculus (Milner et al., 1992)). The team worked out that their implementation of this feature could livelock in a distributed (multi-processor) setting. By the time Richard had worked out a possible solution (Bornat, 1986), it was too late. He had buried himself in teaching, and Samson was down the mathematics mineshaft where Richard couldn't (and in those days wouldn't) follow him. Samson's thesis, which eventually followed, had very little to do with Richard's supervision: it, like his later dazzling success, was all his own.

R. Bornat (✉) · R. Nagarajan
Department of Computer Science, Middlesex University, London, UK
e-mail: richard@bornat.me.uk

A. Palmigiano and M. Sadrzadeh (eds.), *Samson Abramsky on Logic and Structure in Computer Science and Beyond*, Outstanding Contributions to Logic 25,
https://doi.org/10.1007/978-3-031-24117-8_12

In the 1990s Raja studied for a Ph.D. at Imperial College London under Samson's supervision. He was part of the Theory and Formal Methods group and it was a hotbed of activity in theoretical computer science at that time. Raja had a very fruitful collaboration with Samson (and Simon Gay) on Interaction Categories, a new foundational paradigm for concurrency. This was also the time Samson, together with Radha Jagadeesan and Pasquale Malacaria, solved Full Abstraction for PCF, a long-standing open problem in theoretical computer science. Samson moved to Edinburgh and then to Oxford; Raja to Warwick and then to Middlesex. They both started working on different aspects of quantum information processing, independently.

At Middlesex University, in 2016, Richard started talking to Raja's research group, who were studying the formal analysis of quantum security protocols.

Existing security protocols are threatened by the possibility of quantum computing. For example, the RSA mechanism, the basis of most widely-used encryption, and thus the basis of many security protocols, relies for its safety on the presumed difficulty of factoring the products of large prime numbers. Famously, there is a quantum algorithm for factorization in polynomial time (Shor, 1994). No quantum computer yet exists that has anything like the number of qubits required to undermine the protocols used in contemporary practice, but we can expect one in a decade or three. Classical encryption is by no means finished: there are various schemes which are thought to be able to withstand attacks by quantum computers (Chen et al., 2016; Status Report, 2020).

Quantum security protocols offer interest as their safety depends only on the laws of quantum physics, albeit with a statistical guarantee. That might seem enough, but experience with the internet and with processor design (Kocher et al., 2019) shows how implementations, actual programs running on actual machines, can be undermined by clever opponents attacking the details of execution. That is: security depends on safety, but it isn't the whole game.

Raja's group was attempting to import the advantages of formal proof to the matter of quantum security protocols. They were trying to simulate quantum protocols using Microsoft Q# (The Q# Programming Language, 2023) in order to understand them better and were working towards a Coq (Boender et al., 2015) model of quantum circuit computation. Q# was not that suited to simulating quantum protocols and to Richard it seemed that Raja and Simon's language CQP (Gay & Nagarajan, 2005) was a much better fit to the problem. Eager to exorcise the ghosts of his Pascal-m misdesign, and keen to provide himself with a mechanical calculator that would illuminate the ways that the protocols work, he decided to implement it. In the end, because he was building a simulator, he included mixed choice in the new design.

So this is the future illuminated by a blast from the past: quantum protocols simulated with concurrent programming, actually using some of the design features and implementation mechanisms of Pascal-m.

12.2 Qtpi: A Language of Protocol Steps and Calculations

Protocols are carried out by *agents* which send each other messages but share no other information; they are presumed to be physically separated. Protocol designers often call their agents Alice, Bob, Eve and so on. Typical quantum-protocol steps from the literature are

- create a qubit, perhaps initialised to a basis-vector value such as $|0\rangle$, $|1\rangle$, $|+\rangle$ or $|-\rangle$;
- put a qubit through a quantum gate such as I, H, X, etc.;
- measure a qubit, in the computational basis or some other basis, delivering a bit (0 or 1);
- send (or receive) a qubit, or qubits, to (or from) another agent;
- send (or receive) a classical value, such as a list of numbers or bits, to (or from) another agent.

In addition an agent may perform a calculation, such as generating a list of random bits or encrypting/decrypting a message or checking that the values read in messages from another agent fit with other values calculated locally. Calculations aren't protocol steps and don't affect qubit state, though they often depend on the results of measuring qubits and their results often influence subsequent protocol steps.

Simulating quantum gating and measurement needs some sort of calculation library, but none of the rest of it is particularly difficult. If that were all there was to the problem, quantum protocols could be simulated in any concurrent programming language. But there is more: in the real world qubits can't be created out of nothing, they can't be shared between agents, and they can't be cloned by an agent. That is similar to the pointer problem addressed in separation logic (O'Hearn et al., 2001), and our approach to that problem is one of the two interesting things about qtpi (the other is the symbolic quantum calculation engine).

As a language, CQP was based on the pi calculus (Milner et al., 1992) with types and with two additions:

- a declaration form `(qbit q)` which creates a new value of type `Qbit`[1] much as `(new c:T)` creates a new channel of type `T`;
- an expression-step `{...}` in which calculations and quantum steps of measurement and gating take place, with a notation based on Selinger's QPL (Selinger, 2004).

Qtpi (qtp) puts quantum steps in processes P and makes expressions E pure functional calculations. This is an abbreviated description of processes:

[1] Note that CQP programs talk of 'qbits' rather than 'qubits'. Qtpi preserves this nomenclature.

$$
\begin{array}{rcl}
P & ::= & qstep\ .\ P \mid IO\ .\ P \mid (binder)\ .\ P \\
 & & \mid N(E, \ldots, E) \mid (P) \mid par \mid alt \mid cond \mid \_0 \\
binder & ::= & \texttt{new}\ c \mid \texttt{newq}\ q \mid \texttt{newq}\ q = E \mid \texttt{let}\ pat = E \\
par & ::= & P \mid \ldots \mid P \\
alt & ::= & IO\ .\ P + \ldots + IO\ .\ P \\
cond & ::= & \texttt{if}\ E\ \texttt{then}\ P\ \texttt{else}\ P \mid \texttt{match}\ E\ .\ pat.P + \ldots + pat.P \\
qstep & ::= & E, \ldots, E \gg E \mid E \not\nearrow (x) \mid E \not\nearrow [E]\,(x) \\
IO & ::= & E\ !\ E, \ldots, E \mid E\ ?\ (x, \ldots, z) \\
pat & ::= & x \mid \_ \mid [\ pat; \ldots; pat\ ] \mid (\ pat, \ldots, pat\ )
\end{array}
$$

Qsteps put one or more qbits through a gate ($\gg$) or measure ($\not\nearrow$) a single qbit in the $|0\rangle/|1\rangle$computational basis, binding the 0/1 result to a name x; the alternative form puts the qbit through a gate before measuring it (so `q` $\not\nearrow$ `[H](b)` effectively measures `q` in the $|+\rangle/|-\rangle$basis). IO steps send (!) a tuple of values or receive (?) a tuple, binding the result to a tuple of names. In binders 'new' binds a channel to a name; 'newq' binds a new qbit, which can be initialised to a ket value; 'let' binds one or more names. Vertical bars separate a parallel composition of processes (and, as in OCaml, an initial vertical bar is optional); plus signs separate a sum of IO-step guarded processes. In conditionals there is a conventional if-then-else, and a pattern-matching 'match'.

A process definition is '`proc` $N(x, \ldots, z) = P$'. A function definition is '`fun` $f\,pat \ldots pat = E$'.

A program is a collection of process and function definitions. Programs start execution with a zero-argument 'System' process.

The syntax of expressions E is not shown here: it's a fairly ordinary functional notation using 'where' rather than 'let', partly inspired by Turner's Miranda (1985) and, somewhat confusingly, partly by Milner's ML (Milner et al. 1997).

The language is typed but in most circumstances explicit typing is unnecessary because the implementation includes a Hindley-Milner typechecker (Milner, 1978), and typing is omitted from the grammar above. Types T are `num`, `bit`, `qbit`, `gate`, `[`T`]` (lists), `(`$T, \ldots, T$`)` (tuples, including a zero-tuple), and T `->` ... `->` T (functions). Recent developments (see Sect. 12.8) have added `bra`, `ket`, `matrix` and `sxnum` types.

Process invocation can be recursive. Because conditionals apply to processes rather than process steps, a process is tree-rather than dag-shaped. Quantum-measurement steps have a probabilistic outcome: in CQP the result of an execution was a tree of possible outcomes, each labelled by the probability of its occurrence. In qtpi the execution mechanism makes probabilistic measurements but settles pseudo-probabilistically on one of the outcomes.

12.2.1 No Cloning, No Sharing

In the real quantum world there is no way of cloning a qubit—you can't start with a qubit in some arbitrary state and finish up with that qubit unchanged and another in the same state. So in a programming language which simulates quantum effects cloning must either be prohibited in the language, or there must be a run-time check. Language prohibition means something like typing. CQP used linear typing (Kobayashi et al., 1999), but their description didn't quite cover the entire language. Qtpi uses resource-checking rather like that used in separation logic (O'Hearn et al., 2001): at any point a process owns a number of qbits and can send them to others or bequeath them to its successors in a parallel composition or via a process invocation. Processes begin with the qbits passed as arguments in their invocation, and may receive others in messages. A straightforward symbolic execution can check ownership conditions:

- once a qbit has been sent in a message it cannot be mentioned again in the sending process (although the name may be in scope, its value can't be used);
- in a tuple of values, the qbits mentioned in each element must be distinct from all the others (this includes process invocation argument tuples);
- in a parallel-composition tuple of processes, the qbits used in each process must be distinct from all the others;
- measurement may destroy a qbit (the default is that it does) and a destroyed qbit can't be mentioned again;
- qbits, or values containing qbits, can't be passed as arguments in function calls or returned as results of function calls;
- 'let' can only bind a classical value;
- qbits cannot be compared.

To simplify the checking process a qtpi channel can carry either a single qbit or a classical value; there's a similar restriction on process arguments. The effect is that there is only ever a single name for a qbit in a process. It would be possible (and indeed qtpi's cloning-check mechanism implements it) to have a more permissive naming convention for qbits, but the language is simpler without it, and the restrictions easier to understand.

It's very convenient for tracing and diagnostic purposes to allow a simulation to show the user the value of a qbit. A real-world protocol could never do such a thing, so the printable state of a qbit must not be a value which a simulation can use. There is a `qval` function in qtpi which delivers a `qstate` value from a qbit, but that value can't be used in any way other than sending it down the special `outq` channel, whence it's printed.[2] In particular you can't compare `qstate` values.

12.3 Implementation

Qtpi (Qtpi) is an interpreter written in OCaml (OCa) with a straightforward implementation except for one interesting point. Choices of partner in a concurrent language raise questions of fairness. Qtpi uses synchronous message-passing, so for example attempting to read from an empty channel causes the reader to wait until some other process offers to write—and vice-versa writing to an empty channel

[2] It wouldn't have done to send the qbit itself down a printing channel, because qtpi's no cloning, no sharing restrictions would mean that it couldn't be used again.

causes the writer to wait until there is a reader. A channel can hold many offers before a partner comes along, so which offer should it take? Strict temporal order would ensure fairness, but would make a simulation grimly deterministic; one might hope for more random choice, but that raises the possibility of infinite unfairness, when a process might always overlook some offer. In Pascal-m, all that time ago, Steve Cook implemented what he called *lust*: an overlooked offer to communicate has its lust increased, and partners prefer lustier offers. Qtpi uses Cook's technique to choose the next process to run at each protocol step and to choose partners in sends, receives and guarded choices. Thus it is not temporal-order fair, but neither is it infinitely unfair, and it isn't straightforwardly deterministic.

In building a quantum calculator, since Richard was a novice he followed the instruction of Riefell and Polak (2000) and the overall quantum state is implemented in terms of amplitude vectors. But the amplitudes are symbolic numbers (the type *snum*) rather than numerical approximations. Much of the calculation is based on powers of $\sqrt{1/2}$ which the calculator calls `h` and which is also equal to $\sin\frac{\pi}{4}$ and $\cos\frac{\pi}{4}$. A great deal of formulae can be expressed in terms of powers of `h`: for example $\cos\frac{\pi}{8} = \sqrt{(1+h)/2}$.

When measuring a qbit, we must square symbolic amplitudes and compare them with a (pseudo-randomly generated) numerical probability. That will introduce some minor inaccuracy, but measurement is relatively infrequent, and it is part of a statistical calculation in any case and one might hope that its effects would average out.

The symbols used by the calculator were at first invisible to the programmer, but some recent developments (see the simulation of Grover's algorithm in Sect. 12.8) have allowed programs to construct symbolic values.

12.4 Quantum Cider

In his story "The Idyll of Miss Sarah Brown", Damon Runyon reports the advice received by Obadiah Masterson (The Sky) from his father:

> "Son," the old guy says, "no matter how far you travel, or how smart you get, always remember this: some day, somewhere," he says, "a guy is going to come to you and show you a nice brand-new deck of cards on which the seal is never broken, and this guy is going to offer to bet you that the jack of spades will jump out of this deck and squirt cider in your ear. But, son," the old guy says, "do not bet him, for as sure as you do you are going to get an ear full of cider."

Meyer (1999) describes just such an encounter:

> The starship *Enterprise* is facing some imminent—and apparently inescapable—calamity when Q appears on the bridge and offers to help, provided Captain Picard can beat him at penny flipping: Picard is to place a penny heads up in a box, whereupon they will take turns (Q, then Picard, then Q) flipping the penny (or not), without being able to see it. Q wins if the penny is heads up when they open the box.

```
proc P(s:^qbit) = + s?(y) . y>>X . s!y . _0
                  + s?(y) . y>>I . s!y . _0

proc Q(x: qbit, s:^qbit) = x>>H . s!x . s?(z) . z>>H . _0

proc System() = (newq q = |0>) (new s:^qbit) | P(s)
                                             | Q(q,s)
```

Fig. 12.1 Coin tossing as a protocol

Picard, knowing about pennies and game theory, but never having seen *Guys and Dolls*, takes the bet.

Q has no winning strategy if the box really contains a penny and all either of them can do is flip it or not. In Meyer's story the trick is that the penny is really a qubit, which means that Q can do more than Picard realises. A single unentangled qubit can be described as a column vector $\begin{pmatrix} a \\ b \end{pmatrix}$ of complex amplitudes a and b where $|a|^2$ is the probability of finding 0 when the qubit is measured and $|b|^2$ the probability of measuring 1. Q chooses $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ to stand for heads and $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ to stand for tails; $|0\rangle$ is certain to be measured as 0 and $|1\rangle$ to be measured as 1. A flip is represented by multiplication by the matrix $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ and a non-flip by multiplication by $I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. Clearly $X|0\rangle = |1\rangle$, $X|1\rangle = |0\rangle$, and $Iq = q$.

Up his sleeve Q has, instead of cider, the Hadamard matrix $H = \begin{pmatrix} h & h \\ h & -h \end{pmatrix}$, writing h for $\sqrt{1/2}$, as in qtpi's calculator. Multiplication by H, if we start with heads or tails, gives a result which is neither heads nor tails: $H|0\rangle = \begin{pmatrix} h \\ h \end{pmatrix} = |+\rangle$ and $H|1\rangle = \begin{pmatrix} h \\ -h \end{pmatrix} = |-\rangle$. These new states are each equally likely to be measured as 0 or 1: they are said to be each a quantum superposition of $|0\rangle$ and $|1\rangle$. But it's really the fact that $X|+\rangle = |+\rangle = I|+\rangle$ that's the basis of the trick. Cheating Q applies H first to make $|+\rangle$; whether Picard flips it or not it's still $|+\rangle$; then Q applies H again; $H|+\rangle = |0\rangle$ and Q wins. If Picard measures the qubit in the box he will always see 0. An ear full of cider every time.

A qtpi simulation of this game as a protocol is shown in Fig. 12.1: process P is Picard, process Q is Q. Simulation begins with the process System, which creates a qbit `q` in the chosen 'heads' state $|0\rangle$, a channel `s` for P and Q to use for communication, and splits into two parallel subprocesses. One subprocess becomes P, the other Q.

Q first sends the qbit `x` through the `H` gate (`x>>H`), calculating Hx, then sends the modified `x` down channel `s` (`s!x`) and receives a qbit `z` back (`s?(z)`) which it also multiplies by H.

P makes a random choice between two subprocesses (a guarded choice in pi-calculus terms). One subprocess receives a qbit `y` (`s?(y)`) and flips it (`y>>X`) before sending the modified qbit back down the same channel (`s!y`). The other subprocess receives `y`, doesn't flip it (`y>>I` does nothing to `y`, and could have been omitted), and sends it back unmodified.

Because P and Q share channel `s`—System gives it to them both as an argument—we can see that Q sends a qbit to P, P receives it from Q and then sends it back to Q, and Q receives it again.

All that this simulation does is to manipulate a single qbit, and in terms of quantum calculation all that happens is either $HXH|0\rangle$, if P flips the qbit-penny, or $HIH|0\rangle$, if P doesn't flip it. We'd get the right answer to those calculations if P and Q shared the qbit `q` as well as the channel `s`, provided that they took it in turns to manipulate `q`. They'd still have to send messages to take turns, but they wouldn't have to send `q` in those messages.

But qtpi doesn't allow qbit sharing, for the sake of realism. The System process gives `q` to process Q to start the game, and therefore cannot also give it to P. Q, then, begins with sole possession of the qbit which it calls `x`. Q multiplies `x` by `H`, and sends it to P. Q isn't able to use that qbit after it had sent it away, to keep part possession, so the steps that follow `s!x` aren't allowed to mention `x` at all. That is, even though there is only one qbit in the game, qtpi wouldn't accept the following as part of a program:

```
proc Q(x: qbit, s:^qbit) = x>>H . s!x . s?(z) . x>>H . _0
```

In a conventional programming language this would make sense and it would produce the right final result: the qbit value that Q receives back from P is the same one that it sent, so we could use either `x` or `z` to name it in the multiplication. But qtpi doesn't allow it: once qbit `x` has been sent away in `s!x` then it can't be mentioned again in Q, so the second `x>>H` is a resourcing error.

This simulation doesn't produce any output, but qtpi can be persuaded to provide a trace of significant events such as exchange of a message and manipulation of qbit states. Figure 12.2 shows a trace in which P happens to flip the qbit-penny using the X gate. The trace shows that Q puts qbit 0 through an `H` gate ('`0:|0>`' means 'qbit 0 with state $|0\rangle$'), and shows the result (still qbit 0 with state `(h|0>+h|1>)`, i.e. $|+\rangle$). Then channel 0 carries qbit 0 from Q to P, P puts it through an `X` gate, producing no change, and sends it back to Q. Then Q puts qbit 0 through an `H` gate again, and we finish up with qbit 0 in the same state, $|0\rangle$, that we started with.

It is possible, as we shall see below, to use output channels to show the progress of a protocol with more carefully-designed messages.

In the end this example is really trivial: it's all about the properties of $|0\rangle$, $|+\rangle$, and the gates H, X and I. It is of interest only to game theorists and novice quantum dabblers.

```
Q 0:|0> >> H; result 0:(h|0>+h|1>)
Chan 0: Q -> P Qbit 0
P 0:(h|0>+h|1>) >> X; result 0:(h|0>+h|1>)
Chan 0: P -> Q Qbit 0
Q 0:(h|0>+h|1>) >> H; result 0:|0>
```

Fig. 12.2 Sample trace of the cointoss protocol, with P carrying out a flip

12.4.1 Offside

In designing qtpi Richard borrowed Miranda's offside-parser mechanism (Turner, 1989): when parsing a phrase the parser can use page layout to determine where the phrase ends. The guarded choice in process P in Fig. 12.1, for example, uses the offside rule: each guard and its process must not extend to the space below-and-left of the guard (in the example each guard is `s?y`). That means that a '+' symbol on a following line, below and to the left of the guard, will be read as a separator and not as an addition operator continuing a previous line. The mechanism also ensures that the match doesn't need an opening or closing bracket, which is nice. Qtpi requires a '+' for each guard, a device inherited from OCaml, to emphasise the similarity between the lines. It looks ok, we think.

Qtpi uses the same offside technique to describe parallel processes: see the '`|`' operators in the System process in Fig. 12.1. Again, it allows you to stack the components of a parallel composition vertically, lining up the '`|`' separators vertically, and it means that parallel compositions don't need bracketing.

12.5 Straightforward Description

The design aim of qtpi was a programming language in which protocol programs are transparently easy to read, as easy as or easier than the original description. For example, Fig. 12.3 shows teleportation (Bennett et al., 1993) using three processes: Alice and Bob which carry out the protocol, and System which sets up the communication between them. For simplicity we first discuss teleportation of the quantum state $|+\rangle$, which is complicated enough to be interesting and simple enough to describe. A trace of the process is shown in Fig. 12.4.

The System process creates qbits `x` and `y` initialised to $|+\rangle$ and $|0\rangle$ and puts them through a CNot gate. Gating starts by calculating the tensor product `x`$\otimes$`y` of `x` and `y`, which is $(h|00\rangle + h|10\rangle)$, expressing the fact that one is equally likely to measure both qbits as 0, or the first as 1 and the second as 0—i.e. `x` could be measured as 0 or 1, `y` must be measured as 0, which is exactly what we knew already. Then the CNot

```
proc Alice (x:qbit, c:^bit*bit) =
    (newq z = |+>)
    z,x>>CNot . z>>H . z↛(vz) . x↛(vx) .
    c!vz,vx .
    _0

proc Bob(y:qbit, c:^bit*bit) =
    c?(b1,b2) .
    y >>  match b1,b2 . + 0b0,0b0 . I
                        + 0b0,0b1 . X
                        + 0b1,0b0 . Z
                        + 0b1,0b1 . Z*X .
    _0

proc System () =
  (newq x=|+>, y=|0>) x,y>>CNot .
  (new c:^bit*bit) | Alice(x,c)
                   | Bob(y,c)
```

Fig. 12.3 Teleportation of the quantum state $|+\rangle$

```
System (0:(h|0>+h|1>),1:|0>) >> Cnot;
       result (0:[0;1](h|00>+h|11>),1:[0;1](h|00>+h|11>))
Alice.1 (2:(h|0>+h|1>),0:[0;1](h|00>+h|11>)) >> Cnot;
       result (2:[2;0;1](h(2)|000>+h(2)|011>+h(2)|101>+h(2)|110>),
              0:[2;0;1](h(2)|000>+h(2)|011>+h(2)|101>+h(2)|110>))
Alice.1 2:[2;0;1](h(2)|000>+h(2)|011>+h(2)|101>+h(2)|110>) >> H;
result 2:[2;0;1](h(3)|000>+h(3)|001>+h(3)|010>+h(3)|011>+
               h(3)|100>-h(3)|101>-h(3)|110>+h(3)|111>)
Alice.1: 2:[2;0;1](h(3)|000>+h(3)|001>+h(3)|010>+h(3)|011>+
                h(3)|100>-h(3)|101>-h(3)|110>+h(3)|111>) ↛ ;
       result 1 and (0:[0;1](h(2)|00>-h(2)|01>-h(2)|10>+h(2)|11>),
                  1:[0;1](h(2)|00>-h(2)|01>-h(2)|10>+h(2)|11>))
Alice.1: 0:[0;1](h(2)|00>-h(2)|01>-h(2)|10>+h(2)|11>) ↛ ;
       result 0 and 1:(h|0>-h|1>)
Chan 0: Alice.1 -> Bob (1,0)
Bob 1:(h|0>-h|1>) >> Z; result 1:(h|0>+h|1>)
```

Fig. 12.4 Trace of $|+\rangle$ teleportation gate $\begin{pmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&0&1 \\ 0&0&1&0 \end{pmatrix}$ converts the $|10\rangle$ part of the tensor product into $|11\rangle$, producing $(h|00\rangle + h|11\rangle)$. This result is an *entanglement*: it cannot be expressed as the tensor product of two independent qbits and has the magical property that the two qbits are equally likely to be measured as both 0 or both 1—i.e. if you measure one of the pair you know for certain that you will get the same result when you measure the other.

Having entangled its two qbits, System creates a channel `c` which carries pairs of bits, and splits into two parallel subprocesses: one half becomes Alice, taking one of the qbits and the channel; the other half becomes Bob, with the other qbit and the same channel.

The Alice process creates a new qbit `z` in the state $|+\rangle$. Then she puts `z` and `x` through a CNot gate, producing quite a complicated three-way entanglement which we can see in the trace in Fig. 12.4. One of the purposes of qtpi is to carry out complicated symbolic calculation, so we shan't spend time explaining every value in the trace.[3]

Next Alice rotates `z`, now part of a three-way entanglement (effectively putting the entanglement through $H \otimes I \otimes I$), and then she measures `z`. In the trace of Fig. 12.4 Alice measures `z` as 1, recorded in `vz`, and then `x` as 0, which she records in `vx`. Measured qbits are destroyed, so the only one left is Bob's `x`, and we can see in the trace that its value is simplified as a side-effect of the measurement of `y`, becoming $|-\rangle$. Finally she sends the values she recorded to Bob on the `c` channel. Bob, sent (1,0),[4] chooses gate `Z`, multiplication by `Z` converts `y`'s $|-\rangle$ into $|+\rangle$, and the original state of Alice's `z` has been teleported to Bob's `y`.

All this is very complicated, and very specific to one execution of the protocol starting with `z` in one particular state. But qtpi allows us to experiment: what if `z` started in an unknown state $\begin{pmatrix} a \\ b \end{pmatrix}$? Could we teleport it then? The title of (Bennett et al., 1993) is "Teleporting an unknown quantum state ...": it ought to work.

If Alice begins `(newq z)`, not specifying an initial value for `z`, qtpi will create the vector $(a2|0\rangle+b2|1\rangle)$ (the '2' suffix simply indicating that `z` is qbit 2 in the simulation). All that the simulation knows about $a2$ and $b2$ is that $|a2|^2 + |b2|^2 = 1$—otherwise each is a genuinely unknown complex amplitude, and the state of `z` is therefore equally unknown. The trace which follows is shown in Fig. 12.5. The

---

[3] If you would like to follow the calculation, then there is a point worthy of mention. The three-way entanglement in the fourth and fifth lines of Fig. 12.4—`[2;0;1](h(2)|000>+h(2)|011>+h(2)|101>+`..., the result of the second Cnot step in the trace—is less complicated than it looks. First of all h^2 is just $1/2$, but the entanglement itself can be simplified. The *computational basis* $(|0\rangle,|1\rangle)$ can be viewed as a pair of unit vectors, $|0\rangle$ pointing up the y axis, $|1\rangle$ along the x axis. The *Hadamard basis* $(|+\rangle,|-\rangle)$ can be viewed as a pair of unit vectors at an angle of 45° and 135°. Viewed in the Hadamard basis, Alice's three-way entanglement is just `(h|000>+h|111>)`, a GHZ state which is an obvious three-way entanglement, and the entanglement in the trace is just that one projected into the computational basis.

[4] The trace says 1,0 but the program uses the bit constants `0b1` and `0b0`. It hasn't been possible to make bits a subtype of integers in qtpi.

```
System (0:(h|0>+h|1>),1:|0>) >> Cnot;
        result (0:[0;1](h|00>+h|11>),1:[0;1](h|00>+h|11>))
Alice.1 (2:(a2|0>+b2|1>),0:[0;1](h|00>+h|11>)) >> Cnot;
        result (2:[2;0;1](h*a2|000>+h*a2|011>+h*b2|101>+h*b2|110>),
                0:[2;0;1](h*a2|000>+h*a2|011>+h*b2|101>+h*b2|110>))
Alice.1 2:[2;0;1](h*a2|000>+h*a2|011>+h*b2|101>+h*b2|110>) >> H;
        result 2:[2;0;1](h(2)*a2|000>+h(2)*b2|001>+h(2)*b2|010>+h(2)*a2|011>+
                         h(2)*a2|100>-h(2)*b2|101>-h(2)*b2|110>+h(2)*a2|111>)
Alice.1: 2:[2;0;1](h(2)*a2|000>+h(2)*b2|001>+h(2)*b2|010>+h(2)*a2|011>+
                   h(2)*a2|100>-h(2)*b2|101>-h(2)*b2|110>+h(2)*a2|111>)  ;
        result 1 and (0:[0;1](h*a2|00>-h*b2|01>-h*b2|10>+h*a2|11>),
                      1:[0;1](h*a2|00>-h*b2|01>-h*b2|10>+h*a2|11>))
Alice.1: 0:[0;1](h*a2|00>-h*b2|01>-h*b2|10>+h*a2|11>)  ;
        result 1 and 1:(-b2|0>+a2|1>)
Chan 0: Alice.1 -> Bob (1,1)
Bob 1:(-b2|0>+a2|1>) >> { 0  1
                         -1  0 }; result 1:(a2|0>+b2|1>)
```

Fig. 12.5 Teleporting a random quantum state

calculation is far more complicated, but nevertheless it works: the result in the last step is just the state that we can see being entangled in the second.

But traces are hard to read. It's possible to print out the initial state of `z` in Alice and the final state of `y` in Bob by using output channels. Channel `out` accepts a list of strings; `outq` accepts a single `qstate`. Figure 12.6 shows Alice and Bob with output commands. The output of the expanded program is always

```
initially Alice's z is 2:(a2|0>+b2|1>)
finally Bob's y is 1:(a2|0>+b2|1>)
```

(Alice's `z` is qbit 2 in an unknown state; Bob's `y` is qbit 1 in the same state.) But the output commands in Fig. 12.6 somewhat obscure the protocol simulation, especially because they are complicated by the strange dance needed to output a message which includes the state of a qbit. So it's possible to put the output commands in separate subprocesses, using an insertion mark '$\wedge$*number*' and an associated monitor process labelled '*number*:'. The main processes are just what they were originally, with markers that can be ignored. We gain in that the monitor processes can be severely restricted in what they can do: essentially, they are only allowed to calculate and to output classical values (Fig. 12.7).

12.6 Transparent Description Isn't Easy

It's common to justify the existence of programming languages by complaining about the informality of descriptions in natural language. The description of BB84 QKD (Bennett & Brassard, 1984) begins

```
proc Alice (x:qbit, c:^bit*bit) =
     (newq z)
    out!["initially Alice's z is "] .
    outq!(qval z) . out!["\n"] .
    z,x>>CNot . z↛[H](vz) . x↛(vx) .
    c!vz,vx .
    _0

proc Bob(y:qbit, c:^bit*bit) =
    c?(b1,b2) .
    y >> match b1,b2 . + 0b0,0b0 . I
                       + 0b0,0b1 . X
                       + 0b1,0b0 . Z
                       + 0b1,0b1 . Z*X     .
    out!["finally Bob's y is "] .
    outq!(qval y) . out!["\n"] .
    _0
```

Fig. 12.6 Teleportation with output steps

```
proc Alice (x:qbit, c:^bit*bit) =
     (newq z) . ⋀1
    z,x>>CNot . z↛[H](vz) . x↛(vx) .
    c!vz,vx .
    _0
                        with 1: out!["initially Alice's z is "] .
                                outq!(qval z) . out!["\n"] .
                                _0

proc Bob(y:qbit, c:^bit*bit) =
    c?(b1,b2) .
    y >> match b1,b2 . + 0b0,0b0 . I
                       + 0b0,0b1 . X
                       + 0b1,0b0 . Z
                       + 0b1,0b1 . Z*X   .
    ⋀2
    _0
                        with 2: out!["finally Bob's y is "] .
                                outq!(qval y) . out!["\n"] .
                                _0
```

Fig. 12.7 Annotated teleport protocol

```
Alice (n, M, qcA, ...) =
      ...
    (let bs = randbits n)
    (let vs = randbits n)
    (new sent)
    | SendQbits (zip bs vs, qcA, sent)
    | sent?(_)                                      .
      ...

proc SendQbits (bvs, qcA, sent) =
    match bvs .
    + (b,v)::bvs . (newq q = ket_of_bits b v)
                   qcA!q     .
                   SendQbits (bvs, qcA, sent)
    + []         . sent!() . _0

fun ket_of_bits b v = match b,v .
                        + 0b0,0b0 . |0>
                        + 0b0,0b1 . |1>
                        + 0b1,0b0 . |+>
                        + 0b1,0b1 . |->
```

Fig. 12.8 BB84 QKD Alice invents *n* random bases and values, commands qbits to be sent

```
Alice (n, M, qcA, ...) =
      ...
   (let bs = randbits n)
   (let vs = randbits n)
   ‖: (b,v) <- zip bs vs: (newq q = ket_of_bits b v) qcA!q . _0
     ...
```

Fig. 12.9 An iteration which sends Alice's qbit train

> ... one user ('Alice') chooses a random bit string and a random sequence of polarization bases (rectilinear or diagonal). She then sends the other user ('Bob') a train of photons, each representing one bit of the string in the basis chosen for that bit position, a horizontal or 45-degree photon standing for a binary zero and a vertical or 135-degree photon standing for a binary one. As Bob receives the photons he decides, randomly for each photon and independently of Alice, whether to measure the photon's rectilinear polarization or its diagonal polarization, and interprets the result of the measurement as a binary zero or one. ...

Actually that's clear and reasonably straightforward, once you understand that 'rectilinear' is what we've been calling the computational basis and 'diagonal' the Hadamard basis. But its description in qtpi—see Figs. 12.8 and 12.10—seems to us to be more obscure. The simulated Alice first calculates (not shown) a number `n` of qbits probably sufficient to safely run the protocol on some pre-determined message `M`. Then she calculates lists of `n` random bases (each 0/1) and `n` random values (also each 0/1), zips those lists together to make a list of (base,value) pairs,

```
Bob (n,qcB,bscB) =
    ...
    (new received)
    | ReceiveQbits(n, [], qcB, received)
    | received?(bs)                     .
      received?(vs)                     .
      ...

proc ReceiveQbits (n, bvs, qcB, r) =
  if n=0 then (let bs, vs = unzip bvs)
              r!bs . r!vs . _0
        else qcB?(q) .
             (let b = randbit ())
             q ⌢̸ [if b=0b0 then I else H fi] (v) .
             ReceiveQbits(n-1, bvs@[(b,v)], qcB, r)
  fi
```

Fig. 12.10 BB84 QKD Bob receives *n* qbits, guesses bases and measures values

and splits into two subprocesses: the first invokes `SendQbits` to translate the pairs and send qbits down channel `qcA`; the second waits for a termination signal from `SendQbits` on channel `sent` to say that the job has been done, and then carries on (not shown) with the rest of the protocol by sending and receiving tagged protocol messages through channel `bscA`.

`SendQbits` gets the list of pairs plus channels `qcA` and `sent`. It takes the first pair in the list, converts it into a basis-vector constant using the function `bv_of_bits`, and creates a qbit `q` initialised according to that constant. Then it sends `q` down the channel `qcA`, and recurses to process the rest of the list. Once the list is empty, it sends a signal (the zero-tuple or 'unit' ()) on the channel `sent`, and terminates.

As a piece of programming that's fine: a straightforward recursive implementation of an iterative evaluation of a list, with the signal down the `sent` channel working rather like a continuation. But it's not a *simple* description: one would hope for something to convey the fact that Alice processes the base and values lists element by element before moving on. Pairing, pattern matching and termination signal are all unnecessary implementation detail, and a special syntactic form might convey the meaning more clearly. Recently qtpi has incorporated a repetition form—essentially a 'for' loop—illustrated in Fig. 12.9. Each iteration takes a value `(b,v)` from a list of pairs and then executes the process following the colon, which computes a qbit from `b` and `v` and sends it down channel `qcA`. Behind the scenes this uses a recursive mechanism like Fig. 12.8, but as a program it's much clearer.[5] Iterative processes need careful resource-checking: essentially, they can't do anything which affects qubit ownership such as sending a qubit down a channel or measuring it.

In Fig. 12.10 Bob is told the number `n` of qbits that will be in the train. To process them he splits into two subprocesses: `ReceiveQbits` is told the number `n`, given an empty list of information recorded so far, plus the channel `qcB` and a result channel

---

[5] Clearer, perhaps, but certainly less efficient in practice. Ho hum.

```
proc Bob (w, hks, qc, bsc) =
     (new received)
     | ReceiveQBits([],qc,bsc,received) .
     | received?(bs)   .                  (* bs, vs is what I saw *)
       received?(vs)   .

       received?(h0) . received?(bAs) . (* receive Alice's tagged bases *)
         ...

proc ReceiveQBits (bvs, qcB, bsc, r) =
  + qcB?(q) .
         (let b = randbit ())
         q ↛ [if b=0b0 then I else H fi] (v) .
         ReceiveQBits((b,v)::bvs,qcB,bsc,r)
  + bscB?(tag,bits) .
         (let bs, vs = unzip (rev bvs))
         r!bs . r!vs . r!tag . r!bits . _0
```

Fig. 12.11 Bob reads qbits until he sees the first tagged message on the classical channel

`received`. So long as `n` isn't zero `ReceiveQbits` receives a qbit on `qcB`, guesses a measurement basis and measures in that basis. Then it records its basis-guess and the measured value, putting them as a pair at the end of the list `bvs`, and recurses to receive the remaining `n`-1 photons. Once `n` reaches zero it can report to the waiting half of Bob the list of bases it guessed and the values it measured.

Again this is surely an iterative process—a tabulation of an input stream of `n` qbits—described recursively. So why does it involve a list of pairs? Why must `ReceiveQbits` be given an empty list as argument? As programmers we'd have no difficulty answering: it's because those choices make the recursive implementation easier to write. But perhaps it ought not to be like that; perhaps a protocol description ought first to be easy to read and program-minimalist only after that. We haven't yet come up with a nice iterative formulation for Bob.

That's not quite the end of this example. In the BB84 QKD simulation from the qtpi website Bob isn't told the number `n`, which Alice calculates. Instead `ReceiveQbits`—see Fig. 12.11—offers a guarded choice, reading qbits from `qcB` or the first of Alice's tagged classical messages from `bscB`. That makes it even more obscure as a protocol description. It even accumulates its measurement results in the wrong order and reverses them into the right order before sending it back to Bob.

It's not all bad vibes: the simulated Bob isn't told the number of qbits Alice will send for what seems to be the very good reason that it makes it easier to introduce an interfering Eve. To intervene, provided she uses the same `ReceiveQbits` mechanism from Fig. 12.11, all Eve must do is to read from the channel which Alice calls `qcA` and write to the channel which Bob calls `qcB`, and likewise read from Alice's `bscA` and write to Bob's `bscB`. So if the System process sets Alice and Bob up with

```
| Alice (M,qc,bsc)
| Bob   (qc,bsc)
```

```
proc Alice(M, w, hks, cMin, nSigma, qc, bsc) =

  (* Decide on a number of qbits large enough to generate the code bits
     to encrypt M, plus enough to generate 5 Wegman-Carter hash keys
     (each w bits, one for each protocol message). We don't want the
     protocol to fail because we pick too few qbits, and nSigma is the
     number of standard deviations we want to be away from that
     possibility.

     If we set nSigma, cMin and w so that Alice doesn't always calculate
     enough qbits, then she may send short messages.

     The protocol uses about n/8 checkbits, but that's allowed for in the
     formula calculation.
   *)
  (* for the basis of the calculation in min_qbits, see QKD_results.md *)
  . (let n = min_qbits (length M + 5*w) nSigma cMin)

  (* choose the basis and value for each qbit at random *)
  . (let bs = randbits n) . (let vs = randbits n)

  (* send Bob the qbits I chose, tell me when it's done *)
  . (new sent)
  | . SendQbits (zip bs vs, qc, sent)
  | . sent?(_)
```

Fig. 12.12 The actual Alice process (part 1)

they will communicate directly, but if it uses

```
| Alice (M,qcAE,bscAE)
| Eve   (qcAE,bscAE,qcEB,bscEB)
| Bob   (qcEB,bscEB)
```

then Alice and Bob will communicate via Eve. That's clear enough: the overall description is simpler, I think, at the cost of a convoluted explanation elsewhere.

But why does Bob produce a backwards list and then reverse it? That's because, as a programmer, Richard can't forget that that is the most efficient way to do it. He'd love to hide all that complexity and ingenuity inside a simple iterative description, if he could think of one.

12.6.1 The Complete Alice

Figures 12.12, 12.13 and 12.14 contain the code for Alice in the BB84 QKD simulation, shorn of its test points and logging/monitoring code; Fig. 12.15 contains the auxiliary functions she uses. Figure 12.12 has mostly been described already; the comment and the 'let' preceding the calculation of `bs` are new. Justification of the calculation of the minimum qbits is intricate, and not explained here.

```
(* tell each other the qbit bases we used - me first *)
. (let h0 = hwc bs hks 0 w) . bsc!h0,bs . bsc?(h1,bBs)

(* pick out the values for which our bases agree *)
. (let rvs = reconcile bBs bs vs)

(* Now we both know the same _number_ of values. Bob sends me a
   mask of that number of bits, and a list of the values it picks
   out from his list.
 *)
. bsc?(h2,mask) . bsc?(h3,checkbitsB)

(* test to see if Bob and I agree on the bits selected by his mask *)
. (let checkbitsA = mask_filter 0b1 mask rvs)
. (let q_check = checkbitsB=checkbitsA)

(* test for classical interference *)
. (let c_check =
      forall (checkhash hks w)
        (zip [1;2;3] (zip [h1;h2;h3] [bBs;mask;checkbitsB])))

(* Because we allow experimentation with number of checkbits, nSigma
   etc. to provoke statistical variation, it's possible to end up with
   fewer codebits that we need. So we test to see if subtracting Bob's
   checkbits has left us enough to make new hash codes and to encrypt M.
 *)
. (let enough = length rvs >= length checkbitsA + 5*w + length M)
```

Fig. 12.13 The actual Alice process (part 2)

In the BB84 QKD protocol there is a classical channel connecting Alice and Bob, called `bsc` in this process. It carries authenticated bitstring messages: a message together with a hash value calculated using a one-time hash code. The hash calculation, shown as function `hwc` in Fig. 12.15, is not at all that of Wegman and Carter (1981), but it is enough for this simulation since we don't simulate an Eve capable of trying to circumvent the authentication. Note that the messages are sent unencrypted, in the clear, but this doesn't help Eve because to fool Alice or Bob she would have to know the hash codes they are using, which are refreshed after each run of the protocol, in order to inject a message of her own.

Once the photon exchanges are complete, Alice and Bob exchange publicly, over the classical channel, the bases they use—see Fig. 12.13. They then pick out the values for which they happened to use the same basis—rectilinear/computational or diagonal/Hadamard—and discard the rest. The values are still secret. In the simulation they use the same `reconcile` function to do this. If there has been no interference they will share the same sequence of secret values.

Bob picks out a subsequence—in this simulation about a quarter—of those secret values and shares his choice publicly with Alice by sending her a mask indicating the positions in the secret value sequence which he has picked and the values he has at those positions. Then if the mask picks out the same values from Alice's sequence as from Bob's, there is some (statistical) confidence that there has been no interference

```
length of message? 4000
length of a hash key? 40
minimum number of checkbits? 500
number of sigmas? 10
number of trials? 100
(* If Alice detects interference or doesn't have enough codebits she should go silent.
   But in this simulation she sends an empty message, to allow Bob to log what he's
   done and terminate.
 *)
. if not (q_check && c_check && enough) then
    . bsc!tagged hks 4 w [] . _0
  else
    (* The (secret) code bits are the ones Bob didn't mask. We take new WC hash codes
       and a secret message code from the secret code bits.
     *)
    . (let codebits = mask_filter 0b0 mask rvs)
    . (let hks' = take (5*w) codebits)
    . (let code = take (length M) (drop (5*w) codebits))
    . (let encryptedM = xor_mask code M)
    . (let h4 = hwc encryptedM hks 4 w)
    . bsc!h4,encryptedM                  (* send the encrypted message *)
    . _0
  fi
```

Fig. 12.14 The actual Alice process (part 3)

on the quantum channel between them. And if the hash codes she received from Bob are the same in every case as the hash codes she would have calculated for the same messages, there has been no detectable interference on the classical channel.

Figure 12.14 shows what happens next. If Alice has detected interference she stops: to avoid Bob waiting for ever, which would mess up the simulation, she sends him an empty message. She does the same thing if subtracting the checkbits from the secret value sequence hasn't left enough to generate new hash codes and then encrypt `M`. Otherwise she takes the bits to construct new hash codes for the next iteration of the protocol and enough bits to encrypt `M` from the remaining secret bits, those not matched by Bob's checkbit mask.

Is the encoding transparent? Perhaps it is as transparent as it can be: even the simplest protocols require a surprising amount of code.

12.7 Performance on Examples

Qtpi can run various simulations of BB84 QKD, with various different Eve processes. Nothing remarkable is revealed: if Eve doesn't know the one-time hash codes which authenticate Alice and Bob's messages on the classical channel then there is a high probability that her interference will be detected; if she knows those hash codes then she isn't interfering, she's acting as a relay in the transmission chain.

In order to generate a one-time code to encrypt an `m`-bit message, Alice needs to send many more bits than `m`, and our simulation allows us to experiment with various parameters of her calculation to see what happens. Here is the output of an example simulation without an interfering Eve:

```
fun pos_root a b c = (-b+sqrt(b*b-4*a*c))/(2*a)

fun min_qbits k s cmin
  = ceiling (max (rootn*rootn) (rootnmin*rootnmin))
    where rootn = pos_root (3/8) (-s*(sqrt(3/32)+1/2)) (-k)
    where rootnmin = pos_root (1/8) (-s*sqrt(3/32)) (-cmin)

fun mask_filter (m:''a) (mask:''a list) (vs:'b list) : 'b list
   = vs where _, vs = unzip mvs
        where mvs = filter (λ (me,_) . me=m) (zip mask vs)

fun xor_mask code message
  = map (λ (b1,b2) . if b1=b2 then 0b0 else 0b1 fi) (zip code message)

fun packets rs (size:num) (bits:bit list) : bit list list
  = match bits .
    + [] . rs
    + _  . packets (take size bits::rs) size (drop size bits)

fun hwc message keys i w : bit list
  = if w=0 || message=[] then [0b0] else
      hwc1 message
        where hwc1 bits =
                match hps .
                + [hash] . hash
                + ps     . hwc1 (concat hps)
                  where hps = map (λ p . num2bits (bitand mask (key*bits2num p)))
                                  ps
                  where ps = packets [] size bits
        where key = nth keys i
        where size = 2*s
        where mask = bits2num (tabulate s (const 0b1)) (* 2 ^ s-1 *)
        where s = ceiling ((w+1)/3*4+1) (* odd, at least, and bigger than w *)
    fi

fun checkhash hks w (i,(h,m)) = hwc m hks i w = h

(* pick out the bits that are in the same bases *)
fun reconcile b1s b2s vs = mask_filter 0b0 (xor_mask b1s b2s) vs

fun split_codebits M w codebits =
  if 5*w+length M <= length codebits then take (5*w) codebits, drop (5*w) codebits
                                     else []                 , codebits
  fi
```

Fig. 12.15 Alice's auxiliary functions

```
length of message? 4000
length of a hash key? 40
minimum number of checkbits? 500
number of sigmas? 10
number of trials? 100

13718 qbits per trial
all done: 0 interfered with; 100 succeeded

31.20s user 0.32s system 45% cpu 1:09.51 total
```

It takes about 0.3s for each trial,[6] and overall it makes 1.3M qbit transfers and measurements in 30 CPU seconds.[7] With a naive (intercept and resend) Eve process interfering the same exchanges take 42s, and interference is detected every time. With a very short message and very few checkbits we can show that even a naive Eve can sometimes win, as we should expect.

One thing that the multi-chain version of the protocol—the one in which Eve knows the hash codes and transmits false but authenticated messages to Alice and Bob—reveals to a programmer is that in a chain Alice $->$ relay1 $->$ ... $->$ relayN $->$ Bob, all the participants including the relays see the unencrypted message, and each communicating pair uses a unique one-time encryption code. At each relay the message must be received and decrypted, using the incoming key, before being re-encrypted with the outgoing key and sent onwards. That would excite a criminal mind, we think, because it's quite unlike a classical router, which can forward a classically-encrypted message without being able to decrypt it. Classical interference with a quantum-correct relay would seem to be a criminal way forward.

The simulation of E92 QKD (Ekert et al., 1992) uses 20 000 qbit pairs per trial for the same-size problem. Because the calculations are more complicated and the calculation language is interpreted rather than compiled, it takes about 3 CPU minutes.

12.8 Some Recent Developments

Once we'd had our fill of protocols, we wondered how qtpi would handle larger entanglements. In about 3 s it's able to set up and measure one 'brick' (ten qbits, all CZ-entangled) of the measurement based quantum computing mechanism in Ferracin et al. (2018). Larger entanglements are exponentially slower: in about half an hour it can do something, but not much, with eighteen CZ-entangled qbits, just fitting into the 8GB of RAM of our test laptop. Hoping for a bigger or faster machine would, of course, be exponentially ridiculous.

2020s Covid-19 lockdown gave Richard time to focus on the implementation of quantum gating and measurement. Deploying sparse matrix techniques made an enormous difference to calculation speeds, which allowed some interesting experiments.

---

[6] Timing measurements made on an eight-year-old MacBook Air with 8GB RAM. No supercomputer involved.

[7] None of these timings is to be considered by comparison with other quantum-protocol simulators, because so far as we know there aren't any. They are included merely to illustrate that qtpi makes it possible, on a small classical machine, to simulate some reasonably large problems.

12.8.1 Grover's Algorithm

Qtpi is able to deal with Grover's algorithm (Grover, 1996), a mechanism which exploits quantum parallelism to search a database. Abstracted, a database can be seen as a function which will supply a desired item when presented with the index to that item. Abstracting further, suppose that a function f takes a vector of bits, size n, and replies 1 for one of the combination of bit values, 0 for all the rest, signalling success or failure of the search. Then to find the vector for which f will reply 1 is an unguided search through 2^n possibilities. Grover's algorithm utilises quantum parallelism to solve this problem in quadratic time, which is very many fewer iterations than a classical approach could.

An implementation in qtpi is in Fig. 12.16. The algorithm constructs two diagonal matrices `G` and `U` and uses them to rotate a vector of n qbits; after about $\pi\sqrt{2^n/4}$ iterations they probably represent the answer to the search problem. Once measured, their value can be submitted to the oracle, and if we don't have the right result then in real life we could try again.

The function `groverG`, given vector size `n`, computes the matrix `G` = $2((|+\rangle^n)(\langle+|^n)) - I^n$ and turns it into a gate. In the code $\mathtt{sx}_1$ is qtpi's symbolic representation of 1; addition, subtraction and multiplication are extended to symbolic numbers and to matrices, kets and bras made up of symbolic numbers; $\bigotimes\bigotimes$ is tensor exponentiation; `engate` and `degate` convert matrix to gate and gate to matrix. `G` has nothing to do with the particular problem being searched, except that its size is $2^n \times 2^n$.

`U`, on the other hand, is a diagonal matrix in which one particular element of the diagonal distinguishes the oracular answer to our question: `U`$|x\rangle$ is $-|x\rangle$ iff $f(x) = 1$, and $|x\rangle$ otherwise. The function `groverU`, given a bit list `bs` interpreted as the 'answer' vector, uses a special function[8] to initialise a diagonal matrix which is 1 on the diagonal except that at the position indicated by `bs` interpreted as a big-endian binary numeral it is -1.

The System process asks for the size of the problem, then computes a qbit collection—rather like an array, but treated for resource-checking purposes as a single resource—and initialises it to a ket value which represents all the possible values of its n qbits. The fourth, fifth and sixth lines calculate `G`, the answer vector, and `U`. In the seventh line `iters` is the nearest integer to the ideal number of iterations.

In the iteration on the tenth line the operator >>> puts the entire qbit collection `qs` through the `U` and then the `G` gates.[9] On the twelfth line the operator $\#$ measures the whole qbit collection `qs`, binding the bit-list result to the bit list `bs'`. Note that measuring a qbit collection is a sequence of single-qbit measurements: collections are merely a linguistic construct designed to allow resource checking.

The output is straightforward, listing first the answer and then the value computed by the algorithm:

---

[8] I.e. a filthy hack. Sorry, Samson.

[9] Nerdy fact: both `U` and `G` are diagonal matrices, but `G*U` isn't. So it's faster to do the calculation piecewise.

```
fun groverG n = engate ((sx_1+sx_1)*((|+>⊗⊗n)*(<+|⊗⊗n))-(degate I⊗⊗n))

   groverU bs = engate (tabulate_diag_m (2**n) tf
                             where n = length bs
                             where tf i = if i=address then -sx_1 else sx_1
                             where address = bits2num (rev bs) (* big-endian *)
                        )
proc
  System () =
    . (let n = read_min_int 1 "number of bits")
    . (newqs qs = |+>⊗⊗n)

    . (let G = groverG n)
    . (let bs = randbits n)
    . (let U = groverU bs)
    . (let iters = floor (pi*sqrt(2**n)/4+0.5))
    . out!["grover "; show n; " bs = "; show bs; "; "; show iters; " iterations"; "\n"]

    . ‖: i←tabulate iters (λ i. i): qs>>>U . qs>>>G . _0

    . qs⫽(bs')
    . out!["measurement says "; show bs';
         if bs=bs' then " ok" else " ** WRONG **"; "\n"]
    . _0
```

Fig. 12.16 Grover's algorithm in qtpi

```
number of bits? 10
grover 10 bs = [1;0;0;0;1;0;1;0;1;1]; 25 iterations
measurement says [1;0;0;0;1;0;1;0;1;1] ok
```

Performance is slow but perhaps acceptable for investigative purposes: a 12 qbit problem—notionally using matrices of size $2^{12} \times 2^{12}$ but in practice, because of sparse matrix techniques, using vectors no bigger than 2^{12}—can be dealt with in 10 s. 13 qbits take 40 s, and exponential increase swamps us after that.

12.8.2 W States

A W state is an entanglement between three or more qubits, in which only one qubit can measure 1 and the others must measure 0 (Dür et al., 2000). The three-qubit W state, for example, is $\sqrt{1/3}(|100\rangle + |010\rangle + |001\rangle)$: either the first qubit measures 1 and the others 0; or the second qubit 1 and the others 0; or the third qubit 1 and the others zero; and no other outcome is possible. The problem of producing an n-qbit W state can be done surprisingly quickly in qtpi, since the calculation uses matrices and vectors whose elements are mostly zero. Figure 12.17 shows the algorithm, which was adapted from one found in Microsoft's Q# repository (2020).

The `W` process is given a channel `c` and a number `n` which must be a power of 2 and $\geq$ 1; it creates a qbit collection of size `n` in the W state and sends it back through

```
(* W-state algorithm taken from the Q# Kata on superposition
    (https://github.com/microsoft/QuantumKatas/tree/main/
     Superposition/ReferenceImplementation.qs),
    task 16, WState_PowerOfTwo_Reference
 *)

fun ixs k = tabulate k (λ i. i)

fun powerceiling b n =
  pwc 1 where pwc c = if c>=n then c else pwc (b*c)

proc W (c,n) =
  if   n<=0 then (let _ = abandon ["W "; show n; " is impossible"]) . _0
  elsf n=1  then (newqs qs = |1>) c!qs . _0
  else . (let k = n/2)
       . (new c1)
       | W (c1,k)
       | . c1?(q0s)
         . (newqs q1s = |0>⊗⊗k) . (newq anc = |+>)
         . ‖: i←ixs k: anc,q0s↓i,q1s↓i>>F . _0
         . ‖: i←ixs k: q1s↓i,anc>>CNot . _0
         . dispose!anc
         . (joinqs q0s, q1s → qs)
         . c!qs
         . _0

proc Wmake (c,n) =
   (let k = powerceiling 2 n)
   | W (c,k)
   | . c?(qs)
     . out!["W "; show k; " = "] . outq!qvals qs . out!["\n"]
     . if k=n then _0
       else
         . out!["discarding "; show (k-n); " qbits "]
         . (splitqs qs → q0s(k-n),q1s)
         . q0s⫽(bs)
         . out!["which measured "; show bs; ", leaving "] . outq!qvals q1s
         . if forall (λ b . b=0b0) bs then out!["\n"] . _0
                                      else out!["  -- round again!\n"] . Wmake (c,n)

proc System () =
  . (new c) . (let n = read_num "how many qbits") . Wmake (c,n)
```

Fig. 12.17 W-state calculation

`c`.[10] If `n` is 1 it replies with a singleton qbit collection initialised to $|1\rangle$[11]; otherwise it generates a new channel `c1` and splits into two: one half recursively asks for a W-state qbit collection sized `n/2`, while the other waits to receive that collection as `q0s`.

Once it receives `q0s`, the second half makes a second `n/2`-sized qbit collection `q1s`, all $|0\rangle$, and proceeds to entangle the two collections. On the seventh and eighth lines of the `W` process are two iterative constructs. In the first iteration the qbit collections are each indexed (↓) to select a single qbit, and those qbits, plus the ancillary `anc`, are put through the Fredkin gate `F`. In the second iteration a single qbit from `q1s`,

[10] Channels may now carry a single qbit, a qbit collection, or a classical value.

[11] We found it necessary to distinguish between single qubits and singleton collections. Regrettably, we don't have a means of generating an empty collection.

plus the ancillary, is put through the CNot gate. Then the ancillary qbit is discarded, and the two collections are joined by the `joinqs` declaration. The resource treatment of qbit collections demands that after `joinqs` the input collections `q0s` and `q1s` are resource-unavailable; only the output collection `qs` can be used, and it is sent back through the reply channel `c`.

The `Wmake` process calls `W` to make a collection of qbits, entangled in a W state, whose size `k` is a power of 2. Then, if the requested size `n` is smaller than `k`, it detaches `k-n` of them using the declaration `splitqs`. Again, qtpi's resource treatment of qbit collections demands that after `splitqs` the input collection `qs` can't be used; only the two outputs `q0s` and `q1s` are available—but note that the two collections are still in a single entangled W state: `splitqs` only affects the resource checking of qbit collections. If measuring `q0s` finds that it was all $|0\rangle$, the remainder in `q1s` will be `n` qubits in a W state. The output is mostly self-explanatory (though «`1-h(4)`» in the sixth line is 3/4, the modulus of the state, displayed because in this case qtpi can't normalise):

```
how many qbits? 6
W 8 = [#0;#1;#2;#3;#4;#5;#6;#7]:
 [#7;#6;#5;#4;#3;#2;#1;#0](h(3)|00000001>+h(3)|00000010>+h(3)|00000100>+h(3)|00001000>+
                           h(3)|00010000>+h(3)|00100000>+h(3)|01000000>+h(3)|10000000>)
discarding 2 qbits which measured [0;0], leaving [#2;#3;#4;#5;#6;#7]:
 [#7;#6;#5;#4;#3;#2]<<1-h(4)>>(h(3)|000001>+h(3)|000010>+h(3)|000100>+h(3)|001000>+
                               h(3)|010000>+h(3)|100000>)
```

On larger examples the algorithm runs acceptably fast: a 256-qbit W state can be built in less than 3 s; 512 qbits takes 23 s; 1024 qbits can be processed in 186 s.[12] The matrices used in the calculations would be enormous, if they existed as conventional collections of elements. $2^{1024} \times 2^{1024}$ is far too many matrix elements to fit in the memory of a early-21st-century laptop, but sparse matrix techniques, in this case mostly representing matrices as functions, make the representation tractable. In passing we note facetiously that no quantum computer currently envisaged has anything like 1024 qubits, but when it happens we'll be ready.

12.9 Conclusions

Qtpi (qtpi) is a simulator which allows description of quantum protocols with multiple agents. It has protection, built from well-understood computer science foundations, against cloning and sharing of qbits within a simulation. Its symbolic calculator is adequate for the protocols we have examined, and can be pushed to simulate some non-protocol algorithms.

But does the world need a quantum-protocol simulator? Perhaps not yet: there don't seem to be very many protocols to simulate, or very many implementations which we might be allowed to investigate. Certainly, so far we haven't come across

[12] All CPU times: it takes much longer to print out the enormous result, and indeed at the time of writing qtpi crashes when it tries to print out the result in the 512 or 1024 qubit cases.

a protocol with bugs. But we have hopes that qtpi will become useful if quantum protocols become more popular and when distributed quantum computing becomes a thing. Whatever happens, it was fun to make it, and the symbolic calculator can help those who, like Richard when he began, quail at complicated quantum calculations.

The CQP authors decided to base their language on the pi calculus, and Richard followed suit. Experience with slightly complicated protocols suggests that, if there is a need for a protocol-description language, it ought to be a little less spare than the one that qtpi implements.

Acknowledgements After Richard had mostly implemented his own resource treatment, Simon Gay politely pointed out CQP's linear-typing treatment of no-cloning and inspired Richard to treat qbits in qtpi as fragile heavy values. Samson Abramsky thought of Pascal-m (without the mixed-choice mistake), and Steve Cook gave us lust. This research at Middlesex University was supported by UK National Cyber Security Centre through the VeTSS project "Formal Verification of Quantum Security Protocols using Coq". Raja Nagarajan was also partially funded by EU Cost Action IC1405 "Reversible Computation—Extending Horizons of Computing".

References

Abramsky, S., & Bornat, R. (1983). Pascal-m: a language for loosely-coupled distributed systems. In Y. Paker & J.P. Verjus (Eds.), *Distributed Computing Systems (Synchronisation, Control and Communication)* (pp. 163–189). Academic Press.

Bennett, C. H., & Brassard, G. (1984). Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of IEEE International Conference on Computers, Systems, and Signal Processing*, India (p. 175).

Bennett, C. H., Brassard, G., Crépeau, C., Jozsa, R., Peres, A., & Wootters, W. K. (1993). Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels. *Physical Review Letters, 70*(13).

Boender, J., Kammüller, F., & Nagarajan, R. (2015). Formalization of quantum protocols using Coq. *EPTCS, 195*, 71–83. https://doi.org/10.4204/EPTCS.195.6

Bornat, R. (1986). A protocol for generalized occam. *Software: Practice & Experience, 16*(9), 783–799.

Chen, L., Jordan, S., Liu, Y.-K., Moody, D., Peralta, R., Perlner, R., & Smith-Tone, D. (2016). *Report on post-quantum cryptography* (Vol. 12). US Department of Commerce, National Institute of Standards and Technology.

Dür, W., Vidal, G., & Cirac, J. I. (2000). Three qubits can be entangled in two inequivalent ways. *Physical Review A, 62*(6), 062314.

Ekert, A. K., Rarity, J. G., Tapster, P. R., & Palma, G. M. (1992). Practical quantum cryptography based on two-photon interferometry. *Phys. Rev. Lett., 69*, 1293–1295. https://doi.org/10.1103/PhysRevLett.69.1293.

Ferracin, S., Kapourniotis, T., & Datta, A. (2018). Reducing resources for verification of quantum computations. *Physical Review A, 98*(2), 022323.

Gay, S. J., & Nagarajan, R. (2005). Communicating quantum processes. In *32nd Symposium on Principles of Programming Languages (POPL 2005)* (pp. 145–157). https://doi.org/10.1145/1040305.1040318. Also arXiv:quant-ph/0409052.

Grover, L. K. (1996). A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing* (pp. 212–219).

Kobayashi, N., Pierce, B. C., & Turner, D. N. (1999). Linearity and the pi-calculus. *ACM Transactions on Programming Languages and Systems (TOPLAS), 21*(5), 914–947.

Kocher, P., Horn, J., Fogh, A., Genkin, D., Gruss, D., Haas, W., Hamburg, M., Lipp, M., Mangard, S., Prescher, T. et al. (2019). Spectre attacks: Exploiting speculative execution. In *2019 IEEE Symposium on Security and Privacy (SP)* (pp. 1–19). IEEE.

Meyer, D. A. (1999). Quantum strategies. *Physical Review Letters, 82*(5), 1052.

Microsoft. Q# kata on superposition, task 16, WState_PowerOfTwo_Reference, Aug 2020. Retrieved October 08, 2020, from https://github.com/microsoft/QuantumKatas/tree/main/Superposition/ReferenceImplementation.qs.

Milner, R. (1978). A theory of type polymorphism in programming. *Journal of Computer and System Sciences, 17*(3), 348–375.

Milner, R., Parrow, J., & Walker, D. (1992). A calculus of mobile processes, i. *Information and Computation, 100*(1), 1–40.

Milner, R., Tofte, M., Harper, R., & MacQueen, D. (1997). *The definition of standard ML: revised*. MIT Press.

Ocaml. https://www.ocaml.org.

O'Hearn, P., Reynolds, J. C., & Yang, H. (2001). Local reasoning about programs that alter data structures. In L. Fribourg (Ed.), *CSL 2001* (pp. 1–19). Springer. LNCS 2142.

Qtpi quantum simulator. https://www.github.com/mdxtoc/qtpi.

Rieffel, E. G., & Polak, W. (2000). An introduction to quantum computing for non-physicists. *ACM Computing Surveys, 32*(3), 300–335.

Selinger, P. (2004). Towards a quantum programming language. *Mathematical Structures in Computer Science, 14*(4), 527–586.

Shor, P. W. (1994). Algorithms for quantum computation: Discrete logarithms and factoring. In *FOCS* (pp. 124–134). IEEE Computer Society.

Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process, July 2020. Retrieved October 10, 2020, from https://csrc.nist.gov/publications/detail/nistir/8309/final.

The Q# Programming Language. https://www.docs.microsoft.com/en-us/quantum/quantum-qr-intro.

Turner, D. A. (1989). *Miranda system manual*. Retrieved from October 08, 2020, from https://www.cs.kent.ac.uk/people/staff/dat/miranda/manual/.

Turner, D. A. (1985). Miranda: A non-strict functional language with polymorphic types. In *Proceedings of a Conference on Functional Programming Languages and Computer Architecture*, New York, NY, USA (pp. 1–16). New York: Springer. ISBN 3-387-15975-4.

Wegman, M. N., & Carter, J. L. (1981). New hash functions and their use in authentication and set equality. *Journal of Computer and System Sciences, 22*(3), 265–279.

Chapter 13
Closing Bell Boxing Black Box Simulations in the Resource Theory of Contextuality

Rui Soares Barbosa, Martti Karvonen, and Shane Mansfield

Abstract This chapter contains an exposition of the sheaf-theoretic framework for contextuality emphasising resource-theoretic aspects, as well as some original results on this topic. In particular, we consider functions that transform empirical models on a scenario S to empirical models on another scenario T, and characterise those that are induced by classical procedures between S and T corresponding to 'free' operations in the (non-adaptive) resource theory of contextuality. We proceed by expressing such functions as empirical models themselves, on a new scenario built from S and T. Our characterisation then boils down to the non-contextuality of these models. We also show that this construction on scenarios provides a closed structure in the category of measurement scenarios.

Prologue

Among the many and varied facets of Samson Abramsky's work have been his contributions to the foundations of quantum mechanics. Approaching the subject through the lenses of computer science, he has brought its modes of thought and mathematical tools to bear on the analysis of natural systems, providing fresh perspectives that have illuminated the fundamental structures of the quantum world and their interaction with notions of information and computation.

In particular, some of Samson's major contributions over the past decade or so have been to the study of non-locality and contextuality. These are key phenomena that set quantum theory apart from classical physical theories and that can be shown to relate closely to quantum-over-classical advantage in computation and informa-

R. S. Barbosa (✉)
INL–International Iberian Nanotechnology Laboratory, Braga, Portugal
e-mail: rui.soaresbarbosa@inl.int

M. Karvonen
Department of Mathematics and Statistics, University of Ottawa, Ottawa, Canada
e-mail: martti.karvonen@uottawa.ca

S. Mansfield
Quandela, Palaiseau, France
e-mail: shane.mansfield@quandela.com

A. Palmigiano and M. Sadrzadeh (eds.), *Samson Abramsky on Logic and Structure in Computer Science and Beyond*, Outstanding Contributions to Logic 25,
https://doi.org/10.1007/978-3-031-24117-8_13

tion processing. Samson, his collaborators, and others have developed a general, unifying framework that shines light on structural and logical aspects at the core of these phenomena Abramsky et al. (2011), Abramsky et al. (2012a, b), Abramsky (2013), Abramsky et al. (2013), Abramsky et al. (2014a, b, c), Abramsky et al. (2015), Abramsky (2014), Abramsky et al. (2016a, b), Abramsky et al. (2017a, b, c), Abramsky et al. (2018), Abramsky et al. (2019a, b, c), Abramsky (2020), Abramsky and Barbosa (2021), Wang et al. (2021), Abramsky (2018), Abramsky (2022), Barbosa (2014), Gogioso and Zeng (2019), Barbosa et al. (2022), Booth (2021), Karvonen (2019, 2021), Aasnæss (2020), Carù (2017, 2018), Kishida (2014, 2016), Mansfield and Barbosa (2014), Mansfield (2017a, b), Mansfield and Kashefi (2018), Mansfield and Fritz (2012), Raussendorf (2013), de Silva (2018) Perhaps surprisingly, the crystallised formulation that emerges requires remarkably little from the formalism of quantum theory. Stripped down to its essentials, contextuality arises in the tension between local consistency and global inconsistency which is made possible by the limitation that one only has access to partial views of a quantum system. As it turns out, and as Samson has duly pointed out elsewhere, similar structures and concerns crop up in many other subjects, from relational databases and constraint satisfaction to natural language Abramsky (2014).

Our contribution to this volume has two main objectives. On the one hand, it presents some new results that are suggestive of how the research programme might evolve over the coming years (Sects. 13.4 and 13.5). In addition to the results themselves we provide detailed discussion of some of the open questions and research directions that arise from them (Sect. 13.6). On the other hand, it is partly aimed at providing an up-to-date exposition of some of the main ideas of the framework (Sects. 13.2 and 13.3). The adopted perspective is—we hope—somewhat original. It differs from previous expositions in that it focuses on the resource theory of contextuality developed in our recent work with Samson.

This resource-theoretic perspective has placed the emphasis on simulations, or transformations, between contextual behaviours rather than on individual instances of such behaviours; i.e. on morphisms rather than objects, to adopt the language of category theory. We will show that a novel upshot of this perspective is a uniform treatment of some important concepts from the literature: non-local games, for example, arise as particular instances of simulations.

The new contributions concern precisely the structure of this resource theory of contextuality. The 'free' transformations (i.e. the classical simulations) between contextual behaviours are characterised by regarding transformations as empirical behaviours in their own right and reducing the question of 'free'-ness to that of non-contextuality of the corresponding behaviour. In categorical language, this is achieved through internalising the hom-sets, finding a closed structure in the category of simulations.

The technical contents of the chapter are preceded by a lengthy introductory section, which aims to give a broad overview that motivates the approach and the basic ingredients of the framework. The impatient and technically-minded reader may safely skip it and jump straight into the weedier pastures of definition–theorem–proof land. The more leisurely reader at the opposite extreme may be tempted to stop at the introduction, in which we have aimed to convey the central ideas, and our hope is that they won't leave empty-handed.

We have endeavoured to make the chapter accessible to a broad readership including logicians, computer scientists, physicists of a foundational bent, and superpositions thereof. In particular, no knowledge whatsoever of quantum mechanics is assumed or even used: it really is all just about partial information. We hope that everyone—including Samson—will be able to find something of interest here.

13.1 Introduction

A behavioural lens

Systems will be considered from a purely *observational* or *operational* perspective. As such, a system will simply be treated as a black box with which an external agent can interact. Computer scientists may think of these interactions as the posing of queries and the obtaining of responses, like in the calling of functions in a programming language. Physicists may prefer to think of interactions as the performing of measurements and obtaining of outcomes. From this perspective the states of a system are not defined *intrinsically* or *a priori*, but rather they are descriptions of the empirically observable behaviour in every allowed interaction.[1]

In this, computer scientists may be familiar with the terms observational or behavioural as opposed to *state-space based* approaches, while physicists may recognise a similar distinction between operationalism and realism. At the same time, it may be worth commenting that adopting this perspective need not entail any deep philosophical commitment, but merely a methodological one. For on the one hand 'operationalism is, at least, a useful exercise for freeing the mind from the baggage of preconceptions about the world' Hardy and Spekkens (2010), while on the other it is an apt approach when one's primary concern is the practical one of understanding how best to utilise the systems under consideration, as is typical in quantum information and computation.

Attention will be limited to *single-use* black boxes. The agent is permitted only one round of interaction with the system, which is subsequently unusable. This kind of 'one-shot' limitation is typical in interactions with quantum systems, where measurements are often destructive, rendering the system unavailable for future use. In terminology more familiar to computer scientists, there is no state update or continuation. Equivalently, we may think of black boxes that are reinitialised after each round of interaction. Despite the apparent poverty of a setting shorn of sequential interactions and transformations, it is in fact already rich enough to capture salient

[1] At the risk of provoking a relapse into an erstwhile indulgence of Samson's, who has been known to self-identify as a recovering philosopher, we remark that this chapter adopts an approach that is somewhat in the empiricist tradition of philosophy. As someone whose research interests and contributions are ever-evolving and indeed ever-relevant, one former student of his has also pointed out that Samson has clearly distinguished himself from another famous Samson, the dinosaur.

features that set quantum systems apart from classical ones (indeed to capture the most well-loved and well-studied of these features).

The interface, or *type*, of a black box specifies a finite set of basic queries or measurements and their respective sets of possible responses or outcomes.[2] The agent consumes the box by simultaneously performing a subset of these measurements. Of specific interest will be situations in which not all subsets of measurements can be performed jointly. Such a limitation could be imposed as a design feature: e.g.security concerns may dictate that some combinations of database attributes not be simultaneously accessible. Similarly, it could result from a lack of experimental resources: e.g.having a limited number of measurement devices or detectors in a physics experiment. However, our primary motivation comes from quantum theory, where the limitation reflects a more fundamental restriction stemming from the incompatibility of certain combinations of measurements, in the sense of the uncertainty principle. Performing one measurement may spoil the possibility of performing another by disturbing its outcome statistics.

In such a situation it is not possible for the agent to freely examine *all* the observable properties of the system, as would be matter of course in the setting of classical physics. Instead, their access to the system is limited, mediated by sets of jointly-measurable observables, called **contexts**. Each context provides a partial (classical) window into the (quantum) system. And it is only through the varied collection of partial points of view afforded by these windows that one may infer about the system as a whole.

This limitation calls to mind the situation described in the opening passage of Peter Johnstone's compendium on topos theory, *Sketches of an Elephant* Johnstone (2002):

> Four men, who had been blind from birth, wanted to know what an elephant was like; so they asked an elephant-driver for information. He led them to an elephant, and invited them to examine it; so one man felt the elephant's leg, another its trunk, another its tail and the fourth its ear. Then they attempted to describe the elephant to one another. The first man said 'The elephant is like a tree'. 'No', said the second, 'the elephant is like a snake'. 'Nonsense!' said the third, 'the elephant is like a broom'. 'You are all wrong,' said the fourth, 'the elephant is like a fan'. And so they went on arguing amongst themselves, while the elephant stood watching them quietly. [...] But the important thing about the elephant is that 'however you approach it, it is still the same animal'...

We might imagine that the debate would soon resolve itself if only each of the participants were to move around the elephant and bit-by-bit—or sketch-by-sketch—build up a more global picture of the elephant. Yet, the limitation that only partial empirical information about the system can ever be obtained at once gives rise to an altogether more intriguing set of possibilities. After all, the elephant in this story might be thought of as an essentially classical beast. And so in some respects this is the point at which the real fun begins.

[2] From now on, we shall primarily adopt the terminology measurements and outcomes.

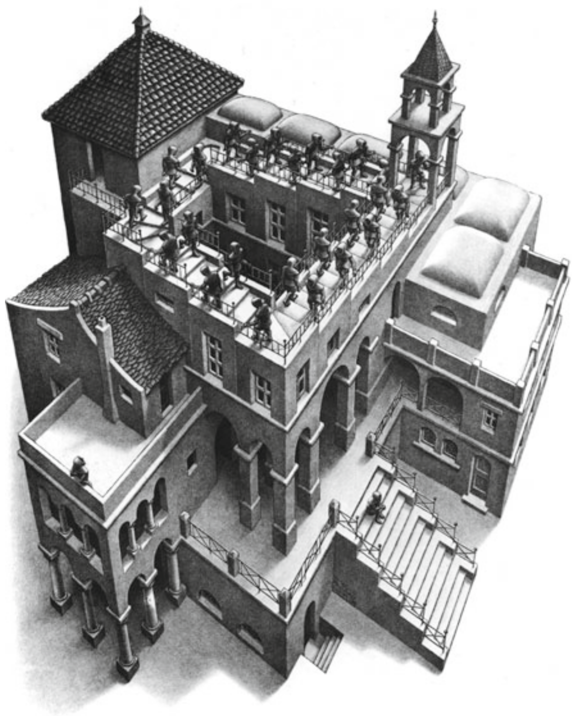


Fig. 13.1 M. C. Escher, *Klimmen en dalen* (Ascending and descending), 1960. Lithograph, 285 mm × 355 mm

Contextuality: 'at the Borders of Paradox'

The idea is that even though the viewpoints afforded by overlapping contexts may fit nicely together there can nevertheless be situations in which it is impossible to paste *all* of them into a consistent global picture. Such a gap between **local consistency** and **global inconsistency** gives rise to an apparent paradox, an instance of which is beautifully illustrated by M.C. Escher's lithograph *Klimmen en dalen* reproduced here in Fig. 13.1.[3]

More concretely, the behaviour of a black box is described by an **empirical model** e which details its (probabilistic) response to any allowed interaction. For each allowed context of measurements C, it specifies a probability distribution e_C over the set of available joint outcomes. We will encounter several examples of empirical models in what follows (see Examples 8–12).

The distributions that make up an empirical model are naturally constrained to be 'locally' compatible in the following sense. If U is a subset of a context C, one could jointly perform the measurements in C but then forget the outcomes of those measurements not in U. This would yield joint outcomes to the measurements in U

[3] This is similarly illustrated by the 'impossible biscuit' in the poster for the 2018 Lorentz Centre workshop *Logical Aspects of Quantum Information*, which was co-organised by Samson: https://www.lorentzcenter.nl/logical-aspects-of-quantum-information.html.

with probabilities given by the marginal distribution $e_C|_U$. The local compatibility requirement is that for any two contexts C and C' containing U, the marginals $e_C|_U$ and $e_{C'}|_U$ coincide. In other words, e_C and $e_{C'}$ *agree on their overlap*, $e_C|_{C\cap C'} = e_{C'}|_{C\cap C'}$. Thus, the probabilistic empirical behaviour observed for the measurements in U is the same regardless of whether they arise as a subset of the context C or of the context C'.

It is like saying that sketches of the system should agree, or fit together, wherever they overlap. This is a property that holds of empirical models that arise in physics. It both ensures a compatibility with basic tenets of relativity in certain scenarios (see discussion of Example 10) and justifies the independent labelling of individual measurements.

A more '*global*' notion of compatibility asks that these locally compatible probability distributions can be 'glued' together into a global probability distribution over joint outcomes to *all* of the measurements at once. Concretely, this global distribution would yield the empirical probability distribution e_C when marginalised to each context C.

Existence of such a global distribution allows one to think of the state of the system as a probabilistic mixture of deterministic states that assign definite values to all observables. Such deterministic states could be empirically inaccessible, which is why they are often referred to as *hidden* variables or as *ontic* states (as opposed to epistemic or empirical ones). But even if the agent is not in fact allowed to perform all the measurements at once, they would have no reason to doubt that these have predetermined would-be outcome values at a more fundamental, though inaccessible, level, and that these values exist independently of the agent's choice of measurement context. This is like asking that taken together the sketches provide a coherent picture of the system as a whole—or, to borrow Einstein's idiom, asserting that the moon is there even when one is not looking at it Pais (1979), Mermin (1985). Yet, examples of violation of the global consistency condition can be found. In the case of quantum systems, they preclude such a conceptually neat and intuitive understanding of the underlying states of the system, and thus of the physical world we inhabit.

Empirical models assign probabilities to observable events. When these empirical probabilities satisfy local but not global consistency they are said to be **contextual**. As has been pointed out by Pitowsky (1994), and further illuminated by Samson (2020), one could say that contextuality was anticipated as far back as in Boole's work on the 'conditions of possible experience' Boole (1862). Boole derived inequalities that the probabilities of logically related events must satisfy in order to be, in our terms, globally consistent. Of course, as the terminology indicates, Boole, sitting in his rain-pelted quarters in what was then Queen's College, Cork, would have believed that these conditions must be satisfied by real-world experiments. However, we now know that this is not the case.

It was roughly a century later in the surroundings of CERN that John Bell, an alumnus of another Queen's on the island of Ireland, showed that the laws of quantum theory predict empirical models that exhibit what we here call contextuality. This is the surprising content of the celebrated Bell (1964) and Bell–Kochen–Specker (1966), Kochen et al. (1967) theorems. By now it has been confirmed that the measurement statistics obtained in a variety of experiments witness a violation of Boole's conditions, recast and renamed as Bell inequalities: e.g.Aspect (1982), Kirchmair et al. (2009), Hensen et al. (2015), Shalm et al. (2015), Giustina et al. (2015).

A strong result connecting Boole to Bell, proved by Samson and Lucien Hardy, shows that a complete set of inequalities characterising the polytope of non-contextual empirical models can be derived from logical consistency conditions Abramsky et al. (2012a). This should strike the reader as rather alarming: the observed behaviour of physical systems can apparently satisfy an inconsistent set of logical formulae! The saving grace in this drastic situation is that, since observations may only arise in context, one can never observe enough events at once to manifest the inconsistency. In relation to this Samson has often been heard to say that quantum systems skirt the borders of logical paradox.[4]

It is worth stressing that contextuality is a property of the observable behaviour of a system. It is independent of whatever theory, quantum or otherwise, that might be conjectured to account for the behaviour. The power of the aforementioned seminal theorems of quantum foundations is that contextuality cannot simply be dismissed as some bizarre artifice of an incomplete or inadequate mathematical formulation of a physical theory Einstein et al. (1935)—it is an unavoidable feature in *any* theory that accounts for the empirical behaviours that have been observed in experiments. This is what grants contextuality its *phenomenological* status, and what justifies its being tested experimentally. No matter how skilled the sketch artists are, a picture of the whole elephant will always elude them—a true elephant in the room.

Laying foundations for quantum information

While Bell's theorem laid bare this counter-intuitive aspect of quantum behaviours, less conclusive clues about the theory's counter-intuitive nature had already been noticed for some time and had been a source of philosophical or interpretational unease for several of its founding contributors: e.g.the EPR paradox Einstein et al. (1935), or the Schrödinger's cat thought experiment Schrödinger (1935).

But surprising, or troubling, as quantum theory may be in this respect, perspectives on the matter have broadened in recent decades. It was only a matter of time before one of the great slogans of programming and hacker culture came to be applied to quantum theory too: 'it's not a bug, it's a feature!'. The idea is that the use of quantum systems to carry and manipulate information opens up the possibility of exploiting their non-classical *weirdness* to attain advantage in computational or other information-processing tasks.

This perspective has led to a renewal of interest in foundational aspects of quantum theory, as we strive for a systematic and effective understanding of quantum advantage. To get there requires being able to reason about information processing at the quantum level, with all the common-sense-defying possibilities it offers.

Contextuality, as the archetypal non-classical feature of quantum phenomenology, has drawn particular attention in this regard. This has led to the develop-

[4] This also brings to mind the words of Álvaro de Campos, as if quantum systems were contriving to realise the motto from his futurist phase, 'to be sincere contradicting oneself'. In the original: 'Ser sincero contradizendo-se'. From the poem *Passagem das horas* (22–05–1916), in Campos (1944).

ment of general, structural frameworks for treating contextuality, including the sheaf Abramsky et al. (2011), graph Cabello et al. (2014), hypergraph Acín et al. (2015) and contextuality-by-default Dzhafarov et al. (2014) approaches. These interrelated frameworks go well beyond the *ad hoc* analysis of particular instances of the phenomenon or the hunt for 'small proofs' of the Bell–Kochen–Specker theorem, topics that had been the main preoccupation of 'all hitherto exisiting' literature on contextuality. Instead, the focus is on developing a general theory that distils the essence of contextuality and reveals its structural and compositional aspects. The framework introduced by Samson and Adam Brandenburger Abramsky et al. (2011) is a particularly potent distillate that emphasises these aspects and characterises contextuality as obstructions to the passage from local to global. The framework gives elegant expression to this idea through the mathematics of sheaf theory.[5,6]

These structural frameworks have provided the basis for a range of recent results that establish links between contextuality and quantum advantage Raussendorf (2013), Howard et al. (2014), Abramsky et al. (2017c), Bermejo-Vega et al. (2017), Abramsky et al. (2017b), Mansfield and Kashefi (2018), Karanjai et al. (2018), which have prompted further investigation into the rôle of contextuality as a resource.

Resources: from Objects to Transformations

In previous work with Samson we built upon the sheaf-theoretic framework to develop a compositional resource theory of contextuality Abramsky et al. (2017c), Karvonen (2019), Abramsky et al. (2019a) (cf. Amaral et al. (2018), Amaral (2019)). The central notion is that of simulation between empirical behaviours, which rests on an underlying notion of experimental procedure between black boxes.

The scenario to have in mind is the following. Imagine an agent who has access to a black box of the kind described before. They can perform experiments by interacting with the box. A recipe for such an experiment specifies the agent's actions surrounding their interaction with the box.

As we are only considering single-use boxes, the instructions are of a rather simple form: they specify a set C of compatible measurements to be performed and a post-processing function mapping the set of possible joint outcomes of these measurements to a (new) set of outcomes for the experiment. For example, a recipe for an experiment could read something like 'perform the compatible, dichotomic measurements a and b simultaneously, obtain their (Boolean) outcome values, and combine them with XOR to yield the (also dichotomic) result of the experiment'. One could also include probabilistic mixtures of such deterministic experiments.

The agent may then package a collection of such experiments as a new (outer) box, offering these derived measurements to external users through an interface (Fig. 13.2). They must of course be careful to ensure that this external interface only labels a set of (new) measurements as compatible if the agent is able to perform

[5] Within these frameworks, the phenomenon of non-locality as discussed by Bell may be seen as a special case of contextuality that arises in distributed or multi-party scenarios. Note that locality in Bell's sense differs from our use of the term earlier in relation to local compatibility.

[6] Subsequent developments are to be found in many papers including in particular the local-consistency-versus-global-inconsistency picture in Abramsky et al. (2015).

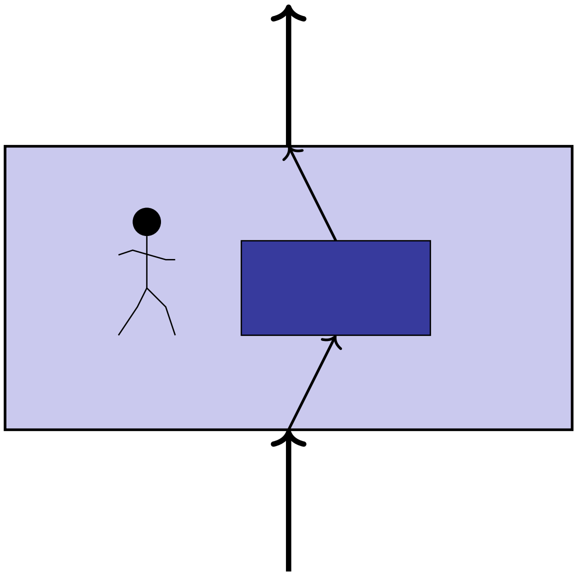


Fig. 13.2 An experimental procedure uses black box of one type to simulate a black box of another type

all the corresponding experiments in parallel without running into the compatibility limitations of the original black box.

In a slightly synecdochical abuse of terminology, we call such a collection of instructions for experiments an **experimental procedure** (or just procedure, for short). It is a procedure for using the original box in building the new, wrapper box. Through such a procedure, any behaviour (i.e. empirical model) of the original box is converted into a behaviour of the new box. However, it should be kept in mind that not every possible behaviour of the new type is necessarily achievable in such a manner.

Setting the physics language aside for a moment, the kind of situation just described will be very familiar to computer scientists. It arises all the time, for example, in modular programming: one uses the functions or methods provided through an API by a library (or by an object in object-oriented programming), whose implementation might be hidden from us, in order to implement new functions or methods, which may in turn be packaged as a new program module or library (or as a 'wrapper' object) and provided to other programmers.

Simulations: Inside-Out and Outside-In

There is an alternative way to think about the notion of procedure, approaching it 'from the outside in', rather than 'from the inside out'. The difference is that between synthesis and analysis: instead of 'A is converted to B', one says that 'B is simulated from A' or that 'B reduces to A'.

From the perspective of someone external to the new box, for whom it is just a black-box system, the procedure followed by an internal agent may be posited as a (typically incomplete) explanatory device for the empirical behaviour of the system.

This could be helpful, for example, in reducing the box's behaviour to the behaviour of another black box, perhaps one of a simpler kind or one that is more familiar and well-understood. Indeed, as the agent's actions are fully classical (non-contextual), they essentially describe a simulation of the empirical behaviour of the outer box from the behaviour of a posited inner box.

It may be noted that the agent in this story plays a double rôle. In relation to the original black box they play the rôle of a user or of the environment—they may choose measurements and then obtain the respective outcomes. But in relation to the wider world outside of the new box, they play the rôle of a system—they are prompted with measurement requests and must produce an outcome.[7]

To What End?

In summary, taking the perspective of resource theory, the emphasis is no longer placed on the individual black box behaviours but on simulations between different instances of such behaviours. There are a number of reasons that recommend this approach in the study of contextuality.

- At least implicitly or informally, the notion of simulation has been central to a number of results in the non-locality and contextuality literature, e.g.Barrett and Pironio (2005), Barrett et al. (2005), Jones and Masanes (2005), Dupuis et al. (2007), Allcock et al. (2009), Forster and Wolf (2011). A more explicit and structural formalisation of the concept can be useful for proving further results of this kind. Good examples of this are the no-copying result in Abramsky et al. (2019a, Theorem 22) and the more general no-catalysis result in Karvonen (2021).
- More broadly, resource theories provide a versatile setting that has already proved useful in exploring a variety of other resources, such as entanglement, in the field of quantum information, e.g.Horodecki and Oppenheim (2013), Chitambar and Gour (2019). A more general mathematical framework encompassing these and many other examples can be found in Coecke et al. (2016), Fritz (2017). As with classical notions of reducibility, the existence of a simulation between one behaviour and another provides a way of comparing their degrees of contextuality. The induced preorder is richer, more expressive, and provides more structural and fine-grained distinctions than the linear order induced by a 'measure of contextuality' such as the contextual fraction or others Abramsky et al. (2017c), Grudka et al. (2014). In fact, there is not just one but a hierarchy of such preorders, analogous to the Abramsky–Brandenburger hierarchy of probabilistic, logical, and strong contextuality Abramsky et al. (2011). These preorders are determined by how flexible a notion of simulation one one wishes to consider. Ultimately, it is these preorders that should be regarded as being the fundamental concepts, while the various 'measures' (or at least the linear orders they induce) are just somewhat arbitrary linear extensions of them.

[7] One cannot help but be reminded of the reversal of Player and Opponent rôles in games of function type in game semantics. Player in such a game plays simultaneously, and inter-dependently, two simpler games, corresponding to the output and the input types, and adopts a different rôle in each of them Abramsky et al. (1999).

- The resource-theoretic framework of simulations provides a unifying language in which several important concepts from the contextuality literature find common expression. The most striking examples are provided by two ‘corner cases’: non-contextual models and Bell inequalities.

  – Within our framework, non-contextual behaviours are naturally characterised as those that can be simulated from ‘nothing’, i.e.from the unique model on the empty black box that allows for no measurements. More precisely, non-contextual models on any given box are in one-to-one correspondence with (probabilistic) procedures from the empty box.[8]
  – A novel, and perhaps surprising, fact that we explore in this chapter is that the well-studied notions of Bell inequalities and non-local games, suitably generalised to apply not just to non-locality but to contextuality more broadly, can also be described as experimental procedures. The key rôle here is played by the box ‘[2]’ that admits a single measurement with binary outcome. Its possible behaviours are thus characterised by a single number in the unit interval, specifying the probability that the outcome is 1. A procedure from a given box to the box [2] corresponds to a (normalised) Bell functional (the ‘left-hand side’ of a Bell or non-contextuality inequality) or to a non-local or contextual game. An empirical model on the given box is mapped through such a procedure to a number in [0, 1]—this is the value of the functional or the winning probability of the game.

- Last but not least, the shift in emphasis from single boxes to transformations between them is very much in the spirit of category theory. Category theory stems from the recognition that mathematical objects, to reappropriate and paraphrase the words of John Donne[9], are best studied not as islands entire of themselves but as pieces of the continent, parts of the main. The same reasoning can be applied to ‘systems’, physical or otherwise. In other words, what matters is not so much what things are, but how they stand in relation to one another. This perspective has proved to be immensely fruitful not only in mathematics but also in computer science, as a way to systematise concepts, recognise structure, and build bridges between *a priori* disparate subjects. Having a category—of boxes and procedures, or of behaviours and simulations, in this case—provides us with a powerful tool to think about the structure of the theory at hand. As an example, the results in this chapter were in a sense motivated from categorical considerations—but nevertheless they may still be stated more concretely, without the high-flown language.

---

[8] Note that in Abramsky et al. (2019a, b), where we first made an observation to this effect, the source of these simulations was the trivial scenario with one measurement and a single outcome. The difference arises due to the kind of simulations we allow in each case. It is related to the fact that ‘the’ singleton set is the terminal object in the category of sets and functions (i.e. there is exactly one function from any given set to a singleton set) whereas in the category of sets and relations the terminal object is the empty set.

[9] From *Meditation XVII*, in Donne (1624).

Summary of Results

A Question: from Objects to Maps

We now turn to an overview of the main results in this chapter. A procedure f that uses a box of type S to build one of type T determines a *state transformation*, a function $\mathsf{EMP}(f)\colon \mathsf{EMP}(S) \longrightarrow \mathsf{EMP}(T)$ mapping any empirical model for S to one for T. It is natural to ask which functions arise in this way. We are thus led to the following question:

$$\textit{Given a function} F\colon \mathsf{EMP}(S) \longrightarrow \mathsf{EMP}(T), \textit{can it be realised by an experimental procedure? I.e. is there a procedure } f\colon S \longrightarrow T \textit{ such that } F = \mathsf{EMP}(f)? \tag{A}$$

In light of the remarks made above, when S is the empty box, its set of empirical modes $\mathsf{EMP}(S)$ is a singleton. So, a function $\mathsf{EMP}(S) \to \mathsf{EMP}(T)$ simply picks out an element of $\mathsf{EMP}(T)$. The question then reduces to a more familiar one:

$$Given an empirical model, is it non-contextual? \tag{B}$$

This latter question of detecting contextuality of a given empirical model, i.e. from the observable probabilistic behaviour of a black box, has been extensively studied and is by now well understood. The more general question (A) can be seen as a relativised version of (B). The generalisation is somewhat in the spirit of the 'relative point of view' Bartels (2010), advocated by Grothendieck in the context of algebraic geometry. The idea is that the consideration of properties of objects gives way to the consideration of properties relativised to morphisms. This results in fact in a strict generalisation, as properties of objects can be then viewed as properties of the unique morphism from that object to the terminal object.

An Answer: from Maps to Objects

We provide an answer to question (A) in the form of necessary and sufficient conditions for a map between empirical models to be realisable by an experimental procedure (Theorem 44). We now give a brief summary of the main ingredients.

As a preliminary observation, note that the set $\mathsf{EMP}(S)$ of empirical models, or possible behaviours, of any given box S is a convex set. In other words, it is closed under probabilistic mixtures: for any two conceivable behaviours, a mixture of them is also a behaviour. Operationally, one can think that whoever prepares the single-use copies of the black box does so by first throwing a (biased) coin and then preparing it to behave according to one or the other. Moreover, any behaviour transformation that is induced by a procedure preserves convex combinations. We may thus restrict attention, in answering question (A), to functions $F\colon \mathsf{EMP}(S) \longrightarrow \mathsf{EMP}(T)$ with this property.

Past this initial hurdle, we get to the crux of our characterisation. The short summary is that (A) is answered by reducing it back to (B). This is achieved by representing behaviour transformations as behaviours themselves. Modulo a few details, this goes as follows. For any boxes S and T, we build a new box of (function) type, $[S, T]$. Its possible behaviours, the empirical models in $\mathsf{EMP}([S, T])$, represent convex-combination-preserving transformations $\mathsf{EMP}(S) \to \mathsf{EMP}(T)$. And those empirical models turn out to be non-contextual precisely when these transformations are realisable as procedures $S \to T$.

In order to make this work, however, we need a refinement of our notion of box. It consists of an added specification that restricts the allowed behaviours of the box, somewhat akin to a type or class invariant in programming. One may think of it as a contracted promise that comes attached to the box interface and which any behaviour must fulfil. Concretely, this is given as a *predicate*, implemented as a two-valued experiment to which the allowed behaviours are pledged to always return the outcome 1. In terms familiar in the quantum foundations and information literature, the behaviours must be perfect strategies for a given (non-local or contextual) game. So, the construction of the box $[S, T]$ also involves specifying such a predicate, $g_{S,T}$.

The box $[S, T]$ has the following interface: measurements correspond to those of T, while outcomes specify procedures for interacting with the box S in order to obtain an outcome in O_T (the outcome set of T). The rôle of the predicate $g_{S,T}$ is to ensure that these procedures never lead to an invalid use of S. In particular, following any set of procedures obtained as joint outcomes from the box $[S, T]$ should only ever require measuring a context of compatible measurements in S.

The full answer to (A) is then stated as follows. Given a map $F \colon \mathsf{EMP}(S) \longrightarrow \mathsf{EMP}(T)$ that preserves convex combinations we find a corresponding empirical model e_F of $[S, T]$. The transformation F is realised by a procedure $S \to T$ if and only if e_F satisfies the predicate $g_{S,T}$ and is non-contextual.

The construction outlined above suggests that one regard the pair $\langle [S, T], g_{S,T} \rangle$ as somewhat akin to a function space, or more accurately a space of procedures between black boxes. And indeed, if one considers the category whose objects are boxes with invariant specifications, then this construction provides it with a closed structure Eilenberg and Kelly (1966), Laplaza (1977), Street (1974), Manzyuk (2012) (Theorem 46). This is an internalisation of the notion of hom-set, whereby the collection of morphisms between any two objects is in a precise sense represented as an object in the category itself.[10]

13.2 The Framework: Objects

We now introduce more carefully the framework that will be used throughout this chapter. The basic setting that we outline in this section is that of the sheaf-theoretic approach to contextuality introduced by Abramsky and Brandenburger (2011). The

[10] At the risk of overstretching the use of poetic metaphor, one is reminded of Blake's, '[to] hold infinity in the palm of your hand, and eternity in an hour.' From the poem 'Auguries of Innocence' (c. 1803), in The Ballads (or Pickering) Manuscript, published in Gilchrist and Gilchrist (1863).

formalism provides general notions of measurement scenarios and empirical models, respectively the *types* or *interfaces* and the *states* or *behaviours* of our black boxes. In addition, a central rôle will be played by morphisms between these basic objects, which were introduced in our previous work to underlie a resource theory of contextuality Karvonen (2019), Abramsky et al. (2017c, 2019a), on which we elaborate further in the next section.[11]

13.2.1 Measurement Scenarios

The first ingredient is the notion of measurement scenario. This is a specification of the interface of a black box: which queries (measurements) are available to the agent and which type of response (outcome) they can elicit. As such, measurement scenarios serve as the *types* (in the computer science sense) in our framework.

As discussed in the introduction, the compatibility structure of measurements plays a central rôle. The scenario specifies which subsets of measurements form contexts and can thus be jointly performed. Note that if C is a context, then any subset U of C must also be a context: the agent might as well jointly perform all the measurements in C and disregard the outcomes of those in $C \setminus U$. Moreover, it must be possible to perform any individual measurement on its own. These two desiderata are neatly encapsulated in the notion of abstract simplicial complex, which is important in algebraic topology and combinatorics.[12]

Definition 1 An (abstract) **simplicial complex** Σ on a set of vertices X is a family of finite subsets of X, called faces, which is non-empty, downwards-closed, and contains all the singletons. That is:

- $\emptyset \in \Sigma$;
- for all $x \in X$, $\{x\} \in \Sigma$;
- if $\sigma \in \Sigma$ and $\sigma' \subseteq \sigma$, then $\sigma' \in \Sigma$.

Besides the measurements and their compatibility structure, a measurement scenario must also specify the available outcomes for each measurement. Putting these ingredients together, we arrive at the following definition.

Definition 2 A **measurement scenario** is a triple $S = \langle X_S, \Sigma_S, O_S \rangle$ where:

- X_S is a finite set ,whose elements are called **measurements**;
- $O_S = (O_{S,x})_{x \in X_S}$ is a family that specifies, for each measurement $x \in X_S$, a finite non-empty set $O_{S,x}$, whose elements are the **outcomes** of x;
- Σ_S is a simplicial complex on X_S, whose faces are called the **measurement contexts**.

[11] In fact, the notion of morphism considered here differs slightly from those of Karvonen (2019) and Abramsky et al. (2019a). It is the appropriate notion to capture *non-adaptive* simulations. The minor discrepancies are discussed and explained at the end of Sect. 13.3.2.

[12] Otherwise it would be difficult to justify calling it a measurement in the first place.

We now introduce a couple of simple examples that will be useful for illustrating the main concepts.

Example 3 The (Auld) **Triangle** scenario $\triangle$ has three measurements (or queries),

$$X_{\triangle} := \{\textit{`pint?'}, \quad \textit{`wine?'}, \quad \textit{`grub?'}\}.$$

The contexts admit no more than a pair of measurements to be performed at once; i.e. the maximal faces of $\Sigma_{\triangle}$ are

$$\{\textit{`pint?'}, \textit{`wine?'}\}, \quad \{\textit{`wine?'}, \textit{`grub?'}\}, \quad \{\textit{`pint?'}, \textit{`grub?'}\}.$$

The outcomes (or responses) to each measurement $x \in X_{\triangle}$ take values in a two-element set,

$$O_{\triangle,x} := \{\textit{`yes'}, \quad \textit{`no'}\}.$$

This scenario can be found, albeit with different labels for measurements and outcomes, in many other articles. It is the simplest scenario in which contextuality can arise. In particular, it is the scenario for Specker's parable of the overprotective seer Liang et al. (2011) (Fig. 13.3).

Example 4 The **Four Candles** scenario $\square$ has four measurements (queries),

$$X_{\square} := \{\textit{`SammyA'}, \quad \textit{`GeorgieB'}, \quad \textit{`JohnnyB'}, \quad \textit{`EvilG'}\}.$$

The maximal contexts, i.e. the maximal faces of $\Sigma_{\square}$, are

$$\{\textit{`SammyA'}, \textit{`GeorgieB'}\}, \quad \{\textit{`GeorgieB'}, \textit{`JohnnyB'}\}, \quad \{\textit{`JohnnyB'}, \textit{`EvilG'}\}, \quad \{\textit{`SammyA'}, \textit{`EvilG'}\}.$$

The outcomes (or responses) to each measurement $x \in X_{\square}$ take values in a two-element set,

$$O_{\square,x} := \{\textit{`grain'}, \quad \textit{`grape'}\}.$$

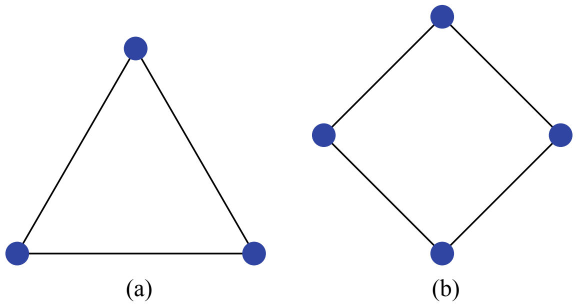


Fig. 13.3 Simplicial complexes representing measurement compatibility in **a** the $\triangle$ scenario and **b** the $\square$ scenario

We might think of the scenario as occurring when an agent needs to buy a round of drinks for his four friends, but only gets to interrupt the conversation enough to ask neighbouring pairs what they would like.

This scenario, again with different measurement and outcome labels, is better known in the physics literature as the CHSH scenario. It is the scenario concerned by perhaps the most well-known of Bell inequalities, originally formulated by Clauser, Horne, Shimony, and Holt Clauser et al. (1969).

We now introduce an extremely simple family of measurement scenarios that will play a significant rôle in our treatment to follow.

Example 5 For each positive integer n, we will denote by $[n]$ the scenario that has a single measurement with n possible outcomes, i.e.

$$X_{[n]} := \{*\} \qquad \text{and} \qquad O_{[n],*} : +\{0, \ldots, n-1\}\,.$$

Note that $\Sigma_{[n]}$ is uniquely determined.

13.2.2 Empirical Models

Having formalised how to describe the interface of the black boxes of interest, we now turn our attention to their possible behaviours. These are described by empirical models on the given measurement scenario.

We start by considering the atomic empirical events that represent a single interaction with a black box.

Definition 6 Let S be a scenario. For any $U \subseteq X_S$, we write

$$\mathcal{E}_S(U) := \prod_{x \in U} O_x$$

for the set of assignments of outcomes to each measurement in the set U. When U is a valid context, these are the joint outcomes one might obtain for the measurements in U.

The mapping above extends to a sheaf $\mathcal{E}_S \colon \mathcal{P}(X_S)^{\mathrm{op}} \longrightarrow \mathsf{Set}$, called the **event sheaf**, with restriction maps

$$\mathcal{E}_S(U \subseteq V) \colon \mathcal{E}_S(V) \longrightarrow \mathcal{E}_S(U)$$

given by the obvious projections. When it does not give rise to ambiguity, we often omit the subscript and denote the event sheaf more simply by $\mathcal{E}$.

An account of the black box's behaviour ought to specify its response to any allowed interaction (measurement context), as a probability distribution over corresponding atomic events.

Let D denote the (discrete) probability distribution functor. That is, for any set X, $\mathsf{D}(X)$ is the set of (finitely-supported) probability distributions on X, and for any function $f\colon X \longrightarrow Y$, $\mathsf{D}(f)$ maps a distributions d on X to its push-forward along f, a distribution on Y. Similarly, we will denote by $\mathsf{D}_{\mathbb{B}}$ the Boolean distribution functor; i.e. $\mathsf{D}_{\mathbb{B}}(X)$ is the set of Boolean-valued distributions on X. Note that this is the covariant non-empty powerset functor.[13]

Definition 7 A (probabilistic) **empirical model** e on a scenario S, written $e\colon S$, is a compatible family for Σ on the presheaf $\mathsf{D} \circ \mathcal{E}_S$. More explicitly, it is a family $(e_\sigma)_{\sigma \in \Sigma_S}$ where, for each $\sigma \in \Sigma_S$,

$$e_\sigma \;\in\; \mathsf{D} \circ \mathcal{E}(\sigma) \;=\; \mathsf{D}\left(\prod_{x \in \sigma} O_x\right)$$

is a probability distribution over the joint outcomes for the measurements in the context σ. These distributions are required to be compatible in the sense that for any $\sigma, \tau \in \Sigma_S$ with $\tau \subseteq \sigma$, one must have $e_\tau = e_\sigma|_\tau$ where

$$e_\sigma|_\tau \;:=\; \mathsf{D} \circ \mathcal{E}(\tau \subseteq \sigma)(e_\sigma)\;,$$

is the marginalisation of e_σ to the smaller context τ; i.e. for any $t \in \mathcal{E}(\tau)$,

$$e_\tau(t) \;:=\; \sum_{s \in \mathcal{E}(\sigma), s|_\tau = t} e_\sigma(s)\;.$$

The set of all probabilistic empirical models on S is denoted $\mathsf{EMP}(S)$.

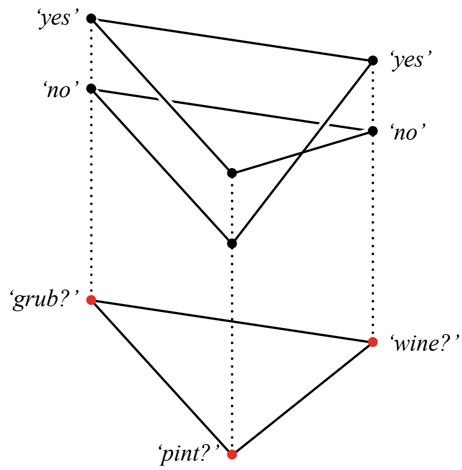
Note that compatibility can equivalently be expressed as the requirement that for all facets (i.e. maximal contexts) C and C' of Σ,

$$e_C|_{C \cap C'} = e_{C'}|_{C \cap C'}\;.$$

Compatibility holds for all quantum-realisable behaviours Abramsky et al. (2011). It generalises a property known as **no-signalling** Ghirardi et al. (1980), as will be illustrated shortly in Example 10.

[13] Note that for Boolean distributions the restriction to finite support is unnecessary. But here we only deal with finite sets of events, anyway.

Fig. 13.4 Bundle diagram for the empirical model of Example 8



Example 8 The following table provides an example of an empirical model in the $\triangle$ scenario. The rows of the table specify probability distributions for the maximal contexts, and all other distributions can simply be obtained by marginalisation.

		'yes' 'yes'	'yes' 'no'	'no' 'yes'	'no' 'no'
'pint?'	*'wine?'*	0	$\frac{1}{2}$	$\frac{1}{2}$	0
'wine?'	*'grub?'*	$\frac{1}{2}$	0	0	$\frac{1}{2}$
'pint?'	*'grub?'*	$\frac{1}{2}$	0	0	$\frac{1}{2}$

This is a black box behaviour that solves the following conundrum: one might wish to have a glass of wine if and only one is also eating; ditto for a pint of beer. However, one might want to have either beer or wine but not both. An empirical model realising this scenario permits all of these constraints to be satisfied as long as no more than two questions are asked at once.

Another useful way to represent the above empirical model is as a bundle diagram (Fig. 13.4). This method of representing models was introduced in Abramsky et al. (2015). In the 'downstairs' part of this diagram lives the simplicial complex $\Sigma_\triangle$ that captures measurement compatibility. Above each measurement is a fibre consisting of its available outcome values. Finally, the faces (just edges in this case, as the complex is 1-dimensional) in the 'upstairs' of the diagram represent possible joint outcomes. Note that while the empirical model associates probabilities with the joint outcomes, the bundle representation in general carries less information, as it simply includes faces for for those joint outcomes that have non-zero probability and omits the others. As we will elaborate on shortly, in many cases such (possibilistic) information about the support of the probability distributions already suffices to pick out interesting features of an empirical model. As it happens, in this specific example, the supports of the distributions uniquely determine the probabilities, by compatibility.

Example 9 Similarly, the following table provides an example of an empirical model in the $\square$ scenario. Here we notice that all context pairs give correlated outcomes except the context consisting of *'SammyA'* and *'EvilG'*, whose outcomes are anti-correlated.

		'grape' 'grape'	*'grape' 'grain'*	*'grain' 'grape'*	*'grain' 'grain'*
'SammyA'	*'EvilG'*	0	$\frac{1}{2}$	$\frac{1}{2}$	0
'SammyA'	*'GeorgieB'*	$\frac{1}{2}$	0	0	$\frac{1}{2}$
'JohnnyB'	*'EvilG'*	$\frac{1}{2}$	0	0	$\frac{1}{2}$
'JohnnyB'	*'GeorgieB'*	$\frac{1}{2}$	0	0	$\frac{1}{2}$

Example 10 The following table gives an empirical model on the $\square$ scenario, whose measurements and outcomes have been relabelled. This is the **CHSH model**.

		00	01	10	11
X_A	X_B	$\frac{1}{2}$	0	0	$\frac{1}{2}$
X_A	Y_B	$\frac{3}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{3}{8}$
Y_A	X_B	$\frac{3}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{3}{8}$
Y_A	Y_B	$\frac{3}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{3}{8}$

Here we can understand the black box as being shared by two parties A and B; each party can independently choose to make an X or a Y measurement, and for all measurements the outcomes take values 0 or 1. Notice, for example, that the probability that X_A returns 0 is independent of which measurement party B chooses to perform ($\frac{1}{2} + 0 = \frac{1}{2}$ when X_B is measured and $\frac{3}{8} + \frac{1}{8} = \frac{1}{2}$ when Y_B is measured). This is the no-signalling property at work. It implies that the black box cannot be used by the parties to transmit information instantaneously to one another by means of their choice of measurement.

If we relabel and revisit Example 9, viewing it as a bipartite scenario as we have interpreted Example 10, it corresponds to an empirical model known as the PR box Popescu and Rohrlich (1994).

13.2.3 Possibilistic Collapse

As already mentioned, sometimes it is enough to consider the possibilistic information present in an empirical model.

Definition 11 A **possibilistic** empirical model e on a scenario S, also written $e\colon S$, is a compatible family for Σ on the presheaf $\mathsf{D}_{\mathbb{B}} \circ \mathcal{E}_S$. The set of possibilistic empirical models on S is denoted $\mathsf{EMP}_{\mathbb{B}}(S)$.

Since Boolean distributions on a finite set can be identified with its non-empty subsets, one may describe a possibilistic empirical model more explicitly as a family

$(e_\sigma)_{\sigma\in\Sigma_S}$ where each e_σ is a non-empty subset of $\mathcal{E}(\sigma)$ and for any $\sigma, \tau \in \Sigma_S$ with $\tau \subseteq \sigma$,

$$e_\tau = e_\sigma|_\tau := \{s|_\tau \mid s \in e_\sigma\} .$$

The set $\mathsf{EMP}_{\mathbb{B}}(S)$ of possibilistic empirical models on S is equipped with a partial order, whereby $d \leq e$ if $d_\sigma \subseteq e_\sigma$ for all $\sigma \in \Sigma_S$. The structure of this partial order was more thoroughly investigated in Abramsky et al. (2016a). We will make use of it when defining weak simulations in Definition 26 below.

There is a natural map $\mathsf{EMP}(S) \to \mathsf{EMP}_{\mathbb{B}}(S)$ which sends a (probabilistic) model e to its **possibilistic collapse**. This is the possibilistic model e' defined by $e'_\sigma := \mathsf{supp}(e_\sigma)$. We will often abuse notation by denoting both a model and its possibilistic collapse by the same letter. As an aside, note that not all models in $\mathsf{EMP}_{\mathbb{B}}(S)$ arise via possibilistic collapse from a model in $\mathsf{EMP}(S)$ Abramsky (2013), Abramsky et al. (2016a), a fact that has echoes of the main problematic set out in this chapter.

Example 12 The following table gives the possibilistic collapse of the CHSH model.

	0 0	0 1	1 0	1 1
X_1 X_2	1	0	0	1
X_1 Y_2	1	1	1	1
Y_1 X_2	1	1	1	1
Y_1 Y_2	1	1	1	1

13.2.4 Contextuality

We now come to the definition of contextuality.

Definition 13 An empirical model $e\colon S$ is said to be non-contextual if it is extendable to a global section for $\mathsf{D} \circ \mathcal{E}$. In other words, e is non-contextual if there exists a distribution $d \in \mathsf{D} \circ \mathcal{E}(X_S)$ on global assignments of outcomes to all measurements in X_S such that $d|_\sigma = e_\sigma$ for every context $\sigma \in \Sigma_S$. Otherwise, the empirical model is said to be **contextual**.

An equivalent formulation of non-contextuality of an empirical model e, which will be used throughout this chapter, is to say that e can be written as a convex combination (taken contextwise) of deterministic empirical models. By a deterministic empirical model, we mean an empirical model $d\colon S$ in which d_σ is a delta distribution for every context $\sigma \in \Sigma_S$, or equivalently, a compatible family for Σ on the presheaf $\mathcal{E}$. Since $\mathcal{E}$ is in fact a sheaf, deterministic empirical models are in one-to-one correspondence with global assignments $s \in \mathcal{E}(X_S)$. We write δ_s for the deterministic model corresponding to the global assignment s.

As already mentioned, sometimes it is possible to witness contextuality at the possibilistic level, i.e. by looking only at the supports of the empirical probability distributions.

Definition 14 An empirical model $e\colon S$ is said to be logically non-contextual if its possibilistic collapse is non-contextual over $\mathbb{B}$, i.e. if there is a Boolean distribution $d \in \mathsf{D}_{\mathbb{B}} \circ \mathcal{E}(X)$ over joint outcomes that marginalises to (the possibilistic collapse of) e. Otherwise, the empirical model is said to be **logically contextual**.

Note that in the case of logical contextuality there is a canonical candidate for the global distribution, namely that corresponding to the set of global value assignments that are compatible with the supports of e,

$$\mathcal{S}_e := \{s \in \mathcal{E}(X_S) \mid \forall \sigma \in \Sigma_S.\ s|_C \in \operatorname{supp} e_C\} \ .$$

That is, a model e is logically non-contextual if and only if $\mathcal{S}_e|_\sigma$ equals to the support of e_σ for all contexts $\sigma \in \Sigma_S$. Equivalently, e is logically contextual if and only if there is an assignment $s \in \mathcal{E}_S(\sigma)$ in the support of e_σ which cannot be extended to a global assignment element in $\mathcal{S}_e$.

This leads us to an even stronger notion of contextuality.

Definition 15 An empirical model $e\colon S$ is said to be **strongly contextual** if $\mathcal{S}_e = \emptyset$.

In other words, e is strongly contextual if there is not even a single global assignment $s \in \mathcal{E}(X_S)$ that is compatible with the support of e in every context.

These *strengths* of contextuality form a strict hierarchy Abramsky et al. (2011). Strong contextuality implies logical contextuality, which in turn implies probabilistic contextuality (for probabilistic empirical models). Moreover, it is possible to find empirical models that separate each of these classes.

The model from Example 8 is strongly contextual. In fact this can be deduced simply by inspecting its bundle diagram in Fig. 13.4 and noticing that any attempt at finding a univocal global value assignment consistent with the observed outcomes must fail—if one traces a path from jointly possible outcome to jointly possible outcome it is impossible to close while only ascribing a single outcome value to each measurement. Similarly the model from Example 9 is strongly contextual, a fact that can also be deduced from a bundle diagram as was previously illustrated in Abramsky et al. (2015). The CHSH model of Example 10 is probabilistically contextual but it is not logically contextual. We refer the interested reader to Abramsky et al. (2011) for more details on the hierarchy of contextuality.

13.3 The Framework: Morphisms

We move to considering transformations between measurement scenarios. The various notions of morphism $S \to T$ introduced in this section describe procedures that can be followed by a classical agent with access to a measurement scenario S in order to implement the measurements of a new scenario T. Such procedures naturally induce simulations between empirical models on the scenarios at hand. They can be regarded as the 'free' operations in a (non-adaptive) resource theory of contextuality.

We discuss how some well-studied notions in the literature, notably non-local games, admit a neat description in terms of procedures and simulations.

We will generally refer to morphisms $S \to T$ as experimental procedures or just **procedures**, with additional adjectives (deterministic, probabilistic, possibilistic) added as needed. However, some particular instances of these concepts warrant specific terminology. An **experiment** on a scenario S refers to a procedure of type $S \to [n]$, where $[n]$ is the scenario with a single measurement and n possible outcomes from Example 5. When $n = 2$, we also speak of a **predicate**.[14]

13.3.1 Deterministic Procedures

We first consider deterministic procedures. A procedure $S \to T$ will consist of two parts: a (pre-processing) map of inputs in the backward direction and a (post-processing) map of outputs in the forward direction. More concretely, a procedure for implementing a measurement x in X_T must specify a context of measurements in S to be performed and a way to map the outcomes obtained into values in the outcome set of x. Importantly, it must be ensured that in the implementation of a context (i.e. compatible set of measurements) of T, only a compatible set of measurements in S is performed. This compatibility condition is captured by the concept of simplicial relation.

We start by fixing some notation. Given a relation $R\colon X \longrightarrow Y$, the image of an element $x \in X$ under R is the set

$$R(x) := \{y \in Y \mid xRy\} \ .$$

Similarly, if U is a subset of X then the image of U under R is the set

$$R(U) := \bigcup_{x \in U} R(x) = \{y \in Y \mid \exists x \in \sigma.\ xRy\} \ .$$

Definition 16 Let Σ and Δ be simplicial complexes. A **simplicial relation** $R\colon \longrightarrow \Sigma\Delta$ is a relation between the vertices of Σ and those of Δ that maps faces to faces, i.e.such that for all $\sigma \in \Sigma$, $R(\sigma) \in \Delta$.

Definition 17 A **deterministic procedure** $f\colon S \longrightarrow T$ between measurement scenarios S and T is a pair $f = \langle \pi_f, \alpha_f \rangle$ consisting of:

[14] While the word “predicate” does not quite fit with the experimental imagery evoked by much of our terminology, there is a reason for introducing an alternative word for two-valued experiments. Later, it will be useful to consider whether a given model always returns the outcome 1 in such an experiment, and to restrict attention to those models on a scenario which do so. Hence calling it a predicate serves the purpose of indicating a change of viewpoint, where we will be restricting attention to models that always satisfy a given property.

- a simplicial relation $\pi_f : \Sigma_T \longrightarrow \Sigma_S$, which specifies for each measurement x of T a context $\pi_f(x)$ of S;
- a family $\alpha_f = (\alpha_{f,x})_{x \in X_T}$ of functions $\alpha_{f,x} : \mathcal{E}_S(\pi_f(x)) \longrightarrow \mathcal{E}_T(x)$, which map joint outcomes of $\pi_f(x)$ to outcomes of x.

The category of measurement scenarios and deterministic procedures is denoted by $\mathsf{Scen}_{\mathrm{Det}}$.

Remark 18 Specifying a family of functions $(\alpha_x : \mathcal{E}_S(\pi(x)) \longrightarrow \mathcal{E}_T(x))_{x \in X_T}$ is equivalent to specifying a natural transformation $\alpha : \mathcal{E}_S(\pi -) \longrightarrow \mathcal{E}_T(-)$ of presheaves on $\mathcal{P}(X_T)$. This is because both of these presheaves are in fact sheaves (on X_T with the discrete topology) and morphisms of sheaves can be glued together along any cover.

As remarked above, a **deterministic experiment** on a scenario S is a deterministic procedure $S \to [n]$, and a **deterministic predicate** is a deterministic procedure $S \to [2]$.

Remark 19 Note that a deterministic predicate on S is given by a choice of $\sigma \in \Sigma_S$ and a subset of $\mathcal{E}_S(\sigma)$.

Example 20 We consider an example of a deterministic experimental procedure $f : \triangle \longrightarrow \square$ where $\triangle$ and $\square$ are the scenarios from Examples 3 and 4 respectively. We define π and α pointwise as follows. We send *'SammyA'* to query for *'pint?'* and to heed to the answer, i.e. to choose *'grain'* if the answer is *'yes'* and *'grape'* otherwise. We send *'EvilG'* to query for *'wine?'* and instruct him to disobey the answer, i.e. to go for *'grain'* if the answer is *'yes'* and *'grape'* otherwise. Finally, we send both *'GeorgieB'* and *'JohnnyB'* to the query *'grub?'*, and instruct them to choose *'grain'*, if the answer is *'yes'* and to choose *'grape'* otherwise. In effect, each party sends *'yes'* to *'grain'* and *'no'* to *'grape'*; they just perform different queries on the triangle.

13.3.2 Probabilistic and Possibilistic Procedures

We now introduce a more general notion of morphism, which allows for some classical randomness.

Definition 21 A **probabilistic procedure** $S \to T$ is a convex mixture of deterministic procedures $S \to T$, i.e. an element of $\mathsf{D}(\mathsf{Scen}_{\mathrm{Det}}(S, T))$. The category of measurement scenarios and probabilistic procedures is denoted by Scen.

We often denote such a probabilistic procedure as $\sum_{i \in I} r_i f_i$ where I is a (necessarily non-empty) finite set, r_i are positive reals summing to 1, and each $f_i : S \longrightarrow T$ is a deterministic procedure.

Similarly, when we care only about the possible events and not their specific probabilities, we are led to a possibilistic version of procedures.

Definition 22 A **possibilistic procedure** $S \to T$ is a Boolean mixture of deterministic procedures $S \to T$, i.e. an element of $\mathsf{D}_{\mathbb{B}}(\mathsf{Scen}_{\mathrm{Det}}(S, T))$. The category of measurement scenarios and possibilistic procedures is denoted by $\mathsf{Scen}_{\mathbb{B}}$.

We often write such a possibilistic procedure as $\bigvee_{i \in I} f_i$ where I is non-empty finite set and each f_i is a deterministic procedure $S \to T$.

Analogously to the deterministic case, a **probabilistic experiment** (resp. **possibilistic experiment**) on a scenario S is a a probabilistic (resp. possibilistic) procedure $S \to [n]$, and this is also called a **probabilistic predicate** (resp. **possibilistic predicate**) on S when $n = 2$.

13.3.3 Simulations

The morphisms of measurement scenarios introduced above yield maps between their sets of empirical models. The idea is that by following a procedure $f : S \longrightarrow T$, an empirical model $e : S$ is used to simulate a new empirical model on T, denoted $\mathsf{EMP}(f)\, e$.

Mathematically, a helpful analogy is to think of empirical models as probability distributions on a 'contextual space', and to regard the simulation maps induced by (deterministic) procedures as analogous to the push-forward of a probability measure along a (measurable) function. Indeed, this is precisely how the map is defined at each context.

Definition 23 Given a deterministic procedure $f : S \longrightarrow T$ we define a function $\mathsf{EMP}(f) : \mathsf{EMP}(S) \longrightarrow \mathsf{EMP}(T)$ by setting

$$(\mathsf{EMP}(f)\, e)_\sigma := \alpha^*_{f,\sigma}(e_{\pi_f \sigma}) \ ,$$

i.e. the probability distribution that $\mathsf{EMP}(f)\, e$ gives at context $\sigma \in \Sigma_T$ is obtained by pushing forward $e_{\pi_f \sigma}$ along the map $\alpha_{f,\sigma} : \mathcal{E}_S(\pi_f \sigma) \longrightarrow \mathcal{E}_T(\sigma)$. This is extended to probabilistic procedures, which are convex mixtures $\sum_i r_i f_i$ of deterministic procedures, by defining

$$\mathsf{EMP}\left(\sum_i r_i f_i\right) e := \sum r_i (\mathsf{EMP}(f_i) e) \ .$$

In this way, EMP defines a functor with domain Scen. As for its codomain, note that $\mathsf{EMP}(S)$ has more structure than just that of a bare set. Namely, one can naturally form convex combinations $\sum_{i=1}^n r_i e_i$ of empirical models, and so $\mathsf{EMP}(S)$ has the structure of a convex set; i.e. it is an algebra of the distribution monad on Set. The following lemma shows that $\mathsf{EMP}(f)$ preserves this structure.

Lemma 24 *If $f : S \longrightarrow T$ is a probabilistic procedure, then $\mathsf{EMP}(f)$ preserves convex combinations.*

Proof We observe first that the claim holds whenever f is a deterministic procedure. Indeed, as $\mathsf{EMP}(f)\, e$ is defined for $\sigma \in \Sigma_T$ by

$$(\mathsf{EMP}(f)\, e)_\sigma = \alpha^*_\sigma(e_{\pi\sigma})$$

this follows from the fact that α^*_σ preserves convex combinations for each σ. Moreover, when $f = \sum r_i f_i$ is a convex combination of deterministic procedures f_i the function $\mathsf{EMP}(f)$ is defined via $\sum r_i \mathsf{EMP}(f_i)$, and so it follows that $\mathsf{EMP}(f)$ preserves convex combinations since each $\mathsf{EMP}(f_i)$ does.

Therefore, we can think of EMP as a functor whose codomain is the category of convex sets (with convex-combination-preserving functions). However, at times we find it convenient to abuse notation and compose this with the forgetful functor to Set without comment.

Formally speaking, the action of EMP on probabilistic procedures follows inevitably from its action on deterministic procedures. This is essentially because EMP is a convex set. The passage from $\mathsf{Scen}_{\mathrm{Det}}$ to Scen can be seen as freely enriching $\mathsf{Scen}_{\mathrm{Det}}$ in convex sets. As EMP already gives a functor from $\mathsf{Scen}_{\mathrm{Det}}$ to convex sets, it induces an enriched (i.e. convex) functor from Scen to convex sets.

There is an similar construction for possibilistic procedures, to which the analogous remarks apply (replacing convex sets by sup-lattices).

Definition 25 A deterministic procedure f induces a function $\mathsf{EMP}_{\mathbb{B}}(f)\colon \longrightarrow \mathsf{EMP}_{\mathbb{B}}(S)\mathsf{EMP}_{\mathbb{B}}(T)$ defined in the same way as for probabilistic empirical models (Definition 23). This is extended to possibilistic procedures by setting

$$\mathsf{EMP}_{\mathbb{B}}\left(\bigvee_i f_i\right) e := \bigvee_i (\mathsf{EMP}_{\mathbb{B}}(f_i)e)\ .$$

We now have all the necessary ingredients in place to introduce the relevant notions of simulation between empirical models.

Definition 26 The category Emp of probabilistic empirical models and (probabilistic) simulations is defined as the category of elements of EMP. Explicitly, the objects of Emp are probabilistic empirical models $e\colon S$, and morphisms $e\colon S \to d\colon T$ are **probabilistic simulations**, i.e. probabilistic procedures $f\colon S \longrightarrow T$ such that $\mathsf{EMP}(f)\, e = d$.

The category $\mathsf{Emp}_{\mathbb{B}}$ of possibilistic empirical models and possibilistic simulations is defined as the category of elements of $\mathsf{EMP}_{\mathbb{B}}$. Explicitly, the objects of $\mathsf{Emp}_{\mathbb{B}}$ are possibilistic empirical models $e\colon S$, and morphisms $e\colon S \to d\colon T$ are **possibilistic simulations**, i.e. possibilistic procedures $f\colon S \longrightarrow T$ such that $\mathsf{EMP}_{\mathbb{B}}(f)\, e = d$.

The category $\mathsf{Emp}^{\leq}_{\mathbb{B}}$ of possibilistic empirical models and weak simulations is defined as the lax category of elements of $\mathsf{EMP}_{\mathbb{B}}$. Explicitly, the objects of $\mathsf{Emp}^{\leq}_{\mathbb{B}}$ are possibilistic empirical models $e\colon S$, and morphisms $e\colon S \to d\colon T$ are given by **weak simulations**, i.e. possibilistic procedures $f\colon S \longrightarrow T$ such that $\mathsf{EMP}(f)_{\mathbb{B}}\, e \leq d$.

While the names of the morphisms above are chosen to elicit a helpful intuition, let us explain this in some more detail. We recommend that one thinks of a morphism $e \to d$ as a way of simulating d from e. The underlying morphism of scenarios gives an operational description of the simulation procedure, whereas the choice of category amounts to choosing a notion of *correctness* for such a procedure.

For $f: S \longrightarrow T$ to define a probabilistic simulation $e \to f$ in Emp, the statistics of e need to be taken to statistics of d exactly. The adjective probabilistic refers to the fact that the procedure can be probabilistic. In other words, when simulating d from e one obtains exactly d. In contrast to this, for f to be a possibilistic simulation $d \to e$ in $\mathsf{Emp}_{\mathbb{B}}$ it is sufficient that $\mathsf{EMP}(f)\, e$ and d have the same support. Finally, for a morphism $S \to T$ to define a weak simulation $e \to d$, it is enough that one never observes an outcome that d deems impossible when running the simulation using e.

While weaker and weaker forms of simulation may seem too weak to be of practical interest, note that the corresponding notions of **non-simulability** between empirical models become stronger as one relaxes the type of simulation considered. With this in mind, we say that an empirical model e is probabilistically/possibilistically/strongly non-simulable from d if there are no morphisms $d \to e$ in $\mathsf{Emp}/\mathsf{Emp}_{\mathbb{B}}/\mathsf{Emp}_{\mathbb{B}}^{\leq}$. Theorem 29 below says that probabilistic, possibilistic, and strong contextuality of a model e correspond precisely to probabilistic, possibilistic, and strong non-simulability from a trivial model.

Example 27 A straightforward calculation shows that the procedure $f: \triangle \longrightarrow \square$ from Example 20 takes the model $e: \triangle$ from Example 8 to the model $d: \square$ from Example 9, so that f defines a simulation $e \to d$ in Emp.

One can in fact show that there is no simulation in the opposite direction. Indeed, consider an arbitrary simplicial relation $\pi: \Sigma_{\triangle} \longrightarrow \Sigma_{\square}$. As any pair of measurements is compatible in the scenario $\triangle$, this must also be true in the image of π. This implies that this image must in fact be a face of $\Sigma_{\square}$. Thus, a procedure $f: \square \longrightarrow \triangle$ can only make use of a context of $\square$. Note that the restriction to a context of any empirical model $d: \square$ is necessarily non-contextual. Consequently, so is $\mathsf{EMP}(\pi, \alpha)d$. However, the model $e: \triangle$ from Example 8 is strongly contextual, and so it cannot be equal to $\mathsf{EMP}(f)\, d$ for any $d: \square$. In fact, this model cannot even be weakly simulated from a model on $\square$, or for that matter from any quantum-realisable model. This statement remains true even if one allows for probabilistic and adaptive procedures, but establishing this in detail is beyond the scope of this chapter. However, the crux of the argument is the same: in quantum-realisable scenarios a set of measurements is compatible if and only if it is pairwise compatible, a fact known as Specker's principle. Therefore, any attempt at simulating an empirical model on $\triangle$ from a model on such a quantum-realisable scenario ends up using only a fully compatible—and thus non-contextual—part.

Further examples of simulations can be found in the literature. For example, Protocols 1, 2, 5, and 6 in Barrett et al. (2005) define simulations between various empirical models. The other protocols in that article go slightly beyond the present framework by being adaptive, approximately correct, or allowing for some limited classical communication. Similarly, the proof Corollary 2 of Barrett and Pironio (2005), which states that any two-output bipartite box can be simulated by sufficiently many PR boxes, does not require adaptivity, hence it also holds true in our current framework.

Remark 28 The definitions of morphisms presented here differ slightly from those in earlier expositions in Karvonen (2019) or Abramsky et al. (2019a). Our deterministic morphisms are the same as in Karvonen (2019), whereas in Abramsky et al. (2019a) we required the basic morphisms to have an underlying simplicial map rather than a simplicial relation. The bigger differences occur afterwards: in Karvonen (2019) the expressive power of deterministic morphisms was increased by letting the components of α be stochastic maps. Our present notion of probabilistic procedure is more general in that it allows both π and α to behave stochastically. The earlier limitation was just due to not seeing the current definition as a possibility. In contrast to this, Abramsky et al. (2019a) extends the deterministic morphisms differently, by passing to a coKleisli category of a certain comonad on Scen in order to allow **adaptive** protocols instead of mere joint measurements as in our current setup. Note that the presence of adaptivity elides the difference in expressive power between simplicial maps and simplicial relations: measuring a context with several measurements can be achieved through a measurement protocol that measures one measurement at a time (with a trivial form of adaptivity). We would happily work in the same adaptive setting if only we knew how to generalise our current results, specifically Theorem 44, to such adaptive morphisms. As we currently don't know how to do this, we focus on the non-adaptive case and discuss the issues raised by adaptivity in Sect. 13.6.2. A further difference is how randomness is dealt with: here we obtain shared, classical randomness by allowing convex mixtures of morphisms, whereas in Abramsky et al. (2019a) we defined a morphism $d \to e$ to be a deterministic co-Kleisli morphism $d \otimes c \to e$ for some non-contextual c. In the presence of adaptivity, there is no difference in expressive power, but only a difference in viewpoint. The setup in Abramsky et al. (2019a) suggests defining more general morphisms $d \to e$ as maps $d \otimes c \to e$ where the resource c is in some fixed class of interest (e.g.the quantum-realisable empirical models). A benefit of our current framework is that it makes it easier to discuss probabilistic morphisms at the level of scenarios already, rather than only at the level of empirical models.

13.3.4 Contextuality as Non-simulability

The following result shows how the familiar notion(s) of contextuality can be neatly expressed in terms of simulations. As discussed below, this justifies the relative point of view, allowing in particular to regard our question (A) as a generalisation of the question of determining whether a model is contextual (question (B)).

Theorem 29 *Let Z denote the unique scenario with an empty set of measurements, and let z denote the unique empirical model on it. A probabilistic model $e : S$ is contextual if and only if it is probabilistically non-simulable from z, i.e. if and only if there is no morphism $z \to e$ in* Emp.

Moreover, a possibilistic model e is logically (resp. strongly) contextual if and only if it is possibilistically (resp. strongly) non-simulable from z, i.e. if and only if there is no morphism $z \to e$ in $\mathsf{Emp}_{\mathbb{B}}$ (resp. $\mathsf{Emp}^{\leq}_{\mathbb{B}}$).

Proof These characterisations of probabilistic and logical contextuality were already proved in Karvonen (2019, Theorem 4.1) but we include a proof here for the sake of completeness.

The crucial observation is that deterministic procedures $Z \to S$ are in one-to-one correspondence with global assignments for S, i.e. with elements of $\mathcal{E}_S(X_S)$, and thus with deterministic models on S. If $f : Z \longrightarrow S$ is a deterministic procedure, then π_f is necessarily the empty relation, so the only choice is in choosing $\alpha_{f,x} : \mathcal{E}_Z(\emptyset) \longrightarrow \mathcal{E}_S(x)$ for each $x \in X_S$. Noting that $\mathcal{E}_Z(\emptyset)$ is a singleton and $\mathcal{E}_S(x) = O_{S,x}$, we see that $\alpha_{f,x}$ is determined by an element of the outcome set of x, and so α_f precisely specifies a global assignment $s \in \mathcal{E}_S(X_S)$. Writing f_s for the deterministic procedure corresponding to s in this way, note that $\mathsf{EMP}(f_s)\, z = \delta_s$, where δ_s is the deterministic model determined by s. So, in particular, every deterministic model is simulable from z.

We start with the probabilistic case and prove the implications in both directions by proving their contrapositives. If e is non-contextual, then it can be expressed as a convex combination of deterministic models $e = \sum_{s\in\mathcal{E}(X_S)} r_s\delta_s$, and the procedure $\sum r_s f_s$ simulates e. Conversely, suppose e is simulable from z, say by a probabilisitc procedure $f : Z \longrightarrow S$. Then f is a convex combination of deterministic procedures, which are of the form f_s for a $s \in \mathcal{E}(X_S)$, and so we can write $f = \sum_{s\in\mathcal{E}(X_S)} r_s f_s$. Consequently,

$$e \;=\; \mathsf{EMP}(f)\, z \;=\; \sum_{s\in\mathcal{E}(X_S)} r_s \mathsf{EMP}(f_s)\, z \;=\; \sum_{s\in\mathcal{E}(X_S)} r_s\delta_s \;,$$

thus e is non-contextual. The case of logical contextuality is obtained by replacing convex sums with possibilistic combinations in the above.

We now move on to strong contextuality, again proving the two directions by showing their contrapositives. If e is weakly simulable from z, the underlying procedure is necessarily of the form $\bigvee_{s\in A} f_s$ for some subset of global assignments $A \subseteq \mathcal{E}(X_S)$. Then any of the deterministic models $\mathsf{EMP}_{\mathbb{B}}(f_s)\, z = \delta_s$ for $s \in A$ is a witness of e not being strongly contextual. Conversely, if e is not strongly contextual, there is a global assignment $s \in \mathcal{E}(X_S)$ consistent with it. The corresponding deterministic procedure $f_s : Z \longrightarrow S$ yields a possibilistic simulation $z \to \delta_s$ in $\mathsf{Emp}_{\mathbb{B}}$, which is a weak simulation $f : z \longrightarrow e$ in $\mathsf{Emp}^{\leq}_{\mathbb{B}}$.

Thus $e : S$ is contextual if and only if the map $\mathsf{EMP}(Z) = \{z\} \to \mathsf{EMP}(S)$ corresponding to it arises from a procedure $Z \to S$. This suggests the following, more general question: which functions $\mathsf{EMP}(S) \to \mathsf{EMP}(T)$ are induced by a procedure $S \to T$? That is, which such functions are equal to $\mathsf{EMP}(f)$ for some $f: S \longrightarrow T$?

Such generalisation can be seen as an instance of Grothendieck's relative point of view Bartels (2010), which, roughly speaking, suggests that properties of mathematical objects should be seen as arising from properties of morphisms, so that a geometric object X is defined to have a property precisely when the canonical morphism $X \to 1$ has the corresponding property of morphisms. As, heuristically speaking, one expects geometry to be dual to algebra, we find ourselves in a dual situation: a model can be defined to be contextual if the map $\mathsf{EMP}(Z) = \{z\} \to \mathsf{EMP}(S)$ corresponding to it is contextual. To make this precise, we need to answer the question of when a general map $\mathsf{EMP}(S) \to \mathsf{EMP}(T)$ is contextual.

13.3.5 Non-local Games as Experiments

Another familiar concept from the literature that admits a neat formulation in our framework is that of non-local games (or more generally, contextual games as considered in Abramsky et al. (2017c, Appendix E)). Note that these can also be thought of as linear inequalities on the probabilities predicted by empirical models, and thus encompass in particular Bell locality inequalities or non-contextuality inequalities.

One usually thinks of non-local games as follows: there are n spatially separated parties who can agree on a strategy beforehand but cannot communicate once the game is in play. A referee sends to the i-th party a question from an input set X_i, and expects back an answer from some output set O_i. The referee draws the questions according to a joint probability distribution on inputs. Afterwards, the referee collects the answers and applies a rule

$$W: X_1 \times \cdots \times X_n \times O_1 \times \cdots \times O_n \longrightarrow \{0, 1\}$$

to determine whether the players win or lose. Both the distribution on inputs and this winning condition are known a priori by all players. The goal of the players is to maximise their winning probability.

Such games turn out to be readily formalised as probabilistic experiments: one first builds a (Bell-type) measurement scenario S whose the measurements are given by $\coprod_i X_i$ and where the maximal measurement contexts are given by a choice of a measurement $x_i \in X_i$ for each party. The outcome set of each measurement in X_i is O_i. If the referee asks a joint question $q \in X_1 \times \cdots \times X_n$ with probability r_q, then we can model the game as a probabilistic experiment on S, which with probability r_q chooses the measurement context corresponding to q, and then uses W to obtain an outcome in $\{0, 1\}$, which is the outcome set of the only measurement of the scenario [2]. Thus, the game setup (namely, the distribution on inputs and the winning condition) can be packaged into a probabilistic experiment $g: S \longrightarrow [2]$. From this point of view, a strategy for the players to answer the questions is given by an empirical model e on the scenario S. Finally, the **winning probability** achieved by that strategy arises as the model $\mathsf{EMP}(g)\,e\colon [2]$, as an empirical model over [2] is uniquely determined by the probability of obtaining the outcome 1.

Note that we can also encode a more general class of games, where the referee attributes a pay-off to each combination of questions and answers, rather than a discrete win-or-lose valuation. Such is in particular the general form of a Bell inequality. If we normalise such pay-offs, this amounts to generalising W to take values in $[0, 1]$ instead of $\{0, 1\}$. These games can still be seen as probabilistic experiments $g\colon S \longrightarrow [2]$, as the extra flexibility can be modelled as a further mixture of deterministic experiments.

Example 30 Recall the Four Candles scenario $\square$ from Example 4. We will represent the CHSH game as a probabilistic experiment $\sum_{1=1}^{n} \frac{1}{4} f_i \colon \square \longrightarrow [2]$ with each f_i a deterministic experiment. Recall that a deterministic experiment on $\square$ is determined by specifying a measurement context $\sigma \in \Sigma_\square$ and a subset of $\mathcal{E}_\square(\sigma)$ that gets sent to 1, which we should think of as the winning condition. With this in mind, we let f_1 be determined by the context { *'SammyA'*, *'EvilG'* } and the winning subset be given by { (*'grape'*,*'grain'*),(*'grain'*,*'grape'*) }. In other words, for the players to win, *'SammyA'* and *'EvilG'* must anti-coordinate in their choice of beverage. The other experiments f_2, f_3 and f_4 correspond to the remaining maximal contexts, with the winning constraint given now by coordination, i.e. by the set { (*'grape'*,*'grape'*),(*'grain'*,*'grain'*) }.

If all our agents have stubbornly, i.e. deterministically, chosen what their chosen beverage is, they can satisfy the referee in at most three out of the four possible questions. As $\mathsf{EMP}(f)$ preserves convex combinations, no classical (i.e. non-contextual) strategy can do any better than this. So the classical value of the game, i.e. the maximum winning probability achievable by a classical strategy, is $\frac{3}{4}$. This bound can be saturated by e.g.everyone going for *'grain'*.

However, if the players can coordinate using quantum resources, they can implement the model from Example 10. This results in a strategy that wins with probability $\frac{13}{16} \approx 81\%$. In fact, with a more judicious choice of shared state and measurements one can design a quantum strategy that wins with probability $\frac{(2+\sqrt{2})}{4}$, or approximately 85% of the time. If 'super-quantum' models are allowed, the players can follow a *perfect strategy*, one which allows them to win the game with certainty, namely by using the empirical model from Example 9. This super-quantum model, usually known as the PR box, can in fact be characterised as the unique empirical model that yields a perfect strategy for this game.

13.3.6 Predicates and Possibilistic Models

When one only cares about perfect strategies, i.e. winning a game with certainty, the only relevant information is contained in the supports of the game and the model, i.e. their possibilistic collapses. As it happens, it will be useful later to restrict attention to only those models that win a particular game with certainty, thus interpreting a (possibilistic) game as a predicate on empirical models. We now make some remarks on this situation.

Definition 31 A possibilistic model $e\colon S$ is said to satisfy a possibilistic predicate $g\colon S \longrightarrow [2]$ if $\mathsf{EMP}_{\mathbb{B}}(g)\, e$ is equal to the empirical model on $[2]$ corresponding to the deterministic outcome 1. A probabilistic model $e\colon S$ is said to satisfy a possibilistic predicate $g\colon S \longrightarrow [2]$ if its possibilistic collapse does so.

We write $e\colon \langle S, g\rangle$ to indicate that $e\colon S$ satisfies g. The set of all probabilistic (resp. possibilistic) models on S that satisfy g is denoted by $\mathsf{EMP}(S, g)$ (resp. $\mathsf{EMP}_{\mathbb{B}}(S, g)$).

There is a preorder on possibilistic predicates $S \to [2]$ according to which $g \le h$ whenever $\mathsf{EMP}_{\mathbb{B}}(S, g) \subseteq \mathsf{EMP}_{\mathbb{B}}(S, h)$. This yields an equivalence relation on predicates: g and h are said to be equivalent, written $g \sim h$, if $g \le h$ and $h \le g$, i.e. if $\mathsf{EMP}_{\mathbb{B}}(S, g) = \mathsf{EMP}_{\mathbb{B}}(S, h)$.

Definition 32 A possibilistic model $e\colon S$ induces a predicate $g(e)\colon S \longrightarrow [2]$ given by $g(e) := \bigvee_{\sigma\in\Sigma} g(e)_\sigma$ where $g(e)_\sigma$ is the deterministic procedure consisting of the relation mapping the unique measurement $*$ to the context σ and the characteristic function $\mathcal{E}(\sigma) \to \{0, 1\} = O_*$ of $e_\sigma \subseteq \mathcal{E}(\sigma)$.

In other words, the predicate $g(e)$ is satisfied by a model $d\colon S$ if for each context σ the support of d_σ is contained in that of e_σ, i.e. if $d \le e$ as possibilistic models (note that we may need to take the possibilistic collapse of d).

Example 33 Kochen–Specker models, studied in Abramsky et al. (2011, Sect. 7) and in Mansfield and Barbosa (2014), are certain possibilistic empirical models that are inspired by and closely connected to the original formulation of the Kochen–Specker theorem Kochen et al. (1967). These models are easy to characterise in the current framework as those satisfying a certain possibilistic predicate. Let S be a scenario in which each measurement is dichotomic, i.e. has outcome set $\{0, 1\}$. For each maximal context $\sigma \in \Sigma_S$ of S, define a deterministic predicate g_σ corresponding to the subset of $\mathcal{E}_S(\sigma)$ consisting of the local assignments over σ which assign outcome 1 to exactly one measurement in σ. More formally, π_{g_σ} maps the unique measurement $* \in X_{[2]}$ to σ, while α_{g_σ} sends the subset $\{s \in \mathcal{E}(\sigma) \mid s^{-1}(1) \text{ is a singleton}\}$ to 1 and the rest of $\mathcal{E}(\sigma)$ to 0. Define the Kochen–Specker predicate g_{KS} on S as $\bigvee_\sigma g_\sigma$ where σ ranges over all maximal contexts. For a scenario S, we then say that e is a Kochen–Specker model if it satisfies the predicate g_{KS}. From this point of view, the import of the Kochen–Specker theorem is that, for suitably chosen scenarios of quantum measurements:

1. any quantum state gives rise to a Kochen–Specker model on the scenario;
2. any Kochen–Specker model on the scenario is strongly contextual.

This results in a **state-independent strong contextuality** argument. The results of Abramsky et al. (2011, Sect. 7) can then be seen as giving a criterion on a scenario S that implies the strong contextuality of every Kochen–Specker model on it.

Kochen–Specker models[15] are defined somewhat differently in Mansfield and Barbosa (2014). In contrast to the viewpoint above, there is single Kochen–Specker model on each scenario according to the definition there. Namely, this is defined to be the possibilistic empirical model whose support for each context is given exactly by the winning constraint of the Kochen–Specker predicate above. Importantly, this support of the predicate is guaranteed to satisfy (possibilisitic) no-signalling, and thus it yields a well-defined possibilistic model. In other words, it turns out that the Kochen–Specker predicate g_{KS} is induced by a possiblistic model in the sense of Definition 32; that is, $g_{\mathrm{KS}} = g(e)$ for some possibilistic model e.

This is no accident. In fact, it is true of every possibilistic predicate, at least up to equivalence in the predicate preorder (Definition 31).

Proposition 34 *If a possibilistic predicate $g\colon S \longrightarrow [2]$ is satisfied by some model, then it is equivalent to a predicate $g(e)$ induced by some possibilistic model $e\colon S$.*

Proof As long as $\mathsf{EMP}_{\mathbb{B}}(S, g) \neq \emptyset$, the predicate g is equivalent to g_e where $e = \bigvee \{d \mid d \in \mathsf{EMP}_{\mathbb{B}}(S, g)\}$.

This result means that for the purpose of satisfaction by empirical models, we might as well assume that all predicates are either the predicate *false* (which constantly outputs the outcome 0) or of the form $g(e)$. Note that not only are all satisfiable predicates equivalent to one of this form, but indeed there is a canonical representative for every equivalence class, given by the choice of $e\colon S$ in the proof above.

In a sense, one can think of an arbitrary predicate as defining a ‘theory’ that we are interested in having satisfied by empirical models. Then, passing to the canonical form corresponds to closing this theory under implication assuming only no-signalling as an axiom.

Note that if one instead works with the definition that $g \leq h$ when $\mathsf{EMP}(S, g) \subseteq \mathsf{EMP}(S, h)$, i.e. when any *probabilistic* model satisfying g also satisfies h, one can prove that each satisfiable predicate $g\colon S \longrightarrow [2]$ is equivalent to $g(e)$ where e is the possibilistic collapse of some probabilistic model. One can just take $e = \bigvee \{d \mid d \in \mathsf{EMP}(S, g)\}$.

13.4 When Is a Function Induced by a Procedure?

We now turn our attention towards answering the central question: which functions $F\colon \mathsf{EMP}(S) \longrightarrow \mathsf{EMP}(T)$ are induced by a probabilistic procedure $f\colon S \longrightarrow T$? We will build up to to the answer in stages: we start with the case where $T \cong [n]$ and the function F preserves deterministic models, and then slowly add structure until we culminate with the full solution in Theorem 44.

[15] See Abramsky et al. (2017b, Sect. 4) for a more general account of state-independent contextuality phrased in similar language.

13.4.1 The Affine Span of Deterministic Models

Before diving in, we make some useful preliminary observations. First, note that Lemma 24 implies that it is necessary for F to preserve convex combinations for it to be induced by a procedure. The next two results enable us to deduce that a convex-combination-preserving $F\colon \mathsf{EMP}(S) \longrightarrow \mathsf{EMP}(T)$ is uniquely determined by its action on deterministic empirical models on S, which we can identify with $\mathcal{E}(X_S)$.

A central result of Abramsky et al. (2011) is that the no-signalling condition is equivalent to realisability by a non-contextual hidden variable model with 'negative probabilities'. In other words, any (no-signalling) probabilistic empirical model $e\colon S$ can be extended to a quasiprobability distribution on global assignments; i.e. there is function $d\colon \mathcal{E}(X_S) \longrightarrow \mathbb{R}$ with $\sum_{s\in\mathcal{E}(X_S)} d(s) = 1$ which marginalises to yield $d|_\sigma = e_\sigma$ for every context $\sigma \in \Sigma_S$. We state this result here in the form we will require.

Theorem 35 (Theorem 5.9 of Abramsky et al. (2011) rephrased) *Any probabilistic empirical model $e\colon S$ can be written as an affine combination of deterministic empirical models, i.e. $e = \sum_{s\in\mathcal{E}(X_S)} r_s\delta_s$ for some $r_s \in \mathbb{R}$ with $\sum_s r_s = 1$.*

Lemma 36 *If $F\colon \mathsf{EMP}(S) \longrightarrow \mathsf{EMP}(T)$ preserves convex combinations, then it preserves existing affine combinations. That is, if $e = \sum r_i e_i$ where $e, e_i\colon S$ and $r_i \in \mathbb{R}$ satisfy $\sum r_i = 1$, then $F(e) = \sum r_i F(e_i)$.*

Proof We prove this when $e = r_1e_1 + r_2e_2$, the general case following by induction. Without loss of generality we may assume that $r_1 \geq r_2$. If $r_1 \leq 1$, then this is an ordinary convex combination and there is nothing to prove. If $r_1 > 1$, then rearranging the equation into $e_1 = (\frac{1}{r_1})e - (\frac{r_2}{r_1})e_2$ results in an ordinary convex combination. Plugging this into F, which we know to preserve convex combinations, and then rearranging back gives $F(e) = r_1F(e_1) + r_2F(e_2)$ as desired.

From the last two results we immediately obtain the following.

Theorem 37 *A convex-combination preserving function $F\colon \mathsf{EMP}(S) \longrightarrow \mathsf{EMP}(T)$ is uniquely determined by its restriction to $\mathcal{E}(X_S)$; i.e. if $F, G\colon \mathsf{EMP}(S) \rightrightarrows \mathsf{EMP}(T)$ preserve convex combinations and agree on deterministic models, then $F = G$.*

13.4.2 The Case of Deterministic Experiments

We first consider the crucial special case of functions $F\colon \mathsf{EMP}(S) \longrightarrow \mathsf{EMP}(T)$ that preserve deterministic models. As deterministic models can be identified with global assignments, such an F induces a function $\mathcal{E}(X_S) \to \mathcal{E}(X_T)$. By Theorem 37, it is in fact determined by this restriction.

Specialising further to the case when $T = [n]$, for which $\mathcal{E}(X_T) = \mathcal{E}(\{*\}) = O_{[n],*} = \{0, \ldots, n-1\}$, it is easy to see when such a function arises from a deterministic experiment $S \to [n]$. This happens precisely when F can be computed within a fixed context.

Proposition 38 *A function $f : \mathcal{E}(X_S) \longrightarrow \{0, \ldots, n-1\}$ arises from a deterministic experiment $S \to [n]$ if and only if it factors through $\mathcal{E}_S(\sigma)$ for some $\sigma \in \Sigma_S$.*

The following lemma will be crucial to our characterisation. It contains a result that can actually be phrased in purely classical (i.e. non-contextual, even deterministic) language. It concerns deterministic functions of the form

$$f : O_1 \times \cdots \times O_k \longrightarrow Q$$

which we think about as a function on k arguments. One might be interested in calculating the result of the function by inspecting as few of the arguments as possible. However, deciding which of them to inspect must be done statically, before any of their values are known. The point of the lemma is that there is always a canonical optimal choice, a least subset of arguments that can get the job done. This no longer holds if one is permitted the extra flexibility of dynamically choosing the next argument to inspect depending on previously observed values. Indeed, this constitutes the main obstacle to answering question (A) for adaptive procedures, which have such flexibility Abramsky et al. (2019a).

Lemma 39 *Let S be a scenario and $F : \mathcal{E}(X_S) \longrightarrow Y$ a function. Then there is a least subset $U \subseteq X_S$ such that F factors through $\mathcal{E}(U)$.*

Moreover, let $(F_i)_{i \in I}$ be a family of functions $F_i : \mathcal{E}(X_S) \longrightarrow Y_i$, and let U_i be the least subset such that F_i factors through $\mathcal{E}(U_i)$. Then the least subset U such that $\langle F_i \rangle_{i \in I} : \mathcal{E}(X) \longrightarrow \prod_{i \in I} Y_i$ factors through $\mathcal{E}(U)$ equals $\bigcup_{i \in I} U_i$.

Proof For the first part it suffices to show that if F factors through $\mathcal{E}(V)$ and through $\mathcal{E}(W)$, then it factors through $\mathcal{E}(V \cap W)$. Note first that F factors through $\mathcal{E}(U)$ if and only if for all $s, t \in \mathcal{E}(X_S)$, $s|_U = t|_U$ implies $F(s) = F(t)$. Now, assume that F factors through $\mathcal{E}(V)$ and through $\mathcal{E}(W)$, and let $s, t \in \mathcal{E}(X_S)$ satisfying $s|_{V \cap W} = t|_{V \cap W}$. Define r by gluing together $s|_V$ and $t|_{X \setminus V}$. This satisfies $r|_V = s|_V$ and $r|_W = t|_W$ by construction. As F factors through $\mathcal{E}(V)$ and through $\mathcal{E}(W)$, we conclude that $F(s) = F(r) = F(t)$, as desired.

For the second part, let F_i, U_i and U be as in the statement. Since $\bigcup_i U_i$ contains each U_i, the function F_i factors through $\mathcal{E}(\bigcup_i U_i)$ and thus so does $\langle F_i \rangle_{i \in I}$. Hence, $U \subseteq \bigcup_i U_i$. On the other hand, each F_i factors through $\mathcal{E}(U)$, whence U contains each U_i and thus it contains $\bigcup_i U_i$, completing the proof.

We discuss a more abstract interpretation of this result in Sect. 13.6, when we examine why its analogue fails in the adaptive case.

13.4.3 The Case of Probabilistic Experiments

We now generalise Proposition 38 in order to answer the question of when a function $F\colon \mathsf{EMP}(S) \to \mathsf{EMP}([n])$ is induced by a probabilistic experiment $S \to [n]$. Assuming that F preserves convex combinations, Theorem 37 implies that we can simplify this situation by restricting F to $\mathcal{E}(X_S)$.

Note that $\mathsf{EMP}([n]) \cong D(O_{[n],*}) \cong D(\{0, \ldots n-1\})$. A function

$$F\colon \mathcal{E}(X_S) \longrightarrow D(\{0, \ldots n-1\})$$

can be uniquely (up to reordering) described as a convex combination $F = \sum_j r_j F_j$ where $r_j > 0$ with $\sum_j r_j = 1$ and where

$$F_j\colon \mathcal{E}(X_S) \longrightarrow \{0, \ldots, n-1\}$$

are (distinct) deterministic functions. Now, let $U_j \subseteq X_S$ be the least subset of X_S such that F_j factors through the projection $\mathcal{E}(X_S) \to \mathcal{E}(U_j)$, as guaranteed by Lemma 39. We show that F is induced by a probabilistic experiment $S \to [n]$ if and only if each U_j is a context in Σ_S.

To see this, suppose first that F is induced by a probabilistic experiment $\sum_k c_k f_k$ with f_k a deterministic experiment. Then, $F = \sum c_k \mathsf{EMP}(f_k)$. Since expressing F as a convex combination of deterministic functions is unique up to rearranging and collecting equal terms, we conclude that each F_j is equal to $\mathsf{EMP}(f_k)$ for some k. That is, F_j is induced by a deterministic experiment, and thus that $U_j \in \Sigma_S$. Conversely, if each U_j is in Σ_S then Proposition 38 implies that each F_j is induced by a deterministic experiment, say f_j. Then, $F = \sum_j r_j F_j = \sum_j r_j \mathsf{EMP}(f_j) = \mathsf{EMP}(\sum_j r_j f_j)$ is induced by a probabilistic experiment.

13.4.4 The General Case and Procedures as Empirical Models

We now wish to answer the general question of when a function $F\colon \mathsf{EMP}(S) \to \mathsf{EMP}(T)$ is induced by a probabilistic procedure $S \to T$. In order to understand F we can study the composites

$$\mathsf{EMP}(S) \to \mathsf{EMP}(T) \to \mathsf{EMP}(T\|_{\sigma})$$

for each context $\sigma \in \Sigma_T$. Note that each of these is an instance of the simpler setting considered in Sect. 13.4.3. As before, we may assume that F preserves convex combinations, so that it is determined by its values on $\mathcal{E}(X_S)$. Since in addition $\mathsf{EMP}(T|_\sigma) \cong D(\mathcal{E}_T(\sigma))$, the composite above can equivalently be described as a convex mixture of (deterministic) functions of type

$$F_i : \mathcal{E}_S(X_S) \longrightarrow \mathcal{E}_T(\sigma) .$$

So, for each context $\sigma \in \Sigma_T$ the function F yields some probabilistic data. This suggests thinking of this data as an empirical model on a new scenario. We now make this idea precise.

Definition 40 Let S and T be measurement scenarios. We define a new scenario $[S, T]$ by setting $X_{[S,T]} := X_T$, $\Sigma_{[S,T]} := \Sigma_T$, and

$$O_{[S,T],x} := \left\{ \langle U, \alpha \rangle \mid U \subseteq X_S, \alpha : \mathcal{E}_S(U) \longrightarrow O_{T,x} \right\} .$$

We equip $[S, T]$ with the possibilistic predicate $g_{S,T}$ defined by $g_{S,T} := \bigvee_{\sigma \in \Sigma_T} g_\sigma$ where g_σ checks that only a compatible part of S is used in the simulation of the context $\sigma \in \Sigma_T$. More formally, g_σ corresponds to the subset

$$\left\{ (\langle U_x, \alpha_x \rangle)_{x \in \sigma} \mid \bigcup_{x \in \sigma} U_x \in \Sigma_S \right\} \subseteq \mathcal{E}_{[S,T]}(\sigma) .$$

Proposition 41 *Deterministic procedures $S \to T$ correspond bijectively to deterministic empirical models of $[S, T]$ satisfying $g_{S,T}$.*

Proof Deterministic models are determined by global assignments. Such an assignment $s \in \mathcal{E}(X_{[S,T]})$ consists of an outcome for each measurement $x \in X_{[S,T]} = X_T$. Each such outcome is a pair $\langle U_x, \alpha_x \rangle$ consisting of a subset of measurements of S, $U_x \subseteq X_S$, and a function $\alpha_x : \mathcal{E}_S(U_x) \longrightarrow \mathcal{E}_T(\{x\})$.

This is almost exactly the data needed to specify a deterministic procedure $f = \langle \pi_f, \alpha_f \rangle : S \longrightarrow T$, with π_f defined so that $\pi_f(x) := U_x$. The only caveat is that the relation $\pi_f : X_T \longrightarrow X_S$ thus defined need not be simplicial (with respect to the complexes Σ_T and Σ_S). It turns out that it is a simplicial relation if and only if the deterministic model δ_s satisfies the predicate $g_{S,T}$.

Corollary 42 *Probabilistic procedures $S \to T$ correspond bijectively to probability distributions on*

$$\{s \in \mathcal{E}_{[S,T]}(X_{[S,T]}) \mid \delta_s \textit{ satisfies } g_{S,T}\} ,$$

and thus give rise to all non-contextual models $e : \langle [S, T], g_{S,T} \rangle$.

Proof Recall that a probabilistic procedure is a convex combination of deterministic procedures. The statement thus follows from applying D to both sides of the bijection from Proposition 41 between deterministic procedures $S \to T$ and deterministic models in $\langle [S, T], g_{S,T} \rangle$.

Explicitly, given a probabilistic procedure $\sum r_i f_i$ with each $f_i : S \longrightarrow T$ a deterministic procedure, and writing $s_i \in \mathcal{E}(X_{[S,T]})$ for the global assignment corresponding to f_i, then $\sum r_i \delta_{s_i} : \langle [S, T], g_{S,T} \rangle$ is the empirical model corresponding to the procedure $\sum r_i f_i$. It satisfies $g_{S,T}$ because each δ_{s_i} does, and moreover, it is non-contextual by construction.

Conversely, any non-contextual model $\langle [S, T], g_{S,T}\rangle$ can be written (not necessarily uniquely) as a convex combination of deterministic models δ_s satisfying $g_{S,T}$, so it arises in this fashion.

We now show how to encode a convex-combination-preserving function $F : \mathsf{EMP}(S) \longrightarrow \mathsf{EMP}(T)$ as an empirical model on $[S, T]$. For each $\sigma \in \Sigma_T$, recall from the discussion above that the composite

$$\mathsf{EMP}(S) \to \mathsf{EMP}(T) \to \mathsf{EMP}(T|\sigma)$$

is determined by a convex mixture $\sum r_i F_i$ of functions

$$F_i : \mathcal{E}_S(X_S) \longrightarrow \mathcal{E}_T(\sigma) .$$

Let $U_i \subseteq X_S$ be the least subset of X_S such that F_i factors through the projection $\mathcal{E}_S(X_S) \to \mathcal{E}_S(U_i)$, and write

$$\tilde{F}_i : \mathcal{E}_S(U_i) \longrightarrow \mathcal{E}_T(\sigma)$$

for the remaining factor. Recalling that $\mathcal{E}_T(\sigma) = \prod_{x\in\sigma} O_{T,x}$, the function $\tilde{F}_i$ is determined by the maps

$$\tilde{F}_{i,x} := \pi_x \circ \tilde{F}_i : \mathcal{E}_S(U_i) \longrightarrow O_{T,x} .$$

Given the signature of this map, note that $\langle U_i, \tilde{F}_{i,x}\rangle$ is an element of $O_{[S,T],x}$. Putting these together for all measurements x in a context σ yields an element of $\mathcal{E}_{[S,T]}(\sigma)$, written $\langle U_i, \tilde{F}_i\rangle := (\langle U_i, \tilde{F}_{i,x}\rangle)_{x\in\sigma}$.

We now define a family $e_F = (e_{F,\sigma})_{\sigma\in\Sigma_T}$ where $e_{F,\sigma}$ is the distribution on $\mathcal{E}_{[S,T]}(\sigma)$ given by $e_{F,\sigma} := \sum r_i \langle U_i, \tilde{F}_i\rangle$.

Lemma 43 *For any function $F : \mathcal{E}(S) \to \mathsf{EMP}(T)$ that preserves convex combinations, the family e_F defines a no-signalling empirical model $e_F : [S, T]$.*

Proof Clearly, each $e_{F,\sigma}$ as defined above is a probability distribution over $\mathcal{E}_{[S,T]}(\sigma)$. It thus remains to check that e_F is no-signalling. When $\tau \subseteq \sigma$, we get that

$$e_{F,\sigma}|_\tau = \sum_i r_i (\langle U_i \tilde{F}_{i,x}\rangle)_{x\in\tau} .$$

As the decomposition of the experiment $\mathcal{E}_S(X_S) \to \mathcal{E}_T(\sigma) \to \mathcal{E}_T(\tau)$ into deterministic experiments is unique, this is equal to $e_{F,\tau}$.

Theorem 44 *A function $F : \mathsf{EMP}(S) \to \mathsf{EMP}(T)$ preserving convex combinations is induced by a probabilistic procedure $S \to T$ if and only if e_F is non-contextual and satisfies $g_{S,T}$.*

Proof ($\Rightarrow$): Assume that $F = \mathsf{EMP}(\sum r_i f_i)$ with $f_i = \langle \pi_i, \alpha_i\rangle$ deterministic procedures. Without loss of generality, we may replace each $\langle \pi_i, \alpha_i\rangle$ with one where $\pi(x)$

is the least subset (as guaranteed by Lemma 39) with the property that $\alpha_{i,x}$ factors through $\mathcal{E}(X_T) \to \mathcal{E}(\pi(x))$, as this does not affect the push-forward of f_i. As these least subsets and the induced factorisations are unique, we now see that e_F admits a convex decomposition as $\sum r_i \delta_{s_i}$, where s_i is the global assignment corresponding to f_i as in Proposition 41. As each s_i satisfies $g_{S,T}$, so does e_F.

($\Leftarrow$): As $e_F : \langle [S, T], g_{S,T} \rangle$ is non-contextual, it arises as some distribution $\sum r_s \delta_s$ for global assignments $s \in \mathcal{E}(X_{[S,T]})$. Each global assignment s corresponds to a deterministic map $f_s = \langle \pi_s, \alpha_s \rangle \colon S \longrightarrow T$ by Proposition 41. By construction F and $\mathsf{EMP}(\sum r_s f_s)$ have the same action when restricted to $\mathcal{E}(X_T)$ and projected to some context $\sigma \in \Sigma_\tau$, so that Theorem 37 implies that $F = \mathsf{EMP}(\sum r_s f_s)$, as desired.

13.5 Getting Closure

The results of the previous section strongly suggest thinking of $[S, T]$ as something like an *internal hom*. However, the construction of $[S, T]$ does not take into account the measurement compatibility structure of S encoded in the simplicial complex Σ_S—that is taken care of by the predicate $g_{S,T}$. This suggests that one should work with pairs $\langle S, g \colon S \longrightarrow [2] \rangle$ as the basic objects of our category.

In this section we make this viewpoint precise and show that it results in a **closed category**, a notion introduced in Eilenberg and Kelly (1966). Instead of the original definition, we work with the axiomatisation given in Manzyuk (2012) and attributed to Laplaza (1977) and Street (1974, Sect. 4). Roughly speaking, closed categories are like monoidal closed categories without the monoidal structure: they axiomatise the notion of a category C where the collection of morphisms $A \to B$ can be given the structure of a C-object. This amounts to defining a bifunctor $[-, -] \colon \mathsf{C}^{\mathrm{op}} \times \mathsf{C} \to \mathsf{C}$ with suitable structure on it satisfying some coherence conditions. The idea is that this structure captures the notions of identity and composition. In the absence of a monoidal product, in contrast to the definition of a monoidal closed category, this internal hom structure must be encoded in a *curried* version. For example, composition is represented by a transformation $L^A_{B,C} \colon [B, C] \longrightarrow [[A, B], [A, C]]$ natural in B and C and dinatural in A.

As the full details are rather technical, we do not recall the complete definition here, even if we do check the conditions in some detail in the proof of Theorem 46. The reader interested in understanding finer points of the proof should be able to do so after consulting the definition in Manzyuk (2012).

Definition 45 We define a new category whose objects are pairs $\langle S, g \colon S \longrightarrow [2] \rangle$ consisting of a scenario and a **structure predicate** on it. We assume that the structure predicate g on S is induced by some (possibilistic) model on S as per Definition 32. A morphism $\langle S, g \rangle \to \langle T, h \rangle$ is given by a probabilistic procedure $f : S \to T$ such that for any probabilistic model $e \colon \langle S, g \rangle$ we have $\mathsf{EMP}(f)\, e \colon \langle T, h \rangle$, i.e. if e satisfies the structure predicate on S, then $\mathsf{EMP}(f)\, e$ satisfies the structure predicate on T. We denote this category by Scen^g. The subcategory of Scen^g with deterministic maps is denoted by $\mathsf{Scen}^g_{\mathrm{Det}}$, while $\mathsf{Scen}^g_{\mathbb{B}}$ denotes the category obtained by replacing probabilistic procedures and models with possibilistic ones in the preceding definition.

Given two objects $\langle S, g\rangle$ and $\langle T, h\rangle$ of Scen^g we define a new object $[\langle S, g\rangle, \langle T, h\rangle]$ by setting

$$[\langle S, g\rangle, \langle T, h\rangle] := \langle [S, T], \bigvee_{\sigma \in \Sigma_T} g_{\sigma,S,T,g,h}\rangle$$

where $g_{\sigma,S,T,g,h}$ is a deterministic predicate corresponding to the subset of $\mathcal{E}_{[S,T]}(\sigma)$ defined by

$$\left\{ (\langle U_x, f_x : \mathcal{E}_S(U_x) \longrightarrow O_{T,x}\rangle)_{x\in\sigma} \mid U := \bigcup_{x\in\sigma} U_x \in \Sigma_S, \forall s \in \mathcal{E}_S(U).\ s \in g_U \Rightarrow f(s) := (f_x(s|_{U_x}))_{x\in\sigma} \in h_\sigma \right\}.$$

In other words, the predicate on $[\langle S, g\rangle, \langle T, h]\rangle$ checks, for each context σ of Σ_T, two things:

1. that only a context of Σ_S is used;
2. that any local assignment satisfying g is mapped to an assignment satisfying h.

The requirement that the structure predicates be induced by possibilistic models is so that one can check locally whether $f: S \to T$ is in fact a map $\langle S, g\rangle \to \langle T, h\rangle$. This enables one to write down the definition of the structure predicate on $[\langle S, g\rangle, \langle T, h\rangle]$. Of course, one should check that this structure predicate is also induced by a possibilistic model. By construction, it is given by a Boolean distribution for each σ, so one need only check that this is non-signalling. That is, we need to show that if $\tau \subseteq \sigma$, then any possible $(\langle U_x, f_x : \mathcal{E}_S(U_x) \longrightarrow O_{T,x}\rangle)_{x\in\tau}$ can be extended to the context σ. Given an assignment $s \in \mathcal{E}_S(U)$, this is already mapped to an assignment $f(s) \in \mathcal{E}_T(\tau)$ given by $f(s) := (f_x(s|_{U_x}))_{x\in\sigma}$. Moreover, if s satisfies g_U then $f(s)$ satisfies h_τ, and $f(s)$ can therefore be extended to some assignment $t_s \in \mathcal{E}_T(\sigma)$ satisfying h_σ because h is non-signalling. For $y \in \sigma \setminus \tau$, we set $U_y := U$ and $f_y : \mathcal{E}_S(U) \longrightarrow O_{T,y}$ given by $f_y(s) := (t_s)_y$ whenever $s \in g_U$ (note that $f_y(s)$ can be set to anything for $s \notin g_U$).

Note that the restriction to predicates induced by possibilistic models is made out of convenience and does not represent a consequential choice. This is because Proposition 34 shows that any predicate is equivalent to one induced by a possibilistic model, in the sense that they have the same set of satisfying models, which is their relevant characteristic here. Therefore, we could have chosen as objects arbitrary pairs of scenarios and predicates and then use the canonical representatives given by Proposition 34 in the definition of $g_{\sigma,S,T,g,h}$.

Note that a plain scenario S can be seen as pair $\langle S, \mathsf{t}\rangle$ where t is a trivial predicate that is always satisfied. Our earlier categories of scenarios are full subcategories of the corresponding categories where objects are scenarios equipped with structure predicates.

Theorem 46 $[-,-]$ *makes* Scen^{g}_{Det} *into a closed category.*

Proof We first show how to make $[-,-]$ functorial in both variables. Given a deterministic morphism $\langle \pi, \alpha\rangle \colon \langle S, e\rangle \longrightarrow \langle T, f\rangle$, we will define a morphism

$$[\mathrm{id}, \langle \pi, \alpha\rangle] \colon [\langle P, g\rangle, \langle S, e\rangle] \longrightarrow [\langle P, g\rangle, \langle T, f\rangle]$$

for any $\langle P, g\rangle$. This boils down to defining a morphism $[\mathrm{id}, \langle \pi, \alpha\rangle] \colon [P, S] \longrightarrow [P, T]$ and explaining why it agrees with the relevant predicates. As $\Sigma_S = \Sigma_{[P,S]}$ and similarly for T, we can define the simplicial relation underlying $[\mathrm{id}, \langle \pi, \alpha\rangle]$ to be π. Now, given $x \in X_T$, we define the function $\mathcal{E}_{[P,S]}(\pi x) \to O_{[P,T],x}$ to be "postcomposition with α_x". More precisely, we send a joint assignment in $\mathcal{E}_{[P,S]}(\pi(x))$, i.e. a family $(\langle U_y, k_y\rangle)_{y\in\pi(x)}$ of functions $k_y \colon \mathcal{E}_P(U_y) \longrightarrow O_{S,y}$, to the element of $O_{[P,T],x}$ given by

$$\left\langle\ U := \bigcup_{y\in\pi(x)} U_y\ ,\quad \alpha_x \circ \langle k_y \circ \rho_y\rangle_{y\in\pi(x)}\ \right\rangle$$

where $\rho_y := \mathcal{E}_P(U_y \subseteq U) \colon \mathcal{E}_P(U) \longrightarrow \mathcal{E}_P(U_y)$ is the obvious projection; i.e. we take the composite at the bottom of the following diagram

$$\begin{tikzcd}
 & \mathcal{E}_P(U_y) \arrow[r, "k_y"] & O_{S,y} & \\
\mathcal{E}_P(U) \arrow[ur, "\rho_y"] \arrow[r, dashed, "\langle k_y\circ\rho_y\rangle_{y\in\pi x}"] & \mathcal{E}_S(\pi x) = \prod\limits_{y\in\pi x} O_{S,y} \arrow[ur] \arrow[r, "\alpha_x"] & \mathcal{E}_T(x) &
\end{tikzcd} .$$

To see that this results in a morphism $[\langle P, g\rangle, \langle S, e\rangle] \to [\langle P, g\rangle, \langle T, f\rangle]$, note that a function $k \colon \mathcal{E}_P(U) \to \mathcal{E}_S(\sigma)$ is an allowed outcome of $[\langle P, g\rangle, \langle S, e\rangle]$ if and only if $U \in \Sigma_P$ and k sends outcomes in the support of g to outcomes in the support of e. Whenever this happens, the fact that $\langle \pi, \alpha\rangle$ is a morphism $\langle S, e\rangle \to \langle T, f\rangle$ implies that postcomposing with α results in a function that sends outcomes in the support of g to outcomes in the support of f, and thus $[\mathrm{id}, \langle \pi, \alpha\rangle]$ is indeed a morphism $[\langle P, g\rangle, \langle S, e\rangle] \to [\langle P, g\rangle, \langle T, f\rangle]$. It is clear that this action on morphisms is functorial in the second variable.

Moving now to the first variable, given a deterministic morphism $\langle \pi, \alpha\rangle \colon \langle S, e\rangle \to \langle T, f\rangle$, we wish to define a morphism

$$[\langle \pi, \alpha\rangle, \mathrm{id}] \colon [\langle T, f\rangle, \langle P, g\rangle] \to [\langle S, e\rangle, \langle P, g\rangle]\,.$$

We define the underlying simplicial relation to be id_P, and define the action on outcomes by "precomposition with $\langle \pi, \alpha\rangle$". More specifically, given an outcome of $[T, P]$ at $x \in X_P$, of the form $\langle U, \mathcal{E}_T(U) \to O_{P,x}\rangle$, we send it to the composite

$\langle\, \pi U,\ \mathcal{E}_S(\pi U) \xrightarrow{\alpha_U} \mathcal{E}_T(U) \to O_{P,x}\,\rangle$. To see that this cooperates with the predicates these scenarios are equipped with, note that a possible outcome at $\sigma \in \Sigma_{[T,P]}$ corresponds to a function $\mathcal{E}_T(U) \to \mathcal{E}_P(\sigma)$ with $U \in \Sigma_T$ that sends outcome assignments allowed by f to assignments allowed by g. Whenever this is the case, the composite $\mathcal{E}_S(\pi U) \xrightarrow{\alpha_U} \mathcal{E}_T(U) \to O_{P,x}$ satisfies $\pi U \in \Sigma_S$ and it sends outcomes allowed by e to outcomes allowed by g. Thus $[\langle \pi, \alpha\rangle, \mathrm{id}]$ indeed is a morphism $[\langle T, f\rangle, \langle P, g\rangle] \to [\langle S, e\rangle, \langle P, g\rangle]$.

Now that $[-,-]$ has been shown to be functorial in each variable, it remains to that it is a bifunctor $(\mathsf{Scen}^g_{\mathrm{Det}})^{\mathrm{op}} \times \mathsf{Scen}^g_{\mathrm{Det}} \to \mathsf{Scen}^g_{\mathrm{Det}}$. But this is clear, since pre- and postcomposition commute with each other, i.e. $(f \circ g) \circ h = f \circ (g \circ h)$ for all functions f, g, h.

We have now defined the bifunctor $[-,-]$. We next move to defining additional structure this bifunctor has, after which we check that these structures satisfy the required axioms. As our unit object, we choose the pair $\langle 0, \mathsf{t}\rangle$, where 0 is the empty scenario and t is the predicate corresponding to the value $1 \in \mathcal{E}_{[2]}(*)$. For any $\langle S, g\rangle$, there is an isomorphism $i_{S,g}\colon \langle S, g\rangle \to [\langle 0, \mathsf{t}\rangle, \langle S, g\rangle]$ whose underlying simplicial relation is just id and which identifies an outcome in $\mathcal{E}_S(X)$ with the function $\mathcal{E}_0(\emptyset) = \{*\} \to \mathcal{E}_S(x)$ (an outcome of $[\langle 0, \mathsf{t}\rangle, \langle S, g\rangle]$ at x) having it as its value at the unique point of the domain. These isomorphisms are clearly natural in $\langle S, g\rangle$.

We define a collection of morphisms $j_{S,g}\colon \langle 0, \mathsf{t}\rangle \to [\langle S, g\rangle, \langle S, g\rangle]$ by setting the underlying simplicial relation to be the empty relation. The function $\mathcal{E}_0(\emptyset) \to \mathcal{E}_{[S,S]}(x)$ then sends the unique element of $\mathcal{E}_0(\emptyset)$ to the function $\mathrm{id}\colon \mathcal{E}_S(x) \to \mathcal{E}_S(x)$. The morphisms $j_{S,g}$ should be dinatural in $\langle S, g\rangle$, which amounts to checking the commutativity of

$$\begin{tikzcd}
\langle 0, \mathrm{t}\rangle \arrow[r, "j_{S,g}"] \arrow[d, "j_{T,h}"'] & {[\langle S, g\rangle, \langle S, g\rangle]} \arrow[d, "{[\mathrm{id}, \langle \pi, \alpha\rangle]}"] \\
{[\langle T, h\rangle, \langle T, h\rangle]} \arrow[r, "{[\langle \pi, \alpha\rangle, \mathrm{id}]}"'] & {[\langle S, g\rangle, \langle T, h\rangle]}
\end{tikzcd}$$

for any $\langle \pi, \alpha\rangle\colon \langle S, g\rangle \to \langle T, h\rangle$. Both of the morphisms have the empty relation as their underlying simplicial relation. Moreover, both of these morphisms have the same functions on outcomes: for $x \in X_T = X_{[S,T]}$ we get the function that sends the unique element of $\mathcal{E}_0(\emptyset)$ to $\mathcal{E}_S(\pi x) \xrightarrow{\alpha_x} \mathcal{E}_T(x)$.

Next we need to define morphisms $L^{P,f}_{S,g,T,h}\colon [\langle S, g\rangle, \langle T, h\rangle] \to [[\langle P, f\rangle, \langle S, g\rangle], [\langle P, f\rangle, \langle T, h\rangle]]$ and check that they are natural in $\langle S, g\rangle, \langle T, h\rangle$ and dinatural in $\langle P, f\rangle$. The intuition is that these morphisms internalise the composition operation on morphisms. We define $L^{P,f}_{S,g,T,h}$ as follows. As on both sides the underlying simplicial complex is (isomorphic to) Σ_T, we can set the underlying simplicial relation to be the identity. To define the functions on outcomes, we need for each $x \in X_T$, a function that sends elements of $\mathcal{E}_{[S,T]}(x) = O_{[S,T],x}$ to elements of $O_{[[P,S],[P,T]],x}$.

Given an element of $O_{[S,T],x}$, i.e. a tuple $\langle U, k\colon \mathcal{E}_S(U) \longrightarrow O_{T,x}\rangle$, we send it to $\langle U, \tilde{k}\rangle \in O_{[[P,S],[P,T]],x}$ where $\tilde{k}\colon \mathcal{E}_{[P,S]}(U) \longrightarrow O_{[P,T],x}$ maps an element $(\langle V_y, m_y\colon \mathcal{E}_P(V_y) \longrightarrow O_{S,y}\rangle)_{y \in U}$ of $\mathcal{E}_{[P,S]}(U)$ to the element of $O_{[P,T],x}$ given by

$$\langle \;\; V := \bigcup_{y \in U} V_y \;, \quad k \circ \langle m_y \circ \rho_y \rangle_{y \in U} \;\; \rangle$$

where the function is the composite at the bottom of the following diagram:

$$\begin{tikzcd}
 & \mathcal{E}_P(V_y) \arrow[r, "m_y"] & O_{S,y} & \\
\mathcal{E}_P(V) \arrow[ur, "\rho_y"] \arrow[rr, dashed, "\langle m_y \circ \rho_y \rangle_{y \in U}"] & & \mathcal{E}_S(U) = \prod_{y \in U} O_{S,y} \arrow[u] \arrow[r, "k"] & O_{T,x}
\end{tikzcd}$$

Thus composition with k defines an element of $O_{[[P,S],[P,T]],x}$. As morphisms of scenarios that are compatible with the structure predicates compose, the morphism of scenarios $[S, T] \to [[P, S], [P, T]]$ is indeed a morphism $[\langle S, g\rangle, \langle T, h\rangle] \to [[\langle P, f\rangle, \langle S, g\rangle], [\langle P, f\rangle, \langle T, h\rangle]]$. Naturality of $L^{P,f}_{S,g,T,h}$ in $\langle S, g\rangle$ and $\langle T, h\rangle$ follows from associativity of function composition. Dinaturality in $\langle P, f\rangle$ amounts to the commutativity of

$$\begin{tikzcd}
{[\langle S,g\rangle, \langle T,h\rangle]} \arrow[r, "L^{P,f}_{S,g,T,h}"] \arrow[d, "L^{Q,e}_{S,g,T,h}"'] & {[[\langle P,f\rangle, \langle S,g\rangle], [\langle P,f\rangle, \langle T,h\rangle]]} \arrow[d, "{[\mathrm{id}, [\langle \pi,\alpha\rangle, \mathrm{id}]]}"] \\
{[[\langle Q,e\rangle, \langle S,g\rangle], [\langle Q,e\rangle, \langle T,h\rangle]]} \arrow[r, "{[[\langle \pi,\alpha\rangle], \mathrm{id}], \mathrm{id}]}"'] & {[[\langle P,f\rangle, \langle S,g\rangle], [\langle Q,e\rangle, \langle T,h\rangle]]}
\end{tikzcd}$$

for any $\langle \pi, \alpha\rangle : \langle Q, e\rangle \longrightarrow \langle P, f\rangle$, which again follows from associativity of function composition.

Next we check that this data satisfies the five axioms required of a closed category. Here we refer to the numbering CC1-CC5 used in Manzyuk (2012).

Axiom CC1 amounts to commutativity of

$$\begin{tikzcd}
\langle 0, \mathsf{t}\rangle \arrow[r, "j_{S,g}"] \arrow[dr, "j_{[\langle T,h\rangle, \langle S,g\rangle]}"'] & {[\langle S,g\rangle, \langle S,g\rangle]} \arrow[d, "L^{T,h}_{S,g,S,g}"] \\
 & {[[\langle T,h\rangle, \langle S,g\rangle], [\langle T,h\rangle, \langle S,g\rangle]]}
\end{tikzcd}$$

which boils down to the fact that composing with id keeps everything else fixed. The diagram

$$\begin{tikzcd}
{[\langle S,g\rangle, \langle T,h\rangle]} \arrow[r, "L^{S,g}_{S,g,T,h}"] \arrow[dr, "i_{[\langle S,g\rangle, \langle T,h\rangle]}"'] & {[[\langle S,g\rangle, \langle S,g\rangle], [\langle S,g\rangle, \langle T,h\rangle]]} \arrow[d, "{[j_{S,g}, \mathrm{id}]}"] \\
 & {[\langle 0, \mathsf{t}\rangle, [\langle S,g\rangle, \langle T,h\rangle]]}
\end{tikzcd}$$

commutes essentially for the same reason, establishing CC2. Axiom CC3 boils down to associativity of composition in $\mathsf{Scen}^g_{\mathrm{Det}}$, and CC4 is straightforward to check.

The final and perhaps most important axiom, CC5, asserts that the function that sends $f\colon \langle S, g\rangle \longrightarrow \langle T, h\rangle$ to the composite $\langle 0, \mathsf{t}\rangle \xrightarrow{j_{S,g}} [\langle S, g\rangle, \langle S, g\rangle] \xrightarrow{[\mathrm{id}, f]} [\langle S, g\rangle, \langle T, h\rangle]$ defines a bijection between morphisms $\langle S, g\rangle \to \langle T, h\rangle$ and morphisms $\langle 0, \mathsf{t}\rangle \to [\langle S, g\rangle, \langle T, h\rangle]$. This essentially follows from Proposition 41: deterministic morphisms $\langle 0, \mathsf{t}\rangle \to [\langle S, g\rangle, \langle T, h\rangle]$ that cooperate with the structure predicates are the same thing as morphisms $\langle S, g\rangle \to \langle T, h\rangle$.

Corollary 47 $[-, -]$ *makes* Scen^g *and* $\mathsf{Scen}^g_{\mathbb{B}}$ *into closed categories.*

Proof We sketch the proof for Scen^g, the case of $\mathsf{Scen}^g_{\mathbb{B}}$ being similar. We extend

$$[-, -]\colon (\mathsf{Scen}^g_{\mathrm{Det}})^{\mathrm{op}} \times \mathsf{Scen}^g_{\mathrm{Det}} \to \mathsf{Scen}^g_{\mathrm{Det}}$$

to a functor

$$[-, -]\colon (\mathsf{Scen}^g)^{\mathrm{op}} \times \mathsf{Scen}^g \to \mathsf{Scen}^g$$

by keeping the definition on objects fixed, and defining

$$\left[\sum_i r_i f_i, \sum_j s_j g_j\right] := \sum_{i,j} r_i s_j \left[f_i, g_j\right] .$$

We define the morphisms j, i, L as before, and their naturality (and dinaturality) for the extended functor follows from them being natural (or dinatural) in the first place.[16] Similarly, the axioms CC1-CC4 follow from those holding in $\mathsf{Scen}^g_{\mathrm{Det}}$. Finally, we wish to check axiom CC5, showing that postcomposition with $[\mathrm{id}, \sum_i r_i f_i] = \sum_i r_i [id, f_i]$ defines a bijection between probabilistic morphisms $\langle S, g\rangle \to \langle T, h\rangle$ and probabilistic morphisms $\langle 0, \mathsf{t}\rangle \to [\langle S, g\rangle, \langle T, h\rangle]$. As we already had a bijection in the deterministic case, and since composition is convex, a convex mixture of morphisms $\langle S, g\rangle \to \langle T, h\rangle$ corresponds bijectively to a convex mixture of morphisms $\langle 0, \mathsf{t}\rangle \to [\langle S, g\rangle, \langle T, h\rangle]$, which completes the proof.

Remark 48 The category $\mathsf{Scen}^g_{\mathbb{B}}$ is isomorphic to $\mathsf{Emp}^{\leq}_{\mathbb{B}}$. To see this, note that one can think of an object $\langle S, g\rangle$ of $\mathsf{Scen}^g_{\mathbb{B}}$ as a pair $\langle S, e_g\rangle$ where $e \in \mathsf{EMP}_{\mathbb{B}}(S)$ induces g. Moreover, saying that a possibilistic procedure $f\colon S \to T$ is a morphism $\langle S, g\rangle \to \langle T, h\rangle$ is equivalent to saying that $\mathsf{EMP}_{\mathbb{B}}(f)e_g$ is included in e_h, i.e. that f defines a weak simulation $e_g \to e_h$, i.e. a morphism $e_g \to e_h$ in $\mathsf{Emp}^{\leq}_{\mathbb{B}}$.

[16] Most of this should follow from the fact that changing the basis of enrichment is a 2-functor and in our case preserves the duality involutions, so one should automatically get a functor $[-, -]$ and the required natural transformations, leaving only dinaturality and some of the axioms to be checked by hand. However, we are not aware of general results guaranteeing that change-of-basis preserves closed structure, so we sketch the hands-on proof.

13.6 What Is to Be Done?

With the results of the previous sections in hand, we now turn to a discussion of some of the open questions and future research directions that naturally suggest themselves. We also offer some unpolished thoughts on how the sheaf and resource approach to contextuality connects to other areas of research to which Samson has made profound contributions.

13.6.1 Using the Closure

Having a closed structure on a category lets one turn relativised questions, i.e. those about morphisms, into questions about objects. This can be especially beneficial when the techniques for studying the objects are relatively well-developed. In our case, it suggest studying the resource theory of contextuality using the plethora of tools developed for studying contextuality, e.g. cohomology Abramsky et al. (2012b, 2015), the contextual fraction Abramsky et al. (2017c), or Bell inequalities. In particular, one is left wondering if, for suitable choices of scenarios, the logical Bell inequalities of Abramsky et al. (2012a) have further logical structure that would help in answering questions of simulability.

13.6.2 Variants of the Main Result

Recall that sending a procedure f to $\mathsf{EMP}(f)$ defines a functor from Scen to the category of (convex) sets. In a sense, Theorem 44 can be seen as characterising the image of this functor. One obtains a family of related problems by asking about the images of other similarly defined functors. Each results in an interesting question that isn't immediately answered by the above.

A natural approach to answering these alternative questions is by following the same strategy: given a variant EMP^* of EMP, and a well-behaved function $f : \mathsf{EMP}^*(S) \to \mathsf{EMP}^*(T)$ (or perhaps $f : \mathsf{EMP}^*\langle S, g\rangle \to \mathsf{EMP}^*\langle T, h\rangle$), one would hope to build a corresponding empirical model e_f over some variant of $[S, T]$ and then show that f is induced by a procedure if and only if e_f is non-contextual and satisfies a certain predicate on $[S, T]$. However, for interesting variants of EMP^*, this strategy runs into trouble right at the outset—it is not obvious that there is a way of building a model e_f, even if there are reasonable variants of $[S, T]$ for the category in question. We will explain the specific issues arising along each axis of variation for our notion of procedure.

Possibilistic procedures

In the possibilistic case, we wish to know when a map $f : \mathsf{EMP}_{\mathbb{B}}(S) \to \mathsf{EMP}_{\mathbb{B}}(T)$ is induced by a possibilistic procedure $S \to T$. We might as well start by assuming that f preserves possibilistic sums as that is a necessary condition. However, while this is an analogue of convexity, there is no obvious way to guarantee that f is determined by its restriction to $\mathcal{E}(X_S)$. This in turn prevents us from from building a possibilistic model e_f that would allow us to recover f. However, possibilistic procedures $S \to T$ readily give rise to possibilistic models on $[S, T]$, so if one can overcome this obstacle one might be able to prove a version of Theorem 44 in the possibilistic case.

Adaptive procedures

The main obstacle for following the same strategy for adaptive procedures is that the analogue Lemma 39 fails in the adaptive case: given a function $F: \mathcal{E}(X_S) \to Y$, there is no clear candidate for the optimal adaptive procedure to implement F. More abstractly, Lemma 39 stems from the fact that each subset $U \subseteq X_S$ defines an equivalence relation $\sim_U$ on $\mathcal{E}(X_S)$ by setting $s \sim_U t$ whenever $s|_U = t|_U$, and such equivalence relations form a sublattice of all equivalence relations on $\mathcal{E}(X_S)$. Similarly, any measurement protocol on X_S induces an equivalence relation on $\mathcal{E}(X_S)$ by setting $s \approx t$ when running the measurement protocol on s and t results in the same outcome. However, there are relatively small examples showing that such equivalence relations no longer form a sublattice of the lattice of all equivalence relations.

This alone does not mean that there cannot be a version of Theorem 44 for adaptive procedures. Indeed, one can easily write down an adaptive version of $[S, T]$, and show that deterministic adaptive procedures correspond exactly to its global assignments that satisfy the predicate checking for simpliciality. Thus, probabilistic and adaptive global sections give rise to non-contextual models over (the adaptive variant of) $[S, T]$ that satisfy the simpliciality predicate, and moreover this adaptive version gives rise to closed structures on the adaptive versions of the categories in Definition 45.

The problem lies in the opposite direction: given a function $f: \mathsf{EMP}(S) \longrightarrow \mathsf{EMP}(T)$ that preserves convex combinations, there seems to be no canonical way to associate a model $e_f: [S, T]$ to it in the adaptive setting. Perhaps one can get around this by nevertheless finding a canonical e_f somehow, or alternatively one might live with this lack of canonicity and work with all possible models over $[S, T]$ that f could correspond to—as long as one of them is induced by an adaptive procedure, then so is f.

Scenarios with structure predicates

If given a function $\mathsf{EMP}(S) \to \mathsf{EMP}(T)$ that preserves convex combinations and also takes models that satisfy g to models that satisfy h, we could readily apply Theorem 44 to understand when such an f arises from a procedure $\langle S, g\rangle \to \langle T, h\rangle$. However, if we are simply given a function $f: \mathsf{EMP}(S, g) \longrightarrow \mathsf{EMP}(T, h)$ that preserves convex combinations, the situation is less clear. Indeed, it could happen that no deterministic model satisfies g and then there is no clear sense in which f is determined by a function $\mathcal{E}(X_S) \to \mathsf{EMP}(T)$. In particular, if g and h are given by strongly contextual models e_g and e_h with minimal supports, then $\mathsf{EMP}(S, g)$ and

$\mathsf{EMP}(T, h)$ are singletons and the unique function $\mathsf{EMP}(S, g) \to \mathsf{EMP}(T, h)$ preserves convex combinations vacuously. Thus characterising which convex functions are induced by procedures $\langle S, g\rangle \to \langle T, h\rangle$ also gives a criterion for deciding if e_g simulates e_h, which in general is a difficult problem.

13.6.3 Monoidal Closure?

Given the closed structure on Scen^g, a natural question is whether it is part of a monoidal closed structure. However, it seems like getting a monoidal structure would require (i) allowing adaptive morphisms (ii) having a "directed" tensor product in the sense that adaptive protocols on $S \otimes T$ should start on S and consume it before moving on to T, never returning back.

To see what suggests this picture, consider first the natural candidate for an evaluation map $[S, T] \otimes S \to T$: to simulate something in T, one first measures the same measurement of $[S, T]$, the answer to which determines what to do in S and how to interpret the outcome. This evaluation procedure is adaptive and directed. Moreover, any directed adaptive procedure $S \otimes T \to V$ seems to correspond to a procedure $S \to [T, V]$—if one just measures the S-half, the rest of the procedure tells what to do in regard to T, resulting in an outcome of $[T, V]$. Conversely, any morphism $S \to [T, V]$ seems to correspond to an adaptive and directed procedure $S \otimes T \to V$. To make this precise one might consider working with scenarios that are equipped with a preorder on measurements, and with adaptive protocols that only go forward along this preorder. Moving to such a setup is tempting not only because of potentially nice categorical properties, but also because it enables one to express other kinds of contextuality that do not quite fit in the Abramsky–Brandenburger framework as-is. Indeed, one of the present authors has elsewhere considered moving to such a framework to discuss *sequential* contextuality, as manifested scenarios in which contexts are sequences of operations rather than sets of compatible one-shot measurements Mansfield and Kashefi (2018), Emeriau et al. (2020), and considered a cruder route to capturing violations Legget-Garg inequalities and macrorealism Leggett and Garg (1985) by adding some extra structure by hand to the framework Mansfield (2017b). A similar approach is developed in Gogioso and Pinzani (2021). However, incorporating morphisms of scenarios in that setting remains to be done.

13.6.4 Towards Duality via (Generalised) Partial Boolean Algebras

Recent work has shown how some of the concepts of the sheaf-theoretic approach to contextuality can be formulated in the language of partial Boolean algebras Abramsky and Barbosa (2021). This provides a dual algebraic-logical picture, in the sense of Stone duality.

One limitation of this result, however, is that it only works for graphical measurement scenarios, i.e. those whose compatibility structure is generated by a binary compatibility relation, so that the simplicial complex of contexts arises as the clique complex of a graph. Extending this to all scenarios will require working with a generalisation of partial Boolean algebras, perhaps as discussed in Czelakowski (1979). Such a structure could be obtained by dualising the sheaf of events $\mathcal{E}$ to a copresheaf of Boolean algebras. There is some hope that this could also be built form the set of procedures $S \to [2]$, but this remains to be checked.

Another observation is that the partial Boolean algebras that correspond to measurement scenarios are in a sense freely built from the Boolean algebras corresponding to each measurement plus a compatibility relation. The theory of partial Boolean algebras is, however, much richer. A speculative suggestion relates to the addition of the predicate g to a scenario S. Its effect, at each context, is to pick out a subset of events. If one thinks of this as a clopen subset of a Stone space, then on the logico-algebraic side it corresponds to a filter (or ideal). One may thus conjecture that the passage from S to $\langle S, g\rangle$ might correspond to the taking quotients of (generalised) partial Boolean algebras.

More generally, one might wonder whether there is a Stone-type duality at work. The scenario [2], which could be identified with the two-element discrete space, could conceivably act as the dualising object on the topological or model side. Note that, suggestively, procedures $S \to [2]$ are naturally endowed with a logico-algebraic structure.

This also suggests a dual question to question (A). Instead of asking which (forward) state transforms arise from procedures, one could inquire which (backward) predicate transforms do.

13.6.5 What Else Can One Do with Predicates?

Changing our basic objects to be pairs $\langle S, g\rangle$ instead of just a scenario S serves the purpose of restricting the kinds of behaviors allowed over an object. As such, it is reminiscent of the definition of basic objects in SProc, a category Samson introduced as a model of concurrency Abramsky et al. Abramsky et al. (1996a, b). Can this superficial similarity be pushed further? In particular, how much further structure does Scen^g or its variants have? Is it e.g.$*$-autonomous?

In Abramsky et al. (2019a) we formalise adaptive procedures via a comonad MP that builds a new scenario $\mathsf{MP}(S)$, the measurements of which correspond to adaptive measurement protocols over S. Similarly, one could start by requiring that procedures have an underlying simplicial function rather than a relation, and then extend the category via a comonad F that takes a scenario S to the scenario $F(S)$ whose measurements are joint measurements in S. Both of these comonads operate well with empirical models. If e is a model on S then there are induced empirical models $\mathsf{MP}(e) : \mathsf{MP}(S)$ and $F(e) : F(S)$. And moreover, these comonads induce comonads on the category of empirical models. However, there is a slight aesthetic issue in that not all empirical models on $\mathsf{MP}(S)$ (or on $F(S)$) are of this form. This is because the bare structure of a measurement scenario does not enforce on all models the intuition that a measurement of the new scenario corresponding to a derived measurement in the original one ought to behave as such. Our expectation is that this could be remedied by equipping $\mathsf{MP}(S)$ and $F(S)$ with structure predicates enforcing this condition. For the adaptive case, the structure predicate should itself be adaptive.

13.6.6 Possibilistic Polytope

Proposition 34 implies that the preorder of possibilistic morphisms $S \to [2]$ is (dually?) equivalent to that of possibilistic models on S. However, different operations seem natural on different sides of this equivalence. It is natural to take a union (i.e. a possibilistic sum) of predicates $g\colon S \longrightarrow [2]$, which restricts the set of models that satisfy it. On the other hand, operation (union/Boolean sum) for possibilistic models increases the set of models that satisfies the corresponding predicate. Thus one might hope that results on one side lead to new results on the other. Given that $\mathsf{EMP}_B(S)$ has been extensively studied in Abramsky et al. (2016a), one may hope either to leverage the results therein to better understand $\mathsf{Scen}_{\mathbb{B}}(S, [2])$, or alternatively, to study $\mathsf{Scen}_{\mathbb{B}}(S, [2])$ in order to answer questions left open therein. In particular, can one understand the image of possibilistic collapse $\mathsf{EMP}(S) \to \mathsf{EMP}_{\mathbb{B}}(S)$ by studying $\mathsf{Scen}_{\mathbb{B}}(S, [2])$?

13.6.7 Other Questions

The fact that procedures $S \to T$ give rise to non-contextual models on $[S, T]$ suggests thinking of contextual models $e : [S, T]$ as "contextual simulations" $S \to T$. The aforementioned evaluation map gives an intuition about how to think of these operationally: given a measurement in T to simulate on S, query the model e on the same measurement—the outcome of this will then indicate what to measure in S and how to interpret it. This seems very closely related to another possible formulation of contextual simulations between empirical models, suggested at the end of Abramsky et al. (2019a): namely, contextual morphisms $d \to e$ could be formalised as adaptive maps $d \otimes c \to e$ without requiring that c be non-contextual. Key differences are that this latter notion is formulated at the level of models rather than at the level of scenarios, and that the adaptivity need not be directed. Despite this, these could still turn out to be similar in terms of expressive power, and may give rise to interesting notions when one still restricts the amount of contextuality allowed, by e.g.requiring that the contextual resources are quantum-realisable.

Another natural question for the computer scientist would be to go beyond single-use black boxes. Here we mean not just allowing for adaptivity but for true multiple use, with an internal state (as in Mealy machines). In the physics side, this would correspond to non-destructive measurement, or measurement with a next state. It might be interesting to extend the formalism to deal with such scenarios.

13.7 Epilogue: A Game of Cat-and-mouse, a Personal Thank-You

A recurrent tip of Samson's is to read Gian-Carlo Rota's *Ten lessons I wish I had been taught* Rota (1997). There, amid an assortment of sage advice, can be found a passage about becoming an institution and the dubious expectations that are bestowed upon one in that event. One could almost look at Samson's career as a game of cat-and-mouse against this prophecy. Rather than settle in to a comfortable and earned existence as 'a piece of period furniture', time after time his relentless scientific curiosity has impelled him to go forth to sow and explore new fields—only to quickly find himself established, there too, as something of an institution.

But plenty has been said of Samson's outstanding scientific contributions, to logic and otherwise, and by others much better qualified than ourselves. So let us take this opportunity to end on a more personal note.

Would-be institutions seem to come in different flavours: there are those that, overbearing, tend to crush and stifle, and there are those that tend to be 'rotten and rotting others', and yet there are also those who help the people around them to flourish and thrive to the best of their abilities. Samson is certainly one of the latter. He has been a central formative influence in our development as (computer?) scientists, and we count ourselves privileged for having worked, and for continuing to work, closely with him. Even as students and fledgling researchers we felt valued and taken seriously, and were afforded the space and encouragement to develop our ideas, with the occasional crucial nudge in the right direction. It may seem little, but it ain't.

We have witnessed up close, on enjoyable afternoons spent exchanging in Samson's office, or over the occasional cheeky pint, his impressive breadth of knowledge, his ability to see further and to spot or establish connections between disparate fields, even across traditional disciplinary boundaries. We have been led to recognise those boundaries not as ontological but as imagined and thus re-imaginable. And in his company we have experienced an environment in which common language can be found and shared among people of very different backgrounds. Testament to this capacity for bringing people together is the present set of authors, including by original academic background a computer scientist, a mathematician, and a physicist.

But alongside his bird-like vision, there is also a frog-like attention to detail Dyson (2009). There are the sudden bursts of activity, when e-mails budding with newly formed ideas are fired afresh every few minutes, when he's engrossed in some intricate detail of work—or swimming with purpose in the milk churn, to borrow a metaphor from one of Samson's favourite quotes found in the preface to Littlewood's miscellany Littlewood (1986). And there are the times when he pins you down on a fine point, forcing you to be precise and organise your own thoughts. Indeed, it is through these sorts of exchanges that the milk is often churned to butter.[17]

While as researchers we may each treasure those moments when, as in Wordsworth's immortal lines about Newton, we find ourselves 'voyaging through strange seas of thought, alone', with Samson one also learns to see science as a conversation. In that spirit we also wish to acknowledge the 'sheaf team' and the many friends and collaborators who have been a part of the conversations and ideas that have nourished the research discussed in this chapter.

We would like to thank Samson and the editors, Mehrnoosh and Alessandra, for giving us the opportunity to contribute a chapter to this volume, and for their patience with us during a tumultuous and drawn-out preparation period. Doing so has been for us a source of great pleasure. We hope, Samson, that you will be find it equally enjoyable to read.

And in the end, so the saying goes, curiosity killed the cat…

Acknowledgements This work was in part carried out while RSB was based at the School of Informatics, University of Edinburgh and while SM was based at the Paris Centre for Quantum Computing, Institut de Recherche en Informatique Fondamentale, University of Paris with financial support from the Bpifrance project RISQ. RSB acknowledges financial support from EPSRC—Engineering and Physical Sciences Research Council, EP/R044759/1, *Combining Viewpoints in Quantum Theory (Ext.)*, and from FCT—Fundação para a Ciência e a Tecnologia, CEECINST/00062/2018.

References

Aasnæss, S. (2020). Cohomology and the algebraic structure of contextuality in measurement based quantum computation. In: B. Coecke & M. Leifer (Eds.), *16th International Conference on Quantum Physics and Logic (QPL 2019)*, *Electronic Proceedings in Theoretical Computer Science* (Vol. 318, pp. 242–253). Open Publishing Association. https://doi.org/10.4204/eptcs.318.15.

Abramsky, S. (2013). Relational hidden variables and non-locality. *Studia Logica, 101*(2), 411–452.

Abramsky, S. (2014). Contextual semantics: From quantum mechanics to logic, databases, constraints, and complexity. *Bulletin of the European Association for Theoretical Computer Science, 113*, 137–163.

Abramsky, S. (2018). Contextuality: At the borders of paradox. In: E. Landry (Ed.), *Categories for the working philosopher* (pp. 262–287). Oxford University Press. https://doi.org/10.1093/oso/9780198748991.003.0011.

Abramsky, S. (2020). Classical logic, classical probability, and quantum mechanics. In *Quantum, Probability, Logic: The Work and Influence of Itamar Pitowsky, Jerusalem Studies in Philosophy and History of Science* (pp. 1–17). Springer International Publishing. https://doi.org/10.1007/978-3-030-34316-3_1.

Abramsky, S. (2022). Notes on presheaf representations of strategies and cohomological refinements of k-consistency and k-equivalence. arXiv:2206.12156 [cs.LO].

[17] Littlewood wrote of his collaboration with Mary Cartwright: *Two rats fell into a can of milk. After swimming for a time one of them realised his hopeless fate and drowned. The other persisted, and at last the milk was turned to butter and he could get out.*

Abramsky, S., & Barbosa, R. S. (2021). The logic of contextuality. In: C. Baier, & J. Goubault-Larrecq (Eds.), *29th EACSL Annual Conference on Computer Science Logic (CSL 2021)*, *Leibniz International Proceedings in Informatics (LIPIcs)* (Vol. 183, pp. 5:1–5:18). Schloss Dagstuhl–Leibniz-Zentrum für Informatik. https://doi.org/10.4230/LIPIcs.CSL.2021.5.

Abramsky, S., Barbosa, R.S., Carù, G., de Silva, N., Kishida, K., & Mansfield, S. (2018). Minimum quantum resources for strong non-locality. In: M. M. Wilde (Ed.), *12th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2017)*, *Leibniz International Proceedings in Informatics (LIPIcs)* (Vol. 73, pp. 9:1–9:20). Schloss Dagstuhl–Leibniz-Zentrum für Informatik. https://doi.org/10.4230/LIPIcs.TQC.2017.9.

Abramsky, S., Barbosa, R. S., Carù, G., & Perdrix, S. (2017). A complete characterization of all-versus-nothing arguments for stabilizer states. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 375*(2106), 20160385.

Abramsky, S., Barbosa, R.S., de Silva, N., & Zapata, O. (2017b). The quantum monad on relational structures. In: K. G. Larsen, H. L. Bodlaender & J. F. Raskin (Eds.), *42nd International Symposium on Mathematical Foundations of Computer Science (MFCS 2017)*, *Leibniz International Proceedings in Informatics (LIPIcs)* (Vol. 83, pp. 35:1–35:19). Schloss Dagstuhl–Leibniz-Zentrum für Informatik. https://doi.org/10.4230/LIPIcs.MFCS.2017.35.

Abramsky, S., Barbosa, R. S., & Mansfield, S. (2017). Contextual fraction as a measure of contextuality. *Physical Review Letters, 119*(5), 050504.

Abramsky, S., Barbosa, R.S., Karvonen, M., & Mansfield, S. (2019a). A comonadic view of simulation and quantum resources. In: *34th Annual ACM/IEEE Symposium on Logic in Computer Science (LiCS 2019)* (pp. 1–12). IEEE. https://doi.org/10.1109/LICS.2019.8785677.

Abramsky, S., Barbosa, R.S., Karvonen, M., & Mansfield, S. (2019b). Simulations of quantum resources and the degrees of contextuality. In: *Early Ideas Talk at 8th Conference on Algebra and Coalgebra in Computer Science (CALCO 2019).*

Abramsky, S., & Carù, G. (2019). Non-locality, contextuality and valuation algebras: A general theory of disagreement. *Philosophical Transactions of the Royal Society A, 377*(2157), 20190036.

Abramsky, S., Barbosa, R. S., Kishida, K., Lal, R., & Mansfield, S. (2015). Contextuality, cohomology and paradox. In: S. Kreutzer (Ed.), *24th EACSL Annual Conference on Computer Science Logic (CSL 2015)*, *Leibniz International Proceedings in Informatics (LIPIcs)* (Vol. 41, pp. 211–228). Schloss Dagstuhl–Leibniz-Zentrum für Informatik. https://doi.org/10.4230/LIPIcs.CSL.2015.211.

Abramsky, S., Barbosa, R. S., Kishida, K., Lal, R., & Mansfield, S. (2016). Possibilities determine the combinatorial structure of probability polytopes. *Journal of Mathematical Psychology, 74*, 58–65.

Abramsky, S., Constantin, C. M., & Ying, S. (2016). Hardy is (almost) everywhere: Nonlocality without inequalities for almost all entangled multipartite states. *Information and Computation, 250*, 3–14.

Abramsky, S., & Brandenburger, A. (2011). The sheaf-theoretic structure of non-locality and contextuality. *New Journal of Physics, 13*(11), 113036.

Abramsky, S., & Brandenburger, A. (2014a). An operational interpretation of negative probabilities and no-signalling models. In: *Horizons of the mind. A tribute to Prakash Panangaden* (pp. 59–75). Springer.

Abramsky, S., Brandenburger, A., & Savochkin, A. (2014b). No-signalling is equivalent to free choice of measurements. In: B. Coecke & M. Hoban (Eds.), *10th International Workshop on Quantum Physics and Logic (QPL 2013)*, *Electronic Proceedings in Theoretical Computer Science* (Vol. 171, pp. 1–9). Open Publishing Association. https://doi.org/10.4204/EPTCS.171.1.

Abramsky, S., & Constantin, C. (2014c). A classification of multipartite states by degree of non-locality. In: B. Coecke & M. Hoban (Eds.), *10th International Workshop on Quantum Physics and Logic (QPL 2013)*, *Electronic Proceedings in Theoretical Computer Science* (Vol. 171, pp. 10–25). Open Publishing Association. https://doi.org/10.4204/EPTCS.171.2.

Abramsky, S., Gay, S., & Nagarajan, R. (1996a). Specification structures and propositions-as-types for concurrency. In: *Logics for Concurrency* (pp. 5–40). Springer.

Abramsky, S., Gay, S.J., & Nagarajan, R. (1996b). Interaction categories and the foundations of typed concurrent programming. In *NATO ASI DPD* (pp. 35–113).

Abramsky, S., Gottlob, G., & Kolaitis, P. G. (2013). Robust constraint satisfaction and local hidden variables in quantum mechanics. In *23rd International Joint Conference on Artificial Intelligence (IJCAI 2013)* (pp. 440–446).

Abramsky, S., & Hardy, L. (2012). *Logical Bell inequalities. Physical Review A, 85*(6), 062114. https://doi.org/10.1103/PhysRevA.85.062114

Abramsky, S., Mansfield, S., & Barbosa, R. S. (2012b). The cohomology of non-locality and contextuality. In: B. Jacobs, P. Selinger & B. Spitters (Eds.), *8th International Workshop on Quantum Physics and Logic (QPL 2011)*, *Electronic Proceedings in Theoretical Computer Science* (Vol. 95, pp. 1–14). https://doi.org/10.4204/EPTCS.95.1.

Abramsky, S., & McCusker, G. (1999). Game semantics. In: *Computational logic* (pp. 1–55). Springer.

Acín, A., Fritz, T., Leverrier, A., & Sainz, A. B. (2015). A combinatorial approach to nonlocality and contextuality. *Communications in Mathematical Physics, 334*(2), 533–628. https://doi.org/10.1007/s00220-014-2260-1

Allcock, J., Brunner, N., Linden, N., Popescu, S., Skrzypczyk, P., & Vértesi, T. (2009). Closed sets of nonlocal correlations. *Physical Review A, 80*, 062107. https://doi.org/10.1103/PhysRevA.80.062107

Amaral, B. (2019). Resource theory of contextuality. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 377*(2157), 20190010. https://doi.org/10.1098/rsta.2019.0010

Amaral, B., Cabello, A., Cunha, M. T., & Aolita, L. (2018). Noncontextual wirings. *Physical Review Letters, 120*(13), 130403. https://doi.org/10.1103/PhysRevLett.120.130403

Aspect, A., Dalibard, J., & Roger, G. (1982). Experimental test of Bell's inequalities using time-varying analyzers. *Physical Review Letters, 49*(25), 1804. https://doi.org/10.1103/PhysRevLett.49.1804

Barbosa, R. S. (2014). On monogamy of non-locality and macroscopic averages: Examples and preliminary results. In: B. Coecke, I. Hasuo & P. Panangaden (Eds.), *11th International Workshop on Quantum Physics and Logic (QPL 2014)*, *Electronic Proceedings in Theoretical Computer Science* (Vol. 172, pp. 36–55). Open Publishing Association. https://doi.org/10.4204/eptcs.172.4.

Barbosa, R. S., Douce, T., Emeriau, P. E., Kashefi, E., & Mansfield, S. (2022). Continuous-variable nonlocality and contextuality. *Communications in Mathematical Physics, 391*(3), 1047–1089. https://doi.org/10.1007/s00220-021-04285-7

Barrett, J., Linden, N., Massar, S., Pironio, S., Popescu, S., & Roberts, D. (2005). Nonlocal correlations as an information-theoretic resource. *Physical Review A, 71*, 022101. https://doi.org/10.1103/PhysRevA.71.022101

Barrett, J., & Pironio, S. (2005). Popescu-Rohrlich correlations as a unit of nonlocality. *Physical Review Letters, 95*(14), 140401.

Bartels, T. (2010). Relative point of view. https://ncatlab.org/nlab/show/relative+point+of+view. Revision 1.

Bell, J. (1964). On the Einstein Podolsky Rosen paradox. *Physics Physique Fizika, 1*(3), 195–200. https://doi.org/10.1103/PhysicsPhysiqueFizika.1.195

Bell, J. S. (1966). On the problem of hidden variables in quantum mechanics. *Reviews of Modern Physics, 38*(3), 447–452. https://doi.org/10.1103/RevModPhys.38.447

Bermejo-Vega, J., Delfosse, N., Browne, D. E., Okay, C., & Raussendorf, R. (2017). Contextuality as a resource for models of quantum computation with qubits. *Physical Review Letters, 119*(12), 120505.

Boole, G. (1862). On the theory of probabilities. *Philosophical Transactions of the Royal Society of London, 152*, 225–252.

Booth, R. I., Chabaud, U., & Emeriau, P. E. (2021). Contextuality and Wigner negativity are equivalent for continuous-variable quantum measurements. arXiv:2111.13218 [quant-ph].

Cabello, A., Severini, S., & Winter, A. (2014). Graph-theoretic approach to quantum correlations. *Physical Review Letters, 112*, 040401. https://doi.org/10.1103/PhysRevLett.112.040401

Campos, Á. D. (1944). Poesias de Álvaro de Campos. Obras completas de Fernando Pessoa (Vol. II). Ática, Lisboa.

Carù, G. (2017). On the cohomology of contextuality. In: R. Duncan & C. Heunen (Eds.) *13th International Conference on Quantum Physics and Logic (QPL 2016)*, *Electronic Proceedings in Theoretical Computer Science* (Vol. 236, pp. 21–39). Open Publishing Association. https://doi.org/10.4204/eptcs.236.2.

Carù, G. (2018). Towards a complete cohomology invariant for non-locality and contextuality. arXiv:1807.04203 [quant-ph].

Chitambar, E., & Gour, G. (2019). Quantum resource theories. *Reviews of Modern Physics, 91*(2), 025001.

Clauser, J. F., Horne, M. A., Shimony, A., & Holt, R. A. (1969). Proposed experiment to test local hidden-variable theories. *Physical Review Letters, 23*(15), 880.

Coecke, B., Fritz, T., & Spekkens, R. W. (2016). A mathematical theory of resources. *Information and Computation, 250*, 59–86.

Czelakowski, J. (1979). Partial Boolean algebras in a broader sense. *Studia Logica, 38*(1), 1–16. https://doi.org/10.1007/bf00493669

de Silva, N. (2018). Logical paradoxes in quantum computation. In: *33th Annual ACM/IEEE Symposium on Logic in Computer Science (LiCS 2018)* (pp. 335–342). IEEE. https://doi.org/10.1145/3209108.32091231.

Donne, J. (1624) *Devotions upon emergent occasions, and severall steps in my sicknes*.

Dupuis, F., Gisin, N., Hasidim, A., Méthot, A. A., & Pilpel, H. (2007). No nonlocal box is universal. *Journal of Mathematical Physics, 48*(8), 082107. https://doi.org/10.1063/1.2767538

Dyson, F. (2009). Birds and frogs. *Notices of the AMS, 56*(2), 212–223.

Dzhafarov, E. N., & Kujala, J. V. (2014). Contextuality is about identity of random variables. *Physica Scripta, 2014*(T163), 014009.

Eilenberg, S., & Kelly, G. M. (1966). Closed categories. In *Proceedings of the Conference on Categorical Algebra* (pp. 421–562). Springer.

Einstein, A., Podolsky, B., & Rosen, N. (1935). Can quantum-mechanical description of physical reality be considered complete? *Physical Review, 47*(10), 777.

Emeriau, P. E., Howard, M., & Mansfield, S. (2020). *Quantum advantage in information retrieval*. arXiv:2007.15643 [quant-ph].

Forster, M., & Wolf, S. (2011). Bipartite units of nonlocality. *Physical Review A, 84*, 042112. https://doi.org/10.1103/PhysRevA.84.042112

Fritz, T. (2017). Resource convertibility and ordered commutative monoids. *Mathematical Structures in Computer Science, 27*(6), 850–938.

Ghirardi, G., Rimini, A., & Weber, T. (1980). A general argument against superluminal transmission through the quantum mechanical measurement process. *Lettere al Nuovo Cimento Series, 2*(1971–1985), *27*(10), 293–298. https://doi.org/10.1007/BF02817189.

Gilchrist, A., & Gilchrist, A. B. (1863). In *Life of William Blake*, "*Pictor ignotus*", *with selections from his poems and other writings* (Vol. II). London: Macmillan and Co.

Giustina, M., Versteegh, M. A., Wengerowsky, S., Handsteiner, J., Hochrainer, A., Phelan, K., Steinlechner, F., Kofler, J., Larsson, J. Å., Abellán, C., et al. (2015). Significant-loophole-free test of Bell's theorem with entangled photons. *Physical Review Letters, 115*(25), 250401.

Gogioso, S., & Pinzani, N. (2021). The sheaf-theoretic structure of definite causality. arXiv:2103.13771 [quant-ph].

Gogioso, S., & Zeng, W. (2019). Generalised Mermin-type non-locality arguments. *Logical Methods in Computer Science, 15*(2), 3:1—3:51. https://doi.org/10.23638/LMCS-15(2:3)2019.

Grudka, A., Horodecki, K., Horodecki, M., Horodecki, P., Horodecki, R., Joshi, P., Kłobus, W., & Wójcik, A. (2014). Quantifying contextuality. *Physical Review Letters, 112*(12), 120401.

Hardy, L., & Spekkens, R. (2010). Why physics needs quantum foundations. *Physics in Canada, 66*(2), 73–76.

Hensen, B., Bernien, H., Dréau, A.E., Reiserer, A., Kalb, N., Blok, M.S., Ruitenberg, J., Vermeulen, R.F., Schouten, R.N., Abellán, C., et al. (2015). Loophole-free Bell inequality violation using electron spins separated by 1.3 kilometres. *Nature, 526*(7575), 682–686.

Horodecki, M., Oppenheim, J.: (Quantumness in the context of) resource theories. *International Journal of Modern Physics B, 27*(01n03), 1345019.

Howard, M., Wallman, J., Veitch, V., & Emerson, J. (2014). Contextuality supplies the 'magic' for quantum computation. *Nature, 510*(7505), 351.

Johnstone, P.T.: *Sketches of an elephant: A topos theory compendium* (Vol. 2). Oxford University Press.

Jones, N. S., & Masanes, L. (2005). Interconversion of nonlocal correlations. *Physical Review A, 72*, 052312. https://doi.org/10.1103/PhysRevA.72.052312

Karanjai, A., Wallman, J. J., & Bartlett, S. D. (2018). *Contextuality bounds the efficiency of classical simulation of quantum processes*. arXiv:1802.07744 [quant-ph].

Karvonen, M. (2018). Categories of empirical models. In P. Selinger & G. Chiribella (Eds.), *15th international conference on quantum physics and logic (QPL 2018), electronic proceedings in theoretical computer science* (Vol. 287, pp. 239–252). https://doi.org/10.4204/EPTCS.287.14.

Karvonen, M. (2021). Neither contextuality nor nonlocality admits catalysts. *Physical Review Letters, 127*(16), 160402. https://doi.org/10.1103/PhysRevLett.127.160402

Kirchmair, G., Zähringer, F., Gerritsma, R., Kleinmann, M., Gühne, O., Cabello, A., Blatt, R., & Roos, C. F. (2009). State-independent experimental test of quantum contextuality. *Nature, 460*(7254), 494–497.

Kishida, K. (2014). Stochastic relational presheaves and dynamic logic for contextuality. In: B. Coecke, I. Hasuo & P. Panangaden (Eds.), *11th International Workshop on Quantum Physics and Logic (QPL 2014)*, *Electronic Proceedings in Theoretical Computer Science* (Vol. 172, pp. 115–132). Open Publishing Association. https://doi.org/10.4204/eptcs.172.9.

Kishida, K. (2016). Logic of local inference for contextuality in quantum physics and beyond. In: I. Chatzigiannakis, M. Mitzenmacher, Y. Rabani & D. Sangiorgi (Eds.), *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, *Leibniz International Proceedings in Informatics (LIPIcs)* (Vol. 55, pp. 113:1–113:14). Schloss Dagstuhl–Leibniz-Zentrum für Informatik . https://doi.org/10.4230/LIPIcs.ICALP.2016.113.

Kochen, S., & Specker, E. P. (1967). The problem of hidden variables in quantum mechanics. *Journal of Mathematics and Mechanics, 17*(1), 59–87.

Laplaza, M. L. (1977). Embedding of closed categories into monoidal closed categories. *Transactions of the American Mathematical Society, 233*, 85. https://doi.org/10.1090/s0002-9947-1977-0480686-8

Leggett, A. J., & Garg, A. (1985). Quantum mechanics versus macroscopic realism: Is the flux there when nobody looks? *Physical Review Letters, 54*(9), 857.

Liang, Y. C., Spekkens, R. W., & Wiseman, H. M. (2011). Specker's parable of the overprotective seer: A road to contextuality, nonlocality and complementarity. *Physics Reports, 506*(1–2), 1–39.

Littlewood, J. E. (1986). In B. Bollobás (Ed.), *Littlewood's miscellany*. Cambridge University Press.

Mansfield, S. (2017). Consequences and applications of the completeness of Hardy's nonlocality. *Physical Review A, 95*(2), 022122. https://doi.org/10.1103/physreva.95.022122

Mansfield, S. (2017b). A unified approach to contextuality and violations of macrorealism. In *Talk at 1st Workshop on Quantum Contextuality in Quantum Mechanics and Beyond (QCQMB 2017)*, Prague, Czech Republic.

Mansfield, S., & Barbosa, R. S. (2014). Extendability in the sheaf-theoretic approach: Construction of Bell models from Kochen–Specker models. arXiv:1402.4827 [quant-ph].

Mansfield, S., & Fritz, T. (2012). Hardy's non-locality paradox and possibilistic conditions for non-locality. *Foundations of Physics, 42*(5), 709–719. https://doi.org/10.1007/s10701-012-9640-1

Mansfield, S., & Kashefi, E. (2018). Quantum advantage from sequential-transformation contextuality. *Physical Review Letters, 121*(23), 230401.

Manzyuk, O. (2012). Closed categories versus closed multicategories. *Theory and Applications of Categories, 26*(5), 132–175.

Mermin, N. D. (1985). Is the moon there when nobody looks? Reality and the quantum theory. *Physics Today, 38*(4), 38–47.

Pais, A. (1979). Einstein and the quantum theory. *Reviews of Modern Physics, 51*(4), 863.

Pitowsky, I. (1994). George Boole's "conditions of possible experience" and the quantum puzzle. *The British Journal for the Philosophy of Science, 45*(1), 95–125.

Popescu, S., & Rohrlich, D. (1994). Quantum nonlocality as an axiom. *Foundations of Physics, 24*(3), 379–385.

Raussendorf, R. (2013). Contextuality in measurement-based quantum computation. *Physical Review A, 88*(2), 022322.

Rota, G. C. (1997). Ten lessons I wish I had been taught. In *Indiscrete thoughts* (pp. 195–203). Springer.

Schrödinger, E. (1935). Die gegenwärtige Situation in der Quantenmechanik. *Naturwissenschaften, 23*(49), 823–828.

Shalm, L. K., Meyer-Scott, E., Christensen, B. G., Bierhorst, P., Wayne, M. A., Stevens, M. J., Gerrits, T., Glancy, S., Hamel, D. R., Allman, M. S., et al. (2015). Strong loophole-free test of local realism. *Physical Review Letters, 115*(25), 250402.

Street, R. (1974). Elementary cosmoi I. In *Category seminar*. Lecture notes in mathematics (Vol. 420, pp. 134–180). Berlin, Heidelberg: Springer. https://doi.org/10.1007/bfb0063103.

Wang, D., Sadrzadeh, M., Abramsky, S., & Cervantes, V. H. (2021). On the quantum-like contextuality of ambiguous phrases. arXiv:2107.14589 [cs.CL].

Chapter 14
Gödel, Escher, Bell: Contextual Semantics of Logical Paradoxes

Kohei Kishida

Abstract Quantum physics exhibits various non-classical and paradoxical features. Among them are non-locality and contextuality (e.g. Bell's theorem or the Einstein-Podolsky-Rosen paradox). Since they are expected to constitute a key resource in quantum computation, several approaches have been proposed to provide high-level expressions for them. In one of these approaches, Abramsky and others use the mathematics of algebraic topology and characterize non-locality and contextuality as the same type of phenomena as M. C. Escher's impossible figures. This article expands this topological insight and demonstrates that logical paradoxes arising from circular references (of the sort formalized by Gödel's fixed-point or diagonalization lemma) share the same topological structure as the quantum paradoxes, by reformatting the topological model of contextuality into a semantics of logical paradoxes. This topological semantics indeed provides a unifying perspective from which previous approaches of philosophers and logicians to logical paradoxes can be understood as diverse ways of fine-tuning topologies to model paradoxes.

Keywords Quantum physics · Contextuality · Logical paradox · Circularity · Topology · Presheaf · Categorical semantics

14.1 Introduction

In quantum physics, when experimenters are interested in a set of observables, they typically cannot measure them all but have to choose a subset of measurements to make, and it is within this context of measurements that they observe outcomes. In each such context the observed outcomes may obey some constraint, but when we combine these constraints across all the contexts, they can be altogether inconsistent, as if nature exhibits inconsistency. This phenomenon, called *contextuality* [Kochen and Specker (1967)], is a distinctly non-classical property of quantum physics. It turns

K. Kishida (✉)
Department of Philosophy, University of Illinois, Urbana-Champaign, U.S.A.
e-mail: kkishida@illinois.edu

A. Palmigiano and M. Sadrzadeh (eds.), *Samson Abramsky on Logic and Structure in Computer Science and Beyond*, Outstanding Contributions to Logic 25,
https://doi.org/10.1007/978-3-031-24117-8_14

out [Abramsky and Brandenburger (2011)] to be a generalization of Bell non-locality [Bell (1964)]—another non-classical, paradoxical feature of quantum physics—and, based on promising results [Raussendorf (2013); Howard et al. (2014)], it is expected to provide computational resources for quantum computation and information.

One approach to formally modelling the paradoxical phenomenon of contextuality is to characterize it using global sections of presheaves, an insight that originated in Isham and Butterfield (1998): a certain type of contextuality of a quantum system was shown to consist in the absence of global sections in presheaves modelling the observational behaviors of the system. The "sheaf-theoretic" approach led by Abramsky [Abramsky and Brandenburger (2011); Abramsky et al. (2012, 2015, 2017), just to list a few] expands this insight by abstracting away from the concrete formalism of quantum physics: regarding contextuality more generally as a matter of *topology* in data of measurements and outcomes, the approach characterizes contextuality as "locally consistent but globally inconsistent". This high-level approach has, on the one hand, classified types and degrees of contextuality and provided various qualitative and quantitative tools to detect and measure contextuality. One of these tools is algebraic topology: cohomology captures many cases of contextuality in literature [Abramsky et al. (2012, 2015)], in the same way Penrose (1992) showed it to capture the impossibility in many impossible figures by M. C. Escher and himself. [For other tools see, e.g., Abramsky et al. (2018) or (2017)]. On the other hand, the high-level perspective has shown that phenomena of the same topological structure as quantum contextuality can be found in various other fields such as constraint satisfaction [see Abramsky et al. (2013)] and relational database theory [see Abramsky (2013a, b)].

One phenomenon that is shown isomorphic to quantum contextuality in Abramsky et al. (2015, 2017) is logical paradoxes of the "liar" type. Consider the following sentence:

(λ) $\boxed{\text{The sentence in the box on p. 1 is not true.}}$

Let's call this sentence λ. It is, by definition, true iff it is not, since the subject of ". . . is not true" in λ refers to λ itself. Now consider whether λ is true or not. If λ is true, it implies that λ is not true, contradicting the assumption. Therefore λ is not true, but from this it follows that λ is indeed true. Thus a contradiction has been derived, with no assumption. This is the *liar paradox*. It should be emphasized that the upshot is not quite that λ is a contradiction. There is nothing paradoxical about contradictory sentences such as "Snow is white and not white", which you can consistently call a contradiction. By contrast, should you deem λ a contradiction, it commits you to the falsity of λ, which then commits you to the truth of λ as well. In other words, the paradox is not that some sentences say inconsistent things (though λ may well); it is that the inference on the truth of λ makes *you* inconsistent!

It may be tenable to ascribe inconsistency to the definition of λ (as opposed to λ itself)—but then the paradox is that we are apparently forced to accept the inconsistent definition (and to thereby become inconsistent ourselves) as soon as it is put forward, due to the self-reference in λ. In fact, exactly the same type of self-referential definitions can be implemented in arithmetic by Gödel's (1931) method of constructing fixed points of predicates. As observed by Tarski (1923) [see also Murawski (1998)], the truth predicate "… is true" cannot be defined in arithmetic in such a way as to define λ, but natural languages apparently admit both the Gödel-style circular reference and the truth predicate, and philosophers and philosophical logicians have been debating how best to reconcile these features as they result in various paradoxical phenomena.

In some cases of such logical paradoxes, circular definitions or constraints make it impossible to consistently assign truth values to sentences; let us refer roughly to these paradoxes as the "liar type". Tarski's theorem states that arithmetic would face this type of paradox if it had the truth predicate. In other cases, there exist consistent assignments of truth values, but circular constraints either rule out or admit assignments in non-standard fashions. The phenomena called Curry's paradox and the "truth-teller" are of these types. (Although Gödel's incompleteness theorems concern the provability predicate rather than the truth predicate, one may also include them as a case of a circular way of ruling out assignments.)

Both the quantum paradoxes of contextuality and the logical paradoxes of the liar type present us with implementable situations (e.g. an experiment and a sentence on a page) that make us inconsistent by means of conventional logic. What Abramsky et al. have shown is that the two sorts of paradoxes share exactly the same structure of inconsistency, and that to provide them with consistent models we need to grasp the topology that underlies this structure. The overall objective of this article is to show that this topological insight is very much relevant to philosophers' and philosophical logicians' discussion of logical paradoxes. The claim we will try to establish is *not* that Abramsky et al. offer a more accurate or correct account than those by philosophers and philosophical logicians as to how to understand paradoxes philosophically (e.g. which truth value we should consider a given paradoxical sentence to have or not have). Instead, we will demonstrate that the topological approach offers a *unifying* framework in which various ideas and proposals by philosophers and logicians can be classified and compared in structural terms.

Most of those philosophers and logicians provide semantic formalisms that deal with full propositional or first-order logic in which a broader range of logical paradoxes arise. By contrast, certainly, Abramsky et al. (2015, 2017) have only dealt explicitly with a subfamily of paradoxes of the liar type in the aforementioned sense, and have not given a fuller account of how paradoxes can arise from Boolean connectives and other vocabulary of logic. Nevertheless, as we will show in this article, methods of categorical logic enable us to reformulate the full standard semantics of propositional and first-order logics in the form of presheaves, in which the standard maps of denotation are exactly the global sections, the compositionality of denotation is a topological property similar to the sheaf condition, and logical paradoxes boil down to contextuality and a breakdown of compositionality. Indeed, given this topological formulation of semantics, we can regard the philosophers and logicians as proposing and debating what topology in these presheaves would provide the best consistent model for logical paradoxes.

The remaining sections of this article are structured as follows. Section 14.2 reviews how the sheaf approach models contextuality using presheaves, covering basic definitions and explaining how they work conceptually. Section 14.3 explains how certain contextual models are isomorphic to the liar-type paradoxes, and reviews examples of logical paradoxes that arise from a broader range of circular constraints. Following these examples, Sect. 14.4 extends topological models to semantics of Boolean propositional logic in general, rewriting the standard semantics in the format of presheaves. Section 14.5 then introduces topological properties of such presheaves that are relevant to paradoxes, viz. local consistency, contextuality, and compositionality; it turns out that the standard semantics without logical paradoxes give non-contextual and compositional presheaves. Section 14.6 briefly reviews how these ideas carry over naturally to first-order logic. Having established a topological, presheaf formulation of semantics for logic, in Sect. 14.7 we introduce circular constraints to the semantics, and investigate their relationship to contextuality and breakdown in compositionality. Then Sect. 14.8 demonstrates that various paradoxes can be characterized in topological terms using presheaves. Section 14.9 briefly reviews how several ideas that philosophers and logicians have proposed as to how to model logical paradoxes can be seen as providing particular implementations within our topological framework. Section 14.10 concludes the article.

14.2 Background: Topological Models of Contextuality

We first review the sheaf-theoretic approach to contextuality [Abramsky and Brandenburger (2011), Abramsky et al. (2015)], which shows contextuality to be topological in nature. While the approach encompasses models for a probabilistic notion of contextuality, we focus the review on models for a logical (i.e. possibilistic or non-probabilistic) notion, which will be relevant to the purpose of this article. We also use the formalism of Kishida (2016)—which is slightly different from but equivalent to the one in Abramsky and Brandenburger (2011), Abramsky et al. (2015)—to emphasize the simplicial and topological aspect of the models.

14.2.1 Presheaf Models

In their full generality, models of contextuality in the sheaf approach concern a general notion of variables and values. A model involves a set X of variables and, for each $x \in X$, a set A_x of possible values of x—in other words, it involves an X-indexed family of sets A_x. This setup can model physical (maybe quantum, though not necessarily) measurements:

- We measure properties $x \in X$ of a physical system and it gives back outcomes $a \in A_x$.

But the formalism can also model various other settings in which we make queries against a system and it answers, such as relational database as observed in Abramsky (2013a, b):

- A relational database has attributes $x \in X$, and $a \in A_x$ are possible data values for x.

Note that we often make a query regarding several variables in combination. A set $U \subseteq X$ of variables the query concerns then forms a *context* in which the system gives back a result. Contexts play essential rôles in the following two kinds of constraints, (a) on queries and (b) on answers.

(a) We have the family $C \subseteq \mathcal{P}X$ of contexts in which queries can be made and answered. We may not be able to make a query in a context $V \subseteq X$ (i.e. $V \notin C$) for reasons such as:

  - Quantum mechanics may deem it impossible to measure all the properties in V at once.
  - A database schema may have no table encompassing all the attributes in V.

As is the case in these examples, we assume that if queries can be made in a context U they can be in any $V \subseteq U$. we also assume that queries can be made in $\{x\}$ for any $x \in X$. When we further assume that all $U \in C$ are finite, all this amounts to assuming that C is an (abstract) *simplicial complex* on X, i.e. a $\subseteq$-downward closed family of finite subsets of X with $\bigcup_{U \in C} U = X$.

(b) When we make a query in a context U, the system returns (one or a set of) tuples $s \in \prod_{x \in U} A_x$ of values. It then has the subset $A_U \subseteq \prod_{x \in U} A_x$ of "admissible" tuples that can be part of query results, and it is often the information on A_U that we want. E.g.,

  - We may measure a physical system in various states and find that some set U of quantities always satisfies a certain equation that characterizes A_U.
  - From a relational database we retrieve data with an attribute list U, and the database returns the relation A_U on sets A_x $(x \in U)$ as a table.

We call such a tuple $s \in \prod_{x \in U} A_x$ a *(local) section* over U. We may refer to s as "$(s(x), s(y), \dots)$ over $U = (x, y, \dots)$". Now let $V \subseteq U \in C$, which implies that we can also make a query in $V \in C$. We assume that—as is the case with the examples above—if a tuple s of values over U is admissible and in A_U, then so is its restriction to V, i.e. $s|_V \in A_V$. Thus, the projection

$$-|_V : \prod_{x \in U} A_x \to \prod_{x \in V} A_x :: s \mapsto s|_V$$

restricts to $A_{V \subseteq U} : A_U \to A_V$. This makes $A : C^{\mathrm{op}} \to \mathbf{Sets}$ a presheaf on the poset C, and indeed a "separated" presheaf.

As was shown in Kishida (2016), the separated presheaves over a simplicial complex C are categorically equivalent to the "non-degenerate" bundles of simplicial

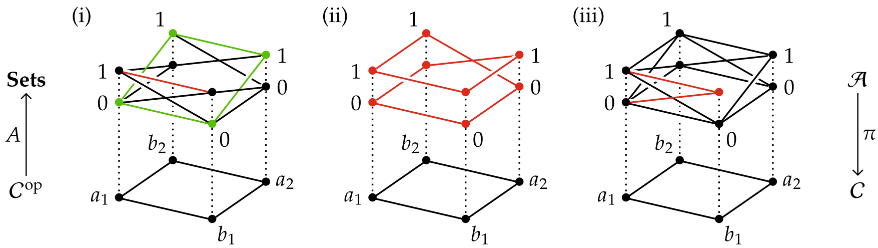


Fig. 14.1 Bundles for (i) the Hardy model and (ii) the PR-box

complexes $\pi : \mathcal{A} \to C$ (where "non-degenerate" means preserving the dimension of every simplex). We can therefore represent presheaf models of contextuality with diagrams of bundles. The diagrams in Fig. 14.1, for instance, exhibit "Bell-type" scenarios in which Alice and Bob measure properties of a system, perhaps a quantum one. The base C expresses constraints of type (a): Alice can make at most one of two measurements a_1 and a_2 at a time, so she chooses one; similarly Bob chooses from b_1 and b_2—the four contexts (a_i, b_j) are indicated by the four edges of C. On the other hand, $\mathcal{A}$ expresses constraints of type (b): while the vertices indicate that each measurement $x \in X = \{a_1, a_2, b_1, b_2\}$ has two possible outcomes 0 and 1, some combinations of them are never obtained, and the edges indicate the possible combinations.

The topological understanding enables us to read (a) and (b) as follows: Each context $U \in C$ is a local, small enough region of the space X of variables. The topology on the space $\mathcal{A}$ of values, sitting over C, then distinguishes those tuples $s \in \prod_{x \in U} A_x$ in A_U from the others and deems the former to be continuous sections.

14.2.2 Contextuality, Topologically

The central notion in the sheaf approach to contextuality is that of *global section*.

Definition 1 Given a separated presheaf $A : C^{\text{op}} \to \mathbf{Sets}$ over a simplicial complex C on X, a *global section* is a tuple $g : \prod_{x \in X} A_x$ over the entire X such that $g|_U \in A_U$ for all $U \in C$. Or, in terms of a simplicial bundle $\pi : \mathcal{A} \to C$, a global section is a simplicial map $g : C \to \mathcal{A}$ such that $\pi \circ g = 1_C$.

$$\begin{tikzcd} C \arrow[r, "g"] \arrow[dr, equal] \arrow[r, phantom, shift right=3ex, "="] & \mathcal{A} \arrow[d, "\pi"] \\ & C \end{tikzcd}$$

A global section is an assignment of values to all the variables that satisfies every constraint on combinations of values. In the physical setting, it may seem natural to think of global sections g as states of the observed system, assigning values to all the observables: although we can only make queries locally in contexts $U \in C$ and although answers to queries can only be observed locally in those contexts, one might assume that the system in a state g actually has a value $g(x)$ assigned to every quantity x, and that the answer we receive in the context U is simply $g|_U$; so, one might think, we can simply make multiple queries in multiple contexts U_i to cover $X = \bigcup_i U_i$ and to recover the global information. This assumption, that any section we observe is part of a context-independent global section, holds in classical physics—but breaks down in quantum physics, precisely when contextuality arises.

The models in Fig. 14.1 all violate the classical assumption above, and are examples of

Definition 2 A separated presheaf model is said to be *logically contextual* if not all of its local sections extend to global ones, and *strongly contextual* if it has no global section at all.

(i) of Fig. 14.1 represents an example of logical contextuality due to Hardy (1993) that is realizable in quantum physics. It has several global sections, e.g. the one marked in green; call it g. So, when Alice and Bob measure (a_1, b_1) and observe $(0, 0)$, the classical explanation is possible that the system was in the state g and had outcomes $g(x)$ assigned to all the measurements $x \in X$, and that Alice and Bob have simply retrieved that information on U. On the other hand, the local section in red, $(1, 1)$ over (a_1, b_1), does not extend to any global section. This means that the classical explanation is simply impossible for this joint outcome. Indeed, all the red sections in Fig. 14.1 fail to extend to a global section. So the classical explanation is never possible in (ii), all its sections in red, making it strongly contextual. This model, called the *Popescu-Rohrlich box* [Popescu and Rohrlich (1994)], or the *PR box* for short, is not quantum-realizable, but it plays an important rôle in the quantum information literature. In fact, quantum physics exhibits many other instances of strong contextuality [e.g. Greenberger et al. (1989), Mermin (1990)].

The upshot is that contextuality consists in the combination of *global inconsistency and local consistency*: a section $s \in A_U$ is consistent locally, in the sense of satisfying the constraint on query results in the context U, but it may be inconsistent globally, in the sense of contradicting all the other constraints and thereby failing to extend to a global section. The general definition of contextuality in terms of global sections can also be applied to relational databases, in which contextuality then corresponds precisely to the absence of a universal relation [Abramsky (2013a, b)], and the set of global sections of a presheaf A is exactly the natural join $\bowtie_{U \in C} A_U$.

14.2.3 No-Signalling Principle

The sense of local consistency that we have used (explicitly) in the presentation so far is that a local section exists although it may not extend to a global one. In the sheaf approach of Abramsky et al., however, presheaf models is usually assumed to satisfy a stronger sense of local consistency—viz., the condition that is called the *no-signalling* principle in the physical setting Ghirardi et al. [(1980)].

Definition 3 We say that a separated presheaf $A : \mathcal{C}^{\mathrm{op}} \to \mathbf{Sets}$ is no-signalling, or locally consistent, if every $A_{U \subseteq V} : A_V \to A_U :: s \mapsto s|_U$ is a surjection, i.e., if $A : \mathcal{C}^{\mathrm{op}} \to \mathbf{Surj}$ for the category **Surj** of sets and surjections.

An example violating this condition is (iii) of Fig. 14.1: $A_{\{b_1\} \subseteq \{a_2, b_1\}} : A_{\{a_2, b_1\}} \to A_{\{b_1\}}$ is not surjective. Reading it in the physical setting, suppose that Alice and Bob make measurements, that Bob chooses to measure b_1, and that he observes 1, which is not in the image of $A_{\{b_1\} \subseteq \{a_2, b_1\}}$. This means that Bob has received the signal from Alice (no matter how far away she may be!) that she has chosen a_1 and not a_2. On the other hand, let us read (iii) in the database setting as well: it has tables $A_{\{a_1, b_1\}}$ and $A_{\{a_2, b_1\}}$, but, when queried about the attribute b_1, the two tables yield different results of projection, disagreeing on whether 1 is in or not—a disagreement or inconsistency that can be witnessed locally. This is how no-signalling also means local consistency [see Abramsky (2013a, b)].

One may note that the following holds, because a section s over U can extend to a global section g only if $U \subseteq V \in \mathcal{C}$ implies $s = A_{U \subseteq V}(g|_V)$.

Fact 1 In any separated presheaf, no section that witnesses violation of no-signalling can extend to a global one. Therefore every locally *in*consistent presheaf is contextual, and every non-contextual presheaf is locally consistent.

14.3 Logical Paradoxes and Contextuality

The principal idea of Abramsky et al. (2015, 2017) on logical paradoxes is that the logical paradoxes of the liar type can be characterized by the failure to obtain global sections, just the same way quantum paradoxes of contextuality can.

14.3.1 From Quantum to Logical Paradoxes

Besides physical measurements and relational databases, presheaf models of contextuality can be applied to constraint satisfaction [Abramsky et al. (2013)].

- $x \in X$ are Boolean variables and valuations assign to them Boolean values $a \in A_x = \mathbf{2}$.
- Each constraint involves a finite set $U \in \mathcal{C}$ of variables.
- Each such context $U \in \mathcal{C}$ has the set A_U of combinations of values of $x \in U$ that satisfy all the constraints on U.

Expanding this application, it is observed in Abramsky et al. (2015) that the strong contextuality of the PR box, (ii) of Fig. 14.1, is topologically isomorphic to logical paradoxes of the liar type.

Imagine a cloister with four corners named c_0, c_1, c_2, and c_3. In each corner c_i, a notice carrying a sentence σ_i is put up, which reads as follows.

- The sentence σ_i on c_i reads "The sentence on c_{i+1} is true", for $i < 3$.
- The sentence σ_3 on c_3 reads "The sentence on c_0 is false."

Presented with these sentences, we are apparently forced to accept their definition so that, e.g., σ_0 is true iff σ_1 is. Although perhaps not strictly *self*-referential as in the liar sentence λ in the Introduction, this definition is circular nonetheless. It then seems to make us inconsistent just as in the liar paradox: assuming σ_0 to be true leads to a contradiction but the falsity of σ_0 also leads to a contradiction, as in

$$\begin{array}{l} \sigma_0 = 1 \xRightarrow{\text{def of } \sigma_0} \sigma_1 = 1 \xRightarrow{\text{def of } \sigma_1} \sigma_2 = 1 \xRightarrow{\text{def of } \sigma_2} \sigma_3 = 1 \xRightarrow{\text{def of } \sigma_3} \sigma_0 = 0 \neq 1 \\ \sigma_0 = 0 \xRightarrow{\text{def of } \sigma_0} \sigma_1 = 0 \xRightarrow{\text{def of } \sigma_1} \sigma_2 = 0 \xRightarrow{\text{def of } \sigma_2} \sigma_3 = 0 \xRightarrow{\text{def of } \sigma_3} \sigma_0 = 1 \neq 0 \end{array} \tag{1}$$

The observation of Abramsky et al. 2015 is that this logical paradox of the cloister is isomorphic to the strong contextuality of the PR box, (ii) of Fig. 14.1. The four constraints due to the definitions of $\sigma_0, \ldots, \sigma_3$ have the same structure as the constraints represented by the edges over the four edges of C in (ii). The inference in (1) then exactly parallels the impossible attempt to find a global section in (ii). The strong contextuality of (ii) thus expresses *our* inconsistency, to which we can contrast the joint inconsistency of $\sigma_0, \ldots, \sigma_3$: the impossibility for us to consistently evaluate $\sigma_0, \ldots, \sigma_3$ amounts to the absence of global section, whereas the joint inconsistency of $\sigma_0, \ldots, \sigma_3$ would merely mean that the tuple $(1, 1, 1, 1)$ is ruled out.

The topological nature of the paradox can also be highlighted as follows. Observe that, although the example above involves four sentences, we can define a paradoxical circular definition with any number n of sentences $\sigma_0, \ldots, \sigma_{n-1}$:

- The sentence σ_i reads "The sentence σ_{i+1} is true", for $i < n - 1$.
- The sentence σ_{n-1} reads "The sentence σ_0 is false."

These "liar cycles" can be modelled using the presheaf models similar to (ii), when $n \geqslant 3$. Such models are not available to the cases of $n = 1$ (i.e. the liar paradox, $\sigma = \lambda$, with self-reference) and $n = 2$, since at least three vertices are needed to represent a circle with a simplicial complex. Nevertheless, abstracting away from the simplicial formalism, the underlying topology would clearly capture these cases, as Fig. 14.2 shows.

14.3.2 No-Signalling and Local Inference

One of the essential features of local consistency in the use of presheaf models as semantic models of logic is that, as observed in Kishida (2016), it supports what is called "local inference" [Kishida (2016)]. Let us regard (iii) as expressing a (say, "epistemic") state where we know the constraint that the values of (a_2, b_1) must be either $(1, 0)$ or $(0, 0)$ but no other constraints. Once we assume local consistency (that the "true" model is no-signalling), we can infer from the first constraint that the value of b_1 must be 0, and then furthermore that values of (a_1, b_1) must be either $(1, 0)$ or $(0, 0)$. This inference removes the red sections from (iii)—because their presence violates no-signalling—and thereby carves out a locally consistent subpresheaf. Let us observe that there are two directions of inference involved here:

- From a small context to a larger one: a constraint (the absence of certain sections) over b_1 entails another over (a_1, b_1). This is due to the projection $A_{\{b_1\}\subseteq\{a_1,b_1\}} : A_{(a_1,b_1)} \to A_{b_1}$ being a function, with each input having an output.
- From a larger context to a smaller one: a constraint over (a_2, b_1) entails another over b_1. This is due to no-signalling, i.e., the the projection $A_{\{b_1\}\subseteq\{a_2,b_1\}}$ being surjective, with each output having an input.

Then we can transport information locally across different contexts, from a smaller context to a larger one, to another smaller one, to another larger one, and on and on.

We should observe that the logic of such local inference [called "inchworm logic" in Kishida (2016)] is properly weaker than standard logic. It is convenient to first note

Fact 2 Any (sub)set G of global sections of a separated presheaf $A : C^{\mathrm{op}} \to \mathbf{Sets}$ forms a non-contextual subpresheaf $\overline{G} :: U \mapsto \{g|_U \mid g \in G\}$ of A. In particular, the subpresheaf formed by the set of all global sections of A is the largest non-contextual subpresheaf of A, which we call the "non-contextual interior" of A and for which we write $\mathrm{NC}(A)$. A separated presheaf A is non-contextual if and only if $\mathrm{NC}(A) = A$.

Now consider the locally consistent but logically contextual (i) of Fig. 14.1, which expresses three constraints: in Boolean formulas,

$$\neg(a_1 \wedge b_2), \quad \neg(a_2 \wedge b_1), \quad a_2 \vee b_2.$$

From these, standard logic derives $\neg(a_1 \wedge b_1)$ and rules out the red section, whereas local inference does not. This is precisely because the red section does not extend to a global one: it is part of the locally consistent model A, but not of the non-contextual $\mathrm{NC}(A)$. Or, the four equations describing the locally consistent but strongly contextual (ii),

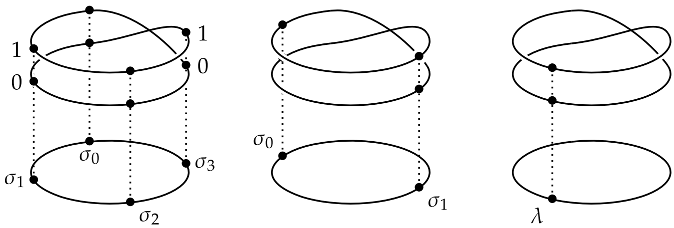


Fig. 14.2 The "liar cycles" of lengths 4, 2, and 1

$$a_1 = b_1, \quad a_1 = b_2, \quad a_2 = b_1, \quad a_2 \neq b_2,$$

are contradictory in standard logic, so that $\mathrm{NC}(A) = \varnothing$ has no global sections, but the equations do have a locally consistent model, viz. (ii). Thus, standard logic is unsound (i.e. too strong) with respect to locally consistent models. Indeed, we can characterize standard logic as the logic of global sections and non-contextual models, and the logic of local inference as the logic of locally consistent models. (See Kishida (2016) for more detail, theorems, and proofs.)

Given the idea of carving out a locally consistent (but perhaps contextual) sub-presheaf from a locally inconsistent presheaf, it is useful to formalize it with

Fact 3 The family of locally consistent presheaves is closed under arbitrary union. Therefore any separated presheaf A has its largest locally consistent subpresheaf, which we call the "locally consistent interior" of A and for which we write $\mathrm{LC}(A)$.

Fact 4 Any separated presheaf A has $\mathrm{NC}(A) \subseteq \mathrm{LC}(A)$ by Fact 1.

These facts as well as Fact 1 lead to the following way to look at contextuality. As Fact 3 states, even if we are first given a locally inconsistent presheaf A, we can always uniquely obtain its locally consistent interior $\mathrm{LC}(A)$. Therefore, although the locally inconsistent A is immediately contextual by Fact 1, that part of contextuality can be removed by taking $\mathrm{LC}(A)$. A more interesting question, then, is whether $\mathrm{LC}(A)$ remains contextual. Hence we introduce the following distinction.

Definition 4 Given a locally inconsistent presheaf A, we call it "genuinely contextual" if $\mathrm{LC}(A)$ is contextual and "trivially contextual" otherwise.

(This distinction is similar to the one made in the contextuality-by-default approach to contextuality [Dzhafarov and Kujala (2016, 2017)]: given a model exhibiting probabilistic contextuality, the approach provides a quantitative measure for the degrees to which the model violates no-signalling and to which its contextuality is due to the violation.)

14.3.3 Circular Definitions with Connectives

As reviewed in Sect. 14.3.1, the logical paradoxes of the liar type can be characterized as isomorphic to contextuality, i.e. the failure to obtain global sections. Besides the liar-type paradoxes, Boolean connectives also give circular constraints from which a broader class of paradoxical phenomena arise. Let us consider examples of such paradoxical phenomena.

For instance, in Curry's paradox, we define

- The sentence γ reads "If γ is true then φ,"

and this definition seems to force both γ and φ to be true. In the "truth-teller",

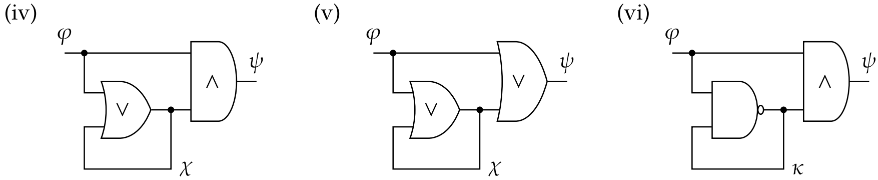


Fig. 14.3 Cyclic logic circuits

- The sentence τ reads "The sentence τ is true,"

no fact whatsoever can seem to determine the value of τ.

Or, to take a family of examples that may be less paradoxical but practically more significant, cyclic logic circuits can be taken as introducing circular definitions [e.g. Kautz (1970), Malik (1994), Halbwachs and Maraninchi (1995), one may also see Berry (2000)]. For instance, (iv) of Fig. 14.3 is one of the circuits in Malik (1994), and expresses the following definitions:

- The sentence ψ reads "Both φ and χ are true."
- The sentence χ reads "At least one of φ and χ is true."

If φ is true, then χ is made true and makes ψ true. On the other hand, if φ is false, then χ behaves the same way the truth-teller τ does. Or, in (v), if φ is false, then the value of ψ is entirely dependent on that of the truth-teller-like χ. Furthermore, in (vi), we can also have what may be called a "controlled liar":

- The sentence κ reads "At most one of κ and φ is true."

This sentence is true if φ is false, but brings about the liar paradox if φ is true. This seems to force φ to be false, similarly to Curry's paradox.

The list of such examples goes on and on. Furthermore, quantifiers also give rise to another class of paradoxical phenomena. One of our goals is to show that these paradoxes, arising from Boolean connectives and quantifiers, are also topological in nature, characterized in terms of global sections.

14.4 Semantics of Logic in the Presheaf Form: The Propositional Case

This article attempts to show a broad range of logical paradoxes to be topological in nature, in a manner expanding the insight of Abramsky et al. (2015, 2017) that the liar-type paradoxes can be characterized in terms of the failure to obtain global sections. To prepare ourselves for this endeavour, the next three sections extend topological models to a setting that can provide general semantics for logic, and show that the standard semantics, from which logical paradoxes are ruled out, can be treated as a presheaf model that exhibits no contextuality. Although this section

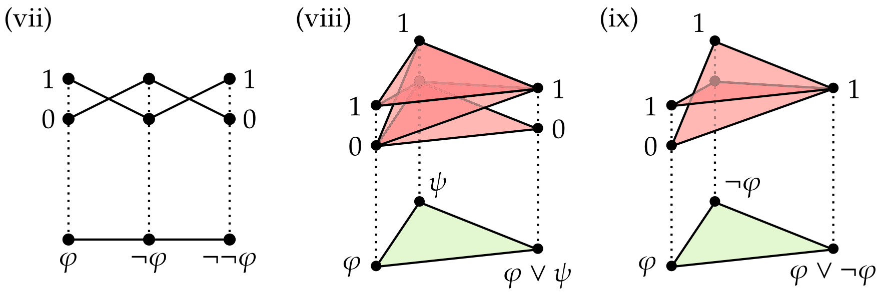


Fig. 14.4 Constraints for Boolean connectives

and Sect. 14.5 focus the explanation and exposition on propositional logic, the ideas can be extended further to quantifiers, as will be laid out briefly in Sect. 14.6.

14.4.1 The Principal Idea

The objective of this section is to reformulate the standard semantics in terms of a presheaf. Before getting to formalism, it may be helpful to briefly explain roughly how our idea works (even though the explanation is merely an approximation to what will come later).

Extending the idea of Sect. 14.3.1—which is to take a presheaf of Boolean values over a space of Boolean variables—we introduce a space of Boolean formulas with connectives and take a presheaf of Boolean values. The base space is formally implemented as something similar to a simplicial complex: we can think of it as having edges $(\varphi, \neg\varphi)$, triangles $(\varphi, \psi, \varphi \vee \psi)$, etc., over which edges, triangles, etc. express the constraints these formulas obey, as shown in Fig. 14.4. For instance, as in (vii), there are two edges over $(\varphi, \neg\varphi)$, viz. $(1, 0)$ and $(0, 1)$. As in (viii), there are four triangles over $(\varphi, \psi, \varphi \vee \psi)$, viz. $(1, 1, 1)$, $(1, 0, 1)$, $(0, 1, 1)$, and $(0, 0, 0)$.

On the other hand, over $(\varphi, \neg\varphi, \varphi \vee \neg\varphi)$ in (ix), we would like to make sure that there are only two triangles, viz. $(1, 0, 1)$ and $(0, 1, 1)$, so that $\varphi \vee \neg\varphi$ must be true, i.e., 1 is its only value. This is where the strong sense of local consistency is supposed to do the job: Because there are (or we need to make sure there are) only two edges $(1, 0)$ and $(0, 1)$ over the edge $(\varphi, \neg\varphi)$, the small-to-large part of local consistency should rule out the triangles $(1, 1, 1)$ and $(0, 0, 0)$ over $(\varphi, \neg\varphi, \varphi \vee \neg\varphi)$. Then the large-to-small part should rule out 0 above $\varphi \vee \neg\varphi$.

14.4.2 An Algebra of Formulas

Although we have just laid out our idea in terms of edges and triangles, we choose not to strictly use a simplicial complex as the base space of a new presheaf model. There are several reasons for this choice: One is that we need to consider, e.g., a triple $(\varphi, \varphi, \varphi \wedge \varphi)$ but that it cannot be a triangle in a simplicial complex. Another is that the Boolean constraints are functions, with the values over e.g. (φ, ψ) as input and those over $\varphi \wedge \psi$ as output, and that it is useful to formally equip the triangle $(\varphi, \psi, \varphi \wedge \psi)$ with this input-output direction. So, instead of a simplicial complex, we use a category, and more precisely an algebra of a Lawvere theory, to represent the space of formulas. The idea is similar to the one in Walters Walters (1989a, b), which uses multigraphs and the free finite-product categories on them to represent context-free grammars and languages.

Definition 5 A *Lawvere theory* is a finite-product category whose objects are all n-fold products $*^n$ of a single object $*$. An *algebra* over a Lawvere theory C, or C-algebra, is a product-preserving functor $A : C \to \mathbf{Sets}$. A homomorphism of C-algebras is a natural transformation.

Let L be the set of formulas generated from a set of propositional variables with Boolean connectives. It is an algebra equipped with Boolean structure, i.e. functions such as $\neg : L \to L$ and $\wedge : L \times L \to L$. ($L$ is however not a Boolean algebra, since $\varphi \wedge \psi$ and $\psi \wedge \varphi$ are two distinct formulas.) This fact can be expressed using the following pair of definitions.

Definition 6 Let $\underline{\Sigma}$ be a multigraph with a single vertex $*$ and with Boolean connectives as edges: $\Sigma(*; *) = \{\neg\}$, $\Sigma(*, *; *) = \{\wedge, \vee, \Rightarrow\}$, and $\Sigma(\ ; *) = \{\top, \bot\}$. We define the Lawvere theory of Boolean structure to be the free finite-product category Σ on $\underline{\Sigma}$.

Typical arrows of Σ are (tuples of) compositions of connectives possibly with duplicated arguments, or in other words, formula schemas, e.g., $(-_1 \wedge -_2) \Rightarrow -_1 : * \times * \to *$. The constants $\top$ and $\bot$ are arrows from the terminal 1 to $*$. Now enter

Definition 7 By an "algebra of Boolean formulas" we mean a free Σ-algebra L on some set.

L_* is then the set of Boolean formulas generated from a set of propositional variables. Arrow components of L apply formula schemas to formulas; e.g., for the arrow $(-_1 \wedge -_2) \Rightarrow -_1$ of Σ,

$$L((-_1 \wedge -_2) \Rightarrow -_1) : L_* \times L_* \to L_* :: (\varphi, \psi) \mapsto (\varphi \wedge \psi) \Rightarrow \varphi.$$

The constants $\top$ and $\bot$ are both elements of L_* and arrows from the terminal $1 = (L_*)^0 = \{()\}$.

$$\top : 1 \to L_* :: () \mapsto \top, \qquad \bot : 1 \to L_* :: () \mapsto \bot.$$

14.4.3 A Presheaf of Semantics

To implement the idea explained above and illustrated in Fig. 14.4, we need to take a presheaf over the set $\mathcal{L}$ of Boolean formulas, which requires us to reformulate the Σ-algebra L as a category. This can be done by

Definition 8 Given a functor $A : C \to \mathbf{Sets}$, its *category of elements*, $\int_C A$ or $\mathcal{A}$, is a category defined as follows.

- Its objects are pairs (x, a) of an object x of C and an element a of A_x. When we can assume that different A_x are all disjoint, we may simply write a instead of (x, a).
- $\int A((x, a), (y, b))$ consists of arrows $f : x \to y$ of C such that $A(f)(a) = b$.

The obvious functor $\pi_A : \mathcal{A} \to C :: (x, a) \mapsto x$, sending each $f : (x, a) \to (y, b)$ to $f : x \to y$ itself, is called the *projection* of $\mathcal{A}$.

If C is a Lawvere theory, then objects of the category of elements $\mathcal{A}$ are tuples of elements of A_*, and $\mathcal{A}$ has finite products given simply by tuples. The projection π_A preserves finite products, and moreover sends elements of A_* to $*$.

Definition 9 By a "category of Boolean formulas" we mean the category of elements $\mathcal{L}$ of an algebra of Boolean formulas L.

Objects U of $\mathcal{L}$ are tuples of Boolean formulas. One type of arrows of $\mathcal{L}$ is given by instances of applying connectives and formula schemas to formulas; e.g., $\wedge_{\varphi,\psi} : (\varphi, \psi) \to \varphi \wedge \psi$ and

$$((-_1 \wedge -_2) \Rightarrow -_1)_{\varphi,\psi} : (\varphi, \psi) \to (\varphi \wedge \psi) \Rightarrow \varphi.$$

This type of arrows provides each formula φ with its parse tree abstractly, in the form of arrows with codomain φ. Not all arrows of $\mathcal{L}$ are of this type, however—more about this in Sect. 14.5.3.

Note that φ and (φ, φ) are two distinct objects of $\mathcal{L}$, and that there are two distinct arrows $\wedge_{\varphi,\varphi} : (\varphi, \varphi) \to \varphi \wedge \varphi$ and $\wedge_{\varphi,\varphi} \circ \Delta_\varphi : \varphi \to \varphi \wedge \varphi$, where $\Delta_\varphi = \langle 1_\varphi, 1_\varphi \rangle : \varphi \to (\varphi, \varphi)$ is the diagonal map of the product (φ, φ). Given this distinction, we introduce this terminology:

- While we refer to an object U of $\mathcal{L}$ as a tuple, we call it a set when it contains no repetition of the same formula. Our notation $U \subseteq V$ does *not* assume U or V to be a set in this sense (unless explicitly noted).

Now consider the following Σ-algebra as well, which specifies which Boolean function interprets a given Boolean connective.

Definition 10 We define the Σ-algebra of Boolean values to be the Σ-algebra $B : \Sigma \to \mathbf{Sets} :: *^n \mapsto \mathbf{2}^n$ that sends, e.g., $\wedge : *^2 \to *$ to $\wedge : \mathbf{2} \times \mathbf{2} \to \mathbf{2}$.

B sends $\top, \bot : *^0 \to *$ to $1 : \mathbf{2}^0 \to \mathbf{2} :: * \to 1$ and $0 : \mathbf{2}^0 \to \mathbf{2} :: * \to 0$. Note that, although $\mathbf{2}$ is a Boolean algebra, B is not a functor to the category of Boolean algebras, since $\wedge : \mathbf{2} \times \mathbf{2} \to \mathbf{2}$, etc., are not Boolean homomorphisms. Then we obtain a (covariant) presheaf $F : \mathcal{L} \to \mathbf{Sets}$ of Boolean values and constraints over $\mathcal{L}$ as follows.

Definition 11 We define the "full presheaf of Boolean values" F for a category of Boolean formulas $\mathcal{L}$ to be the composition $B \circ \pi_L$ of the Σ-algebra of Boolean values B after the projection π_L of $\mathcal{L}$.

$$\begin{array}{rcccl} F : \mathcal{L} & \xrightarrow{\quad \pi_L \quad} & \Sigma & \xrightarrow{\quad B \quad} & \mathbf{Sets} \\ \varphi & \longmapsto & * & \longmapsto & \mathbf{2} \end{array}$$

We write $\mathcal{F}$ and $\pi_F : \mathcal{F} \to \mathcal{L}$ for the category of elements of F and its projection.

This presheaf achieves, e.g., (vii) and (viii) of Fig. 14.4: For (vii), $F_\varphi = F_{\neg\varphi} = \mathbf{2}$, and $F(\neg_\varphi : \varphi \to \neg\varphi) = \neg : \mathbf{2} \to \mathbf{2}$ is a function sending 1 to 0 and 0 to 1; or, regarded as a binary relation on $\mathbf{2}$,

$$F(\neg_\varphi : \varphi \to \neg\varphi) = \neg = \{(1, 0), (0, 1)\} \subseteq \mathbf{2} \times \mathbf{2}.$$

For (viii), $F_\varphi = F_\psi = F_{\varphi\vee\psi} = \mathbf{2}$, and $F(\vee_{\varphi,\psi} : (\varphi, \psi) \to \varphi \vee \psi) = \vee : \mathbf{2} \times \mathbf{2} \to \mathbf{2}$ equals

$$F(\vee_{\varphi,\psi} : (\varphi, \psi) \to \varphi \vee \psi) = \vee = \{(1, 1, 1), (1, 0, 1), (0, 1, 1), (0, 0, 0)\} \subseteq \mathbf{2} \times \mathbf{2} \times \mathbf{2}.$$

14.4.4 Global Sections Are the Standard Semantics

Sections 14.4.2 and 14.4.3 showed how to take a presheaf of Boolean values on a category of Boolean formulas. This style of semantics for Boolean logic may appear peculiar and significantly different from the standard style. It is, however, just a rewriting of the same idea as is used in ordinary semantics. Indeed, maps assigning denotations in these semantics amount precisely to global sections of a presheaf in our style—so that our style of semantics provides a framework that subsumes standard semantics as non-contextual subcases but that is broader and in which, as we will later show, logical paradoxes arise as contextuality.

Generally speaking, in algebraic and categorical semantics, denotations of formulas are given by functors, natural transformations, and other kinds of homomorphisms or structure-preserving maps. In fact, this involves a combination of two types of maps: (I) one assigning a denotation to each formula, and (II) one determining, for each type of formula, a type of denotations it may take. The presheaf F defined in Sect. 14.4.3 is in fact a map of type (II). Then the crucial observation we give here is that the maps of type (I) are exactly the global sections of a map of type (II).

Let us take Boolean logic and its semantics as an example. In its usual presentation, the standard semantics for Boolean logic goes like this: Let L_* be the set of formulas generated from a set PV of propositional variables with Boolean connectives. This makes L_* a "Boolean prealgebra", i.e., an algebra that is equivalent to a Boolean algebra but that may fail antisymmetry (since $\varphi \wedge \psi$ and $\psi \wedge \varphi$ are different formulas). Then denotations $[\![\varphi]\!]$ of formulas φ are given by a function $[\![-]\!] : L_* \to \mathbf{2}$ that is a Boolean homomorphism—i.e. that preserves the Boolean structure, e.g. $[\![\varphi \wedge \psi]\!] = [\![\varphi]\!] \wedge [\![\psi]\!]$. This $[\![-]\!]$ is a map of type (I). From the perspective of Lawvere theories, it is a homomorphism of Σ-algebras, from L of Definition 9 to B of Definition 10.

$$
\begin{tikzcd}
\mathbf{Sets} \\
\Sigma \arrow[u, bend left=60, "L"] \arrow[u, bend right=60, "B"'] \arrow[u, phantom, "\overset{[\![-]\!]}{\Longrightarrow}"]
\end{tikzcd}
\qquad
\begin{tikzcd}
L_* \arrow[rr, "{[\![-]\!]}"] & & \mathbf{2} \\
& * \arrow[ul, mapsto, "L"] \arrow[ur, mapsto, "B"'] &
\end{tikzcd}
$$

On the one hand, a global section of the presheaf $F = B \circ \pi_L : \mathcal{L} \to \mathbf{Sets} :: \varphi \mapsto \mathbf{2}$ is a dependent function $g : \prod_{U \in \mathcal{L}_0} F_U$ such that $F(f)(g(U_1)) = g(U_2)$ for every arrow $f : U_1 \to U_2$ of $\mathcal{L}$; in other words, it is a functor g between categories of elements making

$$
\begin{tikzcd}
\mathcal{L} \arrow[rr, "g"] \arrow[dr, equal] \arrow[drr, phantom, "="] & & \mathcal{F} \arrow[dl, "\pi_F"] \\
& \mathcal{L} &
\end{tikzcd}
$$

commute. Comparing these conditions to the ones above for algebras demonstrates that the denotation maps $[\![-]\!]$ are exactly the global sections g.

To summarize, we have different expressions for the denotation maps $[\![-]\!]$ and the global sections g via the isomorphisms

$$\mathbf{BpA}(\mathcal{L}, \mathbf{2}) \cong \Sigma\text{-}\mathbf{Alg}(L, B) \cong \mathbf{Cat}/\mathcal{L}(1, \pi_F),$$

where **BpA** is the category of Boolean prealgebras".

14.5 Topological Properties of the Standard Semantics

In the previous section we reformulated the standard semantics of propositional logic in terms of presheaves and global sections. This enables us to show that the standard semantics satisfies such topological properties as local consistency and non-contextuality. It also has the advantage that a category of Boolean formulas can be modified in ways that are not available in the algebraic formulation. Such modifications enable us, for instance, to capture compositionality in the standard semantics in topological terms.

14.5.1 Local Consistency

Our presheaf F provides a presheaf format for the standard semantics, but it is not locally consistent in the strong sense explained in Sect. 14.2.3. This is illustrated by the discussion of Fig. 14.4 in Sect. 14.4.1: although F implements the part of the idea there with respect to (vii) and (viii) of Fig. 14.4, F does not by itself implement the other part of the idea, i.e., local consistency in the sense described with respect to (ix). To extend the sheaf approach to contextuality by Abramsky et al. to semantics of logic, we need to discuss what local consistency means in terms of F and moreover in terms of its subpresheaves.

The discussion of (ix) shows that local consistency enables us to make such inferences as the following (2) and (3), locally without invoking global sections (i.e., without considering assignments of Boolean values to all the formulas):

(2) Since it must be the case that one of φ and $\neg\varphi$ is true, $\varphi \vee \neg\varphi$ must be true.
(3) If $\varphi \wedge \psi$ is valid then so is φ.

F does not accommodate these inferences directly, since it maps all the formulas to $\mathbf{2}$. This is why we need to consider subpresheaves of F.

The explanation of how local consistency supports (2) in (ix) suggests that we can define local consistency of F, or any subpresheaf A of F, as all its arrow components being surjections (with each input having an output and each output having an input). This does work, but in a slightly tricky way, since surjections fail to be "multicomposable" in the sense that, even if functions $f_i : X \to Y_i$ $(1 \leqslant i \leqslant n)$ and $g : Y_1 \times \cdots \times Y_n \to Z$ are all surjections, their composition $g \circ \langle f_1, \ldots, f_n \rangle$ may not be. To illustrate the same point with an example, take a subpresheaf A of F and consider:

(4) To the triangle $\vee_{\varphi,\neg\varphi} : (\varphi, \neg\varphi) \to \varphi \vee \neg\varphi$ at the base of (ix), A assigns a function $A(\vee_{\varphi,\neg\varphi}) : A_{(\varphi,\neg\varphi)} \to A_{\varphi\vee\neg\varphi}$. If the domain of this function is the cartesian product $A_\varphi \times A_{\neg\varphi}$ (which is the case with F), it contains not just $(1, 0)$, $(0, 1)$ but also $(1, 1)$ and $(0, 0)$ (assuming $A_\varphi = A_{\neg\varphi} = \mathbf{2}$); then $A(\vee_{\varphi,\neg\varphi})$ contains $(0, 0, 0)$, and therefore the assumption that $A(\vee_{\varphi,\neg\varphi})$ is a surjection could not rule out 0 from $A_{\varphi\vee\neg\varphi}$.

The moral to this example is that we may want inputs in a cartesian product $A_{\varphi_1} \times \cdots \times A_{\varphi_n}$ to obey some constraints, in which case the domain $A_{(\varphi_1,\ldots,\varphi_n)}$ of $A(f : (\varphi_1, \ldots, \varphi_n) \to \psi)$ should not be the full cartesian product but a proper subset. Therefore enter

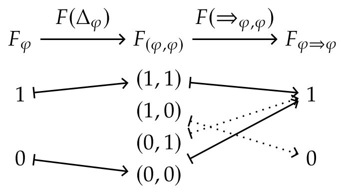
Definition 12 Given a category of Boolean formulas $\mathcal{L}$ with projection $\pi : \mathcal{L} \to \Sigma$, we define a "presheaf of Boolean values" for $\mathcal{L}$ to be any subpresheaf $A : \mathcal{L} \to \mathbf{Sets}$

of the full presheaf of Boolean values $F = B \circ \pi : \mathcal{L} \to \mathbf{Sets}$ for $\mathcal{L}$. We do *not* assume A to preserve products, so that $A_{\varphi_1,\ldots,\varphi_n}$ can be a proper subset of $A_{\varphi_1} \times \cdots \times A_{\varphi_n}$.

Definition 13 Let $A : \mathcal{L} \to \mathbf{Sets}$ be a presheaf of Boolean values for $\mathcal{L}$. Then we say that A is locally consistent if $A : \mathcal{L} \to \mathbf{Surj}$.

Note that this is a straightforward extension of Definition 3. Examples of consequences of local consistency of A are

(5) $A_\top \subseteq \{1\}$, because $F(\top : () \to \top) : \mathbf{2}^0 \to \mathbf{2} :: * \mapsto 1$.
(6) $A_{(\varphi,\varphi)}$ is isomorphic to A_φ, since $A(\Delta_\varphi) : A_\varphi \to A_{(\varphi,\varphi)} :: b \mapsto (b, b)$ has codomain $A_{(\varphi,\varphi)} \subseteq \{(1, 1), (0, 0)\} \subseteq A_\varphi \times A_\varphi$. It follows that the surjection $A(\Rightarrow_{\varphi,\varphi}) : A_{(\varphi,\varphi)} \to A_{\varphi\Rightarrow\varphi}$ has codomain $A_{\varphi\Rightarrow\varphi} = \{1\}$ (assuming $A_\varphi \neq \varnothing$), so that $\varphi \Rightarrow \varphi$ must be true. The following diagram sums this up, where the dotted edges indicate what local consistency rules out.



And here is how local consistency of A accommodates (2) and (3):

(7) Continuing (4), the domain $A_{(\varphi,\neg\varphi)}$ of $A(\vee_{\varphi,\neg\varphi})$ equals $\{(1, 0), (1, 0)\}$, because $A(\langle 1_\varphi, \neg_\varphi\rangle) : A_\varphi \to A_{(\varphi,\neg\varphi)} :: b \mapsto (b, \neg b)$ is a surjection. Hence the surjection $A(\vee_{\varphi,\neg\varphi})$ has codomain $A_{\varphi\vee\neg\varphi} = \{1\}$.
(8) Suppose $A_{\varphi\wedge\psi} = \{1\}$. Then the domain $A_{(\varphi,\psi)}$ of $A(\wedge_{\varphi,\psi}) \subseteq F(\wedge_{\varphi,\psi})$ equals $\{(1, 1)\}$, since $F(\wedge_{\varphi,\psi})$ sends $(1, 0)$, $(0, 1)$, $(0, 0)$ to 0. This implies $A_\varphi = \{1\}$ because, for the projection $p_1 : (\varphi, \psi) \to \varphi$ in $\mathcal{L}$, $A(p_1) : A_{(\varphi,\psi)} \to A_\varphi :: (b_1, b_2) \mapsto b_1$ is a surjection (since $A_{(\varphi,\psi)} \neq \varnothing$).

In this way, locally consistent presheaves of Boolean values capture local inferences on constraints on and among formulas along their parse trees.

It is useful to introduce the following notions and make some observations about them.

Definition 14 Given a presheaf of Boolean values A for $\mathcal{L}$, let us write $\mathrm{dom}(A)$ for the full subcategory of $\mathcal{L}$ of objects $U \in \mathcal{L}_0$ such that $A_U \neq \varnothing$. We call A “total” if $\mathrm{dom}(A) = \mathcal{L}_0$, and “empty” if $\mathrm{dom}(A) = \varnothing$.

Fact 5 If a presheaf of Boolean values A is locally consistent, it is either total or empty.

Definition 15 Let A be a presheaf of Boolean values. Given any assignment $\alpha : T \to \mathbf{2}$ of values to a (perhaps infinite) subset $T \subseteq L_*$ of formulas, let us say that α is consistent with A, or "A-consistent", if, for every finite $U \subseteq T$ and any arrow $f : V \to U$ of $\mathcal{L}$, there is a section $s \in A_V$ such that $A(f)(s) = \alpha|_U$.

Fact 6 A presheaf of Boolean values A is locally consistent iff every assignment $\alpha : T \to \mathbf{2}$ that has $\alpha|_U \in A_U$ for all finite $U \subseteq T$ is consistent with A.

Fact 7 Suppose an assignment $\alpha : T \to \mathbf{2}$ is consistent with a presheaf of Boolean values A. Then $A(f)(\alpha|_U) = \alpha|_V$ for every arrow $f : U \to V$ of $\mathcal{L}$ such that $U, V \subseteq T$.

Proof Take $h = \langle 1_U, f \rangle : U \to (U, V)$. Then the local consistency of α with A implies $A(h)(s) = \alpha_{(U,V)}$ for some $s \in A_U$. But then using the projections p_1, p_2 from (U, V) to U and V we have $s = A(1_U)(s) = A(p_1 \circ h)(s) = (\alpha|_{(U,V)})|_U = \alpha|_U$ and $A(f)(\alpha|_U) = A(p_2 \circ h)(s) = (\alpha|_{(U,V)})|_V = \alpha|_V$. □

14.5.2 *The Standard Semantics Is Non-contextual*

Now that we have defined the concepts of global sections and local consistency in the current setting, we can introduce the same notion of contextuality here as for the simplicial models from Sect. 14.2. An interesting question, then, is whether the Boolean-valued presheaves exhibit any contextuality apart from the trivial one due to local inconsistency. In other words, can a formula have a value that is consistent with constraints along parse trees but that cannot be accounted for by global value assignments?

Let us first observe that the definition of a global section of F extends straightforwardly to its subpresheaves:

Definition 16 A global section of a presheaf of Boolean values A is a global section of the full presheaf F that lands in A (i.e. a dependent function $g : \prod_{U \in \mathcal{L}_0} A_U$ such that $A(f)(g(U)) = g(V)$ for every arrow $f : U \to V$ of $\mathcal{L}$); or equivalently, it is a functor g between categories of elements making the left triangle or (equivalently) the composed one commute in the following diagram.

$$\begin{tikzcd}
\mathcal{L} \arrow[r, "g"] \arrow[dr, equal] & \mathcal{A} \arrow[r, hook] \arrow[d, "\pi_A"'] & \mathcal{F} \arrow[dl, "\pi_F"] \\
& \mathcal{L} &
\end{tikzcd}$$

We can also use Fact 7 to reformulate a homomorphism $[\![-]\!] : L_* \to \mathbf{2}$ so that, while objects of $\mathcal{L}$ are in general tuples of formulas, the behavior of a global section g is entirely determined by its assignment $g|_{L_*}$ of values to formulas in particular:

Fact 8 Given a presheaf of Boolean values A, every A-consistent global assignment $\alpha : L_* \to \mathbf{2}$ extends to a unique global section $\bar{\alpha} : \mathcal{L} \to \mathcal{A}$ of A by $\bar{\alpha}(U) = \alpha|_U$, while each global section g of A is the extension of the assignment $g|_{L_*}$ (which is A-consistent).

Now we can extend the concept of contextuality from Definition 2 of Sect. 14.2.2 to

Definition 17 We say that a presheaf of Boolean values A for $\mathcal{L}$ is non-contextual if for each object U of $\mathcal{L}$ and every $s \in A_U$ there is a global section g such that $g(U) = s$.

Facts 1, 2, 3, and 4 for the simplicial case of Sects. 14.2 and 14.3 carry over to the current case straightforwardly:

Fact 9 We have the following facts:

- Any presheaf of Boolean values has its non-contextual interior $\mathrm{NC}(A)$.
- The family of locally consistent presheaves of Boolean values is closed under arbitrary union. Therefore any presheaf of Boolean values A has its locally consistent interior $\mathrm{LC}(A)$.
- Every non-contextual presheaf of Boolean values is locally consistent. Therefore $\mathrm{NC}(A) \subseteq \mathrm{LC}(A)$ for any presheaf of of Boolean values A.

These facts do not depend much on properties of $\mathcal{L}$. By contrast, the following result is dependent on properties of $\mathcal{L}$ that make F an expression of the standard semantics.

Fact 10 The full presheaf of Boolean values F has a non-contextual $\mathrm{LC}(F)$.

In other words, F is not genuinely contextual, i.e., that it exhibits no contextuality apart from the trivial one due to local inconsistency. More generally, any locally consistent subpresheaf of F is non-contextual, as long as it defines a value for every formula.

Fact 11 If a presheaf of Boolean values A is total then so are $\mathrm{NC}(A)$ and $\mathrm{LC}(A)$.

Fact 12 If a presheaf of Boolean values A is total and locally consistent then A is not contextual.

These are immediate consequences of the following theorem, which roughly states that any combination of values consistent with A extends to a global section of A (as long as A gives values to every formula).

Theorem 1 *Let A be a total presheaf of Boolean values. Then every A-consistent assignment $\alpha : T \to \mathbf{2}$ extends to a global section of A.*

To prove this, it is useful to introduce the following notation based on the *compositionality* of denotation: i.e., given any compound formula, its value is determined by those of the atomic formulas in it.

Notation 1 The parse tree of a formula φ is given by the arrow $\mathrm{par}_\varphi : \mathrm{pv}(\varphi) \to \varphi$ of $\mathcal{L}$ from the set $\mathrm{pv}(\varphi)$ of propositional variables that occur in φ, and the value of φ is a function $F(\mathrm{par}_\varphi)(s)$ of a local valuation $s : \mathrm{pv}(\varphi) \to \mathbf{2}$ or $s \in F_{\mathrm{pv}(\varphi)}$. More generally, given a tuple $U = (\varphi_1, \ldots, \varphi_n)$ of formulas, $\mathrm{pv}(U)$ is the set of propositional variables that occur in at least one φ_i, and a tuple of parse trees par_{φ_i} (after suitable projections p_i) gives $\mathrm{par}_U = \langle \mathrm{par}_{\varphi_i} \circ p_i \rangle_{1 \leqslant i \leqslant n} : \mathrm{pv}(U) \to U$.

Then

Proof for Theorem 1. Given a total presheaf A of Boolean values and an A-consistent assignment $\alpha : T \to \mathbf{2}$, define

$$\mathbb{T} = T_{\alpha,1} \cup T_{A,1} \cup \neg[T_{\alpha,0}] \cup \neg[T_{A,0}]$$

for

$$T_{\alpha,b} = \{\, \varphi \in T \mid \alpha(\varphi) = b \,\}, \qquad T_{A,b} = \{\, \varphi \in L_* \mid A_\varphi = \{b\} \,\}.$$

Fix any finite subset U of $\mathbb{T}$. We want to show that it is modelled by a global valuation (so that the compactness theorem will apply). To do so, define the following finite subset of T.

$$V = \{\, \varphi \in T_{\alpha,1} \mid \varphi \in U \,\} \cup \{\, \psi \in T_{\alpha,0} \mid \neg\psi \in U \,\}.$$

Then, for the projection $p : \mathrm{pv}(U) \to \mathrm{pv}(V)$ followed by $\mathrm{par}_V : \mathrm{pv}(V) \to V$, the local consistency of α with A gives a section $s \in A_{\mathrm{pv}(U)}$ such that $A(\mathrm{par}_V \circ p)(s) = \alpha|_V$. This s models U. Thus every finite subset of $\mathbb{T}$ is modelled by a local valuation. Therefore, by compactness, $\mathbb{T}$ is modelled by a global valuation $v : \mathrm{PV} \to \mathbf{2}$. Its extension $\bar{v}$ to $\mathcal{L}$ is a global section of A extending α. $\square$

14.5.3 Compositionality

Our presheaves of Boolean values also enable us to characterize the compositionality of the semantics in topological terms, with a condition similar to the sheaf condition, once we introduce a suitable topology on $\mathcal{L}$ or a slight modification thereof.

Compositionality in the semantics can be understood as the phenomenon that the value of each formula being determined by the values of the propositional variables in it—or more generally a set of subformulas that contains enough of the subformulas. Taking presheaves over V is a direct expression of this idea: Certain arrows in $\mathcal{L}$ can be seen as abstractly parsing formulas in to subformulas. E.g., $f = ((-_1 \wedge -_2) \Rightarrow -_1)_{\varphi,\psi} : (\varphi, \psi) \to \chi$, where $\chi = (\varphi \wedge \psi) \Rightarrow \varphi$, parses χ into φ and ψ. The function $A(f) : A_{(\varphi,\psi)} \to A_\chi$ then encapsulates how the value of χ is determined by the values of φ and ψ along the parse tree.

Compositionality can also be understood as a "local" version of the following Fact 13: The definition of $\mathcal{L}$ (or L), as a free Σ-algebra generated on a set PV of propositional variables, entails the first isomorphism in

$$\mathbf{Sets}(\mathrm{PV}, \mathbf{2}) \cong \Sigma\text{-}\mathbf{Alg}(L, B) \cong \mathbf{Cat}/\mathcal{L}(1, \pi_F).$$

More generally, for any subpresheaf A of F,

Fact 13 Given a presheaf of Boolean values A for $\mathcal{L}$, each valuation $v : \mathrm{PV} \to \mathbf{2}$ that has $v|_U \in A_U$ for all finite $U \subseteq \mathrm{PV}$ extends to a unique global section $\bar{v} : \mathcal{L} \to \mathcal{A}$ of A, while each global section g of A is the extension of the valuation $g|_{\mathrm{PV}}$ (which has $(g|_{\mathrm{PV}})|_U \in A_U$ for all finite $U \subseteq \mathrm{PV}$).

This fact in itself does not express compositionality, since it refers to global valuations and does not highlight the fact that the value of φ depends on those of subformulas of φ and them alone. To express the contribution of subformulas, therefore, we use arrows of $\mathcal{L}$, which encapsulate the parsing of formulas as mentioned above.

We should note, however, that arrows $f : U \to V$ of $\mathcal{L}$ do not generally satisfy $\mathrm{pv}(U) = \mathrm{pv}(V)$, although they all do $\mathrm{pv}(V) \subseteq \mathrm{pv}(U)$. Projections and other arrows $g : U' \to V$ that have $\mathrm{pv}(U') \neq \mathrm{pv}(V)$, if we saw them as parsing $\varphi \in V$, would make formulas that are not subformulas of φ appear as if relevant to the parsing and evaluation of φ, which they are not. Even in the case of $\mathrm{pv}(U) = \mathrm{pv}(V)$, the projection $p : (\varphi \wedge \psi, \varphi \vee \psi) \to \varphi \wedge \psi$ would not constitute the parsing of $\varphi \wedge \psi$. To rule out these irrelevant formulas, we can take the following subcategory of $\mathcal{L}$.

Definition 18 Given a category of Boolean formulas $\mathcal{L}$, let $\mathcal{L}^-$ be its smallest subcategory that contains the following arrows and that is closed under finite products of arrows:

- Of the arrows required by the finite-product structure, ones that are split monos.
- Instances of connectives.

We call $\mathcal{L}^-$ the "subcategory without projections" of $\mathcal{L}$. We refer to $\mathcal{L}^-$ as a category of Boolean formulas, too, but distinguish between $\mathcal{L}$ and $\mathcal{L}^-$, if needed, by calling them with and without projections.

Without projections, $\mathcal{L}^-$ cannot be the category of elements of a Σ-algebra—but our presheaf formulation of the semantics enables us to consider presheaves of Boolean values for $\mathcal{L}^-$, which we would say is one of the virtues of the formulation.

Definition 19 Given a category of Boolean formulas $\mathcal{L}$ with projections and its subcategory $i : \mathcal{L}^- \hookrightarrow \mathcal{L}$ without projections, a presheaf of Boolean values for $\mathcal{L}^-$ is a functor of the form $A \circ i$ for a presheaf of Boolean values A for $\mathcal{L}$. We say $A \circ i$ is full when A is full, i.e. $A = B \circ \pi_L$.

$$\begin{tikzcd}
\mathcal{L}^- \arrow[r, "i"] & \mathcal{L} \arrow[r, "\pi_L"] & \Sigma \arrow[r, "B"] & \mathbf{Sets} \\
\varphi \arrow[r, mapsto] & \varphi \arrow[r, mapsto] & * \arrow[r, mapsto] & \mathbf{2}
\end{tikzcd}$$

The point of this definition is to require that a presheaf of Boolean values A for $\mathcal{L}^-$ have $(b, b') \in A_{(\varphi,\psi)}$ only if $b \in A_\varphi$, even though $\mathcal{L}^-$ is without projections and there is no $A(p) : A_{(\varphi,\psi)} \to A_\varphi$.

Definition 19 does not affect the ideas of local consistency, global sections, and contextuality, so that Definitions 13, 16 and 17 extend straightforwardly. All the facts carry over, too, except that (8) no longer works, and that instead of Fact 5 we have the following, because a locally consistent A for $\mathcal{L}^-$ is not required to have a surjection $A(p) : A_{(\varphi,\psi)} \to A_\varphi$, so that $A_{(\varphi,\psi)}$ can be empty while A_φ is not.

Fact 14 If a presheaf of Boolean values A over a category $\mathcal{L}^-$ of Boolean formulas without projections is locally consistent, then $\mathrm{dom}(A)$ is also a category of Boolean formulas (built from a subset of propositional variables).

Now, getting back to Fact 13, we can now "localize" it into the following Fact 15, which highlights the contribution of subformulas. We write $\varphi \lhd \psi$ to mean that φ is a subformula of ψ (cf. Walters (1989b)), Section 14.3 and write $\mathrm{subf}(U)$ for the set of subformulas of sentences in U.

Definition 20 Given any presheaf of Boolean values A for $\mathcal{L}^-$ and any object U of $\mathcal{L}^-$, we say A is "locally compositional at U" to mean that there is an isomorphism between the valuations $v \in A_{\mathrm{pv}(U)}$ and the A-consistent assignments $\alpha : \mathrm{subf}(U) \to \mathbf{2}$.

Fact 15 Any presheaf of Boolean values for $\mathcal{L}^-$ is locally compositional at every object.

Fact 13 is a "global" version of this fact. We will therefore refer to Facts 13 and 15 as *global* and *local compositionality*, respectively.

One may note that Fact 15 can be strengthened as follows. Given an object U of $\mathcal{L}^-$ (or $\mathcal{L}$), let us say that a subset S of $\mathrm{subf}(U)$ is "sieve-like on U" if it is closed under subformulas and $\mathrm{pv}(U) \subseteq S$. The sieve-like sets of subformulas of U correspond essentially to the nonempty sieves on U in $\mathcal{L}^-$ [see, e.g., Mac Lane and Moerdijk (1992), p. 37]. Then

Fact 16 Given any presheaf of Boolean values A for $\mathcal{L}^-$, object U of $\mathcal{L}^-$, and sieve-like set S of subformulas of U, there is an isomorphism between the A-consistent assignments to S and those to $\mathrm{subf}(U)$.

Note the similarity between this and the sheaf condition (with respect to the atomic topology; see, e.g., Mac Lane and Moerdijk (1992), pp. 115f.), even though assignments are not natural since sieves are contravariant whereas presheaves of Boolean values are covariant.

14.6 Semantics of Logic in the Presheaf Form: The First-Order Case

The formalism introduced in Sect. 14.4 concerns propositional logic, but extends readily to typed first-order logic. Although too much detail would go beyond the scope of this article, it is worth laying out briefly how the extension goes.

The idea is to replace the Lawvere theory Σ of Boolean structure (Definition 6) with a *multityped* Lawvere theory that, instead of the single "generator" $*$, has infinitely many generators signifying different contexts of formulas. The new Σ will then give us a multityped algebra of α-equivalence classes of formulas-in-contexts $(\, x_1 : X_1, \dots, x_n : X_n \mid \varphi\,)$. The resulting formalism will be similar to a *hyperdoctrine* [Lawvere (1969, 1970)]. In fact what we give can be regarded as a Lawvere-theory version of the hyperdoctrine for first-order logic.

In the same way as in the definition of a hyperdoctrine, we first take the following category T of contexts:

- Objects of T are contexts $\Gamma = [x_1 : X_1, \dots, x_n : X_n]$ based on a set of sorts or types.
- Arrows of T are terms-in-contexts and tuples thereof: $((\,\Gamma \vdash t_1 : Y_1\,), \dots, (\,\Gamma \vdash t_m : Y_m\,))$ is an arrow from Γ to $[y_1 : Y_1, \dots, y_m : Y_m]$.

Concatenations of contexts constitute finite products in this category T, with projections such as

$$(p_1, \dots, p_n) : [x_1 : X_1, \dots, x_n : X_n, y : Y] \to [x_1 : X_1, \dots, x_n : X_n].$$

A hyperdoctrine is then a functor P from T^{op} to a category of some algebraic structure: for each context $\Gamma \in T_0$, $P(\Gamma)$ is an algebra of formulas in Γ. By contrast, our strategy is to express the algebraic structure at the level of T^{op} instead, and to take a functor to **Sets**.

Here is how we add the algebraic structure of first-order formulas to T^{op} and thereby obtain an extension Σ of T^{op}. An arrow $f : \Gamma \to \Delta$ of T is an arrow $f^* : \Delta \to \Gamma$ of T^{op}, signifying term substitution and context extension. Note that concatenations of contexts are coproducts in T^{op}, so we write $\Gamma + \Delta$ for the concatenation of Γ and Δ. Then we add the following to T^{op}:

- Tuples of contexts as new objects; they constitute finite products.
- For each context Γ, Boolean connectives, e.g. $\wedge : (\Gamma, \Gamma) \to \Gamma$.
- For a projection $p : \Gamma + [y : Y] \to \Gamma$ of T, quantifiers $\exists_p, \forall_p : \Gamma + [y : Y] \to \Gamma$.

From these we generate a category Σ, imposing equalities required by syntax, such as follows.

- Boolean connectives commute with term substitution and context extension, e.g.,

$$\begin{tikzcd}
(\Gamma,\Gamma) \arrow[r, "{(f^*,f^*)}"] \arrow[d, "\wedge"'] \arrow[dr, phantom, "="] & (\Delta,\Delta) \arrow[d, "\wedge"] \\
\Gamma \arrow[r, "f^*"'] & \Delta
\end{tikzcd}$$

- (The Beck-Chevalley condition.) For the pullback square of T on the left below, the right square in Σ commutes.

$$\begin{tikzcd}
\Gamma+[y:Y] \arrow[r, "f+[y:Y]"] \arrow[d, "p_\Gamma"'] \arrow[dr, phantom, "\lrcorner", very near start] & \Delta+[y:Y] \arrow[d, "p_\Delta"] \\
\Gamma \arrow[r, "f"'] & \Delta
\end{tikzcd}
\qquad
\begin{tikzcd}
\Gamma+[y:Y] \arrow[d, "\exists_{p_\Gamma}"'] \arrow[dr, phantom, "="] & \Delta+[y:Y] \arrow[l, "f^*+[y:Y]"'] \arrow[d, "\exists_{p_\Delta}"] \\
\Gamma & \Delta \arrow[l, "f^*"]
\end{tikzcd}$$

Then we can extend Definitions 6, 7, 9, 10, 11, and 12 as follows.

Definition 21 We define the Lawvere theory of FOL structure to be the category Σ as above.

Definition 22 By an "algebra of FOL formulas" we mean a free Σ-algebra L, and by a "category of FOL formulas" we mean the category of elements $\mathcal{L}$ of such an L.

The projection $\pi_L : \mathcal{L} \to \Sigma$ sends each formula-in-context $(\,\Gamma \mid \varphi\,) \in L_\Gamma$ to its context Γ. Again, $\mathcal{L}$ gives a parse tree to each formula-in-context abstractly in the form of arrows with codomain φ.

Definition 23 We define a "Boolean-valued Σ-algebra" to be a Σ-algebra (i.e. finite-product-preserving functor) $B : \Sigma \to \mathbf{Sets}$ that

- sends each context Γ to a Boolean algebra $B(\Gamma)$,
- sends, e.g., $\wedge : (\Gamma, \Gamma) \to \Gamma$ to $\wedge : B(\Gamma) \times B(\Gamma) \to B(\Gamma)$, and
- preserves coproducts.

Definition 24 Given a Boolean-valued Σ-algebra B and a category of FOL formulas $\mathcal{L}$, we define the "full B-valued presheaf" F for $\mathcal{L}$ to be the composition $B \circ \pi_L$ of B after the projection π_L of $\mathcal{L}$. We refer to subpresheaves of F (which may not preserve products) as "B-valued presheaves" for $\mathcal{L}$.

$$\begin{array}{rcccl}
F : \mathcal{L} & \xrightarrow{\ \pi_L\ } & \Sigma & \xrightarrow{\ B\ } & \mathbf{Sets} \\
(\,\Gamma \mid \varphi\,) & \longmapsto & \Gamma & \longmapsto & B(\Gamma)
\end{array}$$

The definitions of global sections, local consistency, and contextuality extend straightforwardly, along with all the facts and theorems corresponding to those in Sects. 14.4 and 14.5.

14.7 Expanding the Standard Semantics with References

We have reformulated the standard semantics in terms of presheaves over a category of formulas in Sect. 14.4, and shown the standard semantics to be non-contextual in Sect. 14.5. The primary goal of this article is to show that, as soon as we modify the topology of this category of formulas with circular constraints, logical paradoxes and contextuality arise. In this section we introduce a category of formulas augmented with a reference relation that can express circular constraints.

14.7.1 A Category of Formulas with References

As mentioned in Sect. 14.5.3, one of the virtues of our presheaf formulation of the semantics is that it enables us to consider a category of Boolean formulas that may not be necessarily the category of elements of a Σ-algebra. Let us introduce such a category of formulas by expanding the one from Sect. 14.4 with circular references and constraints of the sort considered in Fig. 14.3. (We will also briefly mention at the end of the subsection how to extend the category from Sect. 14.6.)

A heuristic idea that may help introduce an expanded category of formulas is to obtain the circuit (iv) of Fig. 14.3, for instance, by welding the following two parts together, or perhaps welding the right part to the left.

(9)

While the left part amounts in the standard way to the following composition of arrows,

$$(\varphi, \chi) \xrightarrow{(\Delta_\varphi, 1_\chi)} (\varphi, \varphi, \chi) \xrightarrow{(1_\varphi, \vee_{\varphi,\chi})} (\varphi, \varphi \vee \chi) \xrightarrow{\wedge_{\varphi,\varphi\vee\chi}} \varphi \wedge (\varphi \vee \chi)$$

we need a new sort of arrows to express the right part. While such new arrows can be added in several ways, we adopt

Definition 25 Let $\mathcal{L}$ be the category of Boolean formulas (with or without projections) generated from a set PV of propositional variables. Given another set N (disjoint from L_*), an injection $t : N \rightarrowtail \mathrm{PV}$, and a function $r : N \to L_*$, we define $\mathcal{L}_{t,r}$ to be the category obtained by adding, to $\mathcal{L}$,

- the elements of N as new objects, and,
- for each $\nu \in N$, two arrows $\nu_t : \nu \to t(\nu)$ and $\nu_r : \nu \to r(\nu)$.

We refer to $\mathcal{L}_{t,r}$ as a category of Boolean formulas “with references”.

This definition uses the span of functions (t, r) to introduce spans (ν_t, ν_r) in $\mathcal{L}_{t,r}$ for $\nu \in N$.

$$\begin{array}{ccccc} \mathrm{PV} & \xleftarrow{\ t\ } & N & \xrightarrow{\ r\ } & L_* \\ t(\nu) & \xleftarrow{\ \nu_t\ } & \nu & \xrightarrow{\ \nu_r\ } & r(\nu) \end{array}$$

N can be seen as a reference relation in which $t(\nu)$ is a propositional variable that refers to $r(\nu)$. Or we can see each $\nu \in N$ as a name, $r(\nu)$ as the sentence to which ν refers, and $t(\nu)$ as the atomic sentence "The sentence ν is true". (We will discuss at the end of this subsection how to treat t formally as a predicate in the first-order setting.) The right part of (9) can then be implemented by having $\chi = t(\nu)$ and $\psi \vee \chi = r(\nu)$ for a $\nu \in N$.

$$\chi \xleftarrow{\ \nu_t\ } \nu \xrightarrow{\ \nu_r\ } \varphi \vee \chi$$

It is important to note that, while $t(\nu)$ is a Boolean formula, we do not treat $\nu \in N$ as one: Although $\mathcal{L}_{t,r}$ (or $\mathcal{L}^-_{t,r}$) has all $\nu \in N$ as objects, we do not apply connectives to $\nu \in N$. We do not even let $\mathcal{L}_{t,r}$ have as an object a tuple with any component from N, which will prove, shortly in Sect. 14.8.1, to be crucial in modelling some logical paradoxes.

Now, given a sentence name ν, its referent $r(\nu)$, and $t(\nu)$ stating that ν is true, the natural constraint on the values of $r(\nu)$ and $t(\nu)$ would be what is often called Tarski's (1923) "Convention T" or "T-schema":

- The sentence ν (i.e. "$r(\nu)$") is true if and only if $r(\nu)$.

In short, $r(\nu)$ and $t(\nu)$ have the same value. (Although the statement above involves quotation, we do not let our syntax to have any distinction between an object language and a metalanguage, which Tarski (1923) did; see Sect. 14.9.) Hence enter

Definition 26 Let $\mathcal{L}$ be a category of Boolean formulas with projections, and let $\pi : \mathcal{L} \to \Sigma$ be the projection. Given an extension $\mathcal{L}_{t,r}$ with references, we extend π to $\pi_{t,r}$ by sending all ν_t and ν_r to 1_*. We then define the full presheaf of Boolean values for $\mathcal{L}_{t,r}$ to be $F = B \circ \pi_{t,r}$,

$$\begin{array}{rcccc} F : \mathcal{L}_{t,r} & \xrightarrow{\ \pi_{t,r}\ } & \Sigma & \xrightarrow{\ B\ } & \mathbf{Sets} \\ \nu_t, \nu_r & \longmapsto & 1_* & \longmapsto & 1_2 \end{array}$$

and a presheaf of Boolean values for $\mathcal{L}_{t,r}$ to be any subpresheaf of F. A presheaf of Boolean values for the subcategory $i : \mathcal{L}^-_{t,r} \to \mathcal{L}_{t,r}$ without projections is a functor of the form $A \circ i$ for a presheaf of Boolean values A for $\mathcal{L}_{t,r}$, and it is full if A is.

$F(\nu_t) = F(\nu_r) = 1_2$ then implies that any F-consistent assignment $\alpha : (t(\nu), r(\nu), \nu) \to \mathbf{2}$, and any global section of F, must satisfy $\alpha(t(\nu)) = \alpha(\nu) = \alpha(r(\nu))$. The notions of local consistency, global sections, and contextuality carry over to the presheaves in Definition 26 straightforwardly. Nevertheless, presheaves can now exhibit quite different topological properties than before, and this is how logical paradoxes exhibit themselves, as will be observed in Sect. 14.8.

Before closing this subsection let us observe that, although the formalism above treats references within propositional syntax, we can do the same within first-order syntax as well. Indeed, in philosophers' and logicians' approaches to the liar and other paradoxes, it is common to use the truth predicate $\mathrm{Tr}(x)$, "x is true", as part of the syntax (which t is not). This can of course be done by extending the category and presheaves from Sect. 14.6 as follows. The core idea is simply to replace $t(\nu)$ of the propositional case with $\mathrm{Tr}(\nu)$.

- As before, take a function $r : N \to L_{[]}$ that sends each name $\nu \in N$ to a (closed) sentence $(r(\nu))$.
- To Σ, add a type N and, for each $\nu \in N$, a constant $\nu : N$. Each constant $\nu : N$ comes with an arrow $\nu^* : [x : N] \to []$ (of Σ).
- To $\mathcal{L}$, add a predicate Tr of type N, so that $(\, x : N \mid \mathrm{Tr}(x)\,)$ is an object of $\mathcal{L}$. It comes with arrows $\nu^*_{\mathrm{Tr}} : (\, x : N \mid \mathrm{Tr}(x)\,) \to (\mathrm{Tr}(\nu))$ (of $\mathcal{L}$) that substitute ν for x. Furthermore, add an arrow $\nu_r : (\, x : N \mid \mathrm{Tr}(x)\,) \to (r(\nu))$ for each $\nu \in N$.

$$(\mathrm{Tr}(\nu)) \xleftarrow{\nu^*_{\mathrm{Tr}}} (\, x : N \mid \mathrm{Tr}(x)\,) \xrightarrow{\nu_r} (r(\nu))$$

- Extend $\pi : \mathcal{L} \to \Sigma$ by sending both ν^*_{Tr} and ν_r to ν^*.
- Extend $B : \Sigma \to \mathbf{Sets}$ by sending $[x : N]$ to $\mathbf{2}^N$ and ν^* to

$$B(\nu^*) : \mathbf{2}^N \to \mathbf{2} :: f \mapsto f(\nu).$$

Then define $F = B \circ \pi$. This makes sure that any F-consistent assignment to $((\mathrm{Tr}(\nu)), (r(\nu)), (\, x : N \mid \mathrm{Tr}(x)\,))$ and any global section of F must give the same value to $(\mathrm{Tr}(\nu))$ and $(r(\nu))$.

14.7.2 Non-circularity and Compositionality

In Sect. 14.5.3 we introduced global and local notions of compositionality, and stated that the standard semantics satisfies both notions. Now that we have introduced a reference relation into the category of formulas, we need a slight refinement, or clarification, on the notions of contextuality.

Observe that global compositionality as we defined it in Sect. 14.5.3 (see Fact 13), i.e. each valuation $v : \mathrm{PV} \to \mathbf{2}$ extending to a unique global section, can fail in the full presheaf F for $\mathcal{L}_{t,r}$ in a simple, non-paradoxical way. Let $\chi = t(\nu)$ refer to $r(\nu) = \varphi \Rightarrow \varphi$ for $\varphi \in \mathrm{PV}$ with $\varphi \neq \chi$. This reference involves no circularity.

$$\chi \xleftarrow{\nu_t} \nu \xrightarrow{\nu_r} \varphi \Rightarrow \varphi \xleftarrow{\Rightarrow_{\varphi,\varphi} \circ \Delta_\varphi} \varphi$$

Extending the diagram in (6), F assigns the following values to $(\varphi, \varphi \Rightarrow \varphi, \nu, \chi)$ as follows.

$$\begin{tikzcd}
F_\chi & F_\nu \arrow[l, "F(\nu_t)"'] \arrow[r, "F(\nu_r)"] & F_{\varphi\Rightarrow\varphi} & F_{(\varphi,\varphi)} \arrow[l, "F(\Rightarrow_{\varphi,\varphi} \circ \Delta_\varphi)"'] \\
1 & 1 \arrow[l, mapsto] \arrow[r, mapsto] & 1 & 1 \arrow[l, mapsto] \\
0 & 0 \arrow[l, mapsto] \arrow[r, mapsto] & 0 & 0 \arrow[ul, mapsto]
\end{tikzcd}$$

Hence an assignment α over $(\varphi, \varphi \Rightarrow \varphi, \nu, \chi)$ is consistent with F iff $\alpha = (b, 1, 1, 1)$ for $b \in A_\varphi$. This means that every global section g of F has $g(\chi) = 1$, and that valuations v that have $v(\chi) = 0$ (which do exist) cannot extend to global sections. In this way, the existence condition in global compositionality breaks down even without circular constraints.

This example suggests that, although we technically treat $\chi = t(\nu)$ as a propositional variable, we should not treat it as a base of the compositional (or inductive) assignment of values. Hence we write $\mathrm{PV}_0 = \mathrm{PV} \setminus \mathrm{im}(t)$ and treat those variables in PV_0 as a base, and use it to redefine the notion of global compositionality. For the same reason, we use $\mathrm{pv}_0(U) = \mathrm{pv}(U) \setminus \mathrm{im}(t)$ to redefine local compositionality. Since the subformula relation $\lhd$ plays a crucial rôle in the Definition 20 of local compositionality, we also extend it by setting $\nu \lhd t(\nu)$ and $r(\nu) \lhd \nu$ (and taking the transitive closure). (Note that the direction $r(\nu) \lhd \nu$ is opposite to ν_r.) In the example, the observation above then means an isomorphism between the F-consistent assignments to $\mathrm{pv}_0(\chi) = \{\varphi\}$ and those to $\mathrm{subf}(\chi) = (\varphi, \varphi \Rightarrow \varphi, \nu, \chi)$. This makes F locally compositional at χ, and similarly at ν, according to

Definition 27 Let A be any presheaf of Boolean values for $\mathcal{L}_{t,r}$. We say A is "globally compositional" to mean that there is an isomorphism between the valuations $v : \mathrm{PV}_0 \to \mathbf{2}$ with $v|_U \in A_U$ for all finite $U \subseteq \mathrm{PV}_0$ and the global sections of A. Given any object U of $\mathcal{L}_{t,r}$, we say A is "locally compositional at U" to mean that there is an isomorphism between the valuations $v \in A_{\mathrm{pv}(U)}$ and the A-consistent assignments $\alpha : \mathrm{subf}(U) \to \mathbf{2}$.

With this refined definition, global and local compositionality can be recovered—unless we introduce circular constraints using $\nu \in N$. The expanded subformula relation $\lhd$ can formalize this idea neatly. Clearly, φ is $\lhd$-minimal iff $\varphi \in \mathrm{PV}_0$. Then the idea that the example reviewed in this subsection involves no circularity amounts to the wellfoundedness of $\lhd$ on $\mathrm{subf}(t(\nu))$ (and hence on $\mathrm{subf}(\nu)$, $\mathrm{subf}(r(\nu)) \subseteq \mathrm{subf}(t(\nu))$). On the other hand, (iv) of Fig. 14.3, with $\chi = t(\nu)$ and $\psi \vee \chi = r(\nu)$ as seen in Sect. 14.7.1, involves circularity, meaning that $\lhd$ fails to be wellfounded since $\nu \lhd t(\nu) = \chi \lhd \psi \vee \chi = r(\nu) \lhd \nu$. In this way, the wellfoundedness of $\lhd$ formally expresses non-circularity. Then we have

Fact 17 Let A be a presheaf of Boolean values for $\mathcal{L}_{t,r}$. If $\lhd$ is wellfounded then A is globally compositional. If $\lhd$ is wellfounded on $\mathrm{subf}(\varphi)$ then A is locally compositional at φ. If $\lhd$ is wellfounded and A is locally consistent then A is not contextual.

Thus wellfoundedness entails compositionality and non-contextuality. On the other hand, with circular constraints, both existence and uniqueness conditions in compositionality can fail in paradoxical ways, as will be discussed in the next section.

14.8 A Contextual Semantics of Logical Paradoxes

In Sect. 14.7, we equipped the category of Boolean formulas with a reference relation, and took presheaves of Boolean values over the extended category. Let us see how these presheaves help us model, and characterize in topological terms, the paradoxical phenomena of the types reviewed in Sect. 14.3. Using our presheaf semantics, this section will identify three topological phenomena from which (or from a combination of which) logical paradoxes can arise: (a) One is strong contextuality, as already seen in Sect. 14.3.1. (b) But a weaker notion of contextuality, logical contextuality, can also account for other paradoxes—a fact that requires presheaf models to even express. (c) In addition, there are logical paradoxes that can arise without contextuality, but we can see them as consisting in a breakdown of compositionality in the senses observed in Sect. 14.5.3 and redefined as Definition 27.

A remark may be in order regarding what we mean by "modelling paradoxical phenomena", since it can mean several levels of modelling. At one level, the singleton Boolean algebra for instance, in which $1 = 0$, can be said to model the liar paradox. Our presheaf framework provides finer-grained models than that: as will be shown below, there are presheaves that can assign values to paradoxical sentences. We would however say this is a minimal criterion for successful modelling. The same criterion may be met, for instance, by any semantic structure that assigns the value 1 to the sentence γ in Curry's paradox (Sect. 14.3.3); yet it only models the conclusion of our inference that γ must be true, but not the inference itself. We aim to show that our presheaves can in fact model the inferences we make when faced with logical paradoxes. The notion of consistency of a local assignment will play an essential rôle in modelling these inferences.

In the following, we write $\mathcal{L}_{t,r}$ for a category of Boolean formulas with references (and with or without projections), and F for the full presheaf of Boolean values for $\mathcal{L}_{t,r}$.

14.8.1 The Liar Paradox and Strong Contextuality

The first thing we observe here is that the new presheaves are a direct extension of the presheaves for the liar-type paradoxes seen in Sect. 14.3.1. The liar sentence λ can be introduced in $\mathcal{L}_{t,r}$ by the following arrows, with $\lambda = t(\nu)$ ("The sentence ν is true") and $r(\nu) = \neg t(\nu)$.

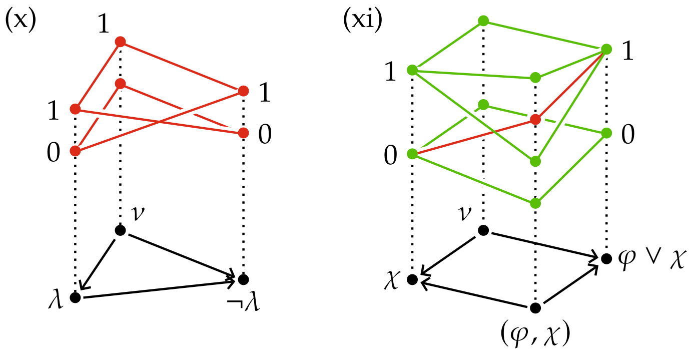


Fig. 14.5 Our bundles for the liar paradox and (iv) of Fig. 14.3

$$\lambda \xleftarrow{\nu_t} \nu \xrightarrow{\nu_r} \neg\lambda \xleftarrow{\neg_\lambda} \lambda$$

Note the circularity here: $\nu \lhd t(\nu) = \lambda \lhd \neg\lambda = r(\nu) \lhd \nu$. Above these arrows, F has $F(\nu_t) = F(\nu_r) = 1_{\mathbf{2}}$ and $F(\neg_\lambda) = \neg$,

$$\begin{tikzcd} F_\lambda & F_\nu \arrow[l, "F(\nu_t)"'] \arrow[r, "F(\nu_r)"] & F_{\neg\lambda} \arrow[r, "F(\neg_\lambda)"] & F_\lambda \\ 1 & 1 \arrow[l, mapsto] \arrow[r, mapsto] & 1 \arrow[dr, mapsto] & 1 \\ 0 & 0 \arrow[l, mapsto] \arrow[r, mapsto] & 0 \arrow[ur, mapsto] & 0 \end{tikzcd} \tag{10}$$

drawing the bundle (x) in Fig. 14.5, which clearly has the same topology as (ii) in Fig. 14.1. Its global inconsistency and local consistency are expressed as follows in our formalism.

- No assignment $\alpha : T \to \mathbf{2}$ to $T = (\lambda, \neg\lambda, \nu)$ can be consistent with F. This means a sort of global inconsistency, with "global" referring to T. Indeed, no assignment to $(\neg\lambda, \nu)$ can be consistent with F, because the two functions $F(\neg_\lambda \circ \nu_t), F(\nu_r) : F_\lambda \to F_\nu$ have no solution b to $F(\neg_\lambda \circ \nu_t)(b) = F(\nu_r)(b)$. From these it follows that F can have no global section (since global sections of F restrict to F-consistent assignments).
- On the other hand, F has a total $\mathrm{LC}(F)$. Some assignments to $(\lambda, \neg\lambda)$ are consistent with F, e.g., $(1, 0)$ and $(1, 0)$. We can also express this with a locally consistent presheaf A for $\mathcal{L}_{t,r}$: while its local consistency implies $A_{(\lambda,\neg\lambda)} \subseteq \{(1, 0), (1, 0)\}$ by the surjectivity of $\langle 1_\lambda, \neg_\lambda \rangle$ [as in (7)], A can have $A_{(\lambda,\neg\lambda)} = \{(1, 0), (1, 0)\}$ with $A_\lambda = A_{\neg\lambda} = A_\nu = \mathbf{2}$. (Note that there is no $A_{(\lambda,\nu)}$ or $A_{(\neg\lambda,\nu)}$, since neither (λ, ν) nor $(\neg\lambda, \nu)$ is an object of $\mathcal{L}_{t,r}$. It is indeed crucial, in order to have a total A, that we define $\mathcal{L}_{t,r}$ not to have as an object any tuple containing ν: e.g., if it had an object (ν, ν) and arrow $\Delta_\nu : \nu \to (\nu, \nu)$, the local consistency would imply $A_{(\lambda,\neg\lambda)} = A_{(\nu,\nu)} \subseteq \{(1, 1), (0, 0)\}$ and hence $A_{(\lambda,\neg\lambda)} = \varnothing$.)

Thus, our presheaves confirm an observation that we made in Sect. 14.3.1 (and perhaps partly in Sect. 14.3.2), namely:

Observation 1 A class of logical paradoxes, including the liar-type, lies in strong contextuality, in which one can use the logic of global sections to derive a contradiction and show that there is no global section.

Given the remark at the beginning of this section on what we mean by modelling, let us reflect on how the presheaves model the inference we make regarding the liar paradox. On the one hand, as observed above, global inferences, such as $\neg b = b$ having no solution, establish global inconsistency. This means that there exists no global section of F. On the other hand, this does not mean that F has no section. Indeed, it accommodates local sections or edges as in (10) and (x), along which our inference goes that there can be no global section, in a manner isomorphic to (1). Local consistency of F thus models our inference on the liar paradox.

14.8.2 The Truth-Teller and a Breakdown of Compositionality

Let us observe how the truth-teller can be modelled and in what exact way it is paradoxical. As a preparation, note that strong contextuality, which characterizes the liar and similar paradoxes as in Observation 1, means a breakdown of compositionality (Facts 13 and 15), and its existence condition in particular. By contrast, the truth-teller and similar paradoxes concern the uniqueness condition.

The truth-teller τ is introduced in $\mathcal{L}_{t,r}$ by

$$\tau \xleftarrow{\nu_t} \nu \xrightarrow{\nu_r} \tau$$

so that $\tau = t(\nu)$ and $r(\nu) = t(\nu)$. Then, since both 1 and 0 are solutions to $F(\nu_t)(b) = F(\nu_r)(b)$, two assignments $(1, 1)$ and $(0, 0)$ to (τ, ν) are both consistent with F. It follows that the uniqueness condition in global compositionality fails: Given a valuation $v : \mathrm{PV}_0 \to \mathbf{2}$, if F has a global section at all, there are global sections g_1 and g_0 both of which extend v but that have $g_1(\tau) = 1$ and $g_0(\tau) = 0$. It is also immediate that the uniqueness condition in local compositionality fails at τ, since $\mathrm{pv}_0(\tau) = \varnothing$ (to which there is a unique, empty assignment).

It may be helpful to compare the case of the truth-teller to the example discussed in Sect. 14.7.2, which was given by $t(\nu) = \chi$ and $r(\nu) = \varphi \Rightarrow \varphi$ for $\varphi \in \mathrm{PV}_0$. As discussed there, the example is locally compositional at ν, meaning that there is an isomorphism between the F-consistent assignments to $\mathrm{pv}_0(\nu) = \{\varphi\}$ and those to $\mathrm{subf}(\nu) = (\varphi, \varphi \Rightarrow \varphi, \nu)$. Indeed, in this example, locally consistent presheaves A must have $A_\chi \subseteq \{1\}$. In this way, both global and local inferences determine the value of ν once we fix the values of the propositional variable in $\mathrm{pv}_0(\nu)$. By contrast, in the case of the truth-teller, there are no propositional variables from whose values even global inference can determine the value of ν. This may mean, for instance if valuations are supposed to model facts of some sort, that facts determine the value of ν when $r(\nu) = \varphi \Rightarrow \varphi$, but that no facts can determine the value of ν when $r(\nu) = t(\nu)$.

To sum up,

Observation 2 A class of logical paradoxes, including the truth-teller, lies in the breakdown of the uniqueness condition in compositionality.

This also applies to the controlled truth-teller in (iv) (or (v)) of Fig. 14.3, in which, informally, the values of φ and χ behave as follows: if φ is true, so is χ; if φ is false, then χ behaves the same way as the truth-teller. As we saw in Sect. 14.7.1, the circular part of (iv) is modelled by the following arrows of $\mathcal{L}_{t,r}$.

$$\chi \xleftarrow{\nu_t} \nu \xrightarrow{\nu_r} \varphi \vee \chi \xleftarrow{\vee_{\varphi,\chi}} (\varphi, \chi) \xrightarrow{p} \chi$$

Over these arrows, F gives the bundle (xi) of Fig. 14.5, modelling the controlled truth-teller in the following way. Given an F-consistent assignment $\alpha : (\varphi, \chi, \varphi \vee \chi, \nu) \to \mathbf{2}$, its local consistency with F implies $\alpha(\chi) = \alpha(\nu) = \alpha(\varphi \vee \chi)$, and that $b = \alpha(\chi)$ must be a solution to the equation

$$(\alpha(\varphi) \vee b) = b,$$

i.e., a fixed point of the Boolean function $\alpha(\varphi) \vee -$, once $\alpha(\varphi)$ is fixed. Therefore, if α has $\alpha(\varphi) = 1$ it must be $(1, 1, 1, 1)$ over $(\varphi, \chi, \varphi \vee \chi, \nu)$. On the other hand, if $\alpha(\varphi) = 0$ then α can be either $(0, 1, 1, 1)$ or $(0, 0, 0, 0)$ over $(\varphi, \chi, \varphi \vee \chi, \nu)$. In this way, if a valuation $v : \mathrm{PV}_0 \to \mathbf{2}$ extends to a global section g such that $g(\varphi) = 0$, it has two extensions g_1 and g_0 such that $g_1(\chi) = 1$ and $g_0(\chi) = 0$, resulting in the breakdown of the uniqueness condition in compositionality as in Observation 2.

One should observe that the case we have just studied exhibits logical contextuality, as is obvious from (xi). In other words, the assignment $(1, 0)$ to (φ, χ) is ruled out by global consistency and global inference, but not by local consistency and local inference. This is a manifestation of circularity, as Fact 17 implies. Logical contextuality does not necessarily mean a paradox, as will be observed in the next subsection.

14.8.3 Curry's Paradox and Logical Contextuality

Let us review how our presheaf model can treat Curry's paradox. The paradox lies in a certain kind of genuine logical contextuality. (By genuine logical contextuality we mean logical contextuality in combination with local consistency; see Definition 4.) We do not claim, however, that all instances of logical contextuality are paradoxical. In this subsection we consider Curry's paradox and another instance of logical contextuality, and investigate an extra condition that makes certain logical contextuality paradoxical.

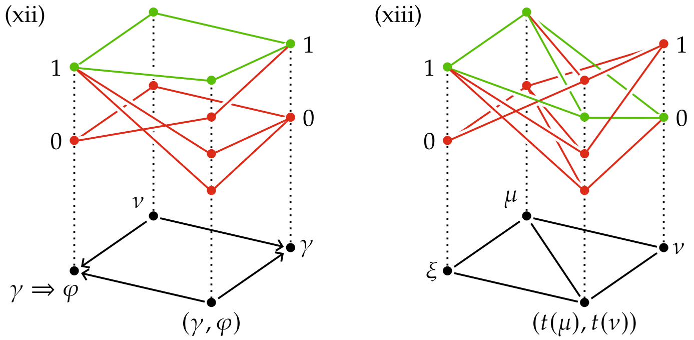


Fig. 14.6 Our bundles for Curry's paradox and Gupta's puzzle

The circular constraint of Curry's paradox is given by the following arrows:

$$\gamma \xleftarrow{\nu_t} \nu \xrightarrow{\nu_r} \gamma \Rightarrow \varphi \xleftarrow{\Rightarrow_{\gamma,\varphi}} (\gamma, \varphi) \xrightarrow{p} \gamma$$

The paradox is that this constraint seems to force not just γ but also φ to be true. F models this inference in the following way, as shown in (xii) of Fig. 14.6. In global terms, every F-consistent assignment α to $(\gamma, \varphi, \gamma \Rightarrow \varphi, \nu)$ has $\alpha(\gamma) = \alpha(\nu) = \alpha(\gamma \Rightarrow \varphi)$, so that $b = \alpha(\gamma)$ is a solution to

$$(b \Rightarrow \alpha(\varphi)) = b,$$

i.e., a fixed point of $- \Rightarrow \alpha(\varphi)$. Therefore $(1, 1, 1, 1)$ is the only F-consistent assignment to $(\gamma, \varphi, \gamma \Rightarrow \varphi, \nu)$. In local terms, however, a locally consistent presheaf A can still have $A_\gamma = A_\nu = A_{\gamma \Rightarrow \varphi} = \mathbf{2}$ and $A_{(\gamma,\varphi)} = \mathbf{2} \times \mathbf{2}$, since every $A(f)$ for $f = \nu_t, \nu_r, \Rightarrow_{\gamma,\varphi}, p$ can be surjective, as is clear from (xii). In short, Curry's paradox exhibits genuine logical contextuality. It should be noted that Curry's paradox is a controlled liar, like (vi) of Fig. 14.3: if φ is false, then $r(\nu) = t(\nu) \Rightarrow \varphi$ amounts to the liar $r(\nu) = \neg t(\nu)$. Indeed, the bundles for Curry's paradox and (vi) are homotopy equivalent.

The next case we consider is "Gupta's puzzle". While its original version (Gupta, 1982, p. 34) involves formulas from PV_0, we present here a simplified version that does not. (The original version can be seen as a controlled version, in which a certain valuation to PV_0 triggers our version of the puzzle.) It is given by the following three formulas named by three names $\mu, \nu, \xi \in N$:

$$r(\mu) = t(\xi), \qquad r(\nu) = \neg t(\xi), \qquad r(\xi) = \neg(t(\mu) \wedge t(\nu)).$$

This definition comes with the following arrows in $\mathcal{L}_{t,r}$.

(11)
$$\begin{tikzcd}
 & t(\mu) \wedge t(\nu) \arrow[rr, "\neg_{t(\mu)\wedge t(\nu)}"] & & \neg(t(\mu) \wedge t(\nu)) & & \xi \arrow[ll, "\xi_r"'] \arrow[dr, "\xi_t"] & \\
(t(\mu), t(\nu)) \arrow[ur, "\wedge_{t(\mu),t(\nu)}"] \arrow[rr, "p_1"] \arrow[dr, "p_2"'] & & t(\mu) & & \mu \arrow[ll, "\mu_t"'] \arrow[rr, "\mu_r"] & & t(\xi) \arrow[dl, "\neg_{t(\mu)}"] \\
 & t(\nu) & & \nu \arrow[ll, "\nu_t"] \arrow[rr, "\nu_r"'] & & \neg t(\xi) &
\end{tikzcd}$$

This gives a circular constraint, making $\varphi \lhd \psi$ for any pair of formulas φ, ψ that appear. Above this diagram (let us identify vertices along μ_t, μ_r, ν_t, ν_r, and ξ_r), F assigns the following constraints, which are shown in (xiii) of Fig. 14.6.

$$\begin{tikzcd}
 & F_\xi \arrow[dr, "1_2"] & \\
F_{(t(\mu),t(\nu))} \arrow[ur, "\neg \circ \wedge"] \arrow[rr, "p_1"] \arrow[dr, "p_2"'] & & F_\mu \arrow[dl, "\neg"] \\
 & F_\nu &
\end{tikzcd}$$

As (xiii) makes clear, $(1, 0, 1)$ is the only assignment to (μ, ν, ξ) consistent with F, but the case exhibits genuine logical contextuality since the functions in the diagram above are all surjective.

Now that we have modelled two instances of genuine logical contextuality over $\mathcal{L}_{t,r}$, let us observe what formal property makes one paradoxical but not the other. On the one hand, like the liar paradox and the truth-teller, Curry's paradox obstructs compositionality: assuming $\varphi \in \mathrm{PV}_0$ for simplicity, we have $\mathrm{pv}_0(\gamma) = \{\varphi\}$ and $\mathrm{subf}(\gamma) = (\gamma, \varphi, \gamma \Rightarrow \varphi, \nu)$; the F-consistent assignment of 0 to the former has no extension to the latter, meaning that F is not locally compositional at γ. (We may note that F is locally compositional at φ.) This captures the informal way in which Curry's paradox seems paradoxical: since φ is a subformula of (the referent of) γ and not the other way around, the value of φ is supposed to be determined independently of γ, but apparently the definition of γ rules out the possibility that φ is false. On the other hand, Gupta's puzzle happens to give no obstruction to compositionality: Let U be the set of formulas in (11). Then, for each $\varphi \in U$, we have $\mathrm{pv}_0(\varphi) = \varnothing$ and $\mathrm{subf}(\varphi) = U$. Therefore the unique F-consistent assignment to $\mathrm{subf}(\varphi)$ corresponds trivially to the unique (F-consistent) assignment to $\mathrm{pv}_0(\varphi)$. Thus we have

Observation 3 A class of logical paradoxes, including Curry's paradox, lies in cases of logical contextuality (as opposed to strong contextuality) in which the existence condition of compositionality fails.

Given Observations 1, 2, and 3, we may classify types of circular constraints and potential paradoxes they may give rise to in the following way.

(I) $\nu \in N$ has $\mathrm{pv}_0(\nu) = \varnothing$. There are three (mutually exclusive) subtypes:

(a) There is no F-consistent assignment to $\mathrm{subf}(\nu)$. The liar paradox is of this type.
(b) There is a unique F-consistent assignment to $\mathrm{subf}(\nu)$. Our version of Gupta's puzzle is of this type.
(c) There are more than one F-consistent assignment to $\mathrm{subf}(\nu)$. The truth-teller is of this type.

(II) $\nu \in N$ has $\mathrm{pv}_0(\nu) \neq \varnothing$. There are three subtypes that are *not* mutually exclusive:

(a) Some valuations to $\mathrm{pv}_0(\nu)$ extend to no F-consistent assignment to $\mathrm{subf}(\nu)$. Curry's paradox is of this type.
(b) Some valuations to $\mathrm{pv}_0(\nu)$ extend to unique F-consistent assignment to $\mathrm{subf}(\nu)$. The original version of Gupta's puzzle is of this type.
(c) Some valuations to $\mathrm{pv}_0(\nu)$ extend to more than one F-consistent assignment to $\mathrm{subf}(\nu)$. (iv) of Fig. 14.3 is of this type.

Cases in (II) can be controlled versions of (I), such as Curry's paradox being a controlled liar paradox. In each of (I) and (II), (a) means an obstruction to the existence condition in compositionality, and (c) to the uniqueness condition, whereas (b) is not paradoxical.

14.9 Comparison to Previous Approaches

The previous sections have expanded Abramsky's topological model of contextuality and presented a new model of logical paradoxes. As was stated in the Introduction, one goal of this article is to demonstrate that this approach offers a unifying framework in which various approaches by philosophers and logicians can be classified and compared in structural terms. Let us take three of these approaches and briefly review how they can be regarded as variants of our model.

Tarski (1923) observed that, if a strong enough language $\mathcal{L}$ had its own truth predicate $\mathrm{Tr}(x)$ satisfying his Convention T (see p. 558), then $\mathcal{L}$ would have the liar sentence $r(\nu) = \neg\mathrm{Tr}(\nu)$ satisfying

- $\mathrm{Tr}(\nu)$ (i.e., ν is true) if and only if $\neg\mathrm{Tr}(\nu)$

and a contradiction would follow. He therefore introduced a hierarchy of languages in which the truth predicate of an object language $\mathcal{L}$ belongs to the metalanguage on $\mathcal{L}$ and not to $\mathcal{L}$ itself. Then the biconditional above cannot be obtained, since ν must name a sentence of $\mathcal{L}$ whereas $\neg\mathrm{Tr}(\nu)$ belongs to the metalanguage. In short, Tarski kept any logical paradox from arising by prohibiting circular definitions.

Tarski's hierarchy can be simulated within our framework in the following way. (Here we use the propositional version of $\mathcal{L}_{t,r}$, but one can also use the first-order version to accommodate the predicate $\mathrm{Tr}(x)$ as part of formalism.) Let $\mathcal{L}_{t,r}$, N, and PV_0 be as in the previous sections, and then define a family of subcategories $\mathcal{L}_0 \subseteq \mathcal{L}_1 \subseteq \cdots \subseteq \mathcal{L}_n \subseteq \cdots$ of $\mathcal{L}_{t,r}$ recursively as follows:

- Let $N_0 = \varnothing$ and $\mathcal{L}_0$ be the category of formulas generated from PV_0.
- Given PV_n and $\mathcal{L}_n$ for an $n \in \mathbb{N}$, let

$$N_{n1} = \{\nu \in N | r(\nu) \in \mathcal{L}_n, \} \subseteq N,$$
$$\mathrm{PV}_{n1} = \mathrm{PV}_n \cup \{t(\nu) \in \mathrm{PV} | \nu \in N_{n1}\},$$

and let $\mathcal{L}_{n+1}$ be the category of formulas generated from PV_{n+1} and augmented with the reference relation of $\mathcal{L}_{t,r}$ restricted to N_{n+1}.

This definition makes sure that $t(\nu)$ belongs to $\mathcal{L}_n$ only if $r(\nu)$ does to $\mathcal{L}_{n-1}$. Therefore a $\nu \in N$ of the liar sentence $r(\nu) = \neg t(\nu)$ cannot be in any of $\mathcal{L}_n$. Indeed, let $\mathcal{L}_\infty$ be the union of all $\mathcal{L}_n$; then $\lhd$ is wellfounded on the formulas (and formula names ν) in $\mathcal{L}_\infty$, making presheaves of Boolean values for $\mathcal{L}_\infty$ non-contextual (or not genuinely contextual) and compositional by Fact 17. This is how our topological model formalizes Tarski's way of preventing logical paradoxes by prohibiting circular definitions.

Kripke (1975) was among the first to model the behavior of the truth predicate in a language that has its own truth predicate. In our setting this amounts to using the entire $\mathcal{L}_{t,r}$ as opposed to a subcategory. (Again, we use the propositional version but one can use the first-order version.) Kripke's idea goes as follows in our formalism: Given a valuation $v : \mathrm{PV}_0 \to \mathbf{2}$, every formula in $\mathcal{L}_0$ has a truth value, but a formula $t(\nu)$ in $\mathrm{PV} \setminus \mathrm{PV}_0$ may or may not have a truth value; therefore we have a valuation $v' : \mathrm{PV}' \to \mathbf{2}$ with $\mathrm{PV}_0 \subseteq \mathrm{PV}' \subseteq \mathrm{PV}$. Then v' extends to a global section $\bar{v}'$ over the category of formulas $\mathcal{L}'$ (without references) generated from PV', determining the values of all the formulas $\varphi \in \mathcal{L}'$. Let us describe this by saying that v' assigns the value $\bar{v}'(\varphi)$ to φ. This may include $r(\nu) = \varphi \in \mathcal{L}'$, so we take a revised valuation $\rho(v') : \rho(\mathrm{PV}') \to \mathbf{2}$ by

$$\rho(\mathrm{PV}') = \mathrm{PV}' \cup \{\, t(\nu) \in \mathrm{PV} \mid r(\nu) \in \mathcal{L}' \,\},$$
$$\rho(v')(\chi) = \begin{cases} v(\chi) & \text{if } \chi \in \mathrm{PV}_0, \\ \bar{v}'(r(\nu)) & \text{if } \chi \notin \mathrm{PV}_0 \text{ and } \chi = t(\nu) \end{cases}$$

(note the similarity between $\mathrm{PV}' \mapsto \rho(\mathrm{PV}')$ here and $\mathrm{PV}_n \mapsto \mathrm{PV}_{n+1}$ in our formulation above of Tarski's idea). Then Tarski's Convention T is satisfied in a valuation v' iff v' is a fixed point of ρ. Kripke shows that ρ has the smallest fixed point, but it amounts to a global extension of v over $\mathcal{L}_\infty$ (again by Fact 17). Kripke deems a formula φ paradoxical if there is no fixed point of ρ that assigns a value to it—this corresponds to types (Ia) and (IIa) in our classification.

We have just simulated Kripke's account using our presheaf of Boolean values, but in Kripke's own formalism he uses the third truth value, "neither", which can be a fixed point of $\neg$ and therefore a solution to the equation $b = \neg b$. Other philosophers have also considered "both", which can also solve $b = \neg b$. We can incorporate these values in our formalism and take presheaves of three values or four values. The resulting bundle for the liar paradox has two global sections, a constant "neither" and

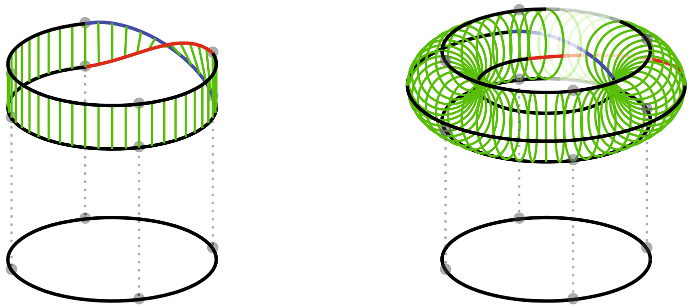


Fig. 14.7 The Möbius strip and the Klein bottle for the liar paradox

a constant "both". This idea can be illustrated by Fig. 14.7, which extends Fig. 14.2: the Möbius strip does not have a global section on its boundary, but the Klein bottle has two global sections (the top and bottom circles in the picture).

Another approach to logical paradoxes is revision theory [Gupta (1982), Gupta and Belnap (1993)], which refuses "neither" or "both" and adheres to the two Boolean values. In terms of our model, revision theory takes global sections over the category of formulas $\mathcal{L}$ without references. With the presence of paradoxical cases like the liar sentence, no global section satisfies Convention T; instead of as a constraint on global sections, we use Convention T as a rule of revision: given a global section g over $\mathcal{L}$, define $\rho(g) = \bar{v}$ for valuation $v : \mathrm{PV} \to \mathbf{2}$ such that

$$v(\chi) = \begin{cases} g(\chi) & \text{if } \chi \in \mathrm{PV}_0, \\ g(r(\nu)) & \text{if } \chi \notin \mathrm{PV}_0 \text{ and } \chi = t(\nu). \end{cases}$$

Each global section restricted to $\mathrm{PV} \setminus \mathrm{PV}_0$ is called a "hypothesis", and ρ revises a hypothesis to another, forming a sequence of hypotheses. Some paradoxical sentences receive unstable values from this sequence: e.g., the value of the liar sentence λ oscillates, with $\rho(g)(\lambda) = \neg g(\lambda)$ for any hypothesis g. Other sentences receive stable values: e.g., the truth-teller τ has $\rho(g)(\tau) = g(\tau)$. In (our version of) Gupta's puzzle, suppose g assigns $(1, 1, 1)$ to $(t(\mu), t(\nu), t(\xi))$. Then $\rho(g)$, $\rho^2(g)$, and $\rho^n(g)$ $(n \geqslant 3)$ respectively assign $(1, 0, 0)$, $(0, 1, 1)$, and $(1, 0, 1)$, coming to and staying at the unique F-consistent assignment to $(t(\mu), t(\nu), t(\xi))$. This process can be seen in (xiii) of Fig. 14.6, although, unlike in our treatment of the puzzle in Sect. 14.8.3, the inference of how the revision goes has a designated direction.

We have compared our presheaf approach to logical paradoxes with some previous approaches by philosophers' and logicians', and shown that the latter can be seen as particular implementations within our general framework. There are more approaches with which to compare ours. For instance, the similarity between our model and valuation algebras [Kohlas et al. (2012)] is notable. Another notable similarity is found in situation semantics [Barwise and Perry (1983), Barwise and

Etchemendy (1987), Devlin (2006)], in which, for instance, the notion of "infon" amounts to objects (that are not tuples) of the category of elements $\mathcal{F}$.

14.10 Conclusion

This article has expanded on the insight of Abramsky et al. (2015, 2017) that logical paradoxes share the same topological structure of inconsistency with quantum paradoxes of contextuality. We have demonstrated that, by rewriting the standard semantics of propositional and first-order logic in the format of presheaves over categories of formulas, we can not just import the topological notions of global and local consistencies to the standard semantics, we can also use these notions to express the idea of compositionality as a topological idea. Furthermore, the flexibility of the categorical formalism enables us to introduce and express circular, potentially paradoxical constraints with presheaves over categories of formulas augmented with references.

In this topological framework, we have shown on the one hand that the standard semantics is non-contextual—since the maps assigning denotations to formulas in the standard semantics are precisely the global sections—and on the other that (genuine) logical contextuality is a manifestation of circularity. In the end we have classified types of logical paradoxes and characterized them in topological terms, with contextuality and compositionality. Some, like the liar paradox, are strongly contextual and obstructs the existence condition in compositionality. Others, like the truth-teller, have the uniqueness condition in compositionality break down. There are controlled versions of these, too, such as Curry's paradox as a controlled liar. Yet our models also make it clear that there are non-paradoxical cases of circularity in which compositionality are preserved.

It is beyond the scope of this article to claim, for example, that we should therefore consider such and such sentences to have (or not to have) such and such values, in the way some philosophers do. We have nevertheless illustrated that our topological framework is a unifying one, in which various approaches to logical paradoxes philosophers and logicians have taken can be seen as variations of topological ideas. This suggests that Abramsky's observation that quantum and logical paradoxes share the same structure is a robust one, and that further research of the shared logical and semantic structure may reveal an interesting novel pathway between the study of quantum contextuality in quantum computation and the study of circular definitions in logic and computation.

References

Abramsky, S., et al. (2013). Relational databases and Bell's theorem. In V. Tannen (Ed.), *In Search of Elegance in the Theory and Practice of Computation: Essays Dedicated to Peter Buneman* (pp. 13–35). Springer.

Abramsky, S. (2013). Relational hidden variables and non-locality. *Studia Logica, 101*, 411–452.

Abramsky, S. (2017). Contextuality: At the borders of paradox. In E. Landry (Ed.), *Categories for the Working Philosopher* (pp. 262–285). Oxford University Press.

Abramsky, S., Soares Barbosa, R., Carù, G., de Silva, N., Kishida, K., and Mansfield, S. (2018). Minimum quantum resources for strong non-locality. In M. W. Wild (Ed.), *TQC 2017*, LIPIcs, pp. 9:1–9:20.

Abramsky, S., Soares Barbosa, R., Kishida, K., Lal, R., and Mansfield, S. (2015). Contextuality, cohomology and paradox. In S. Kreutzer (Ed.), *CSL 2015*, LIPIcs, pp. 211–228.

Abramsky, S., Soares Barbosa, R., & Mansfield, S. (2017). Contextual fraction as a measure of contextuality. *Physical Review Letters, 119*, 050504.

Abramsky, S., & Brandenburger, A. (2011). The sheaf-theoretic structure of non-locality and contextuality. *New J. Phys., 13*, 113036–113075.

Abramsky, S., Gottlob, G., and Kolaitis, P.G. (2013). Robust constraint satisfaction and local hidden variables in quantum mechanics. In F. Rossi (Ed.), *IJCAI-13*, 440–446.

Abramsky, S., Mansfield, S., and Soares Barbosa, R. (2012). The cohomology of non-locality and contextuality'. In B. Jacobs et al. (eds.), *QPL 2011*, EPTCS, pp. 1–14.

Abramsky, S., & Zvesper, J. A. (2015). From Lawvere to Brandenburger-Keisler: interactive forms of diagonalization and self-reference. *Journal of Computer and System Sciences, 81*(5), 799–812.

Barwise, J., & Etchemendy, J. (1987). *The Liar: An Essay on Truth and Circularity*. Oxford University Press.

Barwise, J. and Perry, J. (1983). *Situations and Attitudes*, MIT Press.

Bell, J. S. (1964). On the Einstein-Podolsky-Rosen paradox. *Physics, 1*, 195–200.

Berry, G., et al. (2000). The foundations of Esterel. In G. Plotkin (Ed.), *Proof* (pp. 425–454). Language, and Interaction, MIT Press.

Devlin, K. (2006). Situation theory and situation semantics. In D. M. Gabbay & J. Woods (Eds.), *Handbook of the History of Logic* (Vol. 7, pp. 601–664). Elsevier.

Dzhafarov, E. N., & Kujala, J. V. (2016). Context-content systems of random variables: The Contextuality-by-Default theory. *J. Math. Psychol., 74*, 11–33.

Dzhafarov, E.N. and Kujala, J.V. (2017). Contextuality-by-Default 2.0: systems with binary random variables. In J. A. de Barros et al. (eds.), *QI 2016*, LNCS, pp. 16–32.

Ghirardi, G.-C., Rimini, A., & Weber, T. (1980). A general argument against superluminal transmission through the quantum mechanical measurement process. *Lettere al Nuovo Cimento, 27*, 293–298.

Gödel, K. (1931). Uber formal unentscheidbare Satze der 'Principia Mathematica' und venvandter Systeme. 1. *Monatshefre fur Mathematik und Physik, 38*, 173–198.

Greenberger, D. M., Horne, M. A., & Zeilinger, A. (1989). Going beyond Bell's theorem. In M. Kafatos (Ed.), *Bell's Theorem* (pp. 69–72). Quantum Theory, and Conceptions of the Universe, Kluwer.

Gupta, A. (1982). Truth and Paradox. *Journal of Philosophical Logic, 11*, 1–60.

Gupta, A. and Belnap, N. (1993). *The Revision Theory of Truth*, MIT Press.

Halbwachs, N., and Maraninchi, F. (1995). On the symbolic analysis of combinational loops in circuits and synchronous programs. *Euromicro'95*.

Hardy, L. (1993). Nonlocality for two particles without inequalities for almost all entangled states. *Phys. Rev. Lett., 71*, 1665–1668.

Hermida, C., et al. (2004). Fibrations for abstract multicategories. In G. Janelidze (Ed.), *Galois Theory* (pp. 281–293). Hopf Algebras, and Semiabelian Categories, AMS.

Howard, M., Wallman, J., Veitch, V., & Emerson, J. (2014). Contextuality supplies the 'magic' for quantum computation. *Nature, 510*, 351–355.

Isham, C. J., & Butterfield, J. (1998). Topos perspective on the Kochen-Specker theorem: I. Quantum states as generalized valuations. *Int. J. Theor. Phys., 37*(11), 2669–2733.
Kautz, W.H. (1970) The necessity of closed circuit loops in minimal combinational circuits. *IEEE Transactions on Computers* **C-19**:2, 162–164.
Kishida, K. (2016). Logic of local inference for contextuality in quantum physics and beyond. In I. Chatzigiannakis, et al., (eds.), *ICALP 2016*, LIPIcs, pp. 113:1–113:14.
Kochen, S., & Specker, E. (1967). The problem of hidden variables in quantum mechanics. *Journal of Mathematics and Mechanics, 17*, 59–87.
Kohlas, J., Pouly, M., & Schneuwly, C. (2012). Generic local computation. *J. Comput. Syst. Sci., 78*(1), 348–369.
Kripke, S. (1975). Outline of a theory of truth. *Journal of Philosophy, 72*, 690–716.
Lawvere, F. W. (1969). Adjointness in foundations. *Dialectica, 23*, 281–296.
Lawvere, F.W. (1970). Equality in hyperdoctrines and comprehension schema as an adjoint functor. In A. Heller (Ed.), *Applications of Categorical Algebra*, AMS, pp. 1–14.
Mac Lane, S., & Moerdijk, I. (1992). *Sheaves in Geometry and Logic: A First Introduction to Topos Theory*. Springer.
Malik, S. (1994). Analysis of cyclic combinational circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 13*(7), 950–956.
Mermin, N. D. (1990). Extreme quantum entanglement in a superposition of macroscopically distinct states. *Phys. Rev. Lett., 65*, 1838–1840.
Murawski, R. (1998). Undefinability of truth. The problem of priority: Tarski vs Gödel. *History and Philosophy of Logic, 19*, 153–160.
Nielsen, M. A., & Chuang, I. L. (2011). *Quantum Computation and Quantum Information: 10th* (Anniversary). Cambridge University Press.
Penrose, R. (1992). On the cohomology of impossible figures. *Leonardo, 25*, 245–247.
Popescu, S., & Rohrlich, D. (1994). Quantum nonlocality as an axiom. *Found. Phys., 24*, 379–385.
Raussendorf, R. (2013). Contextuality in measurement-based quantum computation. *Phys. Rev. A, 88*, 022322.
Shulman, M. (2016). *Categorical Logic from a Categorical Point of View*, draft (Jul. 28, 2016) online at http://mikeshulman.github.io/catlog/catlog.pdf
Tarski, A., & Pojęcie prawdy w językach nauk dedukcyjnych, (The Concept of Truth in Languages of Deductive Sciences), Nakladem Towarzystwa Naukowego Warszawskiego, 1933; translation by J. H. Woodger, "The Concept of Truth in Formal Languages", in A. Tarski, Logic, Semantics, Metamathematics: Papers from,. (1923). to 1938. *Oxford University Press, 1956*, 152–278.
Walters, R. F. C. (1989). A note on context-free languages. *Journal of Pure and Applied Algebra, 62*, 199–203.
Walters, R. F. C. (1989). The free category with products on a multigraph. *Journal of Pure and Applied Algebra, 62*, 205–210.

Chapter 15
The Contextuality-by-Default View of the Sheaf-Theoretic Approach to Contextuality

Ehtibar N. Dzhafarov

Abstract The Sheaf-Theoretic Contextuality (STC) theory developed by Abramsky and colleagues is a very general account of whether multiply overlapping subsets of a set, each of which is endowed with certain "local" structure, can be viewed as inheriting this structure from a global structure imposed on the entire set. A fundamental requirement of STC is that any intersection of subsets inherit one and the same structure from all intersecting subsets. I show that when STC is applied to systems of random variables, it can be recast in the language of the Contextuality-by-Default (CbD) theory, and this allows one to extend STC to arbitrary systems, in which the requirement in question (called "consistent connectedness" in CbD) is not necessarily satisfied. When applied to possibilistic systems, such as systems of logical statements with unknown truth values, the problem arises of distinguishing lack of consistent connectedness from contextuality. I show that it can be resolved by considering systems with multiple possible deterministic realizations as quasi-probabilistic systems with epistemic (or Bayesian) probabilities assigned to the realizations. Although STC and CbD have distinct native languages and distinct aims and means, the conceptual modifications presented in this paper seem to make them essentially coextensive.

Keywords Contextual fraction · Contextuality · Consistent connectedness · Dichotomization · Inconsistent connectedness · Measures of contextuality

15.1 Introduction

1.1. Contextuality-by-Default (CbD) and Sheaf-Theoretic Contextuality (STC) are two general approaches to establishing and measuring (non)contextuality of systems of measurements. The word "measurement" is understood very broadly, including various relations between inputs and outputs in a physical entity, between databases and records, between logical statements and their truth values, etc. The two theo-

E. N. Dzhafarov (✉)
Purdue University, West Lafayette, USA
e-mail: ehtibar@purdue.edu

A. Palmigiano and M. Sadrzadeh (eds.), *Samson Abramsky on Logic and Structure in Computer Science and Beyond*, Outstanding Contributions to Logic 25,
https://doi.org/10.1007/978-3-031-24117-8_15

ries employ distinct mathematical languages, and were designed with different aims in mind. With respect to their applicability areas, when dealing with probabilisitic scenarios, STC is confined to special systems, called “strongly consistently connected” in CbD. By contrast, CbD was designed with the primary purpose to apply to arbitrary probabilistic systems. On the other hand, CbD is confined to probabilistic systems, with all deterministic systems being trivially noncontextual. By contrast, STC offers contextuality analysis of systems that are inherently deterministic, finding there interesting cases of contextual systems. These comparative characterizations are incomplete, and I will elaborate them as we proceed.

1.2. The two theories are represented by numerous publications, and their representative exposition can be found in Abramsky et al. (2014, 2017a, b), Abramsky and Brandenburger (2011), Abramsky (2014) for STC, and Dzhafarov et al. (2017), Kujala and Dzhafarov (2016, 2019), Dzhafarov and Kujala (2017a, 2020) for CbD. Familiarity with the two theories would be helpful in reading this paper, but all relevant results and concepts, including those mentioned in this introduction, will be explained below. The reader is to be warned, however, that I will not endeavor here what might be a worthy project for future work: to systematically present the two theories using their native languages and establish correspondences and differences between them. Rather, as the title of the paper suggests, STC will only be presented here from the point of view of CbD, in the CbD language.

1.3. I will show that by doing this one can easily extend STC, including the important notion of (non)contextual fraction, to apply to inconsistently connected systems. More precisely, the CbD language allows one to redefine inconsistently connected systems into consistently connected ones, so that STC can apply to them.

1.4. I will argue that there is a conceptual problem in applying original STC, with its commitment to strong consistent connectedness, to inherently deterministic systems, such as systems of statements with definitive truth values. A way of dealing with this issue I propose is to distinguish completely specified deterministic systems (that are always noncontextual) and systems with multiple possible deterministic realizations. In the latter case, by assigning Bayesian priors to these realizations one renders such systems quasi-probabilistic, disentangling thereby contextuality and inconsistent connectedness. This construction allows one to extend CbD to the contextuality analysis of deterministic systems in the spirit of STC.

15.2 CbD: Conceptual Set-up

2.1. In CbD, the object of contextuality analysis is a *system of random variables* representing what generically can be called measurements. Depending on application, the random variables may describe outputs of physical measurements, responses to inputs, answers to questions, etc. The random variables in a system are assumed to be *dichotomous* (say, $+1/-1$), for reasons discussed in Sects. 5.5 and 5.6. This means that any measurement is presented as simultaneous answers to a set of Yes/No questions, such as "Is the measured value less than 5?".

2.2. The question a random variable answers is referred to as the *content* of the random variable. A set of random variables forms a system if they are labeled both by their contents and by their *contexts*. A context of a variable includes all conditions recorded *together* with this variable, where "together" can mean any empirical procedure by which the observed values of the random variables and conditions within the context are paired. For instance, if a random variable R is recorded together with two other random variables, this fact is part of the context of R. If the order of recording these random variables is itself systematically recorded, this is part of the context of R too.

2.3. The terminology is: if content q is measured in context c, which is written

$$q \prec c,$$

the outcome of the measurement is the (dichotomous) random variable R_q^c.

2.4. If the sets of the contents and contexts are finite (*as we are going to assume throughout this paper*), the system can be represented by a matrix like this:

$$\begin{array}{|c|c|c|c||c|}
\hline
R_1^1 & R_2^1 & & & c^1 \\
\hline
 & R_2^2 & R_3^2 & R_4^2 & c^2 \\
\hline
R_1^3 & & R_3^3 & & c^3 \\
\hline
R_1^4 & & & R_4^4 & c^4 \\
\hline
R_1^5 & R_2^5 & R_3^5 & & c^5 \\
\hline
\hline
q_1 & q_2 & q_3 & q_4 & \mathcal{R} \\
\hline
\end{array}\;. \tag{15.1}$$

This system has four contents variously measured in five contexts, and R_i^j is the abbreviation for $R_{q=q_i}^{c=c^j}$. I will use this system as an example throughout the paper.

2.5. The following are the two basic properties of any system.

CbD1: All random variables sharing a context are jointly distributed (i.e., they can be presented as measurable functions on one and the same probability space).

CbD2: Any two random variables in different contexts are stochastically unrelated, i.e., they are defined on distinct probability spaces.

2.6. If any two random variables in the system that have the same content are identically distributed, writing this as

$$R_q^c \sim R_q^{c'},$$

the system is called (*simply*) *consistently connected*. CbD does not assume this property: generally, a system can be *inconsistently connected* (and this term is often used in the meaning of "not necessarily consistently connected").

2.7. The following is the main definition in CbD.

Definition 1 A system is *noncontextual* if it has a *multimaximally connected coupling*. Otherwise it is *contextual*.

The terminology in the definition is deciphered in the next three sections (2.8–2.10)

2.8. Let

$$\mathcal{R} = \left\{ R_q^c : c \in C, q \in Q, q \prec c \right\} \tag{15.2}$$

be a system (with Q and C being sets of contents and contexts, respectively). A *coupling* of this system is a correspondingly labeled set of random variables

$$\mathcal{S} = \left\{ S_q^c : c \in C, q \in Q, q \prec c \right\}, \tag{15.3}$$

such that

(a) its components are jointly distributed;
(b) its context-wise marginals are distributed in the same way as the corresponding subsets of the original system.

That is, for any $c \in C$,

$$\left\{ R_q^c : q \in Q, q \prec c \right\} \sim \left\{ S_q^c : q \in Q, q \prec c \right\}. \tag{15.4}$$

2.9. For instance, the jointly distributed random variables

$$
\begin{array}{|c|c|c|c||c|}
\hline
S_1^1 & S_2^1 & & & c^1 \\
\hline
 & S_2^2 & S_3^2 & S_4^2 & c^2 \\
\hline
S_1^3 & & S_3^3 & & c^3 \\
\hline
S_1^4 & & & S_4^4 & c^4 \\
\hline
S_1^5 & S_2^5 & S_3^5 & & c^5 \\
\hline
\hline
q_1 & q_2 & q_3 & q_4 & \mathcal{S} \\
\hline
\end{array}
\tag{15.5}
$$

form a coupling of system $\mathcal{R}$ in (15.1) if the corresponding row-wise distributions in $\mathcal{S}$ and $\mathcal{R}$ coincide.

2.10. Now, the coupling $\mathcal{S}$ is *multimaximally connected* if, for any two content-sharing random variables $S_q^c, S_q^{c'}$ it contains, the probability of $S_q^c = S_q^{c'}$ is maximal among all possible couplings of $\mathcal{R}$. Equivalently, the probability of $S_q^c = S_q^{c'}$ in a multimaximal coupling is maximal given the marginal distributions of $S_q^c \sim R_q^c$ and $S_q^{c'} \sim R_q^{c'}$. For system (15.1) with coupling (15.5), this means the simultaneous maximization of the probabilities of

$$S_1^1 = S_1^3, S_1^1 = S_1^4, S_1^3 = S_1^4, S_3^2 = S_3^3, \ldots$$

2.11. Two couplings of the same system that have the same distribution are considered equivalent and are not distinguished. In other words, the domain probability space of a coupling with a given distribution can be chosen arbitrarily. The most economic choice of the domain space is the distribution itself, with all random variables being defined as the componentwise projections of the identity function on this space. That is, (15.3) can be viewed as the identity function on the probability space

$$(X_S, \Sigma_S, \mu_S) = \left\{ \{-1, 1\}^{|\prec|}, 2^{\{-1,1\}^{|\prec|}}, \mu_S : 2^{\{-1,1\}^{|\prec|}} \to [0, 1] \right\}, \tag{15.6}$$

where $|\prec|$ is the cardinality of the relation $\prec$, and μ_S is the probability measure defined by the (joint) probability mass function

$$\Pr\left[S_q^c = s_q^c : c \in C, q \in Q, q \prec c \right],$$

with $s_q^c = -1, 1$. For uncountably infinite Q and/or C the definition should be modified in well-known ways, but we have agreed to consider finite Q, C only.

2.12. Note that any jointly distributed set of random variables is a random variable. So the set of all variables within context c can be written as

$$\mathcal{R}^c = \left\{R_q^c : q \in Q, q \prec c\right\} = R^c,$$

and a coupling $\mathcal{S}$ in (15.3) can be written as a random variable S.

15.3 STC in the CbD Language

3.1. The language of STC is very different, and it manages to avoid even mentioning random variables. Thus, our example system (15.1) could be represented in STC as

$$\begin{array}{|c|c|c|c|c|c|c|c||c|}
\hline
\multicolumn{2}{|c|}{p^1_{\bar{1}\bar{1}}} & \multicolumn{2}{c|}{p^1_{\bar{1}1}} & \multicolumn{2}{c|}{p^1_{1\bar{1}}} & \multicolumn{2}{c||}{p^1_{11}} & (q_1, q_2) \\
\hline
p^2_{\bar{1}\bar{1}\bar{1}} & p^2_{\bar{1}\bar{1}1} & p^2_{\bar{1}1\bar{1}} & p^2_{\bar{1}11} & p^2_{1\bar{1}\bar{1}} & p^2_{1\bar{1}1} & p^2_{11\bar{1}} & p^2_{111} & (q_2, q_3, q_4) \\
\hline
\multicolumn{2}{|c|}{p^3_{\bar{1}\bar{1}}} & \multicolumn{2}{c|}{p^3_{\bar{1}1}} & \multicolumn{2}{c|}{p^3_{1\bar{1}}} & \multicolumn{2}{c||}{p^3_{11}} & (q_1, q_3) \\
\hline
\multicolumn{2}{|c|}{p^4_{\bar{1}\bar{1}}} & \multicolumn{2}{c|}{p^4_{\bar{1}1}} & \multicolumn{2}{c|}{p^4_{1\bar{1}}} & \multicolumn{2}{c||}{p^4_{11}} & (q_1, q_4) \\
\hline
p^5_{\bar{1}\bar{1}\bar{1}} & p^5_{\bar{1}\bar{1}1} & p^5_{\bar{1}1\bar{1}} & p^5_{\bar{1}11} & p^5_{1\bar{1}\bar{1}} & p^5_{1\bar{1}1} & p^5_{11\bar{1}} & p^5_{111} & (q_1, q_2, q_3) \\
\hline
\end{array}\;, \tag{15.7}$$

where p^i is the probability distribution in the ith context, and the subscripts represent combinations of values 1 and $-1 \equiv \bar{1}$. The values $(-1, -1)$ in $p^1_{\bar{1}\bar{1}}$ are not values of the contents (q_1, q_2) (called in STC "measurements", "observables", or simply "variables"). They are values of the random variables not being mentioned.

3.2. The reason this does not lead to complications is that STC is committed to requiring that a system of random variable amenable to contextuality analysis be *strongly consistently connected*. This means that for any pair of contexts c, c', we have

$$\left\{R_q^c : q \prec c, q \prec c'\right\} \sim \left\{R_q^{c'} : q \prec c, q \prec c'\right\}, \tag{15.8}$$

i.e., the joint distributions for identically subscripted random variables are identical. Thus, in our example (15.1),

$$\left\{R_1^5, R_2^5\right\} \sim \left\{R_1^1, R_2^1\right\}$$

and

$$\left\{R_2^5, R_3^5\right\} \sim \left\{R_2^2, R_3^2\right\}.$$

Abramsky and colleagues consider this property fundamental (see, e.g., Abramsky et al., 2014), and it is indeed indispensable if one is to use the language of sheafs.

3.3. It should be noted, however, that in quantum-physical experiments even simple consistent connectedness (which is obviously implied by strong one) is routinely violated (see, e.g., Kujala et al., 2015, and for more references, Dzhafarov, 2019). In some non-physical applications, notably in human behavior, inconsistent connectedness is a universal rule (Cervantes & Dzhafarov, 2018; Dzhafarov et al., 2016).

3.4. Note that random variables may, in particular, be deterministic, i.e. they may attain a single value with probability 1. If the requirement of consistent connectedness is applied to such variables, then it translates into any two content-sharing variables being equal to one and the same value,

$$R_q^c \equiv r \Longleftrightarrow R_{q'}^c \equiv r. \tag{15.9}$$

We will see later, in Sect. 15.6, that this constraint creates a difficulty when STC deals with deterministic systems.

3.5. The property of strong consistent connectedness allows Abramsky and colleagues to define a context simply by the set of contents measured together. Thus, in the example (15.1), c^1 would be defined as the context in which we measure $\{q_1, q_2\}$, c^2 as the context in which we measure $\{q_2, q_3, q_4\}$, etc. In STC, there cannot be distinct contexts with the same set of random variables in them, because their joint distributions would have to be the same.

3.6. To illustrate the effect of the restriction imposed by STC on the CbD framework, consider the system

$$\begin{array}{|c|c||c|}\hline R_1^1 & R_2^1 & c^1 \\ \hline R_1^2 & R_2^2 & c^2 \\ \hline\hline q_1 & q_2 & \mathcal{C}_2 \\ \hline \end{array}\ . \tag{15.10}$$

In STC, it can only represent the same context repeated twice, which makes the system trivially noncontextual. By contrast, in CbD, this so-called cyclic system of rank 2 is the smallest nontrivial system, one that can be contextual or noncontextual depending on the distributions involved (Kujala & Dzhafarov, 2016). In particular, the difference between the two contexts may be the order in which the two contents are measured ($q_1 \to q_2$ and $q_2 \to q_1$) (Dzhafarov et al., 2016).

3.7. The assumption of consistent connectedness (not necessarily strong one) simplifies the definition of (non)contextuality.

Definition 2 [Equivalent of STC definition] A consistently connected system is noncontextual if it has an *identically connected coupling*. Otherwise it is *contextual*.

A coupling $\mathcal{S}$ is identically connected if, for any q, c, c' such that $q \prec c, c'$,

$$\Pr\left[S_q^c = S_q^{c'}\right] = 1. \tag{15.11}$$

3.8. This is, clearly, a special case of a multimaximal coupling: the maximal probability of $S_q^c = S_q^{c'}$ in such a coupling is 1 if and only if $R_q^c \sim R_q^{c'}$. For all practical purposes, it allows one to think of the random variables R_q^c as "context-independent," and many authors would even denote them as R_q. This is, however, a dubious practice that leads to a logical contradiction (Dzhafarov & Kujala, 2017a; Dzhafarov, 2019). STC avoids this difficulty by not mentioning random variables at all, and systematically labeling probability distributions by their contexts (Dzhafarov, 2019).

3.9. Definition 2 does not explicitly require that consistent connectedness of the system be strong. This, however, makes little difference if one is only interested in determining whether a system is contextual. It is easy to show the following.

Theorem 3 *A simply consistently connected system that is not strongly consistently connected is contextual.*

Proof In an identically connected coupling of the system, $\Pr\left[S_q^c = S_q^{c'}\right] = 1$ and $\Pr\left[S_{q'}^c = S_{q'}^{c'}\right] = 1$ imply

$$\Pr\left[\left(S_q^c, S_{q'}^c\right) = \left(S_q^{c'}, S_{q'}^{c'}\right)\right] = 1,$$

which is only possible if $\left(R_q^c, R_{q'}^c\right) \sim \left(R_q^{c'}, R_{q'}^{c'}\right)$. □

One can think of the strong consistent connectedness requirement in STC as a provision excluding this "guaranteed" contextuality from consideration.

15.4 Contextual Fraction

4.1. There are several possible ways of measuring the degree of contextuality in CbD (Kujala and Dzhafarov, 2019), but in STC the measure of choice is contextual fraction. I present it, as everything else in this paper, in the language of CbD. We need a few general probabilistic notions first.

4.2. An *incomplete probability space*, or *α-probability space* (where $0 \leq \alpha \leq 1$) is defined as a measure space (X, Σ, μ) with $\mu(X) = \alpha$. The meaning of the components of the space (set X, sigma-algebra Σ, and sigma-additive measure μ) is standard. Any measurable function Z defined on this space is called an *incomplete (random) variable*, or a *(random) α-variable*. It is essentially an ordinary random variable: for any measurable set D in the codomain of Z, $\Pr[Z \in D]$ is defined as $\mu\left(Z^{-1}(D)\right)$ and referred to as the probability of Z falling in D. The only difference is that if $D = Z(X)$ then $\Pr[Z \in D] = \alpha$. The rest of the concepts related to α-variables (e.g., their joint distribution) are the same as for true random variables (with $\alpha = 1$). In Feller's classical monograph (Feller, 1968) incomplete random variables are called "defective". Other terms, such as "improper", are used too.

4.3. Let $0 \le \alpha \le \beta \le 1$, and let Z_α and Z_β be an α-variable and a β-variable, respectively, with the same codomain. We say that Z_α is *majorized* by Z_β, if for every measurable set D in their common codomain,

$$\Pr[Z_\alpha \in D] \le \Pr\left[Z_\beta \in D\right]. \tag{15.12}$$

We write then

$$Z_\alpha \lesssim Z_\beta. \tag{15.13}$$

4.4. An *incomplete (or α-) coupling* of a system of random variables $\mathcal{R}$ is a correspondingly indexed set

$${}^\alpha\mathcal{S} = \left\{{}^\alpha S_q^c : c \in C, q \in Q, q \prec c\right\} \tag{15.14}$$

of jointly distributed α-variables such that, for any context $c \in C$,

$$\left\{{}^\alpha S_q^c : q \in Q, q \prec c\right\} \lesssim \left\{R_q^c : q \in Q, q \prec c\right\}. \tag{15.15}$$

An α-coupling is identically connected if

$$\Pr\left[{}^\alpha S_q^c \ne {}^\alpha S_q^{c'}\right] = 0, \tag{15.16}$$

for any $q \prec c, c'$. Clearly, an identically connected α-coupling may exist only for a consistently connected system, and the latter is noncontextual if and only if it has an identically connected 1-coupling.

4.5. The following theorem allows one to introduce a measure of contextuality.

Theorem 4 *Any consistently connected system has an identically connected $\alpha_{\max}$-coupling ($0 \le \alpha_{\max} \le 1$), such that the system has no identically connected α-couplings with $\alpha > \alpha_{\max}$.*

This property can be proved by employing the linear programming representation of the relation between an α-coupling and the context-wise distributions in the system. This representation is essentially the same as one routinely used in both STC (Abramsky et al., 2017b; Abramsky and Brandenburger, 2011) and CbD (Kujala and Dzhafarov, 2019; Dzhafarov et al., 2017).

Proof We represent the system $\mathcal{R}$ and an α-coupling by probability vectors $\mathbf{r}$ and $\mathbf{s}$, respectively, such that

$$\mathbf{Bs} \leq \mathbf{r}(\text{componentwise}), \tag{15.17}$$

with the following meaning of the terms. The entries of $\mathbf{r}$ are context-wise joint probabilities

$$\Pr\left[R_q^c = r_q : q \in Q, q \prec c\right], \tag{15.18}$$

across all $c \in C$ and all combinations of $r_q = +1/-1$. The entries of $\mathbf{z}$ are joint probabilities

$$\Pr\left[{}^{\alpha}S_q^c = s_q : c \in C, q \in Q, q \prec c\right], \tag{15.19}$$

across all combinations of $s_q = +1/-1$. $\mathbf{B}$ is a Boolean matrix ("incidence matrix" in Abramsky and Brandenburger, 2011) with rows indexed by the values of c, and, for each c, by the combinations of r_q-values in (15.18). Its columns are indexed by the combinations of s_q-values in (15.19). A cell of $\mathbf{B}$ is filled with 1 if its s_q-combination contains its r_q-combination for the corresponding q-values, otherwise the cell is filled with 0. The set of all α-couplings of $\mathcal{R}$ is represented by the (obviously nonempty) polytope

$$\mathbb{Z} = \{\mathbf{s} : \mathbf{Bs} \leq \mathbf{r}, \mathbf{s} \geq 0, \mathbf{1} \cdot \mathbf{s} \leq 1\}. \tag{15.20}$$

Every linear functional, including $\mathbf{1} \cdot \mathbf{s}$, attains its extrema within this polytope, and the maximum value of $\mathbf{1} \cdot \mathbf{s}$ is taken as $\alpha_{\max}$. □

4.6. The quantity $\alpha_{\max}$ is called *noncontextual fraction*, and $1 - \alpha_{\max}$ is called *contextual fraction*. It is easy to see that if $\alpha_{\max} = 1$, then $\mathbf{Bs} = \mathbf{r}$, and the system is noncontextual. Otherwise it is contextual, and the contextual fraction is a natural measure of the degree of contextuality.

4.7. Abramsky and colleagues single out the case $\alpha_{\max} = 0$, calling such systems *strongly contextual*. In strongly contextual systems, every possible combination of s_q-values has the probability of zero. This is the situation one encounters with the Kochen-Specker systems (Kochen & Specker, 1967) and with the Popescu-Rohrlich boxes (Popescu & Rohrlich, 1994).

15.5 Consistified Systems

5.1. Can STC, and contextual fraction in particular, be generalized to arbitrary, generally inconsistently connected, systems? It turns out this can be done by a simple procedure that converts an inconsistently connected system $\mathcal{R}$ into a *contextually equivalent* consistently connected one, $\mathcal{R}^{\ddagger}$. Contextual equivalence means that $\mathcal{R}$ is contextual if and only if so is $\mathcal{R}^{\ddagger}$, and that if $\mathcal{R}$ is consistently connected, then $\mathcal{R}^{\ddagger}$

has the same value of contextual fraction.

5.2. The discussion of the procedure of *consistification* is helped by two additional CbD terms. Let us call all random variables sharing a context a *bunch*,

$$\mathcal{R}^c = \left\{ R_q^c : q \in Q, q \prec c \right\} = R^c, \tag{15.21}$$

and all random variables sharing a content a *connection*,

$$\mathcal{R}_q = \left\{ R_q^c : c \in C, q \prec c \right\}. \tag{15.22}$$

The terminology is intuitive: a bunch is jointly distributed, and different bunches are disjoint, but the fact that some random variables in different contexts have the same content creates connections between the bunches.

5.3. Within a connection (15.22) the random variables are stochastically unrelated, but they can be coupled by

$$\mathcal{T}_q = \left\{ T_q^c : c \in C, q \prec c \right\} = T_q, \tag{15.23}$$

and among all such couplings one can seek a *multimaximal coupling*, one that maximizes the probabilities for all equalities

$$T_q^c = T_q^{c'}, q \prec c, c'. \tag{15.24}$$

5.4. If, as we have agreed, all random variables in the system are dichotomous, then we have the following result, proved in Dzhafarov and Kujala (2017b).

Theorem 5 *Any connection has one and only one multimaximal coupling. $\mathcal{T}_q$ is a multimaximal coupling of the connection $\mathcal{R}_q$ if and only if any subset of $\mathcal{T}_q$ is a maximal coupling of the corresponding subset of $\mathcal{R}_q$.*

The second statement means that, for any part

$$\mathcal{R}'_q = \left\{ R_q^c : c \in \{c_1, \ldots, c_k\} \subseteq C, q \prec c \right\},$$

of the connection (15.22), the event

$$T_q^{c_1} = \ldots = T_q^{c_k}$$

has the maximal possible probability among all possible couplings of $\mathcal{R}'_q$ (or, equivalently, given the marginal distributions of $T_q^{c_i} \sim R_q^{c_i}$, $i = 1, \ldots, k$).

5.5. The theorem above is one of the two reasons why CbD subjects any set of measurements to dichotomization before making it a system of random variables amenable to contextuality analysis. The consistification procedure to be described is based on the possibility to find unique multimaximal couplings for all connections.

5.6. For completeness, I should mention the second, and main, reason for the dichotomization of all random variables: it prevents the otherwise possible situation when coarse-graining of the random variables in a noncontextual system makes it contextual (Dzhafarov et al., 2017). Clearly, a good theory of contextuality should not have this property. Note that dichotomization does not lose any information extractable from random variables before they are dichotomized. It does not even increase the size of a system, if size is measured by the cardinality of the supports of the system's bunches.

5.7. The idea of consistification is to treat the multimaximal couplings of connections as if they were additional bunches. This is implicit in any CbD-based algorithm for establishing or measuring contextuality (Kujala & Dzhafarov, 2016, 2019; Kujala et al., 2015). Explicitly, however, it was first described by Amaral et al. (2018). However, Amaral and coauthors use maximal couplings instead of the multimaximal ones (as we did in the older version of CbD, e.g., in Dzhafarov and Kujala, 2016), and they allow for multivalued variables. The difference between the two types of couplings of a set $\{X_1, \dots, X_n\}$ is that in the multimaximal coupling $\{Y_1, \dots, Y_n\}$ we maximize probabilities of all equalities $Y_i = Y_j$ (whence it follows, by Theorem 5, that we also maximize the probability of $Y_{i_1} = Y_{i_2} = \cdots = Y_{i_k}$ for any subset of $\{Y_1, \dots, Y_n\}$), whereas a maximal coupling $\{Z_1, \dots, Z_n\}$ only maximizes the probability of the single chain equality $Z_1 = Z_2 = \cdots = Z_n$. Maximal couplings generally are not unique, even for dichotomous variables (if there are more than two of them). Since measures of (non)contextuality generally depend on what couplings are being used, the approach advocated in Amaral et al. (2018) faces the problem of choice. In addition, a system declared noncontextual using maximal rather than multimaximal couplings may have contextual subsystems, obtained by dropping some of the variables. I consider this possibility highly undesirable for a theory of contextuality.

5.8. The particular consistification scheme presented below is an elaboration of one described to me by Janne Kujala (personal communication, November 2018).

5.9. Given an arbitrary system $\mathcal{R}$, the new system $\mathcal{R}^\ddagger$ has a set $Q^\ddagger$ of "new" contents, a set $C^\ddagger$ of "new" contexts, and a "new" is-measured-in relation $\prec^\ddagger$. The corresponding constructs in the original system, Q, C, and $\prec$, will be called "old".

5.10. For each random variable R_j^i in $\mathcal{R}$ we form a new content, denoted q_j^i. The set of all new contents is

$$Q^{\ddagger} = \left\{q_j^i : c^i \in C, q_j \in Q, q_j \prec c^i\right\}. \tag{15.25}$$

The number of the new contents is the cardinality of $\prec$, which cannot exceed $|C \times Q|$.

5.11. New contexts are formed as the set

$$C^{\ddagger} = C \sqcup Q,$$

and their number is $|C| + |Q|$.

5.12. The new is-measured-in relation is

$$\prec^{\ddagger} = \left\{\left(q_j^i, c^i\right) : c^i \in C, q_j \in Q, q_j \prec c^i\right\} \sqcup \left\{\left(q_j^i, q_j\right) : c^i \in C, q_j \in Q, q_j \prec c^i\right\}. \tag{15.26}$$

That is, a new content q_j^i is measured in the new contexts c^i and q_j only.

5.13. Each $\left(q_j^i, c^i\right)$-cell contains the old random variables R_j^i. The new bunch

$$R^i = \left\{R_j^i : q_j^i \in Q^{\ddagger}, q_j^i \prec^{\ddagger} c^i\right\} \tag{15.27}$$

coincides with the old bunch

$$R^i = \left\{R_j^i : q_j \in Q, q_j \prec c^i\right\}. \tag{15.28}$$

5.14. Each $\left(q_j^i, q_j\right)$-cell contains a new random variable V_j^i whose distribution is the same as that of R_j^i. The bunch

$$V^j = \left\{V_j^i : q_j^i \in Q^{\ddagger}, q_j^i \prec^{\ddagger} q_j\right\} \tag{15.29}$$

is the multimaximal coupling of the old connection

$$\mathcal{R}_j = \left\{R_j^i : c^i \in C, q_j \prec c^i\right\}. \tag{15.30}$$

5.15. Using our examples (15.1) and (15.10), the corresponding consistified systems are

R_1^1	R_2^1											c^1
		R_2^2	R_3^2	R_4^2								c^2
					R_1^3	R_3^3						c^3
							R_1^4	R_4^4				c^4
									R_1^5	R_2^5	R_3^5	c^5
V_1^1					V_1^3		V_1^4		V_4^4			q_1
	V_2^1	V_2^2								V_2^5		q_2
			V_3^2			V_3^3					V_3^5	q_3
				V_4^2				V_4^4				q_4
q_1^1	q_2^1	q_2^2	q_3^2	q_4^2	q_1^3	q_3^3	q_1^4	q_4^4	q_1^5	q_2^5	q_3^5	$\mathcal{A}^\ddagger$

(15.31)

and

R_1^1	R_2^1			c^1
		R_1^2	R_2^2	c^2
V_1^1		V_1^2		q_1
	V_2^1		V_2^2	q_2
q_1^1	q_2^1	q_1^2	q_2^2	$\mathcal{C}_2^\ddagger$

. (15.32)

5.16. Note the following properties of all consistified systems.

1. Bunches corresponding to different old contexts, c^i, $c^{i'}$, are disjoint.
2. Bunches corresponding to different old contents, q_j, $q_{j'}$, are disjoint.
3. A bunch corresponding to an old content, q_j, and a bunch corresponding to an old context, c^i, have at most one connection between them, $\left\{R_j^i, V_j^i\right\}$.
4. The connection corresponding to any new content q_j^i contains precisely two random variables, R_j^i and V_j^i, with the same distribution.

5.17. Property 3 above means that in a consistified system the notions of simple consistent connectedness and strong consistent connectedness coincide. As mentioned earlier, in Sect. 3.6, system $\mathcal{C}_2$ in (15.10) will only be considered in STC if the two bunches $\{R_1^1, R_2^1\}$ and $\{R_1^2, R_2^2\}$ are identically distributed, which would make this system trivial. By contrast, if one adopts the CbD-based consistification of $\mathcal{C}_2$, in (15.32), its STC analysis will be the same as CbD's.

5.18. The following fact ensures that the generalization of the contextual fraction coincides with the original one in the case of consistently connected systems.

Theorem 6 *If a system $\mathcal{R}$ is consistently connected, then its contextual fraction is the same as that of $\mathcal{R}^{\ddagger}$. (In particular, $\mathcal{R}$ is contextual if and only if so is $\mathcal{R}^{\ddagger}$.)*

Proof Immediately follows from the observation that any state of an α-coupling in which the values corresponding to R_q^c and V_q^c are different has the probability zero. □

5.19. For completeness, I formulate the following as a formal statement. Recall the definition of contextual equivalence in Sect. 5.1.

Theorem 7 *Any system $\mathcal{R}$ is contextually equivalent to its consistification $\mathcal{R}^{\ddagger}$.*

Proof Follows from the previous theorem, and the obvious fact that $\mathcal{R}$ and $\mathcal{R}^{\ddagger}$ have the same linear programming representation (see Kujala and Dzhafarov, 2019, for a detailed description of the latter). □

15.6 Deterministic Systems

6.1. Deterministic systems can be viewed as systems of random variables whose distributions attain specific values with probability 1. In CbD, therefore, they are treated as a special case of systems of random variables, with the following general result.

Theorem 8 *Any deterministic system is noncontextual.*

Proof A deterministic system

$$\mathcal{R} = \left\{R_q^c \equiv r_q^c : c \in C, q \in Q, q \prec c\right\},$$

where $\equiv$ means equality with probability 1, has a single overall coupling,

$$\mathcal{S} = \left\{S_q^c \equiv r_q^c : c \in C, q \in Q, q \prec c\right\},$$

with all S_q^c defined on an arbitrary probability space. Since $\left\{S_q^c \equiv r_q^c, S_q^{c'} \equiv r_q^{c'}\right\}$ is the only coupling of $\left\{R_q^c \equiv r_q^c, R_q^{c'} \equiv r_q^{c'}\right\}$, the probability of $S_q^c = S_q^{c'}$ (0 or 1) is maximal possible, whence $\mathcal{S}$ is multimaximally connected. □

6.2. This simple observation seems to put CbD at odds with STC, where the theoretical ideas formulated in algebraic and topological terms are not restricted to random variables. Consider, e.g., two deterministic systems that have the same $\prec$-format as the system $\mathcal{C}_2$ in example (15.10):

$$
\begin{array}{|c|c||c|}
\hline
R_1^1 \equiv 1 & R_2^1 \equiv -1 & c^1 \\
\hline
R_1^2 \equiv 1 & R_2^2 \equiv -1 & c^2 \\
\hline
\hline
q_1 & q_2 & \mathcal{C}_{2.1} \\
\hline
\end{array}
\quad \text{and} \quad
\begin{array}{|c|c||c|}
\hline
R_1^1 \equiv 1 & R_2^1 \equiv -1 & c^1 \\
\hline
R_1^2 \equiv 1 & R_2^2 \equiv 1 & c^2 \\
\hline
\hline
q_1 & q_2 & \mathcal{C}_{2.2} \\
\hline
\end{array}\,. \tag{15.33}
$$

System $\mathcal{C}_{2.1}$ is consistently connected, which in a deterministic system means it is strongly consistently connected. It is therefore trivially noncontextual. System $\mathcal{C}_{2.2}$ is inconsistently connected. Strictly speaking, therefore, the original STC analysis should not be applicable to this system, as it violates the fundamental assumption underlying STC.

6.3. If we use the extended version of STC, with the help of consistification, we get

$$
\begin{array}{|c|c|c|c||c|}
\hline
R_1^1 \equiv 1 & R_2^1 \equiv -1 & & & c^1 \\
\hline
 & & R_1^2 \equiv 1 & R_2^2 \equiv 1 & c^2 \\
\hline
V_1^1 \equiv 1 & & V_1^2 \equiv 1 & & q_1 \\
\hline
 & V_2^1 \equiv -1 & & V_2^2 \equiv 1 & q_2 \\
\hline
\hline
q_1^1 & q_2^1 & q_1^2 & q_2^2 & \mathcal{C}_{2.2}^{\ddagger} \\
\hline
\end{array}\,. \tag{15.34}
$$

This system is trivially noncontextual by the STC/CbD definition.

6.4. This reasoning would apply to any deterministic system: if it is consistently connected (or consistified), it is trivially noncontextual, and if it is inconsistently connected, STC should place it outside its sphere of applicability (or consistify it). In other words, STC with consistification and CbD treat deterministic systems identically (finding them noncontextual). There is, of course, a simple way out: to complement STC with the additional stipulation that all inconsistently connected systems are contextual. STC would then have to allow for contextual systems whose degree of contextuality cannot be measured by contextual fraction. I do not think this simple way out is intellectually satisfactory.

6.5. Here is a good place to mention that CbD treats inconsistent connectedness and contextuality as fundamentally different concepts. Inconsistent connectedness, i.e. the difference in the distributions of R_q^c and $R_q^{c'}$, is interpreted as the result of direct influences of the contexts upon the measurements. In the case of physical systems, one can say that some elements of the contexts c and c' differently affect (in the causal sense) the measurement of the content q. In quantum physics this is reflected by such notions as "signaling" or (a better term) "disturbance". Contextuality, by contrast, is non-causal, and reflects the differences between random variables R_q^c and $R_q^{c'}$ that are above and beyond the differences in their distributions.

6.6. This interpretation is philosophically based on the *no-conspiracy principle* (Cervantes & Dzhafarov, 2018), according to which in "not-precariously-unstable" and "not-deliberately-contrived" systems, no differences in the direct influences exerted by the elements of context are hidden. Being hidden means that these differences are present but are not reflected in the differences of the distributions. For instance, if R_q^c and $R_q^{c'}$ attain values 1 and -1 with probability $\frac{1}{2}$ each, and if c' by some causal mechanism reverses (multiplies by -1) each value of $R_q^{c'}$, then this influence will be hidden, as it will not affect the distribution of $R_q^{c'}$. The no-conspiracy principle says this should not be expected to happen, and if it does, should be expected to disappear by slight modifications of the experimental set-up. The principle is closely related to the "no-fine-tuning" principle advocated by Cavalcanti (2018) (see a detailed analysis of these principles by Jones, 2019).

6.7. With this in mind, let us consider an especially elegant application of STC to an inherently deterministic system, described in Abramsky et al. (2017a). This is a system whose contents are statements referencing each other's truth value and forming a version of the Liar antinomy. I will consider the version with three statements, although any larger number will be analyzed similarly:

$$\begin{array}{|c|c|c||c|}
\hline
R_1^1 & R_2^1 & & c^1 \\
\hline
 & R_2^2 & R_3^2 & c^2 \\
\hline
R_1^3 & & R_3^3 & c^3 \\
\hline
\hline
q_1 = \text{"}q_2\, is\, true\text{"} & q_2 = \text{"}q_3\, is\, true\text{"} & q_3 = \text{"}q_1 \text{ is false"} & \mathcal{L}_3 \\
\hline
\end{array}. \tag{15.35}$$

The contexts combine the statements one of which references the other, and the R_q^c is the truth value (1 or -1) of statement q in context c.

6.8. We could have considered the smaller system

R_1^1	R_2^1	c^1
R_1^2	R_2^2	c^2
$q_1 =$ "q_2 is true"	$q_2 =$ "q_1 is false"	$\mathcal{L}_2$

, (15.36)

representing a more familiar classical form of the antinomy, but the contexts in $\mathcal{L}_3$ are easier to interpret, as the direction of inference there need not be specified. The interpretation is even more complicated with the classical form $q =$ "q is false", although the reasoning below is still applicable.

6.9. We can posit that each statement in a given context should have one definitive truth value, and this makes $\mathcal{L}_3$ a deterministic system. In Abramsky et al. (2017a) this system is characterized as strongly contextual, based on the impossibility to assign the truth values in a context-independent way. However, we know that the original version of STC is predicated on the assumption of strong consistent connectedness, whereas any deterministic realization of $\mathcal{L}_3$ (precisely because no context-independent assignment of truth values exists) is inconsistently connected. Consider one of the eight such deterministic versions, corresponding to the usual conceptualization of the Liar Antinomy:

$R_1^1 \equiv 1$	$R_2^1 \equiv 1$		c^1
	$R_2^2 \equiv 1$	$R_3^2 \equiv 1$	c^2
$R_1^3 \equiv -1$		$R_3^3 \equiv 1$	c^3
$q_1 =$ "q_2 is true"	$q_2 =$ "q_3 is true"	$q_3 =$ "q_1 is false"	$\mathcal{L}_{3.1}$

. (15.37)

The arguments related to system $\mathcal{C}_{2.2}$ in (15.33) apply here fully. I see no reasonable way a system like $\mathcal{L}_{3.1}$ can be treated as contextual, either in CbD or in STC.

6.10. There is, however, another way of looking at system $\mathcal{L}_3$. It seems to be very much in the spirit of how it is treated in Abramsky et al. (2017a). Moreover, it corresponds to the traditional presentation of the Liar antinomy: suppose $R_1^1 \equiv 1$, then it follows that $R_1^1 \equiv -1$; now suppose $R_1^1 \equiv -1$, then it follows that $R_1^1 \equiv 1$. With the stipulation that each statement in a given context should have one definitive truth value, $\mathcal{L}_3$ can indeed be just one of the eight deterministic (and inconsistently connected) systems of which $\mathcal{L}_{3.1}$ is one. However, we do not know which of these eight systems to choose, and we can consider all eight of them as variants of $\mathcal{L}_3$:

1	1		c^1
	1	1	c^2
−1		1	c^3
q_1	q_2	q_3	$\mathcal{L}_{3.1}$

−1	−1		c^1
	−1	−1	c^2
1		−1	c^3
q_1	q_2	q_3	$\mathcal{L}_{3.2}$

−1	−1		c^1
	1	1	c^2
−1		1	c^3
q_1	q_2	q_3	$\mathcal{L}_{3.3}$

etc.

6.11. One can assign Bayesian (or epistemic) probabilities to these possibilities, a natural choice here being to assign them uniformly. This renders the system probabilistic in the epistemic sense, with

$$\left\langle R_q^c \right\rangle_B = 0, \tag{15.38}$$

for all the "epistemically-random" variables (indicated by the subscript B), and

$$\left\langle R_1^1 R_2^1 \right\rangle_B = \left\langle R_2^2 R_3^2 \right\rangle_B = -\left\langle R_3^3 R_1^3 \right\rangle_B = 1. \tag{15.39}$$

This is a Bayesian analogue of a rank 3 cyclic system that is consistently connected and forms a Popescu-Rohrlich box. Its contextuality, both in CbD and STC, is maximal. In particular, when measured by contextual fraction, it is strong ($\alpha_{\max} = 0$), in accordance with how Abramsky and colleagues view it.

6.12. This Bayesian procedure can be applied to any deterministic system with more than one possible deterministic realization. The procedure will render the system quasi-probabilistic and, at least in all the simple cases I can think of, consistently connected and contextual. More work is needed to elaborate this approach.

15.7 Conclusion

7.1. We have seen that STC can be extended to apply to inconsistently connected systems, using CbD-based multimaximal couplings to consistify these systems. We have also seen that the Bayesian rendering of the deterministic systems with multiple possible realizations allows STC to circumvent the difficulty associated with inconsistent connectedness of each of these realizations. It simultaneously extends CbD to such systems and allows CbD to treat them in the spirit of STC, forming thereby another bridge between the two theories.

7.2. Together, the consistification and the Bayesian treatment make STC and CbD essentially coextensive, with a major proviso: one has to agree to represent all measurement outcomes in a system as sets of jointly distributed dichotomous random variables. Dichotomization of a system is always possible, so it is more of a language choice than a restriction of applicability. Dealing only with dichotomous variables allows one to avoid a variety of difficulties (Dzhafarov et al., 2017), but no proof exists that they could not be avoided by other means.

7.3. Finally, nothing in this paper implies that CbD can be replaced with STC, or vice versa. Each of the two theories has its own aims and means. Thus, logical aspects of contextuality, especially in the possibilistic proofs of contextuality, are significantly more salient in STC than CbD, adding to the former's aesthetic elegance. Perhaps the use of the Bayesian/epistemic random variables, as discussed above, might offer CbD a way to "catch up" in this respect. STC in turn might benefit from using the language of random variables for proof purposes. For instance, the fact that the existence of a hidden variable model for a system of random variables is equivalent to the existence of their joint distribution (from which it follows, in particular, that nonlocality is a special case of contextuality) is true almost by definition if the language of random variables is used explicitly. It is a non-trivial, perhaps even surprising fact, however, if one considers the systems of random variables in terms of their distributions only (Fine, 1982; Abramsky & Brandenburger, 2011). It would be good if the equivalences established in this paper helped the two theories to more freely borrow from each other's native languages, follow each other's directions of research, and use each other's proof techniques.

References

Abramsky S., Barbosa R.S., Kishida K., Lal R., & Mansfield S. (2017). *Contextuality, cohomology and paradox*. arXiv:1502.03097.

Abramsky, S., Barbosa, R. S., & Mansfield, S. (2017b). The contextual fraction as a measure of contextuality. *Physical Review Letters,119,* 050504.

Abramsky, S., Brandenburger, A., & Savochkin, A. (2014). No-signalling is equivalent to free choice of measurements. In M. J. Hoban, & B. Coecke (Eds.), *10th International Workshop on Quantum Physics and Logic (QPL 2013) EPTCS 171* (pp. 1–9).

Abramsky, S. (2014). Contextual semantics: From quantum mechanics to logic, databases, constraints, and complexity. *Bulletin of the European Association Theoretical Computer Science, 113*, 137–163.

Abramsky, S., & Brandenburger, A. (2011). The sheaf-theoretic structure of non-locality and contextuality. *New Journal of Physics, 13*, 113036–113075.

Amaral, B., Duarte, C., & Oliveira, R. I. (2018). Necessary conditions for extended noncontextuality in general sets of random variables. *Journal of Mathematical Physics, 59*, 072202.

Cavalcanti, E. G. (2018). Classical causal models for Bell and Kochen-Specker inequality violations require fine-tuning. *Physical Review X, 8*, 021018.

Cervantes, V. H., & Dzhafarov, E. N. (2018). Snow Queen is evil and beautiful: Experimental evidence for probabilistic contextuality in human choices. *Decision, 5*, 193–204.

Dzhafarov, E. N., & Kujala J. V. (2017b). Contextuality-by-Default 2.0: Systems with binary random variables. In J. A. de Barros, B. Coecke & E. Pothos (Eds.) *Lecture Notes in Computer Science* (Vol. 10106, pp. 16–32).

Dzhafarov, E. N., & Kujala, J. V. (2017a). Probabilistic foundations of contextuality. *Fortschritte der Physik,65*, 1–11.

Dzhafarov, E. N., Cervantes, V. H., & Kujala, J. V. (2017). Contextuality in canonical systems of random variables. *Philosophical Transactions of the Royal Society A,375*, 20160389.

Dzhafarov, E.N., Kujala, J.V., & Cervantes, V.H. (2020). Contextuality and noncontextuality measures and generalized Bell inequalities for cyclic systems. *Physical Review A, 2020*, 101, 042118. (Erratum notes in *Physical Review A, 2020, 101*, 069902 and in Physical Review A, 2021,103, 059901).

Dzhafarov, E. N. (2019). On joint distributions, counterfactual values, and hidden variables in understanding contextuality. *Philosophical Transactions of the Royal Society A, 377*, 20190144.

Dzhafarov, E. N., & Kujala, J. V. (2016). Context-content systems of random variables: The Contextuality-by- Default theory. *Journal of Mathematical Psychology, 74*, 11–33.

Dzhafarov, E. N., Zhang, R., & Kujala, J. V. (2016). Is there contextuality in behavioral and social systems? *Philosophical Transactions of the Royal Society A, 374*, 20150099.

Feller, W. (1968). *An introduction to probability theory and its applications* (Vol. 1). NewYork: Wiley.

Fine, A. (1982). Joint distributions, quantum correlations, and commuting observables. *Journal of Mathematical Physics, 23*, 1306–1310.

Jones, M. (2019). Causal and probabilistic approaches to contextuality. *Philosophical Transactions of the Royal Society A, 377*, 20190133.

Kochen, S., & Specker, E. P. (1967). The problem of hidden variables in quantum mechanics. *Journal of Mathematical Physics, 17*, 59–87.

Kujala, J. V., & Dzhafarov, E. N. (2016). Proof of a conjecture on contextuality in cyclic systems with binary variables. *Foundations of Physics, 46*, 282–299.

Kujala, J. V., & Dzhafarov, E. N. (2019). Measures of contextuality and noncontextuality. *Philosophical Transactions of the Royal Society A, 377*, 20190149.

Kujala, J. V., Dzhafarov, E. N., & Larsson, J. Å. (2015). Necessary and sufficient conditions for extended noncontextuality in a broad class of quantum mechanical systems. *Physical Review Letters, 115*, 150401.

Popescu, S., & Rohrlich, D. (1994). Quantum nonlocality as an axiom. *Foundations of Physics, 24*, 379–385.

Chapter 16
Putting Paradoxes to Work: Contextuality in Measurement-Based Quantum Computation

Robert Raussendorf

Abstract We describe a joint cohomological framework for measurement-based quantum computation (MBQC) and the corresponding contextuality proofs. The central object in this framework is an element $[\beta_\Psi]$ in the second cohomology group of the chain complex describing a given MBQC. $[\beta_\Psi]$ contains the function computed therein up to gauge equivalence, and at the same time is a contextuality witness. The present cohomological description only applies to temporally flat MBQCs, and we outline an approach for extending it to the temporally ordered case.

Keywords Quantum computation · Contextuality · Cohomology · Measurement-based quantum computation · Kochen–Specker theorem · Temporal order

16.1 Introduction

Spectators may be sent into infinite loops by Zeno, but Achilles catches up with the turtle anyway. Paradoxes do not spell trouble in the way contradictions do, as a contradiction appears in them only when improper assumptions are made. The more reasonable these assumptions seem, the brighter shines the paradox.

Here we investigate the paradox of Kochen and Specker (1967) and Bell (1964), describing a particular property of quantum mechanics by which it is distinguished from classical physics: contextuality (Kochen & Specker, 1967; Bell, 1964; Mermin, 1993; Abramsky & Brandenburger, 2011; Cabello et al., 2014; de Silva, 2017; Anders & Browne, 2009; Isham & Döring, 2011). The statement "quantum mechanics is contextual" means that descriptions of quantum phenomena in terms of classical

R. Raussendorf (✉)
Department of Physics and Astronomy, University of British Columbia, Vancouver, Canada

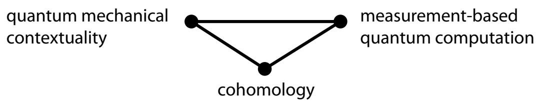
Stewart Blusson Quantum Matter Institute, University of British Columbia, Vancouver, Canada
e-mail: raussen@phas.ubc.ca

A. Palmigiano and M. Sadrzadeh (eds.), *Samson Abramsky on Logic and Structure in Computer Science and Beyond*, Outstanding Contributions to Logic 25,
https://doi.org/10.1007/978-3-031-24117-8_16

statistical mechanics—so-called non-contextual hidden variable models (ncHVMs) (Einstein et al., 1935; Kochen & Specker 1967; Bell, 1964)—are in general not viable. In such models, all observables are assigned pre-existing values which are merely revealed by measurement—in stark contrast with quantum mechanics.

Each paradox invites us to ask what becomes of the glaring discrepancy once the (in hindsight) improper assumption is excised. For the Kochen–Specker paradox, one such inquiry leads to measurement-based quantum computation (MBQC) (Raussendorf & Briegel, 2001), a scheme of universal quantum computation driven by measurement.

We identify the mathematical structures that simultaneously capture the contextuality and the computational output of measurement-based quantum computations. These structures turn out to be cohomological. Put in graphical form, we explore the following triangle.



(16.1)

In the first part of this paper, consisting of Sects. 16.2 and 16.3, we flesh out the above diagram for the simplest case, deterministic temporally flat MBQCs and the corresponding proofs of contextuality. Temporally flat means that these MBQCs have no non-trivial temporal order, which is a restriction. Section 16.2 reviews the necessary background on contextuality and measurement-based quantum computation, and Sect. 16.3 explains how cohomology encapsulates the essence of parity-based contextuality proofs and temporally flat MBQCs. The main results are Theorem 4 (Okay et al., 2017) and Theorem 5 (Raussendorf, 2019), which we restate here.

In the second part of the paper, consisting of Sect. 16.4, we work towards removing the assumption of temporal flatness. MBQCs are typically temporally ordered. Even though the measurements driving the computation commute, measurement bases need to be adjusted depending on the outcomes of other measurements, and this introduces temporal order. This adjustment is necessary to prevent the randomness inherent in quantum measurement from creeping into the logical processing.

While we do not yet tackle temporally ordered MBQCs, we demonstrate that a known contextuality proof exhibiting temporal ordering of measurements, the so-called "iffy" proof (Abramsky et al., 2018), can be described by the *same* cohomological formalism that is used for the temporally flat case. We conjecture that this strategy might also work for general MBQCs.

Section 16.5 is the conclusion, and Sect. 16.6 covers some stations of the author's own journey through the world of quantum computation and paradox.

16.2 Background

16.2.1 Contextuality

We assume that the reader is familiar with the concept of contextuality (Kochen & Specker, 1967; Bell, 1964); see Mermin (1993) for a review. To provide a short summary, contextuality of quantum mechanics signifies that, in general, quantum mechanical phenomena cannot be described by so-called non-contextual hidden variable models (ncHVMs) (Einstein et al., 1935). In an ncHVM, observable quantities have predetermined value assignments; i.e., each observable possesses a value, and those values are merely revealed upon measurement. The statistical character of measurement in quantum mechanics is then sought to be reproduced by a probability distribution over the value assignments. For certain sets of measurements no such probability distribution exists. If that's the case, then the physical setting at hand is contextual.

In this paper, we assume that in each value assignment λ is deterministic; i.e., the value $\lambda(A)$ assigned to each observable A is an eigenvalue of that observable, in accordance with the Dirac projection postulate. More general constructs are conceivable; for example the value assignments may themselves be probability distributions over eigenvalues (Spekkens, 2008); however, we do not consider such generalizations here. We remark that deterministic ncHVMs are equivalent to factorizable probabilistic ones (Anders & Browne, 2009); also see Fine (1982).

The Kochen–Specker (KS) theorem (Kochen & Specker, 1967) says that in Hilbert spaces of dimension 3 and higher, it is impossible to assign all quantum-mechanical observables deterministic non-contextual values in a consistent fashion. A very simple proof of the KS theorem, in dimension 4 and up, is provided by Mermin's square (Mermin, 1993). It is the simplest parity proof of contextuality, where the assumption of existence of a consistent non-contextual value assignment λ leads to a system of mod 2–linear equations with an internal inconsistency. As we will discuss below, the connection between contextuality and MBQC runs through the parity proofs.

For MBQC we employ state-dependent contextuality. In it, consistent value assignments λ do exist, but no probability distribution over them can explain the measurement statistics for the quantum state in question. The reason that value assignments suddenly become possible does not contradict the KS theorem; we merely have shrunk the set of observables considered. Already the original proof (Kochen & Specker, 1967) of the KS theorem and the simpler proof via Mermin's square use a finite number of observables picked from a priori infinite sets; and in the application to MBQC we simply reduce those sets further.

The key example for the connection between contextuality and MBQC is the state-dependent version of Mermin's star (Mermin, 1993), as was observed in Anders and Browne (2009). Consider the eight-dimensional Hilbert space of 3 qubits, a specific state in it, the Greenberger–Horne–Zeilinger (GHZ) state (Greenberger et al., 1989),

$$|\mathrm{GHZ}\rangle = \frac{|000\rangle + |111\rangle}{\sqrt{2}}, \tag{16.2}$$

and furthermore the six local Pauli observables X_i, Y_i, $i = 1, \ldots, 3$. The state-dependent contextuality question is whether those six local observables can be assigned values $\lambda(\cdot) = \pm 1$ in such a way that the measurement statistics for the four non-local Pauli observables $X_1X_2X_3$, $X_1Y_2Y_3$, $Y_1X_2Y_3$, $Y_1Y_2X_3$ is reproduced.

The GHZ state is a simultaneous eigenstate of these observables,

$$X_1X_2X_3\,|\mathrm{GHZ}\rangle = -X_1Y_2Y_3\,|\mathrm{GHZ}\rangle = -Y_1X_2Y_3\,|\mathrm{GHZ}\rangle = -Y_1Y_2X_3\,|\mathrm{GHZ}\rangle = |\mathrm{GHZ}\rangle.$$

The measurement outcomes for the four non-local observables are deterministic and equal to ± 1.

Now note that these non-local observables are products of the local ones X_i, Y_i, namely $X_1X_2X_3 = (X_1)(X_2)(X_3)$, $X_1Y_2Y_3 = (X_1)(Y_2)(Y_3)$, etc. Assuming an ncHVM value assignment λ for the local observables, the above operator constraints translate into constraints on the assigned values $\lambda(\cdot)$, namely $\lambda(X_1)\lambda(X_2)\lambda(X_3) = +1$, $\lambda(X_1)\lambda(Y_2)\lambda(Y_3) = -1$, and two more of the same kind. It is useful to write the value assignments λ in the form $\lambda(\cdot) = (-1)^{s(\cdot)}$. In terms of the binary variables s, the four constraints read

$$\begin{array}{rl} s(X_1) + s(X_2) + s(X_3) & \mod 2 = 0, \\ s(X_1) + s(Y_2) + s(Y_3) & \mod 2 = 1, \\ s(Y_1) + s(X_2) + s(Y_3) & \mod 2 = 1, \\ s(Y_1) + s(Y_2) + s(X_3) & \mod 2 = 1. \end{array} \tag{16.3}$$

Adding those four equations mod 2 reveals a contradiction $0 = 1$, hence no value assignment s (equivalently λ) for the six local observables reproduces the measurement statistics of the GHZ state. The state-dependent Mermin star is thus contextual. We will return to Eq. (16.3) throughout, as it relates to the simplest example of a contextual MBQC (Anders & Browne, 2009).

In preparation for the subsequent discussion we review one further concept, the contextual fraction (Abramsky & Brandenburger, 2011). To define it, consider an empirical model e, i.e., a collection of probability distributions over measurement contexts, and split it into a contextual part e^C and a non-contextual part e^{NC},

$$e = \tau e^{NC} + (1-\tau)e^C,\ \ 0 \leq \tau \leq 1. \tag{16.4}$$

The maximum possible value of τ is called the non-contextual fraction $\mathsf{NCF}(e)$ of the model e,

$$\mathsf{NCF}(e) := \max_{e^{NC}} \tau. \tag{16.5}$$

The contextual fraction $\mathsf{CF}(e)$ is then the probability weight of the contextual part e^C,

$$\mathsf{CF}(e) := 1 - \mathsf{NCF}(e). \tag{16.6}$$

It is a measure of the "amount" of contextuality contained in a given physical setup.

16.2.2 Measurement-Based Quantum Computation

Again, we assume that the reader is familiar with the concept of measurement-based quantum computation, a.k.a. the one way quantum computer (Raussendorf & Briegel, 2001). Here we provide only a very short summary, and then expand on one technical aspect that is of particular relevance for the connection with contextuality–the classical side processing. For a review of MBQC see e.g. Raussendorf and Wei (1989).

In MBQC, the process of quantum computation is driven by local measurement on an initially entangled quantum state; no unitary evolution takes place. Further, the initial quantum state, for example a 2D cluster state, does not carry any information about the algorithm to be implemented—it is universal. All algorithm-relevant information is inputted to that quantum state, processed and read out by the local measurements.

In quantum mechanics, the basis of a measurement can be freely chosen but the measurement outcome is typically random; and this of course affects MBQC. There, the choice of measurement bases encodes the quantum algorithm to be implemented, and the measurement record encodes the computational output. In MBQC every individual measurement outcome is in fact completely random, and meaningful information is contained only in correlations of measurement outcomes. As it turns out, these computationally relevant correlations have a simple structure. To extract them from the measurement record, every MBQC runs a classical side processing.

The need for classical side processing in MBQC also arises in a second place: measurement bases must be adapted according to previously obtained measurement outcomes, in order to prevent the randomness of quantum measurement from creeping into the logical processing.

We confine our attention to the original MBQC scheme on cluster states (Raussendorf & Briegel, 2001), which we will henceforth call $l2$-MBQC. There are other MBQC schemes, for example using AKLT states as computational resources, in which the side processing is more involved.

In $l2$-MBQC, for each measurement i there are two possible choices for the measured observable $O_i[q_i]$, depending on a binary number q_i. The eigenvalues of these observables are constrained to be ± 1. Furthermore, both the bitwise output $\mathbf{o} = (o_1, o_2 \ldots, o_k)$ and the choice of measurement bases, $\mathbf{q} = (q_1, q_2, \ldots, q_N)$ are functions of the measurement outcomes $\mathbf{s} = (s_1, s_2, \ldots, s_N)$. In addition, $\mathbf{q}$ is also a function of the classical input $\mathbf{i} = (i_1, i_2, \ldots, i_m)$. These functional relations are all mod 2 linear,

$$\mathbf{o} = Z\mathbf{s} \mod 2, \tag{16.7a}$$

$$\mathbf{q} = T\mathbf{s} + S\mathbf{i} \mod 2. \tag{16.7b}$$

Therein, the binary matrix T encodes the temporal order in a given MBQC. If $T_{ij} = 1$ then the basis for the measurement i depends on the outcome of measurement j, hence the measurement j must be executed before the measurement i. We remark that Eqs. (16.7) have been discussed with additional constant offset vectors on the r.h.s. (Raussendorf et al., 2012), but we don't need that level of generality here.

16.2.3 Links Between Contextuality and MBQC

The basic result relating MBQC to contextuality is the following.

Theorem 1 (Raussendorf, 2013) *Be $\mathcal{M}$ an l2-MBQC evaluating a function $o : (\mathbb{Z}_2)^m \longrightarrow \mathbb{Z}_2$. Then, $\mathcal{M}$ is contextual if it succeeds with an average probability $p_S > 1 - d_H(o)/2^m$, where $d_H(o)$ is the Hamming distance of o from the closest linear function.*

That is, if the function evaluated by the $l2$-MBQC is non-linear—hence outside what the classical side processing can compute by itself—then the assumption of non-contextuality puts a limit on the reachable probability of success. The reliability of the MBQC can be improved beyond this threshold only in the presence of contextuality. The more nonlinear the computed function (in terms of the Hamming distance $d_H(o)$), the lower the threshold. The lowest contextuality thresholds are reached for bent functions. For m even and o bent, it holds that $d_H(o) = 2^{m-1} - 2^{m/2-1}$ (MacWilliams & Sloane, 1977), and therefore the contextuality threshold for the average success probability p_S approaches $1/2$ for large m. An MBQC can thus be contextual even though its output is very close to completely random.

In particular when comparing the above Theorem 1 to structurally similar theorems on the role of entanglement in MBQC (Van den Nest et al., 2007), we observe that the above only provides a binary "can do vs. cannot do" separation. According to the theorem, in the presence of contextuality a success probability of unity is a priori possible, but without it the stated bound applies. Yet it is intuitively clear that the reachable success probability of function evaluation in MBQC should depend on the "amount" of contextuality present. In this regard, we note the following refinement of Theorem 1, invoking the contextual fraction.

Theorem 2 (Abramsky et al., 2017) *Let $f : (\mathbb{Z}_2)^m \longrightarrow \mathbb{Z}_2$ be a Boolean function, and $\mathbb{H}(f, \mathcal{L})$ its Hamming distance to the closest linear function. For each l2-MBQC with contextual fraction $\mathsf{CF}(\rho)$ that computes f with average success probability $\overline{p}_S$ over all 2^m possible inputs it holds that*

$$\overline{p}_S \leq 1 - \frac{(1 - \mathsf{CF}(\rho))\, \mathbb{H}(f, \mathcal{L})}{2^m}. \tag{16.8}$$

Thus, the larger the contextual fraction, the larger the achievable success probability for function evaluation through MBQC. If the contextual fraction of the resource state becomes unity, then the theorem puts no non-trivial bound on the success probability of the corresponding $l2$-MBQC.

If, on the other hand, the contextual fraction of the resource state becomes zero, i.e., when the resource state can be described by a non-contextual hidden variable model, the threshold in success probability reduces to that of Theorem 1. Theorem 2 interpolates between those two limiting cases.

One important aspect of the MBQC–contextuality relationship is revealed only by the proof of Theorem 1, but not by the statement of the theorem itself. Namely, the contextuality of MBQC is intimately related to the classical side processing Eq. (16.7). Rather than replicating the proof from Raussendorf (2013), here we illustrate the idea through the example of Anders and Browne's GHZ-MBQC (Anders & Browne, 2009), related to Mermin's star. We will return to this example throughout.

Example (GHZ-MBQC). In this scenario, the resource state is a Greenberger–Horne–Zeilinger state of Eq. (16.2), and the local measurable observables $O_i[q_i]$, depending on a binary number q_i, are $O_i[0] = X_i,\ \ O_i[1] = Y_i$, for $i = 1, \dots, 3$. These are precisely the ingredients of the state-dependent version of Mermin's star, as we discussed in Sect. 16.2.1. As before, the measurement outcomes $s_i \in \mathbb{Z}_2$ are related to the measured eigenvalues $\lambda_i = \pm 1$ of the respective local Pauli observables via $\lambda_i = (-1)^{s_i}$. There are two bits y, z of input and one bit o of output, and the computed function is an OR-gate, $o = y \vee z$.

The required linear classical side processing is as follows.

$$q_1 = y,\ q_2 = z,\ q_3 = y + z \mod 2, \tag{16.9a}$$

$$o = s_1 + s_2 + s_3 \mod 2. \tag{16.9b}$$

The two input bits y and z determine the choices q_i of measured observables through Eq. (16.9a), and then the corresponding binary measurement outcomes s_1, s_2, s_3 determine the outputted value of the function, $o(y, z)$.

Let's verify that the output is the intended OR function. First, consider $y = z = 0$. Thus, by Eq. (16.9a), $q_1 = q_2 = q_3 = 0$, and all three locally measured observables are of X-type. While the outcomes s_1, s_2, s_3 are individually random, they are correlated since the product of the corresponding observables X_i is the stabilizer of the GHZ state, $X_1 X_2 X_3 |\text{GHZ}\rangle = |\text{GHZ}\rangle$. Therefore, $s_1 + s_2 + s_3 \mod 2 = 0$. Hence, with Eq. (16.9b), $o(0, 0) = 0$ as required for the OR-gate.

We consider one more input combination, $y = 0$ and $z = 1$. Then, with Eq. (16.9a), $q_1 = 0$ and $q_2 = q_3 = 1$. Hence X_1, Y_2 and Y_3 are measured. Because of the stabilizer relation $X_1 Y_2 Y_3 |\text{GHZ}\rangle = -|\text{GHZ}\rangle$, the three measurement outcomes s_1, s_2, s_3 satisfy $s_1 + s_2 + s_3 \mod 2 = 1$. With Eq. (16.9b), $o(0, 1) = 1$ as required. The discussion of the other two inputs is analogous.

The OR-gate is a very simple function; yet it is of consequence for the above computational setting. Every MBQC requires a classical control computer, to enact the classical side processing of Eq. (16.9). This control computer is constrained to performing addition mod 2, and it is therefore not classically computationally universal. The OR-gate is a non-linear Boolean function. By adding it to the available operations, the extremely limited classical control computer is boosted to classical computational universality (Anders & Browne, 2009).

To understand the connection between contextuality and MBQC classical processing relations, we state Eq. (16.9b) separately for all four input values.

$$\begin{array}{llll} \textbf{input:} & (0,0) & \textbf{output:} & 0 = s(X_1) + s(X_2) + s(X_3) \\ & (0,1) & & 1 = s(X_1) + s(Y_2) + s(Y_3) \\ & (1,0) & & 1 = s(Y_1) + s(X_2) + s(Y_3) \\ & (1,1) & & 1 = s(Y_1) + s(Y_2) + s(X_3) \end{array} \tag{16.10}$$

Note the striking resemblance of Eq. (16.10) to the earlier Eq. (16.3). The only difference is that Eq. (16.10) refers to quantum mechanical measurement record, one context at a time, whereas Eq. (16.3) refers to a noncontextual value assignment in an ncHVM, applying to all contexts simultaneously. Thus, if we assume an ncHVM then we obtain a contradiction; and if we do not assume it then the same equations describe a computation.

This dichotomy exists not only for the GHZ-scenario discussed here, but indeed for all MBQCs satisfying the classical processing relations Eq. (16.7). It is the basis for Theorems 1 and 2.

16.3 Cohomology

In the previous section we found that for $l2$-MBQCs contextuality and computation hinge on the same algebraic structure. If we impose an ncHVM description on top of this structure, we obtain a contradiction; and if we do not impose it, we obtain a computation. This begs the question: *What precisely is this common algebraic structure underlying both parity-based contextuality proofs and measurement-based quantum computation?* This is where cohomology comes in.

Below, we build up the cohomological picture for deterministic, temporally flat MBQCs. The connection between MBQC and contextuality runs through state-dependent parity-based contextuality proofs. In Sect. 16.3.1, we first introduce the cohomological description of the state-independent counterpart. It is based on a chain complex $\mathcal{C}(E)$, and slightly simpler. We then progress to the state-dependent version, described by the relative chain complex $\mathcal{C}(E, E_0)$. In Sect. 16.3.2, we explain the relation between cohomology in $\mathcal{C}(E, E_0)$ and MBQC output.

16.3.1 Cohomology and Contextuality

We begin with the simpler state-independent parity proofs of contextuality, and then move on to their state-dependent cousins which are of more direct interest for MBQC. In all that follows we consider observables whose eigenvalues are all ± 1. We denote these observables by T_a, a notation we now explain.

The basic object in the cohomological discussion of the parity proofs are chain complexes $\mathcal{C}(E) = (C_0, C_1, C_2, C_3)$ consisting of points (0-chains), edges (1-chains), faces (2-chains) and volumes (3-chains), and boundary maps ∂ between those chains. The observables T_a forming the contextuality proof are associated with the edges $a \in E$ in the complex $\mathcal{C}(E)$. More precisely, each edge a corresponds to an equivalence class $\{\pm T_a\}$ of observables, $a := \{\pm T_a\}$. From each equivalence class a, one observable is picked and denoted as T_a.

From the perspective of contextuality, the reason for considering the observables T_a and $-T_a$ as equivalent is the following. If a parity-based contextuality proof can be based on some set of observables $\{T_a, a \in E\}$, then any signed set $\{(-1)^{\gamma(a)} T_a,\ \gamma(a) \in \mathbb{Z}_2, \forall a \in E\}$ produces an equivalent proof. The signs $(-1)^{\gamma(a)}$ in the definition of the observables T_a don't matter for the existence of contextuality proofs; and this leads us to consider the equivalence classes $\{\pm T_a\}$. We will return to this observation once we have set up the appropriate notation, right after Theorem 3.

The 1-chains $c_1 \in C_1$ are linear combinations of the edges $a \in E$ with $\mathbb{Z}_2$ coefficients. The faces of $\mathcal{C}(E)$ are sets $f = (a_1, a_2, \ldots, a_n)$ of edge labels a_i of pairwise commuting operators T_{a_i}, such that for every face f it holds that

$$\prod_{a \in f} T_a = I(-1)^{\beta(f)}, \tag{16.11}$$

for a suitable function β defined on the faces. We denote the set of faces by F, and the 2-chains $c_2 \in C_2$ are linear combinations of the faces $f \in F$ with coefficients in $\mathbb{Z}_2$.

We can now define a boundary operator $\partial : C_2 \longrightarrow C_1$ via $\partial(f) = \sum_{a \in f} a$, for all $f \in F$, and extension from F to C_2 by linearity. We can then also define a coboundary operator $d : C^1 \longrightarrow C^2$ in the usual way; i.e. for every 1-cochain $x \in C^1$ it holds that $dx(f) := x(\partial f)$, for all $f \in F$.

The function $\beta : C_2 \longrightarrow \mathbb{Z}_2$ plays a central role in the cohomological discussion of contextuality. Namely, assume that a non-contextual value assignment λ exists, and as before write $\lambda(\cdot) = (-1)^{s(\cdot)}$. Then, Eq. (16.11) implies that $\beta(f) = \sum_{a \in f} s(a) = s(\partial f)$ for all $f \in F$. We may write this in cochain notation as

$$\beta = ds. \tag{16.12}$$

This equation may be interpreted as a constraint on the value assignment s, given β. But it may as well be regarded as a constraint on β. Namely, not all functions β are of form Eq. (16.12), for any 1-cochain s. Thus, a measurement setting based on $\mathcal{C}(E)$

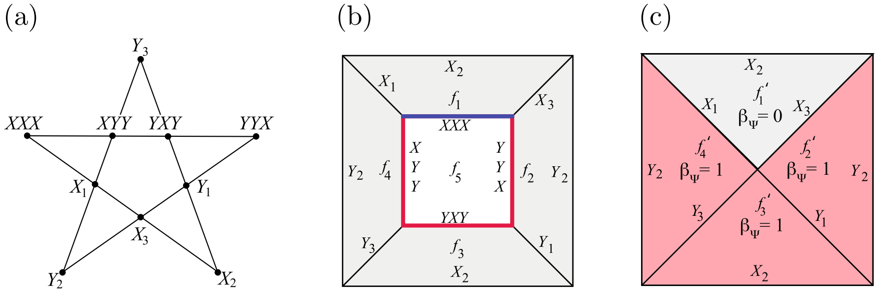


Fig. 16.1 Mermin's star. **a** Standard representation. Each line represents a measurement context, composed of four commuting Pauli observables multiplying to $\pm I$. **b** Mermin's star re-arranged on a surface. The Pauli observables now correspond to edges, and each measurement context to the boundary of one of the four elementary faces. The exterior edges are pairwise identified. The colored edges carry a value assignment, resulting from the GHZ stabilizer. **c** Relative complex $\mathcal{C}(E, E_0)$. The edges corresponding to observables in the GHZ stabilizer are removed by contraction

is non-contextual only if $\beta = ds$ for some $s \in C^1$, or, equivalently, it is contextual if $\beta \neq ds$, for any $s \in C^1$.

We will now slightly reformulate the last statement, to better bring out its cohomological nature. The function β is by definition a 2-cochain. But in fact it is a 2-cocycle, $d\beta = 0$ (Okay et al., 2017). Thus, we may express the above contextuality condition as follows.

Theorem 3 (Okay et al., 2017) *A set of measurements specified by the chain complex $\mathcal{C}(E)$ is contextual if for the cocycle class $[\beta] \in H^2(\mathcal{C}(E), \mathbb{Z}_2)$ it holds that*

$$[\beta] \neq 0.$$

Remark We observed above that no transformation $T_a \longrightarrow (-1)^{\gamma(a)} T_a$, $\forall a \in E$, affects the existence of contextuality proofs. We can now verify this statement in Theorem 3. At the level of the cocycle β, the transformations act as $\beta \longrightarrow \beta + d\gamma$. Hence, $[\beta] \longrightarrow [\beta]$. The parity proofs are thus indeed unchanged. We point out that the transformations discussed here—which we call gauge transformations—have a further use in characterizing MBQC output functions; see Sect. 16.3.2.

Now let's consider the state-independent Mermin star in this framework. The ten Pauli observables T_a therein are assigned to the edges $a \in E$ in a chain complex $\mathcal{C}$; see Fig. 16.1b. For the five faces shown we have $\beta(f_1) = \beta(f_2) = \beta(f_3) = \beta(f_4) = 0$, and $\beta(f_5) = 1$. Further denote $\mathcal{F} := \sum_{i=1}^{5} f_i$, such that $\partial\mathcal{F} = 0$ and $\beta(\mathcal{F}) = 1$. Now assume Mermin's star were non-contextual. Then, $\beta = ds$ for some $s \in C^1$, and we have

$$0 = s(0) = s(\partial\mathcal{F}) = ds(\mathcal{F}) = \beta(\mathcal{F}) = 1.$$

Contradiction. Hence, Mermin's star is contextual.

We now seek a state-dependent version of Theorem 3, preferably formulated in an analogous way. This can be achieved by proceeding from the chain complex $\mathcal{C}(E)$ to a relative chain complex $\mathcal{C}(E, E_0)$. The quantum state $|\Psi\rangle$ now appears, and the set $E_0 \subset E$ consists of those edges a for which the corresponding operator T_a has $|\Psi\rangle$ as an eigenstate,

$$T_a|\Psi\rangle = (-1)^{\mu(a)}|\Psi\rangle, \text{ with } \mu : E_0 \longrightarrow \mathbb{Z}_2. \quad (16.13)$$

Geometrically, $\mathcal{C}(E, E_0)$ is obtained from $\mathcal{C}(E)$ by contracting the edges in E_0. Thereby, the faces of $\mathcal{C}(E)$ whose boundary lives entirely inside E_0 are removed. Under this contraction, the boundary map ∂ changes to a relative boundary map ∂_R defined by $\partial_R(f) = \sum_{a \in f \setminus E_0} a$.

Extending the above function μ to all of E by setting $\mu(a) := 0$ for all $a \notin E_0$, we define a relative 2-cochain

$$\beta_\Psi := \beta + d\mu \mod 2. \quad (16.14)$$

Again, β_Ψ is a 2-cocycle. Also, β_Ψ evaluates to zero on all faces with boundary entirely inside E_0, and it is thus a cocycle in the relative complex $\mathcal{C}(E, E_0)$.

Quantum mechanically, the measurement record in the context corresponding to any face $f \in F$ has to satisfy $s|_{f \cap E_0} = \mu|_{f \cap E_0}$, and $\beta(f) = s(\partial f)$. Then, from the above definitions it follows that

$$\beta_\Psi(f) = s(\partial_R f). \quad (16.15)$$

Now assume a value assignment s exists. It has to satisfy the condition Eq. (16.15) for all faces $f \in F$ simultaneously. We may thus write the constraints on such a global value assignment s as $ds = \beta_\Psi$, with d now being the coboundary operator in the complex $\mathcal{C}(E, E_0)$.

We thus have, in complete analogy with the state-independent case, the following result.

Theorem 4 (Okay et al., 2017) *A set of measurements and a quantum state* $|\Psi\rangle$ *specified by the chain complex* $\mathcal{C}(E, E_0)$ *are contextual iff for the cocycle class* $[\beta_\Psi] \in H^2(\mathcal{C}(E, E_0), \mathbb{Z}_2)$ *it holds that*

$$[\beta_\Psi] \neq 0.$$

Example, Part II. We now apply this to the example of the state-dependent Mermin star. Four faces remain in $\mathcal{C}(E, E_0)$ after contraction of E_0 in $\mathcal{C}(E)$, $f_1', \ldots, f_4'$. We have $\beta_\Psi(f_1') = 0, \beta_\Psi(f_2') = \beta_\Psi(f_3') = \beta_\Psi(f_4') = 1$. Denote $\mathcal{F}' = \sum_{i=1}^4 f_i'$ such that the relative boundary of $\mathcal{F}'$ vanishes, $\partial_R \mathcal{F}' = 0$, and $\beta_\Psi(\mathcal{F}') = 1$.

Now assume that the state-dependent Mermin star is non-contextual. Then, $\beta_\Psi = ds$ for some 1-cochain $s \in C^1(\mathcal{C}(E, E_0), \mathbb{Z}_2)$. And thus

$$1 = \beta_\Psi(\mathcal{F}') = s(\partial \mathcal{F}') = s(0) = 0. \quad (16.16)$$

Contradiction. Hence the state-dependent Mermin star is contextual.

Equation (16.16) is the cohomological version of Eq. (16.3). It describes the exact same system of linear constraints.

16.3.2 Cohomology and Computation

Recall from Sect. 16.2.2 that in MBQC there are two measurable observables at each physical site i, $O_i[q_i]$, $q_i \in \mathbb{Z}_2$. To make use of the cohomological formalism, we now denote these observables as

$$O_i[0] = T_{a_i},\ \ O_i[1] = T_{\overline{a}_i},\ \ \forall i = 1, \ldots, n. \tag{16.17}$$

We define the notion of an input group to import the classical processing relation Eq. (16.7b) into our cohomological picture. The input group is $Q = \langle \mathbf{i}_j,\ j = 1, \ldots, m\rangle \cong \mathbb{Z}_2^m$. The generators of Q act on the observables of Eq. (16.17) as

$$\begin{array}{l} \mathbf{i}_j(a_i) = (a_i),\ \mathbf{i}_j(\overline{a}_i) = (\overline{a}_i),\ \text{if}\ S_{ij} = 0, \\ \mathbf{i}_j(a_i) = (\overline{a}_i),\ \mathbf{i}_j(\overline{a}_i) = (a_i),\ \text{if}\ S_{ij} = 1. \end{array} \tag{16.18}$$

Denoting by $\mathcal{E}_\mathrm{e}$ a reference context corresponding to the trivial input $\mathrm{e} \in Q$, $\mathcal{E}_\mathrm{e} := \{a_j,\ j = 1, \ldots, n\}$, and by $\mathcal{E}_\mathbf{i}$ the measurement context for any input $\mathbf{i} \in Q$, then, with the definitions Eqs. (16.17) and (16.18), the relation

$$\mathcal{E}_\mathbf{i} = \mathbf{i}(\mathcal{E}_\mathrm{e}) := \{\mathbf{i}(a_j),\ j = 1, \ldots, n\} \tag{16.19}$$

reproduces the classical side processing relation Eq. (16.7b) in the limit of temporally flat MBQCs, $T = 0$. This is the limit we are presently interested in.

We have thus far represented computational input by a group Q that maps the complex $\mathcal{C}(E, E_0)$ to itself, and we now turn to the computational output. In terms of the above sets $\mathcal{E}_\mathbf{i}$, the classical side processing relations for output, Eq. (16.7a), read

$$o(\mathbf{i}) = \sum_{a \in \mathcal{E}_\mathbf{i}} s(a) \mod 2,\ \ \forall \mathbf{i} \in Q. \tag{16.20}$$

We note that for any $\mathbf{i} \in Q$, the observables T_a, $a \in \mathcal{E}_\mathbf{i}$, pairwise commute. Furthermore, in the setting of deterministic computation, the input group Q (equivalently, the matrix S in Eq. (16.7b)) is chosen such that the MBQC resource state $|\Psi\rangle$ is an eigenstate of all observables $\prod_{a \in \mathcal{E}_\mathbf{i}} T_a$. That is, $\prod_{a \in \mathcal{E}_\mathbf{i}} T_a = \pm T_x$, with $x \in E_0$; cf. Eq. (16.13). Therefore, the edges $a \in \mathcal{E}_\mathbf{i}$ form the boundary of a face $f_\mathbf{i}$ in the contracted complex $\mathcal{C}(E, E_0)$, i.e., $f_\mathbf{i} \in C_2(\mathcal{C}(E, E_0))$ satisfies $\mathcal{E}_\mathbf{i} = \{\partial_R f_\mathbf{i}\}$. Finally, with Eq. (16.19), $\mathcal{E}_\mathbf{i} = \{\mathbf{i}(\partial_R f_\mathrm{e})\}$, and the face f_e corresponds to $\mathcal{E}_\mathrm{e}$. Therefore, Eq. (16.20) can be rewritten in cohomological notation as

$$o(\mathbf{i}) = s(\mathbf{i}(\partial_R f_\mathrm{e})),$$

where s is the measurement record for the observables in $\mathcal{E}_\mathbf{i}$.

Inserting Eq. (16.15) into the last equation, we obtain the following result.

Theorem 5 (Raussendorf, 2019) *The function $o: Q \longrightarrow \mathbb{Z}_2$ computed in a given deterministic and temporally flat l2-MBQC is related to the cocycle $\beta_\Psi \in C^2(\mathcal{C}(E, E_0))$ via*

$$o(\boldsymbol{i}) = \beta_\Psi(\boldsymbol{i}(f_e)),\ \ \forall \boldsymbol{i} \in Q. \tag{16.21}$$

This relation between the computational output o and the 2-cocycle β_Ψ is the main result of this section. It has been established in greater generality in Raussendorf (2019) (Theorem 4 therein), but we don't need the additional generality here. Theorem 5 is the computational counterpart to Theorem 4 in Sect. 16.3.1. Both results together establish that a single cohomological object, the cocycle β_Ψ, governs contextuality and computational output in MBQC. Jointly, Theorems 4 and 5 thus flesh out the Diagram (16.1).

Example, Part III. For the GHZ-MBQC, Eq. (16.21) may be explicitly verified by inspecting Fig. 16.1c. The reference context is $\mathcal{E}_\mathrm{e} = (a_{X_1}, a_{X_2}, a_{X_3})$, hence $f_\mathrm{e} = f_1'$, w.r.t. the labeling of Fig. 16.1c. The input group is $Q = \mathbb{Z}_2 \times \mathbb{Z}_2$. Its two generators $\mathbf{i}_1, \mathbf{i}_2$ are related to the input bits y, z of the OR-gate via $y \mapsto \mathbf{i}_1,\ z \mapsto \mathbf{i}_2$, and Eq. (16.18) becomes

$$\begin{array}{l} \mathbf{i}_1 : a_{X_1} \leftrightarrow a_{Y_1},\ a_{X_3} \leftrightarrow a_{Y_3},\ a_{X_2} \circlearrowleft,\ a_{Y_2} \circlearrowleft, \\ \mathbf{i}_2 : a_{X_2} \leftrightarrow a_{Y_2},\ a_{X_3} \leftrightarrow a_{Y_3},\ a_{X_1} \circlearrowleft,\ a_{Y_1} \circlearrowleft . \end{array} \tag{16.22}$$

We may now verify in the cohomological calculus established above that this action does indeed lead to the execution of the OR-gate in the corresponding GHZ-MBQC. For example, if $y = z = 0$ then $\mathbf{i} = \mathrm{e}$, and $\mathbf{i}(f_1') = f_1'$; and thus $o(0,0) = \beta_\Psi(f_1') = 0 = \mathrm{OR}(0,0)$. Further, if $y = 1$ and $z = 0$, then $\mathbf{i} = \mathbf{i}_1$, and $\mathbf{i}_1(f_1') = f_3'$. Thus, $o(1,0) = \beta_\Psi(f_3') = 1 = \mathrm{OR}(1,0)$. The other two cases are analogous. See Fig. 16.2 for illustration of the action of the input group given by Eq. (16.18).

One point remains to be discussed. When comparing Theorems 4 and 5, we notice a difference. Theorem 4 invokes the cohomology class $[\beta_\Psi]$ whereas Theorem 5 invokes the cocycle β_Ψ itself. Only the former theorem is therefore truly topological. This prompts the question: *Is there an operationally meaningful way of grouping the MBQC output functions o into equivalence classes $[o]$ that depend only on $[\beta_\Psi]$?*

That is indeed the case. The equivalence classes $[o]$ of MBQC output functions are motivated and constructed as follows. We note that the signs in the observables $\{T_a,\ a \in E\backslash E_0\}$ are a mere convention. If an observable T_a, for some $a \in E\backslash E_0$, is measured in a given MBQC, then a measurement of $-T_a$ is exactly as hard, because the corresponding projectors are the same. To obtain one measurement from the other, only the labels of the two pointer positions of the measurement device need to be switched. Therefore, the change

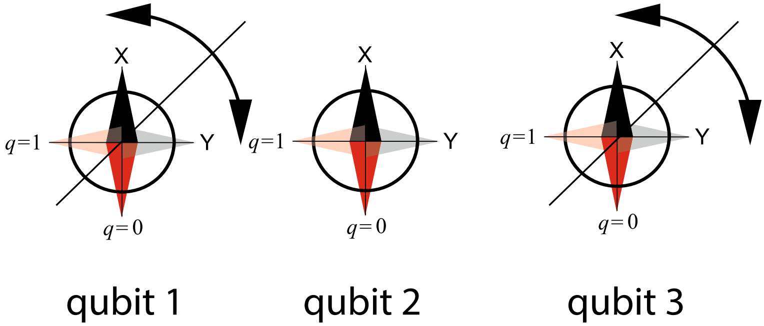


Fig. 16.2 Input group of the GHZ-MBQC. Displayed is the action of the element $\mathbf{i}_1$ of the input group $Q = \mathbb{Z}_2 \times \mathbb{Z}_2$. As described by Eq. (16.22), for qubits 1 and 3, X and Y are interchanged under the given input, and the reference context (X_1, X_2, X_3) is thereby changed into (Y_1, X_2, Y_3)

$$T_a \longrightarrow (-1)^{\gamma(a)} T_a, \;\; \forall a \in E \backslash E_0, \tag{16.23}$$

for any cochain $\gamma : C_1(E, E_0) \longrightarrow \mathbb{Z}_2$ is an equivalence transformation, or, as it is also called, a gauge transformation.

Yet, these transformations have an effect. The cocycle β_Ψ changes, namely

$$\gamma : \beta_\Psi \mapsto \beta_\Psi + d\gamma.$$

And thus, by Theorem 5, the outputted function o changes too. Functions obtained from one another through such a transformation should be considered computationally equivalent, as was argued above. It is thus meaningful to group MBQC output functions o into equivalence classes

$$[o(\cdot)] := \{(\beta_\Psi + d\gamma)(\cdot f_\mathrm{e}), \;\; \forall \gamma \in C^1(\mathcal{C}(E, E_0))\}.$$

With this definition, Theorem 5 has the following corollary.

Corollary 1 *For each deterministic and temporally flat l2-MBQC, the equivalence class* $[o]$ *of output functions is fully determined by* $[\beta_\Psi]$.

Thus, the gauge-invariant information in an MBQC output function is contained in the same cohomological information that also provides the contextuality proof.

Example, Part IV. In the GHZ-MBQC, we may flip $Y_3 \longrightarrow -Y_3$. In result, the new computed function is an AND. Therefore, AND and OR are equivalent w.r.t. MBQC. Considering the whole set of equivalence transformations for this example, we find that there are two equivalence classes of functions on two bits, the non-linear Boolean functions and the linear ones. Each member of the former class boosts the classical control computer of MBQC to computational universality, whereas the second class has no effect on the computational power at all.

From the cohomological perspective, $H^2(\mathcal{C}(E, E_0), \mathbb{Z}_2) = \mathbb{Z}_2$, i.e. there are two equivalence classes of cocycles β_Ψ. The trivial class corresponds to the linear Boolean functions on two bits and the non-trivial class to the non-linear Boolean functions.

16.3.3 On the Probabilistic Case

In the previous sections we focussed to deterministic MBQC. Indeed, powerful deterministic quantum algorithms do exist, notably for the Discrete Log problem (Mosca & Zalka, 2003). However, most known quantum algorithms are probabilistic, i.e., they succeed with a probability smaller than one. A cohomological treatment of probabilistic MBQCs is given in Raussendorf (2019), based on group cohomology. Here we are content with alerting the reader to the additional layer of difficulty posed by the probabilisitic case.

Let's trace the restriction to deterministic MBQCs back to its origin. In Theorem 5, the central result on the computational side, it is present through the cocycle $\beta_\Psi \in C^2(\mathcal{C}(E, E_0), \mathbb{Z}_2)$. This cocycle is defined in Eq. (16.14), in terms of the cocycle $\beta \in C^2(\mathcal{C}(E), \mathbb{Z}_2)$ and the value assignment $\mu : E_0 \longrightarrow \mathbb{Z}_2$. The value assignment μ in turn refers to eigenvalues of certain observables related to computational output, of which the resource state $|\Psi\rangle$ is an eigenstate; cf. Eq. (16.13).

In the probabilistic case, the value assignment μ does in general not exist. Hence, β_Ψ is not defined, and we cannot have straightforward probabilistic counterparts of Theorems 4 and 5.

But the problem is not merely technical; it is conceptual. Consider our running example of the GHZ-MBQC, which executes an OR-gate with certainty. As soon as probabilistic computations are admitted, we may as well say that it evaluates the constant function $y \equiv 1$ with an average success probability of 75 percent. In fact, the same computation executes any 2-bit Boolean function, except ¬OR, with some nonzero probability of success. How can we then say that one particular function is computed while all others are not?

Key to the solution is a group G of symmetry transformations that extends the input group Q, in the group-theoretic sense. G maps the complex $\mathcal{C}(E, E_0)$ to itself, acting on the observables T_a, $a \in E\backslash E_0$ via

$$g(T_a) = (-1)^{\tilde{\Phi}_g(a)} T_{ga}, \quad \forall g \in G. \tag{16.24}$$

Therein, the phase function $\tilde{\Phi}$ is, per construction, a 1-cocycle in group cohomology.

There is a further condition on G. Namely, the action Eq. (16.24) of G on the set of observables $\{\pm T_a, a \in E\backslash E_0\}$ induces an action on the output function o, and we require o to be invariant under this action. It turns out that, given G, this invariance condition constrains o up to an additive constant (Raussendorf, 2019). Thus, the output function o is *defined* through a symmetry group.

Furthermore, o can be expressed in terms of the phase function $\tilde{\Phi}$, and a contextuality proof can be given in terms of a group cohomology class derived from $\tilde{\Phi}$. In result, Theorems 4 and 5 have counterparts in the probabilistic case. They are given as Theorem 5 in Okay et al. (2017) and Theorem 6 in Raussendorf (2019), respectively. The probabilistic counterpart of Corollary 1 is Corollary 2 in Raussendorf (2019).

16.4 Temporal Order

The connection between contextuality and $l2$-MBQC described by Theorem 1 is completely general. It applies to deterministic and probabilistic measurement-based computations, as well as temporally flat and temporally ordered ones. It is only the cohomological description of this connection that is presently restricted to temporally flat computations. This is a technical limitation, and the purpose of this section is to outline an approach for overcoming it.

The idea is to not change the cohomological description at all, but to enlarge the complex $\mathcal{C}(E, E_0)$ by additional observables which take care of the temporal ordering. We illustrate this approach with the setting of the "iffy" proof (Abramsky et al., 2018).

In Sect. 16.4.1 we review the iffy contextuality proof, largely following the original exposition (Abramsky et al., 2018). We then explain why the signature feature of iffiness is incompatible with applications to MBQC. In Sect. 16.4.2 we present a cohomological contextuality proof for the iffy scenario that is MBQC-compatible. This proof includes temporal order, yet is covered by Theorem 4 without any modification.

16.4.1 The "iffy" Contextuality Proof

To get started, we require a simple example for a contextuality proof with temporal order, a counterpart to the non-adaptive GHZ proof. Luckily, Abramsky et al. (2018), Section 6, offers one; in fact, it offers a whole family of examples. We begin by writing them in a stabilizer notation that suits our purpose.

The examples consist of a three-qubit resource state $|\Psi\rangle$, and local measurement settings for the three qubits. For any even integer N, choose

$$|\Psi\rangle \sim |00\rangle|\nu\rangle + |11\rangle|\omega\rangle,$$

where

$$\begin{aligned}|\nu\rangle &= \cos\tfrac{\lambda}{2}|0\rangle + \sin\tfrac{\lambda}{2}|1\rangle,\\ |\omega\rangle &= \sin\tfrac{\lambda}{2}|0\rangle + \cos\tfrac{\lambda}{2}|1\rangle,\end{aligned}$$

and $\lambda = \pi/2 - \pi/N$. This defines the resource state. Now the measurements: qubit 3 will be measured in the eigenbasis of X or Y, and qubits 1 and 2 will be measured in the eigenbases of any of the operators

$$X_k := \cos\left(k\frac{\pi}{N}\right) X + \sin\left(k\frac{\pi}{N}\right) Y, \quad \forall k = 0 \dots 2N-1. \tag{16.25}$$

Note that $X_{N+k} = -X_k$, such that we really only need the observables $X_0, \dots, X_{N-1}$.

Denote by $P_{y,\pm}$ the projector on the eigenstate of Y with positive and negative eigenvalue, respectively, and define the operators

$$\begin{array}{rll} \mathcal{T}_k & := X^{(1)}_{N-1-k} \otimes X^{(2)}_k \otimes P^{(3)}_{y,+} + X^{(1)}_{N+1-k} \otimes X^{(2)}_k \otimes P^{(3)}_{y,-}, & k = 0, \dots, N-1, \\ \overline{X}_k & := X^{(1)}_{N-k} \otimes X^{(2)}_k \otimes X^{(3)}, & k = 0, \dots, N-1. \end{array} \tag{16.26}$$

By direct calculation, we can verify that

$$\overline{X}_k |\Psi\rangle = -|\Psi\rangle, \ \forall k, \tag{16.27a}$$

$$\mathcal{T}_k |\Psi\rangle = -|\Psi\rangle, \ \forall k. \tag{16.27b}$$

The measurement strategies considered in the contextuality proof have temporal order. Namely, first qubit 3 is measured, in the X or Y basis. In the latter case, the further choice of the measurement bases for qubits 1 and 2 depends on the outcome of the measurement at 3.

From Eq. (16.27) we can read off the constraints on the non-contextual hidden variable model, which are provided in Abramsky et al. (2018). Denote by a_k and b_k the binary measurement outcomes on qubits 1 and 2, respectively, given the measured observable X_k, and by c_0 (c_1) the outcome on qubit 3 if the measured observable is X (Y). If these values form the value assignment of an ncHVM, they must satisfy the constraints

$$\begin{array}{c} a_i \oplus b_j \oplus c_0 = 0, \ \forall i, j \text{ s.th. } i + j = 0, \\ a_i \oplus b_j \oplus c_0 = 1, \ \forall i, j \text{ s.th. } i + j = N, \\ \\ a_i \oplus b_j = 0, \ \forall i, j \text{ s.th. } i + j + (-1)^{c_1} = 0, \\ a_i \oplus b_j = 1, \ \forall i, j \text{ s.th. } i + j + (-1)^{c_1} = N. \end{array} \tag{16.28}$$

The contextuality proof proceeds from there, as usual, by adding up equations mod 2. This will be discussed below.

We now show how Eq. (16.28) are derived from the stabilizer relations Eq. (16.27).[1] The two relations at the top of Eq. (16.28) follow straightforwardly from Eq. (16.27a); here we focus on the relations at the bottom of Eq. (16.28), which derive from Eq. (16.27b).

[1] The original derivation of Eq. (16.28) in Abramsky et al. (2018) uses a different formalism which we do not reproduce here.

First, for the observables of Eq. (16.26), with Eq. (16.27) we have the following values

$$s_{\mathcal{T}_k} = s_{\overline{X}_k} = 1, \quad k = 0, \ldots, N-1. \tag{16.29}$$

corresponding to eigenvalues $(-1)^1 = -1$. Now consider separately the two cases of $c_1 = 0$ and $c_1 = 1$, respectively.

Case I: $c_1 = 0$. We now want to argue that, in this case, the observables $\mathcal{T}_k(0) = X^{(1)}_{N-1-k} \otimes X^{(2)}_k$ are also assigned the value 1,

$$s_{\mathcal{T}_k(0)} = 1, \quad k = 0, \ldots, N-1.$$

The argument is as follows. If $c_1 = 0$, then this fact could be established by measuring $Y^{(3)}$. According to quantum mechanics, the post-measurement state would be $|y, +\rangle := P^{(3)}_{y,+}|\Psi\rangle$. For this state it holds that

$$\mathcal{T}_k(0)|y, +\rangle = \mathcal{T}_k|y, +\rangle = \mathcal{T}_k P_{y,+}|\Psi\rangle = P_{y,+}\mathcal{T}_k|\Psi\rangle = -P_{y,+}|\Psi\rangle = -|y, +\rangle. \tag{16.30}$$

For later reference, note that in the above chain of equalities we have used the properties

$$\mathcal{T}_k(0) P_{y,+} = \mathcal{T}_k P_{y,+} \text{ and } [\mathcal{T}_k, P_{y,+}] = 0. \tag{16.31}$$

By Eq. (16.30), $s_{\mathcal{T}_k(0)} = 1$, for all k, as claimed. Further, by standard arguments, $s_{\mathcal{T}_k(0)} = a_{N-1-k} \oplus b_k$. Combining the last two statements,

$$a_{N-1-k} \oplus b_k = 1, \quad \forall k = 0, \ldots, N-1.$$

This provides the lower part of Eq. (16.28) for the case of $c_1 = 0$.

Case II: $c_1 = 1$. A completely analogous argument establishes the bottom half of Eq. (16.28) for $c_1 = 1$.

Equation (16.28) is thus established as a set of constraints that any value assignment $\{a_k, b_l, c_m\}$ needs to satisfy. We now complete the proof, focussing on Case I, $c_1 = 0$. Case II is analogous.

We assume that a value assignment exists. From the upper half of Eq. (16.28), we pick the equation $a_0 + b_0 + c_0 \mod 2 = 0$, and the equations $a_k + b_{N-k} + c_0 \mod 2 = 1$, for $k = 1, \ldots, N-1$. From the lower half we pick the equations $a_l + b_{N-1-l} = 1$, for $l = 0, \ldots, N-1$. Summing those equations, we obtain $Nc_0 + 2\sum_{k=0}^{N-1}(a_k + b_k) = 2N - 1 \pmod 2$. Since N is even, this is a contradiction. □

Now that we have presented the iffy contextuality proof, let's take a step back and ask two questions.

(1) *Where is temporal order in this contextuality proof?*—Suppose one wants to test the correlations of Eq. (16.26) through local measurement. The correlations are labeled by an integer $k \in \mathbb{Z}_N$, and a further binary integer $l \in \mathbb{Z}_2$ that decides whether qubit #3 is measured in the X-basis ($l = 0$) or in the Y-basis ($l = 1$). Given the input (k, l), the pattern of local measurements to test the correlations is fully specified.

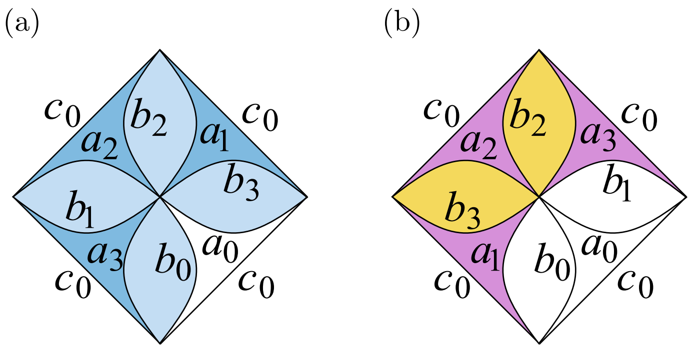


Fig. 16.3 Chain complexes in the iffy proof, for $N = 4$. **a** the complex $\mathcal{C}^{(0)}$ for $c_1 = 0$ and **b** the complex $\mathcal{C}^{(1)}$ for $c_1 = 1$. In either case, the four edges labeled "c_0" correspond to the same observable $X^{(3)}$, and are identified. The faces f on which $\beta_\Psi(f) = 1$ (0) are shown in color (white)

Therein, if $l = 1$, the measurement basis for qubit #1 depends on the outcome c_1 obtained on qubit #3, cf. Eq. (16.26), upper line. Thus, qubit #1 must be measured *after* qubit #3. This is the same temporal ordering due to adaptive measurement as occurs in MBQC.

(2) *Is the iffy proof topological?*—Yes, but with a caveat. The value assignment for c_1 is not part of the topological description. Instead there are two separate topological descriptions, one for $c_1 = 0$ and one for $c_1 = 1$. They are depicted in Fig. 16.3, (a) the complex $\mathcal{C}^{(0)}$ for $c_1 = 0$ and (b) the complex $\mathcal{C}^{(1)}$ for $c_1 = 1$. In both cases there is a surface $\mathcal{F}^{(c_1)}$ comprising all of the faces displayed. Those surfaces have the property that $\partial\mathcal{F}^{(c_1)} = 0$. In both cases it holds that $\beta_\Psi^{(c_1)}(\mathcal{F}^{(c_1)}) = 1$, which, together with the former statement, implies that $\left[\beta_\Psi^{(c_1)}\right] \neq 0$, $\forall c_1 \in \mathbb{Z}_2$. The iffy proof thus has two cohomological parts, conditioned by the value of c_1,

$$\text{Iffy Proof} = \left\{\mathbb{Z}_2 \ni c_1 \mapsto \left(\mathcal{C}^{(c_1)}, \beta_\Psi^{(c_1)}\right)\right\}. \tag{16.32}$$

The conditioning on c_1 is in the way of using the iffy proof as a template for describing temporally ordered MBQCs. To see why this is so, let's recap the earlier topological proofs. There, the assumption of a noncontextual value assignment s is contradicted by $[\beta_\Psi] \neq 0$, and β_Ψ is an object that is well-defined in quantum mechanics. Beyond the contextuality witness (see Theorem 4), β_Ψ also contains the function computed in MBQC (see Theorem 5).

The counterpart of β_Ψ in the present iffy proof is the quantum-classical hybrid structure given by Eq. (16.32). It consists of the quantum-mechanically valid parts $\mathcal{C}^{(c_1)}$, $\beta_\Psi^{(c_1)}$, and one element, c_1, of the non-contextual value assignment, so far assumed to exist. (Recall that ruling out the existence of such a value assignment is the very purpose of the contextuality proof.) Unlike β_Ψ in the former cases, as a whole this hybrid object is not compatible with quantum mechanics. It is thus not suitable to base a description of MBQC on. Now that we have understood this, we seek to modify the iffy proof such that it becomes compatible with measurement-based quantum computation.

16.4.2 Deiffifying the Iffy Proof

Here we present a topological contextuality proof for the above iffy scenario that uses a complex of the type defined in Okay et al. (2017). The proof works in completely the same way as in the temporally flat scenarios it was previously applied to.

We define a couple of extra observables, for all $k \in \mathbb{Z}_{2N}$,

$$\epsilon_k := \frac{I^{(3)} + Y^{(3)}}{2} \otimes X^{(1)}_{k-1} + \frac{I^{(3)} - Y^{(3)}}{2} \otimes X^{(1)}_{k+1}, \tag{16.33a}$$

$$\sigma_k^+ := \frac{I^{(3)} + Y^{(3)}}{2} \otimes X^{(1)}_{k-1} + \frac{I^{(3)} - Y^{(3)}}{2} \otimes I^{(1)}, \tag{16.33b}$$

$$\sigma_k^- := \frac{I^{(3)} + Y^{(3)}}{2} \otimes I^{(1)} + \frac{I^{(3)} - Y^{(3)}}{2} \otimes X^{(1)}_{k+1}. \tag{16.33c}$$

These are correlated observables on qubits #1 and #3. They can also be considered as unitary gates in which qubit #3 is the control and qubit #1 the target. This is how the original iffiness enters into our topological proof, but in a fully quantum fashion.

The stabilizer relations Eq. (16.27) can be expressed in terms of the observables defined in Eq. (16.33). (only the first relation changes),

$$\epsilon_{N-k} \otimes X^{(2)}_k \, |\Psi\rangle = -|\Psi\rangle, \tag{16.34a}$$

$$X^{(1)}_{N-k} \otimes X^{(2)}_k X^{(3)} \, |\Psi\rangle = -|\Psi\rangle. \tag{16.34b}$$

Further, the observables ϵ_k, $\sigma_k^\pm$ satisfy the following *recoupling relations*:

$$\epsilon_k = \sigma_k^+ \sigma_k^-, \tag{16.35a}$$

$$X^{(1)}_k = \sigma_{k+1}^+ \sigma_{k-1}^-, \tag{16.35b}$$

$$-Y^{(3)} = \sigma_k^+ \sigma_{N+k}^+, \tag{16.35c}$$

$$Y^{(3)} = \sigma_k^- \sigma_{N+k}^-. \tag{16.35d}$$

Finally, we note the commutation relations

$$[\sigma_k^+, \sigma_l^-] = 0, \quad \forall k, l \in \mathbb{Z}_{2N}, \tag{16.36a}$$

$$[\sigma_k^\pm, Y^{(3)}] = 0, \quad \forall k, \in \mathbb{Z}_{2N}. \tag{16.36b}$$

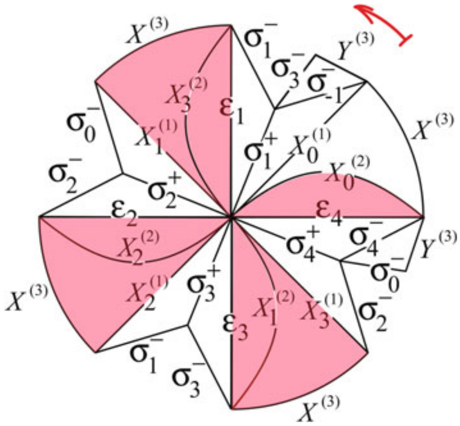


Fig. 16.4 Complex for the cohomological contextuality proof of the iffy scenario. There are four edges corresponding to $X^{(3)}$, and two each for $\sigma_k^{\pm}$, for various values of k, and for $Y^{(3)}$. Such edges are identified. The faces f coloured in red have $\beta_\Psi(f) = 1$, and the white faces g have $\beta_\Psi(g) = 0$

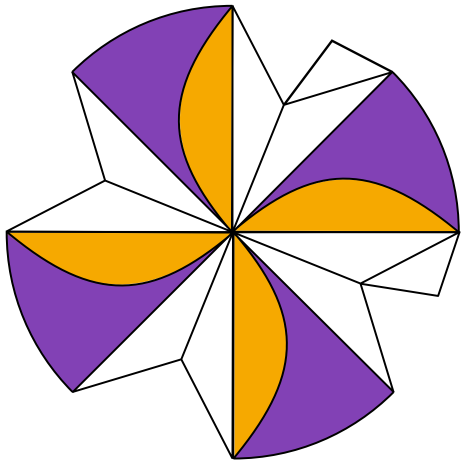


Fig. 16.5 The complex for the cohomological contextuality proof of the iffy scenario, in a different colouring. Orange: faces corresponding to the stabilizer relation Eq. (16.34a), purple: faces stemming from the stabilizer relation Eq. (16.34b), white: faces invoking the recoupling relations Eq. (16.35)

With these relations, the complex shown in Fig. 16.4 is well-composed. I.e., all faces correspond to triples of commuting operators that multiply to $\pm I$. Figure 16.5 shows how the above relations Eqs. (16.4) and (16.5) are represented in the chain complex.

We first consider the case of $N = 4$ which is displayed in Fig. 16.4, and then the general case. Denote $\mathcal{F} := \sum_i f_i$, i.e. $\mathcal{F}$ is the complete surface shown. It is easily verified that, after identifying the outer edges, $\partial\mathcal{F} = 0$. Further, there are 9 faces in $\mathcal{F}$ on which β_Ψ evaluates to 1, hence $\beta_\Psi(\mathcal{F}) = 1 \mod 2$. Both facts together imply that $[\beta_\Psi] \neq 0$, and hence the arrangement is contextual. □

We now turn to the general case of even N. If and only if N is even, there is an even number of edges labeled by $X^{(3)}$ in the boundary of the disc (shown in Fig. 16.4 for $N = 4$). Hence $\partial\mathcal{F} = 0 \mod 2$ if and only if N is even. We still need to establish $\beta_\Psi(\mathcal{F}) = 1 \mod 2$. So let's count the number of faces f with $\beta_\Psi(f) = 1$. Such faces arise through the relation of Eq. (16.34), and there are $2N$ of them. Hence their contribution cancels mod 2.

There is one more contribution to $\beta_\Psi(\mathcal{F})$. For guidance, we look at Fig. 16.4 and follow the red arrow in the counter-clockwise sense. The first observable we encounter that has non-trivial support only on qubit #1 is $X_0^{(1)}$. The next such observable is $X_1^{(1)}$, then $X_2^{(1)}$, and so forth. Going around the disk, we increase the value of k for such observables $X_k^{(1)}$ in increments of 1. Completing the circle, we arrive at $X_N^{(1)}$ which equals $-X_0^{(1)}$ by virtue of Eq. (16.25). $X_0^{(1)}$ already is the label of the start-stop edge, and hence we obtain an additional factor of -1 (That is why, in Fig. 16.4, the color of the last face before completing the circle is white, $\beta_\Psi(f_{\text{last}}) = 0$). We have thus overcounted the contributions stemming from Eq. (16.34) by 1, which we now correct for. There are no other contributions, hence $\beta_\Psi(\mathcal{F}) = 1$.

Now assume the existence of a value assignment $s = (a_k, b_l, c_m)$, i.e., $\beta_\Psi = ds$. Then,

$$1 = \beta_\Psi(\mathcal{F}) = ds(\mathcal{F}) = s(\partial_R \mathcal{F}) = s(0) = 0.$$

Contradiction. Thus, no non-contextual value assignment exists. □

To conclude, let's compare the above proof for the iffy scenario with the original iffy proof. The "iffiness" is gone. The algebraic structure Eq. (16.32) underlying the iffy proof is replaced by a simpler one, namely a relative chain complex $\mathcal{C}(N)$ with 2-cocycle $\beta_\Psi(N)$ living in it (N even). This is exactly the same structure as in the parity-based contextuality proofs without temporal order.

We achieved this reduction to the prior case by introducing additional observables in the chain complex, namely $\{\epsilon_k, \sigma_k^+, \sigma_k^-\}$ as defined in Eq. (16.33), to represent the temporal ordering. We propose this as a blueprint for a general method of constructing cohomological contextuality proofs describing temporally ordered measurement-based quantum computations.

16.5 Conclusion

In this paper, we have explained the contextuality–MBQC–cohomology triangle of Diagram (16.1). Its upper corners, contextuality and measurement-based quantum computation, represent the phenomenology of interest; and the lower corner, cohomology, the mathematical method to describe it. The link between MBQC and contextuality is provided by Theorems 1 and 2, the link between contextuality and cohomology by Theorem 4, and the link between MBQC and cohomology by Theorem 5 and Corollary 1. Finally, in the center of the diagram sits the cocycle class $[\beta_\Psi]$, an element of the second cohomology group of the underlying chain complex. It contains the function computed in a given MBQC up to gauge equivalence, and the corresponding contextuality proof.

A limitation of the cohomological framework established to date is that it only applies to temporally flat MBQCs, which form a small subclass. Here we made a first step towards describing MBQCs and contextuality proofs with temporal order in a cohomological fashion, by providing a cohomological contextuality proof in one concrete temporally ordered setting, the so-called "iffy" scenario (Abramsky et al., 2018). Extending the cohomological formalism to all MBQCs with proper temporal order is a main subject of future research on the MBQC-contextuality connection.

16.6 Travel Log

As I learned over the years, the 8th Conference on Quantum Physics and Logic, held in Nijmegen, the Netherlands in November 2011, is remembered fondly by many participants; for all sorts of reasons. Here I'd like to describe my journey towards this conference, how I spiralled out of it, and my thoughts for the future.

My journey began in Munich in 2003, the final year of my PhD. Hans Briegel and I had discovered the one-way quantum computer, a scheme of measurement-based quantum computation (as it is now known) in 2000, and had answered the obvious first question—universality. Quite naturally, the universality proof was based on a mapping to the circuit model. But, besides proving the point, the mapping seemed inadequate in many ways. For example, the temporal order among the measurements in MBQC was different and more flat than the mapping would suggest: all Clifford gates can be implemented in the first round of measurement, before all other gates, irrespective of where they are located in the simulated circuit. This and similar observations prompted us to look for a description of MBQC outside the realm of circuit simulation, and, in the first place, for the basic structures upon which such a description could be built.

There was, and is, no manual for how to approach this question. We are left to our own intuition and judgement. A structural element we focussed on early were the correlations among measurement outcomes that yield the computational result. Individually, the measurement outcomes in MBQC are completely random, and meaningful information can only be gleaned from certain correlations among them. What made the analysis of these correlations simultaneously difficult and interesting was their non-stabilizerness; i.e. the fact that the correlator observables are in general not mere tensor products of Pauli operators X, Y, Z.

Fault-tolerance seemed a path to make progress on these correlations. I figured that it could not be established for MBQC without understanding the structure of these correlations first. At the time, fault-tolerance with high error threshold was a problem with a price tag. In addition, when solved for MBQC we could sure learn something from the solution—a goldilocks problem.

When first putting non-stabilizer quantum correlations on my map in early 2003, unbeknownst to me, someone in far away Moscow was finding out something about them: Sergey Bravyi. The next year we would be office mates at Caltech.

Having arrived at Caltech in October 2003, it took about two years until, resting upon the scrap of two unsuccessful attempts, I established fault-tolerant universal MBQC with 3D cluster states (Raussendorf et al., 2006; Raussendorf & Harrington, 2007) (joint work with Jim Harrington and Kovid Goyal). Price tag fetched: the fault-tolerance threshold was high, and the whole construction elegant.

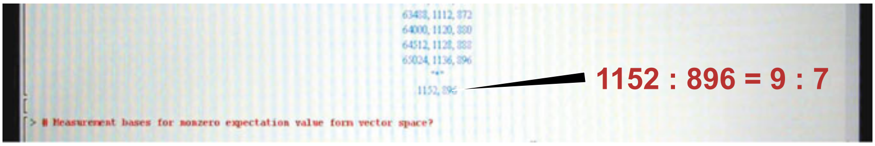


Fig. 16.6 Numerical experiment on toy MBQCs using Reed–Muller quantum code states as computational resources. Shown is the output for the example based on a 31-qubit punctured Reed–Muller code. All tests worked out—the Boolean function computed was total and non-linear

And yet, one thing didn't completely fall into place—the learning-from-the-solution part. As noted above, I had stipulated that in order to establish fault-tolerance for MBQC, the structure of the non-stabilizer correlations would need to be understood first. It panned out differently. Those correlations did not need to be understood, and I hadn't understood them. This realization is one of three waypoints encountered at Caltech on my journey to Nijmegen.

However, some correlations in MBQC—those which provide the error-correction capability for Clifford gates—could be understood very well. Namely, it turned out that those correlations have a cohomological underpinning. 3D cluster states can be described by a pair of three-dimensional chain complexes, related by Poincare duality. The measurement outcomes live on the respective faces, and are thus represented by 2-cochains s. The cluster state stabilizer implies that, in the absence of errors, the measurement record satisfies the constraint $s(\partial v) = 0$, for all volumes v, and hence s is a 2-cocycle. Furthermore, the output of the MBQC is given by evaluations $s(f)$, for non-trivial 2-cycles f. Fault-tolerance and computation on 3D cluster states is thus a matter of cohomology. This finding is the second Caltech waypoint.

In 2004, Sergey Bravyi and Alexei Kitaev developed "magic state distillation" (Bravyi & Kitaev, 2005), an efficient and robust technique for implementing non-Clifford gates fault-tolerantly. It was eventually incorporated into fault-tolerant MBQC, but its main effect on me was a different one. Magic state distillation exploits non-Pauli quantum correlations to operate, as they are found, for example, in Reed–Muller quantum codes. Save the aspect of temporal order, these were precisely the quantum correlations I wanted to understand in the first place!

A shortcut seemed to open: What about using quantum Reed–Muller code states as computational resource states in MBQC—could toy computations exhibiting non-trivial correlations be constructed this way? I was eager to try, and settled on the following conditions for Reed–Muller toy MBQCs: (i) The classical side processing relations Eq. (16.7) have to be obeyed; in particular, the input values form a vector space, as required by Eq. (16.7b). (ii) The outcome is deterministic for every admissible value of input, and (iii) the MBQC is non-Clifford. Further, the criterion for an "interesting" computation was that it computed a non-linear Boolean function. Quite a low bar, but justified as it exceeds what the classical side processing permits by itself.

Armed with those criteria, I got my laptop running. I started with the 15 qubit punctured Reed–Muller quantum code, and it didn't work. So I went on to the 31 qubit punctured Reed Muller code, which, given the next came at 63, I knew was the largest I could handle. I held my breath. There was deterministic output on 2048 inputs—power of 2, good sign. The output was imbalanced, hence the computed function non-linear (Fig. 16.6). A final check remained to be made: did the inputs form a vector space, as required by Eq. (16.7b)? That worked out too! I was over the moon.

Sometime in the subsequent months, while finalizing the fault-tolerance work, it must have trickled in that to be excited about such toy quantum computations needed a very particular taste or preparation. They didn't achieve anything of real computational value. At any rate, the finding of these Reed–Muller toy MBQCs is my third waypoint at Caltech.

In 2008, after I had moved to the University of British Columbia by way of the Perimeter Institute (PI), at a workshop at PI I heard Dan Browne speak about similar toy MBQCs. In a work of Janet Anders and him (Anders & Browne, 2009), they considered MBQCs on a Greenberger–Horne–Zeilinger state, satisfying the above conditions (i) and (ii). Not enforcing condition (iii) (non-Cliffordness) allowed them to get by with 3 qubits rather than 31. But much, much more importantly, they managed to relate their 3 qubit-MBQC to something known and valued in the world of Physics: Mermin's star. Thus the MBQC–contextuality link saw the light of day. Learning of this result I was ready to go to QPL 2011, although the conference was still 3 1/2 years ahead.

Finally, being at QPL 2011 in Nijmegen, what made the day for me was a talk by Samson Abramsky, Shane Mansfield and Rui Barbosa on "The Cohomology of Non-Locality and Contextuality". It had taken me quite a bit of effort to make it to the conference—teaching had to be rescheduled and so on. But I boarded the return plane in Amsterdam with a swagger: very, very worth the trouble. Although, honestly, in actual terms I had not learned all that much. I had understood precisely one slide of Shane Mansfield's presentation, and that was the title slide. What my journey through Caltech and PI had prepared me for was to see significance in the words "contextuality" and "cohomology" appearing side by side. I also somehow managed to not be completely bypassed by Mansfield's cohomological explanation of the GHZ scenario, at least in so far as I noted the argument's existence. Of course I tried to chase down Mansfield and Barbosa after their talk, but they seemed quite busy answering other calls.

For me, the upshot of Nijmegen was that a cohomological theory of MBQC was in range, making sense of all the known toy examples and hopefully beyond. To get started, all I needed to do was to get to grip with the Abramsky–Mansfield–Barbosa paper (Anders & Browne, 2012), which finally happened in the Spring of 2012. Then it turned out that their cohomological explanation of the GHZ example did not quite provide the desired connection to MBQC. The latter required a cohomological interpretation of precisely Mermin's argument for the GHZ-scenario, not merely a cohomological explanation of that scenario. And so, with my collaborators Cihan Okay, Stephen Bartlett, Sam Roberts and Emily Tyhurst, we set out to define our own cohomological framework. I do not need to describe the ensuing work here, since I already did in the previous sections.

This brings me to my thoughts for the future. Regarding measurement-based quantum computation, the recent investigations into its structure—contextuality as we discussed it here, computational phases of matter (Miyake, 2010; Else et al., 2012; Miller & Miyake, 2015; Raussendorf et al., 2019; Devakul & Williamson, 2018; Stephen et al., 2019; Devakul, 2019; Daniel et al., 2020) and temporal order (Raussendorf & Briegel, 2002; Browne et al., 2007; Raussendorf et al., 2012)—have to day remained separate. And yet they share a common ingredient at their cores: symmetry. I'm confident that these investigations will be unified into a single framework in the coming years, and that something new will spring from it.

To think about the future of our field more broadly, let's take a really long run-up and zoom right into the year 1842. Ada, Countess of Lovelace and assistant to the British computing pioneer Charles Babbage, had just invented the notion of the computer program. Also, at a time when everybody around her saw the future of computation in calculating trajectories of cannon balls, she had the fundamental insight that not only numbers can be processed by computers, but rather symbolic information of any kind—musical notes, images, text (Isaacson, 2014). Her insight lives on today in digital radio and television, the internet, Maple, the Google search engine, and countless other inventions of the information age.

But, quantum computation extends beyond this line of thought. Quantum information is not "symbolic". Due to the irreversibility of quantum measurement, it cannot be perceived by looking at it. And with the limits of the reigning paradigm exposed, a new era of computation can begin—at least in the skunkworks. On the theory side of it, whether one is thinking about measurement-based quantum computation or the circuit model, essentially everything boils down to one thing: quantum algorithms. Towering achievements such as Shor's factoring notwithstanding, we seem to have difficulty inventing new quantum algorithms, and it's a matter of intuition.

What would Ada's insight be today?

Acknowledgements The author thanks the Yukawa Institute for Theoretical Physics Kyoto (YITP) for their hospitality. Part of this work was performed there. This work is supported by NSERC.

LastPage

References

Abramsky, S., Barbosa, R. S., Carú, G., de Silva, N., Kishida, K., & Mansfield, S. (2018). Minimum quantum resources for strong non-locality. In *Proceedings of the 12th Conference on the Theory of Quantum Computation, Communication and Cryptography 2017*. Schloss Dagstuhl–Leibnitz-Zentrum für Informatik.

Abramsky, S., Barbosa, R. S., & Mansfield, S. (2017). The contextual fraction as a measure of contextuality. *Physical Review Letters, 119*, 050504.

Abramsky, S., & Brandenburger, A. (2011). The sheaf-theoretic structure of non-locality and contextuality. *New Journal of Physics, 13*, 113036.

Abramsky, S., Mansfield, S., & Barbosa, R. S. (2012). The cohomology of non-locality and contextuality. *EPTCS, 95*, 1.

Anders, J., & Browne, D. E. (2009). Computational power of correlations. *Physical Review Letters, 102*, 050502.

Bell, J. S. (1964). On the Einstein Podolsky Rosen paradox. *Physics, 1*, 195.

Bravyi, S., & Kitaev, A. (2005). Universal quantum computation with ideal Clifford gates and noisy ancillas. *Physical Review A, 71*, 022316.

Browne, D. E., Kashefi, E., Mhalla, M., & Perdrix, S. (2007). Generalized flow and determinism in measurement-based quantum computation. *New Journal of Physics, 99*, 250.

Cabello, A., Severini, S., & Winter, A. (2014). Graph-theoretic approach to quantum correlations. *Physical Review Letters, 112*, 040401.

Daniel, A. K., Alexander, R. N., & Miyake, A. (2020). Computational universality of symmetry-protected topologically ordered cluster phases on 2D Archimedean lattices. *Quantum, 4*, 228.

de Silva, N. (2017). Graph-theoretic strengths of contextuality. *Physical Review A, 95*, 032108.

Devakul, T., & Williamson, D. J. (2018). Universal quantum computation using fractal symmetry-protected cluster phases. *Physical Review A, 98*, 022332.

Devakul, T. (2019). Classifying local fractal subsystem symmetry-protected topological phases. *Physical Review B, 99*, 235131.

Einstein, A., Podolsky, B., & Rosen, N. (1935). Can quantum-mechanical description of physical reality be considered complete? *Physical Review, 47*, 777.

Else, D. V., Schwarz, I., Bartlett, S. D., & Doherty, A. C. (2012). Symmetry-protected phases for measurement-based quantum computation. *Physical Review Letters, 108*, 240505.

Fine, A. (1982). Hidden variables, joint probability, and the Bell inequalities. *Physical Review Letters, 48*, 291.

Greenberger, D. M., Horne, M. A., & Zeilinger, A. (1989) In Kafatos, M., (Eds.), *Bell's theorem, quantum theory, and conceptions of the universe* (pp. 69–72). Kluwer.

Isaacson, W. (2014). *The innovators*. Simon & Schuster.

Isham, C. J., & Döring, A. (2011). What is a thing?: Topos theory in the foundations of physics. In *New structures for physics*. Lecture notes in physics (Vol. 813, p. 753). Berlin: Springer.

Kochen, S., & Specker, E. P. (1967). The problem of hidden variables in quantum mechanics. *Journal of Mathematics and Mechanics, 17*, 59.

MacWilliams, F. J., & Sloane, N. A. (1977). *The theory of error-correcting codes*. Amsterdam: North-Holland.

Mermin, N. D. (1993). Hidden variables and the two theorems of John Bell. *Reviews of Modern Physics, 65*, 803.

Miller, J., & Miyake, A. (2015). Resource quality of a symmetry-protected topologically ordered phase for quantum computation. *Physical Review Letters, 114*, 120506.

Miyake, A. (2010). Quantum computation on the edge of a symmetry-protected topological order. *Physical Review Letters, 105*, 040501.

Mosca, M., & Zalka, C. (2003). ERATO Conference on Quantum Information Science (EQUIS), Kyoto, Japan. arXiv:quant-ph/0301093

Okay, C., Roberts, S., Bartlett, S. D., & Raussendorf, R. (2017). Topological proofs of contextuality in quantum mechanics. *Quantum Information and Computation, 17*, 1135–1166.

Raussendorf, R., Harrington, J., & Goyal, K. (2006). A fault-tolerant one-way quantum computer. *Annals of Physics (N.Y.),321*, 2242.

Raussendorf, R., Sarvepalli, P., Wei, T.-C., & Haghnegahdar, P. (2012). Symmetry constraints on temporal order in measurement-based quantum computation. In *8th on International Workshop Quantum Physics and Logic (QPL 2011), EPTCS 95* (pp. 219–250). https://doi.org/10.4204/EPTCS.95.16

Raussendorf, R. (2013). Contextuality in measurement-based quantum computation. *Physical Review A, 88*, 022322.

Raussendorf, R. (2019). Cohomological framework for contextual quantum computations. *Quantum Information and Computation, 19*, 1141–1170.

Raussendorf, R., & Briegel, H. J. (2001). A one-way quantum computer. *Physical Review Letters, 86*, 5188.

Raussendorf, R., & Briegel, H. J. (2002). Computational model underlying the one-way quantum computer. *Quantum Information and Computation, 6*, 443.

Raussendorf, R., & Harrington, J. (2007). Fault-tolerant quantum computation with high threshold in two dimensions. *Physical Review Letters, 98*, 190504.

Raussendorf, R., Okay, C., Wang, D. S., Stephen, D. T., & Nautrup, H. P. (2019). A computationally universal quantum phase of matter. *Physical Review Letters, 122*, 090501.

Raussendorf, R., & Wei, T.-C. (2012). Quantum computation by local measurement. *Annual Review of Condensed Matter Physics, 3*, 239.

Spekkens, R. W. (2008). Negativity and contextuality are equivalent notions of nonclassicality. *Physical Review Letters, 101*, 020401.

Stephen, D. T., Nautrup, H. P., Bermejo-Vega, J., Eisert, J., & Raussendorf, R. (2019). Subsystem symmetries, quantum cellular automata, and computational phases of quantum matter. *Quantum, 3*, 162.

Van den Nest, M., Dür, W., Vidal, G., & Briegel, H. J. (2007). Classical simulation versus universality in measurement based quantum computation. *Physical Review A, 75*, 012337.

Chapter 17
Consistency, Acyclicity, and Positive Semirings

Albert Atserias and Phokion G. Kolaitis

Abstract In several different settings, one comes across situations in which the objects of study are locally consistent but globally inconsistent. Earlier work about probability distributions by Vorob'ev (1962) and about database relations by Beeri et al. (1983) produced characterizations of when local consistency always implies global consistency. Towards a common generalization of these results, we consider K-relations, that is, relations over a set of attributes such that each tuple in the relation is associated with an element from an arbitrary, but fixed, positive semiring K. We introduce the notions of projection of a K-relation, consistency of two K-relations up to normalization, and global consistency of a collection of K-relations; these notions are natural extensions of the corresponding notions about probability distributions and database relations. We then show that a collection of sets of attributes has the property that every pairwise consistent collection of K-relations over those attributes is globally consistent if and only if the sets of attributes form an acyclic hypergraph. This generalizes the aforementioned results by Vorob'ev and by Beeri et al., and demonstrates that K-relations over positive semirings constitute a natural framework for the study of the interplay between local and global consistency. In the course of the proof, we introduce a notion of join of two K-relations and argue that it is the "right" generalization of the join of two database relations. Furthermore, to show that non-acyclic hypergraphs yield pairwise consistent K-relations that are globally inconsistent, we generalize a construction by Tseitin (1968) in his study of hard-to-prove tautologies in propositional logic.

Keywords Local consistency · Global consistency · Acyclic hypergraphs · Semirings · Contextuality · Non-locality

A. Atserias
Universitat Politècnica de Catalunya, Barcelona, Spain
e-mail: atserias@cs.upc.edu

P. G. Kolaitis (✉)
University of California Santa Cruz and IBM Research, Santa Cruz, CA, USA
e-mail: kolaitis@soe.ucsc.edu

A. Palmigiano and M. Sadrzadeh (eds.), *Samson Abramsky on Logic and Structure in Computer Science and Beyond*, Outstanding Contributions to Logic 25,
https://doi.org/10.1007/978-3-031-24117-8_17

17.1 Introduction

There are many situations, spanning art and science, in which the objects under consideration are locally consistent but globally inconsistent, where the terms “local”, “global”, and “consistent” are used in some intuitive sense but can be made precise in each concrete setting. In art, Escher’s 1960 *Ascending and Descending* and 1961 *Waterfall* lithographs are striking depictions of locally consistent but globally inconsistent situations. Closely related to Escher’s artwork is the work by Penrose and Penrose (1958) on impossible objects, such as the impossible tribar (see also Francis, 2007). In quantum mechanics, the interplay between local consistency and global inconsistency takes the form of non-locality and contextuality phenomena, where collections of empirical local measurements may not admit a global explanation via a hidden variable; prominent results in this area include Bell’s Theorem (Bell, 1964) and Hardy’s paradox (Hardy, 1992). In probability theory, there is work on when a given collection of pairwise consistent probability distributions admits a global distribution whose marginal distributions coincide with the given collection (Vorob’ev, 1962). In computer science, the interplay between local consistency and global consistency arises in such different areas as constraint satisfaction (Dechter, 2003), proof complexity (Chvátal & Szemerédi, 1988), and relational databases (Beeri et al., 1983).

What do the aforementioned situations have in common and is there a unifying framework behind them? Abramsky (2013, 2014) pointed out that there are formal connections between non-locality and contextuality in quantum mechanics on one side and the universal relation problem in database theory on the other side. The latter is the following decision problem: given a collection $X_1, \ldots, X_m$ of sets of attributes (that is, names of columns of relations) and a collection $R_1, \ldots, R_m$ of relations over $X_1, \ldots, X_m$ (that is, X_i is the set of the attributes of R_i, for $i \in [m]$), are the relations $R_1, \ldots, R_m$ globally consistent? In other words, is there a relation R, called a *universal* relation, over $X_1 \cup \cdots \cup X_m$ such that, for every $i \in [m]$, the projection $R[X_i]$ of R on X_i is equal to R_i? Clearly, if such a universal relation exists, then the relations $R_1, \ldots, R_m$ are *pairwise consistent*, i.e., $R_i[X_i \cap X_j] = R_j[X_i \cap X_j]$, for all $i, j \in [m]$, but the converse need not hold. Switching to the quantum mechanics side and by regarding the collection of empirical measurements in Hardy (1992) as a collection of database relations, Hardy’s paradox can be viewed as a negative instance of the universal relation problem: the database relations at hand are pairwise consistent, but globally inconsistent. Note that, since experiments are typically repeated, measurements give rise to probabilities. This way, Bell’s Theorem (Bell, 1964) can be viewed as an instance of a collection of probability distributions that are pairwise consistent, but globally inconsistent. As regards unifying frameworks, (Abramsky & Brandenburger, 2011) used sheaf theory to provide a unified account of non-locality and contextuality. This approach was explored further in Abramsky et al. (2011, 2015).

As mentioned in the preceding paragraph, pairwise consistency is a necessary, but not sufficient, condition for a collection of relations to be globally consistent.

In the setting of relational databases, Beeri et al. (1983) characterized when pairwise consistency is also a sufficient condition for global consistency. If $X_1, \ldots, X_m$ are sets of attributes, we say that the collection $X_1, \ldots, X_m$ has the *local-to-global consistency property* if every collection $R_1, \ldots, R_m$ of pairwise consistent relations over $X_1, \ldots, X_m$ is globally consistent. The main finding in Beeri et al. (1983) is that a collection $X_1, \ldots, X_m$ of sets of attributes has the local-to-global consistency property if and only if the hypergraph with $X_1, \ldots, X_m$ as hyperedges is acyclic, where the notion of hypergraph acyclicity is a suitable generalization of the notion of graph acyclicity. Observe that the local-to-global consistency property is a semantic property (in the sense that its definition involves relations over the sets of attributes), while acyclicity is a syntactic property (in the sense that it describes a structural property of hypergraphs with no reference to relations). In Beeri et al. (1983), several other syntactic conditions on hypergraphs were considered, and each was shown to be equivalent to acyclicity. In the setting of probability theory, Vorob'ev (1962) identified a different syntactic condition on hypergraphs, which we call *Vorob'ev regularity*, and showed that a collection of probability distributions over $X_1, \ldots, X_m$ has the local-to-global consistency property (suitably adapted to probability distributions) if and only the hypergraph with $X_1, \ldots, X_m$ as hyperedges is Vorob'ev regular. It is perhaps worth noting that Vorob'ev's paper (Vorob'ev, 1962) was published much earlier, but Beeri et al. (1983) were apparently unaware of Vorob'ev's work. It is now natural to ask: is there a common generalization of the above results? This question was investigated by Barbosa in his doctoral thesis (Barbosa, 2015, Chap. VI). Barbosa explored the question in the sheaf-theoretic framework for non-locality and contextuality and showed that hypergraph acyclicity implies the local-to-global consistency property in that framework, but did not obtain the reverse direction.

We establish a common generalization of the results by Vorob'ev (1962) and Beeri et al. (1983). Instead of the sheaf-theoretic framework, we work in the algebraic framework of *positive semirings*, which are commutative semirings with no zero-divisors and with the property that $a + b = 0$ holds for two elements a and b of the semiring if and only if $a = b = 0$. Positive semirings were used to study the provenance of relational database queries (Green et al., 2007) and also the provenance of first-order sentences (Grädel & Tannen, 2017); furthermore, commutative semirings were considered by Abramsky (2013) in discussing algebraic databases as a generalization of relational databases.

Let K be a positive semiring. As a common generalization of database relations and probability distributions, we consider *K-relations*, i.e., relations over a set of attributes such that each tuple in the relation has an associated element from K as value. Note that ordinary relations are K-relations where K is the Boolean semiring, while probability distributions are K-relations with K-values adding to 1 and where K is the semiring of the non-negative real numbers. We introduce natural extensions of the notions of projection of a K-relation, pairwise consistency up to normalization, global consistency, and the local-to-global consistency property for K-relations. We then show that a collection $X_1, \ldots, X_m$ of sets of attributes has the local-to-global consistency property for K-relations if and only if the hypergraph with $X_1, \ldots, X_m$ as hyperedges is acyclic. We also show that a hypergraph is

Vorob'ev regular if and only if it is acyclic (this result has been mentioned in passing or has been taken for granted in earlier papers, but we have not found an explicit reference for it). The results by Vorob'ev (1962) and Beeri et al. (1983) then follow as immediate corollaries.

While the proof of our main result about the equivalence between hypergraph acyclicity and the local-to-global consistency property for K-relations bears some similarities and analogies with the earlier proofs of its special cases, it also brings in some new concepts and tools that may be of independent interest. We conclude this section by highlighting some of these concepts and tools.

To prove that hypergraph acyclicity implies the local-to-global consistency property for K-relations, we introduce a *join* operation on K-relations. We make the case that this is the "right" extension to K-relations of the notion of the join of two ordinary relations. In particular, we show that the join of two consistent K-relations witnesses their consistency and also that the basic results about lossless-join decompositions of ordinary relations extend to K-relations. Note that if K is the semiring of non-negative integers, then the K-relations are precisely the *bags* (also known as *multisets*). Our join operation on bags is, in general, different from the standard bag join used in SQL (for bag operations in SQL, see Ullman & Widom, 2002). We point out, however, that unlike the join operation introduced here, the standard bag join does not always witness the consistency of two consistent bags. Furthermore, we show that the join of two consistent probability distributions is the unique probability distribution that maximizes entropy among all probability distributions that witness the consistency.

To prove that the local-to-global consistency property for K-relations implies hypergraph acyclicity, we need to have a systematic way to produce negative instances of the universal relation problem, such as the instances found in Hardy's paradox and related constructions in the study of non-locality and contextuality. Note that, in our setting, we need the relations in the negative instances to be K-relations where K is an *arbitrary* positive semiring, instead of ordinary relations over the Boolean semiring or probability distributions over the semiring of nonnegative real numbers; furthermore, we need to be able to produce such negative instances for *any* given cyclic collection $X_1, \ldots, X_m$ of sets of attributes. For the special cases of ordinary relations and probability distributions, Beeri et al. (1983) and Vorob'ev (1962) provided suitable such constructions, which, as far as we can tell, do not generalize to arbitrary positive semirings. For our construction, which works for an arbitrary positive semiring, we adapt an idea that can be traced to Tseitin (1968) in his study of hard-to-prove tautologies in propositional logic. In brief, Tseitin constructed arbitrarily large sets of propositional clauses such that any fixed number of them are satisfiable, but, when taken jointly, they are unsatisfiable. The combinatorial principle underlying Tseitin's construction is the following basic *parity principle*: for every undirected graph and for every labeling of the vertices of the graph with 0's and 1's with an odd total number of 1's, there is no subset of the edges that touches every vertex a number of times that is congruent to the label of the vertex modulo 2. To generalize this to arbitrary cyclic hypergraphs and to arbitrary semirings, we resort to a similar modular counting principle for a modulus $d \geq 2$ that depends on the structure of the hyperedges $X_1, \ldots, X_m$. While similar but different variations of Tseitin's construction have been used in other contexts (see, e.g., Buss et al., 2001 and Atserias et al., 2009), we are not aware of any other construction that simultaneously generalizes the results in Beeri et al. (1983) and Vorob'ev (1962). Furthermore, it is worth noting that our construction contains as a special case the most basic Popescu-Rorhlich box (Popescu & Rorhlich, 1994), which is another well-known example of non-locality and contextuality (see, e.g., Abramsky & Brandenburger, 2011). Specifically, the support of the Popescu-Rorhlich box is precisely the special case of our construction in which the hypergraph $X_1, \ldots, X_m$ is the 4-cycle AB, BC, CD, DA on the four vertices A, B, C, D.

17.2 Valued Relations up to Normalization

In this section we define the notion of *valued relation*, or *K-relation* for a positive semiring K of values, as a generalization of the database-theoretic notion of relation. We study its most basic properties and discuss some examples. Besides the standard concept of ordinary relation from database theory, two other canonical examples will be the *bags* and the *probability distributions*.

17.2.1 Basic Properties of Valued Relations

We start by recalling some basic terminology and notation from the theory of databases. While most of our notation is standard and well-established, we refer to the textbooks Ullman (1988) and Abiteboul et al. (1995) for further elaboration.

Attributes, Tuples, and Relations An *attribute* A is a symbol with an associated set $\mathrm{Dom}(A)$ called its *domain*. If X is a finite set of attributes, then we write $\mathrm{Tup}(X)$ for the set of *X-tuples*; i.e., $\mathrm{Tup}(X)$ is the set of maps that take each attribute $A \in X$ to an element of its domain $\mathrm{Dom}(A)$. Note that $\mathrm{Tup}(\emptyset)$ is non-empty as it contains the *empty tuple*, i.e., the unique map with empty domain. If $Y \subseteq X$ is a subset of attributes and t is an X-tuple, then the *projection of t on Y*, denoted by $t[Y]$, is the unique Y-tuple that agrees with t on Y. In particular, $t[\emptyset]$ is the empty tuple.

A *relation over X* is a subset of $\mathrm{Tup}(X)$; it is a finite relation if it is a finite subset of $\mathrm{Tup}(X)$. In what follows, we will often refer to such relations as *ordinary* relations to differentiate them from K-relations, where K is a positive semiring other than the Boolean semiring. We write $R(X)$ to emphasize the fact that the relation R has *schema* X. In this paper all sets of attributes and all relations are finite, so we omit the term. If $Y \subseteq X$ and R is a relation over X, then the *projection of R on Y*, denoted $R[Y]$, is the relation over Y made of all the projections $t[Y]$ as t ranges over R. If R is a relation over X and S is a relation over Y, then their *join* $R \bowtie S$ is the relation over $X \cup Y$ made of all the $X \cup Y$-tuples t such that $t[X]$ is in R and $t[Y]$ is in S.

If X and Y are sets of attributes, then we write XY as shorthand for the union $X \cup Y$. Accordingly, if x is an X-tuple and y is a Y-tuple with the property that $x[X \cap Y] = y[X \cap Y]$, then we write xy to denote the XY-tuple that agrees with x on X and on y on Y. We say that x *joins with* y, and that y *joins with* x, to *produce* the tuple xy.

Positive Semirings A *commutative semiring* is a set K with two binary operations $+$ and $\times$ that are commutative, associative, have 0 and 1, respectively, as identity elements, $\times$ distributes over $+$, and 0 *annihilates* K, that is, $0 \times a = a \times 0 = 0$ holds for all $a \in K$. We assume that $0 \neq 1$, that is, the semiring is *non-trivial*. The identity of multiplication 1 is also called the *unit* of the semiring. We write multiplication $a \times b$ by concatenation ab or with a dot $a \cdot b$. If there do not exist non-zero a and b in K such that $a + b = 0$, then we say that K is *plus-positive*. If there do not exist non-zero a and b in K such that $ab = 0$, then we say that K *has no zero-divisors*, or that K is a semiring *without zero-divisors*. A plus-positive commutative semiring without zero-divisors is called *positive*. In the sequel, K will always denote a non-trivial positive commutative semiring.

We introduce some examples. The *Boolean semiring* $\mathbb{B} = (\{0, 1\}, \vee, \wedge, 0, 1)$ has 0 (false) and 1 (true) as elements, and disjunction ($\vee$) and conjunction ($\wedge$) as operations. This is a commutative semiring that is plus-positive and has no zero-divisors, hence it is positive; it is not a ring since disjunction does not have an inverse. The non-negative integers $\mathbb{Z}^{\geq 0}$, the non-negative rationals $\mathbb{Q}^{\geq 0}$, and the non-negative reals $\mathbb{R}^{\geq 0}$ with their usual arithmetic operations $+$ and $\times$ and their identity elements 0 and 1 are also positive semirings. In contrast, the full integers $\mathbb{Z}$, the rationals $\mathbb{Q}$, or the reals $\mathbb{R}$ are commutative semirings without zero-divisors that are not plus-positive. The semiring of non-negative integers $\mathbb{Z}^{\geq 0}$ is also denoted by $\mathbb{N}$, and it is called the *bag semiring*. For an integer $m \geq 2$, the semiring $\mathbb{Z}_m$ of arithmetic mod m, also denoted by $\mathbb{Z}/m\mathbb{Z}$, is a commutative semiring that is not plus-positive, and that has no zero-divisors if and only if m is prime; $\mathbb{Z}_1$ is not even non-trivial.

Under the convention that $0 < 1$, the disjunction $\vee$ and conjunction $\wedge$ operations of the Boolean semiring can also be written as max and min, respectively. Semirings over arithmetic ordered domains that combine the max or min operations with the usual arithmetic operations are called *tropical* semirings. The *min-plus* semiring has the extended reals $\mathbb{R} \cup \{+\infty, -\infty\}$ as elements, and the standard operations of minimum and addition for $+$ and $\times$, with $+\infty$ playing the role of the identity for min. The *positive min-plus* semiring has the extended positive reals $\mathbb{R}^{\geq 0} \cup \{+\infty\}$ as elements, and again standard minimum and addition for $+$ and $\times$, with $+\infty$ playing again the role of identity for min. The *Viterbi* semiring has elements ranging over the unit interval $[0, 1]$, and the standard operations of maximum and multiplication for $+$ and $\times$, respectively. The *rational tropical* semirings are based on the extended rational numbers in place of the extended real numbers.

Definition of K-relations and their Marginals Let $K = (K^*, +, \times, 0, 1)$ be a semiring and let X be a finite set of attributes. A *K-relation over X* is a map R : $\mathrm{Tup}(X) \to K$ that assigns a value $R(t)$ in K to every X-tuple t in $\mathrm{Tup}(X)$. Note that this definition makes sense even if X is the empty set of attributes; in such a case, a K-relation over X is simply a single value from K that is assigned to the empty tuple. Note also that the ordinary relation are precisely the $\mathbb{B}$-relation, where $\mathbb{B}$ is the Boolean semiring.

The *support* of the K-relation R, denoted by Supp(R), is the set of X-tuples t that are assigned non-zero value, i.e.,

$$\text{Supp}(R) := \{t \in \text{Tup}(X) : R(t) \neq 0\}. \tag{17.1}$$

Whenever this does not lead to confusion, we write R' to denote Supp(R). Note that R' is an ordinary relation over X. A K-relation is *finitely supported* if its support is a finite set. In this paper, all K-relations are finitely supported and we omit the term. When R' is empty, we say that R is the empty K-relation over X. For $a \in K$, we write aR to denote the K-relation over X defined by $(aR)(t) = aR(t)$ for every X-tuple t. It is always the case that Supp$(aR) \subseteq$ Supp(R), and, as the proof of the next lemma shows, the reverse inclusion Supp$(R) \subseteq$ Supp(aR) also holds in case a is a non-zero element of K and K has no zero-divisors.

If $Y \subseteq X$, then the *marginal of* R *on* Y, denoted by $R[Y]$, is the K-relation over Y such that for every Y-tuple t, we have that

$$R[Y](t) := \sum_{\substack{r \in R': \\ r[Y]=t}} R(r). \tag{17.2}$$

The value $R[Y](t)$ is called the *marginal of* R *over* t. In what follows and for notational simplicity, we will often write $R(t)$ for the marginal of R over t, instead of $R[Y](t)$. It will be clear from the context (e.g., from the arity of the tuple t) if $R(t)$ is indeed the marginal of R over t (in which case t must be a Y-tuple) or $R(t)$ is the actual value of R on t as a mapping from Tup(X) to K (in which case t must be a X-tuple).

Note that if R is an ordinary relation (i.e., R is a $\mathbb{B}$-relation), then the marginal $R[Y]$ is the projection of R on Y, so the notation for the marginal is consistent with the one introduced for the projection earlier. It is always the case that Supp$(R[Y]) \subseteq$ Supp$(R)[Y]$, and, as the proof of the next lemma shows, the reverse inclusion also holds in case the semiring K is plus-positive.

From now on, we make the blanket assumption that K is a positive semiring. This hypothesis will not be explicitly spelled out in the statements of the various lemmas in which K-relations are mentioned.

Lemma 1 *Let $R(X)$ be a K-relation. The following statements hold:*

1. *For all non-zero elements a in K, we have $(aR)' = R'$.*
2. *For all $Y \subseteq X$, we have $R'[Y] = R[Y]'$.*
3. *For all $Z \subseteq Y \subseteq X$, we have $R[Y][Z] = R[Z]$.*

Proof For 1, the inclusion $(aR)' \subseteq R'$ holds for all semirings since if $t \in (aR)'$, then $aR(t) \neq 0$, so $R(t) \neq 0$ since 0 annihilates K, and hence $t \in R'$. For the converse, if $t \in R'$, then $R(t) \neq 0$, so $aR(t) \neq 0$ since a is non-zero and K has no zero-divisors, and hence $t \in (aR)'$. For 2, the inclusion $R[Y]' \subseteq R'[Y]$ is obvious and holds for all semirings. For the converse, assume that $t \in R'[Y]$, so there exists r such that $R(r) \neq 0$ and $r[Y] = t$. By (17.2) and the plus-positivity of K we have that $R(t) \neq 0$. Hence $t \in R[Y]'$. For 3, we have

$$R[Y][Z](u) = \sum_{\substack{v \in R[Y]': \\ v[Z]=u}} R[Y](v) = \sum_{\substack{v \in R'[Y]: \\ v[Z]=u}} \sum_{\substack{w \in R': \\ w[Y]=v}} R(w) = \sum_{\substack{w \in R': \\ w[Z]=u}} R(w) = R[Z](u) \tag{17.3}$$

where the first equality follows from (17.2), the second follows from Part 2 of this lemma to replace $R[Y]'$ by $R'[Y]$, and again (17.2), the third follows from partitioning the tuples in R' by their projection on Y, together with $Z \subseteq Y$, and the fourth follows from (17.2) again. □

Some examples follow.

Example 1 When K is the Boolean semiring $\mathbb{B}$, a $\mathbb{B}$-relation over X is simply an ordinary relation over X; its support is the relation itself, and its marginals are the ordinary projections. When K is the bag semiring $\mathbb{N}$, the $\mathbb{N}$-relations are called *bags* or *multi-sets*. If T is a bag and t is a tuple in its support, then $T(t)$ is called the *multiplicity* of t in T. When K is the semiring of non-negative reals $\mathbb{R}^{\geq 0}$, the finite $\mathbb{R}^{\geq 0}$-relations T that satisfy

$$T[\emptyset] = \sum_{t \in T'} T(t) = 1 \tag{17.4}$$

are the probability distributions of finite support over the set $\mathrm{Tup}(X)$ of X-tuples, or, in short, the *probability distributions over* X. Conversely, to every finite non-empty $\mathbb{R}^{\geq 0}$-relation T one can associate a probability distribution T^* through *normalization*; this means that if we set $N_T := \sum_{t \in T'} T(t)$ and $n_T = 1/N_T$, then the $\mathbb{R}^{\geq 0}$-relation $T^* := n_T T$ is a probability distribution. Finally, when K is the semiring of non-negative rationals $\mathbb{Q}^{\geq 0}$, the corresponding probability distributions are called *rational* probability distributions.

17.2.2 Equivalence of K-Relations

We introduce a notion of equivalence between two K-relations over the same set of attributes that will play an important role in the later sections of this paper. To motivate this definition, let us look again at the probability distributions seen as the $\mathbb{R}^{\geq 0}$-relations that satisfy the normalization Eq. (17.4) from Example 1.

Derivation of the Equivalence Relation Recall that to every non-empty $\mathbb{R}^{\geq 0}$-relation T one can associate a probability distribution T^* via normalization $T \mapsto T^*$. More generally, a normalization operation can be defined for any semiring K that is actually a *semifield*, which is a semiring whose multiplication operation $\times$ admits an inverse $\div$. Note that both $\mathbb{R}^{\geq 0}$ and $\mathbb{Q}^{\geq 0}$ are semifields. Formally, if K is a semifield and T is a non-empty K-relation over a set of attributes X, then we define T^* as the K-relation defined by $T^*(t) := (1/N_T)T(t)$ for every X-tuple t, where $N_T := T[\emptyset] = \sum_{t \in T'} T(t)$, and $1/N_T$ is the multiplicative inverse of N_T in the semifield K. Note that T was assumed non-empty, so $N_T \neq 0$ since K is plus-positive, and the multiplicative inverse $1/N_T$ exists. When T is the empty K-relation, we let T^* be the empty K-relation itself.

With this definition in hand, still assuming that K is a semifield, we can define an equivalence relation $R \equiv S$ to hold between two K-relations R and S if and only if $R^* = S^*$. An important observation that follows from the definitions is that if R and S are K-relations over the same set of attributes, then $R^* = S^*$ holds if and only if $aR = bS$ for some non-zero a and b in K. For the *only if* direction just take $a = 1/N_R$ and $b = 1/N_S$ if both R and S are non-empty K-relations, and $a = b = 1$ otherwise. For the *if* direction, assuming that R and S are both non-empty K-relations over X, for every X-tuple t we have

$$R^*(t) = \frac{1}{N_R}R(t) = \frac{a}{aN_R}R(t) = \frac{b}{bN_S}S(t) = \frac{1}{N_S}S(t) = S^*(t), \tag{17.5}$$

where the first equality follows from the fact that R^* is defined from R through normalization, the second follows from the assumption that a is non-zero, the third follows from the assumption that $aR = bS$, so, in particular, $aR(t) = bS(t)$ and also $R' = S'$ and $aN_R = bN_S$, the fourth follows from the assumption that b is non-zero, and the last follows from the definition of S^* through normalization. This observation motivates the following definition of the equivalence relation $\equiv$ for arbitrary positive semirings that are not necessarily semifields.

Definition of the Equivalence Relation for Positive Semirings Let K be a positive semiring. Two K-relations R and S over the same set of attributes are *equivalent up to normalization*, denoted by $R \equiv S$, if there exist non-zero a and b in K such that $aR = bS$. It is obvious that $\equiv$ is reflexive and symmetric. The next lemma collects a few easy facts about $\equiv$, the first of which states that $\equiv$ is also transitive, and hence an equivalence relation. We write $[R]$ for the equivalence class of R under $\equiv$.

Lemma 2 *Let $R(X)$, $S(X)$, $T(X)$ be K-relations over the same set X of attributes. The following statements hold:*

1. *If $R \equiv S$ and $S \equiv T$, then $R \equiv T$.*
2. *If $R \equiv S$, then $R' = S'$.*
3. *If $R \equiv S$ and $Y \subseteq X$, then $R[Y] \equiv S[Y]$.*

Proof For 1, assume that $aR = bS$ and $cS = dT$ for non-zero a and b in K, and non-zero c and d in K. Since K has no zero-divisors we have that ac and bd are non-zero. Moreover,

$$acR = caR = cbS = bcS = bdT, \tag{17.6}$$

where the first equality is commutativity, the second follows from $aR = bS$, the third is commutativity, and the last follows from $cS = dT$. For 2, assume that $aR = bS$ for non-zero a and b, and that $R(t) \neq 0$ for some X-tuple t. Then $aR(t) \neq 0$ because K has no zero-divisors, hence $bS(t) \neq 0$ by the assumption that $aR = bS$, and $S(t) \neq 0$ since 0 annihilates K. This shows $R' \subseteq S'$ and the reverse inclusion follows from symmetry. For 3, assume that $aR = bS$ for non-zero a and b and that $Y \subseteq X$. For every Y-tuple u we have

$$aR(u) = a \sum_{\substack{r \in R': \\ r[Y]=u}} R(r) = \sum_{\substack{r \in R': \\ r[Y]=u}} aR(r) = \sum_{\substack{r \in R': \\ r[Y]=u}} bS(r) = \sum_{\substack{s \in S': \\ s[Y]=u}} bS(r) = b \sum_{\substack{s \in S': \\ s[Y]=u}} S(r) = bS(u), \tag{17.7}$$

where the first equality follows from (17.2), the second is distributivity, the third follows from the assumption that $aR = bS$, the fourth follows from point 2 in this lemma, the fifth is again distributivity, and the sixth is (17.2). □

17.3 Consistency of Two K-Relations

For ordinary relations $R(X)$ and $S(Y)$, there are several different ways to define the concept of R and S being *consistent*, and all these concepts turn out to be equivalent to each other. One way is to say that R and S arise as the projections $T[X]$ and $T[Y]$ of a single relation T over the union of attributes XY. Another way is to say that R and S agree on their projections to the set $Z = X \cap Y$ of their common attributes. Yet a third way is to say that their ordinary join $R \bowtie S$ projects to R on X and to S on Y. In this section, we study the analogous concepts for K-relations with consistency defined up to normalization. Along the way, we will also define a notion of $\bowtie$ for two K-relations.

17.3.1 Consistency of Two K-Relations up to Normalization

Let K be an arbitrary but fixed positive semiring. We start with the definition of consistency up to normalization

Let $R(X)$ and $S(Y)$ be two K-relations. We say that R and S are *consistent up to normalization* if there is a K-relation $T(XY)$ such that $R \equiv T[X]$ and $S \equiv T[Y]$. We also say that T *witnesses* their consistency up to normalization. Two equivalence classes $[R]$ and $[S]$ of K-relations are *consistent up to normalization* if their representatives R and S are consistent up to normalization. It is easy to see that this notion of consistency among equivalence classes is well defined in that it does not depend on the chosen representatives R and S. Indeed, if $[R]$ and $[S]$ are consistent up to normalization and T witnesses the consistency of R and S up to normalization, then for every $R_0 \equiv R$ and every $S_0 \equiv S$, we have that T also witnesses the consistency of R_0 and S_0 up to normalization, by the transitivity of $\equiv$. Conversely, if T_0 witnesses the consistency of $R_0 \equiv R$ and $S_0 \equiv S$ up to normalization, then it also witnesses the consistency of R and S up to normalization, again by the transitivity of $\equiv$.

Note that it is also natural to define the following alternative notion of consistency: we say that two K-relations $R(X)$ and $S(Y)$ are *strictly consistent* if there is a K-relation $T(XY)$ such that $R = T[X]$ and $S = T[Y]$. Clearly, if $R(X)$ and $S(X)$ are strictly consistent, then they are consistent up to normalization. Observe also that for the Boolean semiring $\mathbb{B}$, these two notions coincide, that is, two ordinary relations are consistent up to normalization if and only if they are strictly consistent. In contrast, this is not true for the bag semiring $\mathbb{Z}^{\geq 0}$, that is, there are bags that are consistent up to normalization but they are not strictly consistent. For example, consider the bags $R(AB) = \{12 : 3\}$ and $S(BC) = \{23 : 2\}$, i.e., $R(AB)$ is the bag in which the AB-tuple $(1, 2)$ has multiplicity 3 and all other AB-tuples have multiplicity 0, while $S(BC)$ is the bag in which the BC-tuple $(2, 3)$ has multiplicity 2 and all other BC-tuples have multiplicity 0. These two bags are consistent up to normalization with the bag $T(ABC) = \{123 : 6\}$ witnessing their consistency up to normalization, but, clearly, they are not strictly consistent. Structural and algorithmic aspects of strict consistency for bags have recently been explored in Atserias and Kolaitis (2021).

In this paper, we will focus exclusively on the notion of consistency up to normalization. From now on and for the sake of simplicity, we will use the term *consistency* to refer to the notion of consistency up to normalization.

17.3.2 The Join of Two K-Relations

Naive Join Operation of Two K-Relations We want to define a join operation $R \bowtie S$ for K-relations R and S with the property that if R and S are consistent, then their join witnesses the consistency. A natural candidate for such an operation would be to define $(R \bowtie S)(t)$ by $R(t[X])S(t[Y])$ for every XY-tuple t, where X and Y are the sets of attributes of R and S, respectively. This is the straightforward generalization of the ordinary join of ordinary relations since for the Boolean semiring both definitions give the same operation. Moreover, this is the way the join of bags is defined in SQL (see Ullman & Widom, 2002). As we show below, however, this naive generalization does not work: in fact, even for bags, the bag defined this way does not always witness the consistency of two consistent bags.

Example 2 Let $R(AB)$, $S(BC)$, $J(ABC)$, $U(ABC)$ be the four bags given by the following tables of multiplicities (the #-column is the multiplicity):

$R(AB)$ #	$S(BC)$ #	$J(ABC)$ #	$U(ABC)$ #
1 2 : 6	2 3 : 2	1 2 3 : 12	1 2 3 : 6
2 3 : 3	2 4 : 2	1 2 4 : 12	1 2 4 : 6
	3 4 : 2	2 3 4 : 6	2 3 4 : 6

Consider also the marginals of J and U on AB and BC:

$J[AB]$ #	$J[BC]$ #	$U[AB]$ #	$U[BC]$ #
1 2 : 24	2 3 : 12	1 2 : 12	2 3 : 6
2 3 : 6	2 4 : 12	2 3 : 6	2 4 : 6
	3 4 : 6		3 4 : 6

The bags R and S are consistent since U witnesses their consistency: $2R = U[AB]$ and $3S = U[BC]$. The bag J is actually the naive join of R and S defined by $J(t) = R(t[AB])S(t[BC])$, and there are no non-zero a and b in $\mathbb{N}$ such that $aR = bJ[AB]$, and also there are no non-zero c and d in $\mathbb{N}$ such that $cS = dJ[BC]$.

This example has shown that the *naive* join need not witness the consistency of R and S. We need a different way of defining the join operation.

Derivation of the New Join Operation To arrive at the definition of the join operation that will work for arbitrary semirings, we turn again to probability distributions from Example 1 as the motivating example. Recall that a probability distribution is a $\mathbb{R}^{\geq 0}$-relation that satisfies (17.4). As in the discussion for defining the equivalence relation, this motivation will generalize to any semifield beyond $\mathbb{R}^{\geq 0}$. From there, generalizing the definition to arbitrary semirings will be a small step.

Let $R(X)$ and $S(Y)$ be probability distributions. It is easy to see that if X and Y were disjoint, then the $\mathbb{R}^{\geq 0}$ relation given by $t \mapsto R(t[X])S(t[Y])$ would again be a probability distribution. This, however, fails badly if X and Y are not disjoint as can be seen from turning the example bags $R(AB)$ and $S(BC)$ from Example 2 into probability distributions through normalization (when seen as $\mathbb{R}^{\geq 0}$-relations). The catch is of course that if $Z = X \cap Y$ is non-empty, then two independent samples from the distributions R and S need not agree on their projections on Z. The solution is to define the join $R \bowtie S$ as the probability distribution on XY that is sampled by the following different process: first sample r from the distribution R, then sample s from the distribution S *conditioned* on $s[Z] = r[Z]$, finally output the tuple rs which is well defined since $s[Z] = r[Z]$. This leads to the expression

$$(R \bowtie_{\mathrm{P}} S)(t) := R(t[X])S(t[Y])/S(t[Z]) \tag{17.8}$$

defined for all XY-tuples t, with convention that $0/0 = 0$. Observe that, by writing $r := t[X]$, $s := t[Y]$, $u := t[Y \setminus Z]$ and $v := t[Z]$, the factor $R(t[X])$ in (17.8) is the probability $R(r)$ of getting r in a sample from the distribution R, and the factor $S(t[Y])/S(t[Z])$ is the probability $S(s)/S(v) = S(uv)/S(v)$ of getting s in a sample from the distribution S conditioned on $s[Z] = v = r[Z]$.

Naturally, we could have equally well considered the reverse sampling process that first samples s from S, and then samples r from R *conditioned* on $r[Z] = s[Z]$. This would lead to the alternative expression

$$(R \;{}_{\mathrm{P}}\!\bowtie S)(t) := S(t[Y])R(t[X])/R(t[Z]). \tag{17.9}$$

It is clear from the definitions that $R \bowtie_{\mathrm{P}} S = S \;{}_{\mathrm{P}}\!\bowtie R$, but to have that $R \bowtie_{\mathrm{P}} S = R \;{}_{\mathrm{P}}\!\bowtie S$, we need to have that $R[Z] = S[Z]$. Luckily, this can actually be seen to hold in case R and S are consistent probability distributions since if T is a $\mathbb{R}^{\geq 0}$-relation that satisfies $T[X] = R$ and $T[Y] = S$, then also $T[Z] = R[Z] = S[Z]$ by Part 3 of Lemma 1. It will follow from the lemmas below that in such a case we also have that $(R \bowtie S)[X] \equiv R$ and $(R \bowtie S)[Y] \equiv S$ hold for both $\bowtie = \bowtie_{\mathrm{P}}$ and $\bowtie = {}_{\mathrm{P}}\!\bowtie$, which is what we want.

It is clear that the expression in (17.8), in addition to being defined for $K = \mathbb{R}^{\geq 0}$, could have been defined for any semiring K that is actually a semifield where a division operation is available. On the other hand, to obtain an expression that works for arbitrary semirings, we need to eliminate the divisions. A natural approach for this would be to multiply the expression in (17.8) by the product $\prod_{s \in S[Z]'} S(s[Z])$ of all the values that appear in the denominator. This way, the denominator would cancel, yet the resulting K-relation would remain equivalent up to normalization because the multiplying products do not depend on the tuple t. We are now ready to formally define this.

Definition of the Join of Two K-Relations For a K-relation $T(X)$, a subset $Z \subseteq X$, and a Z-tuple u, define

$$c^*_{T,Z} := \prod_{v \in T[Z]'} T[Z](v) \quad \text{and} \quad c_{T,Z}(u) := \prod_{\substack{v \in T[Z]':\\ v \neq u}} T[Z](v), \tag{17.10}$$

with the understanding that the empty product evaluates to 1, the unit of the semiring K. Observe that $c_{T,Z}(u)$ and $c^*_{T,Z}$ are always non-zero because $T[Z]'$ is precisely the set of Z-tuples v with non-zero $T[Z](v)$, and K has no zero-divisors. The *join* of two K-relations $R(X)$ and $S(Y)$ is the K-relation over XY defined, for every XY-tuple t, by

$$(R \bowtie S)(t) := R(t[X])S(t[Y])c_{S,X\cap Y}(t[X \cap Y]). \tag{17.11}$$

It is worth noting at this point that the identity $c^*_{T,Z} = c_{T,Z}(u)T[Z](u)$ holds, which means that whenever K is a semifield such as $\mathbb{R}^{\geq 0}$, we have

$$c^*_{T,Z}/T[Z](u) = c_{T,Z}(u) \tag{17.12}$$

for all $u \in T[Z]'$, and therefore

$$R \bowtie S = c^*_{S[Z],Z}(R \bowtie_{\mathrm{P}} S), \tag{17.13}$$

where $\bowtie_{\mathrm{P}}$ is defined as in (17.8). In other words, $R \bowtie S$ coincides, up to normalization, with $R \bowtie_{\mathrm{P}} S$.

For ordinary relations, the join operation just introduced coincides with the (ordinary) join operation in relational databases. Note also that the definition of $\bowtie$ is asymmetric. Thus, on the face of it, the definition of the join of two K-relations need not be commutative, i.e., there may be K-relations R and S such that $R \bowtie S \neq S \bowtie R$. As a matter of fact, something stronger holds: typically, $R \bowtie S \not\equiv S \bowtie R$; furthermore, as we shall see next, this happens even for bags. Nonetheless, we will show later that $R \bowtie S \equiv S \bowtie R$ does hold in case R and S agree on their common marginals; moreover, in that case, both joins $R \bowtie S$ and $S \bowtie R$ witness the consistency of R and S.

The example that follows illustrates the definition of the join and its associated quantities $c^*_{T,Z}$ and $c_{T,Z}(u)$, and also shows that the join operation need not be commutative, not even up to equivalence.

Example 3 Consider the bags $R(AB)$ and $S(BC)$ from Example 2, where they were shown to be consistent using the bag $U(ABC)$ from the same example. The joins $V := R \bowtie S$ and $W := S \bowtie R$ defined by (17.11) are the bags given by the following tables of multiplicities. We display V and W alongside their marginals on AB and BC.

$V(ABC)$ #	$W(ABC)$ #	$V[AB]$ #	$V[BC]$ #	$W[AB]$ #	$W[BC]$ #
1 2 3 : 24	1 2 3 : 36	1 2 : 48	2 3 : 24	1 2 : 72	2 3 : 36
1 2 4 : 24	1 2 4 : 36	2 3 : 24	2 4 : 24	2 3 : 36	2 4 : 36
2 3 4 : 24	2 3 4 : 36		3 4 : 24		3 4 : 36

For example, according to the expression (17.11), the entry $V(234)$ is computed as

$$V(234) = R(23) \cdot S(34) \cdot c_{S,B}(3), \tag{17.14}$$

which equals $3 \cdot 2 \cdot (2 + 2) = 24$ since $c_{S,B}(3) = S[B](2) = 2 + 2$. To illustrate the quantity $c^*_{T,Z}$, let $\bar{R}$ and $\bar{S}$ denote the probability distributions that are obtained from normalizing R and S, i.e.,

$\bar{R}(AB)$ #	$\bar{S}(BC)$ #
1 2 : 2/3	2 3 : 1/3
2 3 : 1/3	2 4 : 1/3
	3 4 : 1/3

Setting $\bar{V} = \bar{R} \bowtie \bar{S}$ and using Eq. (17.12), the value of $\bar{V}(234)$ is computed as

$$\bar{V}(234) = \bar{R}(23)\cdot\bar{S}(34)\cdot c^*_{\bar{S},B}/\bar{S}[B](3) \quad (17.15)$$

$$= \bar{R}(23)\cdot(\bar{S}(34)/\bar{S}[B](3))\cdot c^*_{\bar{S},B} \quad (17.16)$$

$$= (1/3)\cdot((1/3)/(1/3))\cdot c^*_{\bar{S},B}. \quad (17.17)$$

In other words, $\bar{V}(234)$ is a fixed (i.e., independent of the tuple 234) multiple of the probability of getting $AB = 23$ in a sample of $\bar{R}$, times the probability of getting $BC = 34$ in a sample of $\bar{S}$ conditioned on the event that $B = 3$. Observe that the constant of proportionality is independent of the actual tuple 234 and is computed as

$$c^*_{\bar{S},B} = c_{\bar{S},B}(2)\cdot c_{\bar{S},B}(3) = \bar{S}(3)\cdot\bar{S}(2) = (1/3)\cdot(1/3 + 1/3) = 2/9. \quad (17.18)$$

Returning to the bags, by inspection we have that $8R = V[AB]$ and $12S = V[BC]$, so V witnesses the consistency of R and S, and $24R = W[AB]$ and $18S = W[BC]$. Similarly, W also witnesses the consistency of R and S. Indeed, $3V = 2W$, which shows that $R \bowtie S \equiv S \bowtie R$ holds for these two bags R and S.

Next, we use the bag R and another bag T to show that $\bowtie$ need not be commutative, not even up to equivalence. Consider the bag $T(BC)$ given below by its table of multiplicities together with $J_1 = R \bowtie T$ and $J_2 = T \bowtie R$:

$T(BC)$ #	$J_1(ABC)$ #	$J_2(ABC)$ #
2 3 : 2	1 2 3 : 48	1 2 3 : 36
2 4 : 2	1 2 4 : 48	1 2 4 : 36
3 4 : 4	2 3 4 : 48	2 3 4 : 72

Clearly, there are no non-zero a and b in $\mathbb{N}$ such that $aJ_1 = bJ_2$, thus $R \bowtie T \not\equiv T \bowtie R$. Note that the pair of K-relations R and T that gave this has $R[B] \not\equiv T[B]$. This is no coincidence, since, in what follows, we will show that if the two K-relations have equivalent common marginals, then their join is commutative up to equivalence. This was the case, for example, for the pair of bags R and S considered also in this example, which had $R[B] \equiv S[B]$ and $R \bowtie S \equiv S \bowtie R$.

Properties of the Join Operation The first property we show about the join of two K-relations is that it is well-defined in the sense that its equivalence class does not depend on the representatives. In other words, we show that the join operation $\bowtie$ is a congruence with respect to the equivalence relation $\equiv$ on K-relations.

Lemma 3 *Let R, R_0 be two K-relations over a set X and let S, S_0 be two K-relations over a set Y. If $R \equiv R_0$ and $S \equiv S_0$, then $R \bowtie S \equiv R_0 \bowtie S_0$.*

Proof Let X be the set of attributes of R and R_0, and let Y be that of S and S_0. Write $Z = X \cap Y$. Let a and b be non-zero elements in K such that $aR = bR_0$, and let c and d be non-zero elements in K such that $cS = dS_0$. First note that $R[Z]' = R_0[Z]'$ by Parts 3 and 2 in Lemma 2. Let m be the cardinality of $R[Z]' = R_0[Z]'$ and set $a^* = ac^m$ and $b^* = bd^m$. We argue that $a^*(R \bowtie S)(t) = b^*(R_0 \bowtie S_0)(t)$ for every XY-tuple t. Fix an XY-tuple t and distinguish the cases $t[Z] \notin R[Z]'$

from $t[Z] \in R[Z]'$. In the first case, we have $R(t[Z]) = 0$ and hence $R(t[X]) = 0$ by Part 2 in Lemma 1, so $R_0(t[X]) = 0$ by Part 2 in Lemma 2. It follows that $(R \bowtie S)(t) = (R_0 \bowtie S_0)(t) = 0$ in this case. In the second case, we have $m \geq 1$ and

$$a^*(R \bowtie S)(t) = aR(t[X]) \cdot cS(t[Y]) \cdot \prod_{\substack{r \in R[Z]': \\ r \neq t[Z]}} cS(r) \tag{17.19}$$

on one hand since $a^* = acc^{m-1}$, and

$$b^*(R_0 \bowtie S_0)(t) = bR_0(t[X]) \cdot dS_0(t[Y]) \cdot \prod_{\substack{r \in R_0[Z]': \\ r \neq t[Z]}} dS_0(r), \tag{17.20}$$

on the other since $b^* = bdd^{m-1}$. The right-hand sides of (17.19) and (17.20) are equal by $aR = bR_0$ and $cS = dS_0$, and $R[Z]' = R_0[Z]'$, so the lemma is proved. □

Next, we show that the support of the join is the ordinary join of the supports.

Lemma 4 *For all K-relations R and S, we have that $(R \bowtie S)' = R' \bowtie S'$.*

Proof Let X be the set of attributes of R, and let Y be that of S. Write $Z = X \cap Y$ and $T = R \bowtie S$. Fix an XY-tuple t. If t is in T', then $T(t) \neq 0$ and in particular $R(t[X]) \neq 0$ and $S(t[Y]) \neq 0$ by (17.11). It follows that $t[X]$ is in R' and $t[Y]$ is in S'; i.e., t is in the relational join of R' and S'. Conversely, if $T(t) = 0$, then by (17.11) again either $R(t[X]) = 0$ or $S(t[Y]) = 0$ or $c_S(t[Z]) = 0$ since K has no zero-divisors. The third case is absurd: we already argued that $c_S(t[Z]) \neq 0$ since $S[Z]'$ is precisely the set of Z-tuples v with $S(v) \neq 0$. In the first two cases, we can conclude that either $t[X]$ is not in R' or $t[Y]$ is not in S', so t is not in their join. □

The *left semijoin* $R' \ltimes S'$ of two ordinary relations $R'(X)$ and $S'(Y)$ is the set of X-tuples in R' that join with some Y-tuple in S', i.e., $R' \ltimes S' = (R' \bowtie S')[X]$. We use Lemma 4 to show that the asymmetric join behaves like a left semijoin up to equivalence, in a strong sense (with $a = 1$).

Lemma 5 *For all K-relations R and S and all $r \in R' \ltimes S'$, we have that $(R \bowtie S)(r) = c_S^* R(r)$.*

Proof Let X and Y be the sets of attributes of R and S, respectively, and write $Z = X \cap Y$ and $T = R \bowtie S$. Fix an X-tuple $r \in R' \ltimes S'$ and write $u = r[Z]$. We have

$$T(r) = \sum_{\substack{t \in T': \\ t[X]=r}} T(t) = \sum_{\substack{t \in T': \\ t[X]=r}} R(t[X])S(t[Y])c_S(t[Z]) = c_S(u)R(r) \sum_{\substack{t \in T': \\ t[X]=r}} S(t[Y]), \tag{17.21}$$

where the first equality follows from (17.2), the second follows from (17.11), and the third follows from the condition that $t[X] = r$ because $Z \subseteq X$ implies $t[Z] =$

$t[X][Z] = r[Z] = u$. At this point, we use the fact that $r \in R' \ltimes S'$ and hence $r \in R'$, together with Lemma 4, to argue that the map

$$f : \{t \in T' : t[X] = r\} \to \{s \in S' : s[Z] = u\} :: t \mapsto t[Y] \tag{17.22}$$

is a bijection. Indeed, since by Lemma 4 each $t \in T'$ comes from the relational join of R' and S', for each $t \in T'$ such that $t[X] = r$ there exists $s \in S'$ with $t[Y] = s$ and $s[Z] = t[Y][Z] = t[Z] = t[X][Z] = r[Z] = u$. Clearly this $s = t[Y]$ is uniquely determined from t. Conversely, if $s \in S'$ is such that $s[Z] = u = r[Z]$, then the join tuple t of r and s exists, it is in T' by Lemma 4 and the fact that $r \in R'$, and moreover $t[X] = r$. This t is uniquely determined from s (and the fixed r). This proves that (17.22) is a bijection. Therefore, continuing from (17.21), we have

$$c_S(u)R(r) \sum_{\substack{t \in T': \\ t[X]=r}} S(t[Y]) = c_S(u)R(r) \sum_{\substack{s \in S': \\ s[Z]=u}} S(s) = c_S(u)R(r)S(u), \tag{17.23}$$

where the first equality follows from the fact that (17.22) is a bijection, and the second follows from (17.2). Recall now that $u = r[Z]$ and $r \in R' \ltimes S'$, which means that $u \in (R' \bowtie S')[Z]$. In particular, $u \in S'[Z]$, so $u \in S[Z]'$ by Part 2 of Lemma 1. Thus, by (17.10), we have $c_S^* = c_S(u)S(u)$, and Eqs. (17.21) and (17.23) actually show that $T(r) = c_S^* R(r)$. □

Next we show that if two K-relations are consistent in the sense that their marginals on the common attributes are equivalent, then their join commutes up to equivalence. Later we will use this to argue that this sense of consistency in terms of marginals is equivalent to the one defined earlier in this section, and thus that if two K-relations are consistent, then their join commutes up to equivalence.

Lemma 6 *For all K-relations $R(X)$ and $S(Y)$, if $R[X \cap Y] \equiv S[X \cap Y]$, then $\bowtie$ commutes on R and S up to equivalence, i.e., $R \bowtie S \equiv S \bowtie R$.*

Proof Write $Z = X \cap Y$. Let a and b be non-zero and such that $aR[Z] = bS[Z]$. First note that $R[Z]' = S[Z]'$ by Part 3 and 2 of Lemma 2. Let m be the cardinality of $R[Z]' = S[Z]'$. If $m = 0$, then $(R \bowtie S)(t) = R(t[X])S(t[Y]) = S(t[Y])R(t[X]) = (S \bowtie R)(t)$ for every XY-tuple t, and we are done. Assume then that $m \geq 1$ and set $a^* = a^{m-1}$ and $b^* = b^{m-1}$. We argue that $b^*(R \bowtie S)(t) = a^*(S \bowtie R)(t)$ for every XY-tuple t. Fix an XY-tuple t and distinguish the cases $t[Z] \notin S[Z]'$ from $t[Z] \in S[Z]'$. In the first case we have $S(t[Y]) = 0$ by Part 2 of Lemma 1 and it follows that $b^*(R \bowtie S)(t) = 0 = a^*(S \bowtie R)(t)$ in this case. In the second case we have

$$b^*(R \bowtie S)(t) = R(t[X])S(t[Y]) \prod_{\substack{s \in S[Z]': \\ s \neq t[Z]}} bS(r) \tag{17.24}$$

on one hand since $b^* = b^{m-1}$ and $t[Z] \in S[Z]'$, and

$$a^*(S \bowtie R)(t) = S(t[Y])R(t[X]) \prod_{\substack{r \in R[Z]': \\ r \neq t[Z]}} aR(r) \tag{17.25}$$

on the other since $a^* = a^{m-1}$ and $t[Z] \in S[Z]' = R[Z]'$. Now, given that $aR(r) = bS(r)$ for every Z-tuple r, the right-hand sides of (17.24) and (17.25) are equal, and the lemma is proved. □

We are ready to show that the join witnesses the consistency of any two consistent K-relations. Along the way, we also prove that two K-relations are consistent if and only if their marginals on the common attributes are equivalent. This result says that the join operation on two K-relations introduced here possesses most of the desirable properties that the join of ordinary relations in relational databases does.

Lemma 7 *Let $R(X)$ and $S(Y)$ be K-relations. The following statements are equivalent:*

(a) *R and S are consistent.*
(b) *$R[X \cap Y] \equiv S[X \cap Y]$.*
(c) *R' and S' are consistent and $R \bowtie S \equiv S \bowtie R$.*
(d) *$R \equiv (R \bowtie S)[X]$ and $S \equiv (R \bowtie S)[Y]$.*

Proof Write $Z = X \cap Y$. For (a) implies (b), let T witness that R and S are consistent, so $R \equiv T[X]$ and $S \equiv T[Y]$. Then, by Part 3 of Lemma 2, we have $R[Z] \equiv T[X][Z]$ and $S[Z] \equiv T[Y][Z]$. Since by Part 3 of Lemma 1 we also have $T[X][Z] = T[Z] = T[Y][Z]$, we get $R[Z] \equiv S[Z]$, as was to be shown. For (b) implies (c) first apply Part 2 of Lemma 1 followed by Part 2 of Lemma 2 to conclude that $R'[X \cap Y] = S'[X \cap Y]$ and hence that R' and S' are consistent as ordinary relations. By Lemma 6 we also have $R \bowtie S \equiv S \bowtie R$. For (c) implies (d) first note that the consistency of R' and S' implies that $R' = R' \ltimes S'$ and $S' = S' \ltimes R'$. Thus, Lemma 5 gives $R \equiv (R \bowtie S)[X]$ and $S \equiv (S \bowtie R)[Y]$. Together with the assumption that $R \bowtie S \equiv S \bowtie R$ this also gives $S \equiv (R \bowtie S)[Y]$ by Part 3 of Lemma 2. That (d) implies (a) is direct since (d) says that $R \bowtie S$ witnesses the consistency of R and S. □

17.3.3 Justification of the Join of Two K-Relations

In this section, we address the question whether the join operation on two relations that we defined in Sect. 17.3 is well motivated. For the rest of this section, fix a finite set of attributes and let Tup denote the set of all tuples over these attributes, which we assume is a computable set through the appropriate encodings. We also assume that the positive semiring K is a computable structure in the sense that the elements of its domain admit a computable presentation that makes its operations be computable functions. The bag semiring $\mathbb{N}$, as well as the semiring $\mathbb{Q}^{\geq 0}$ of non-negative rationals and many others, are of course computable in this sense. Furthermore, we require the equivalence relation $\equiv$ to be decidable; in other words, we require that the following computational problem is decidable:

Given two K-relations R and S over the same set, does $R \equiv S$ hold?

We note that for the bag semiring $\mathbb{N}$, as well as for the semiring $\mathbb{Q}^{\geq 0}$ of non-negative rationals, this problem is very easily decidable, even polynomial-time solvable through what we call the *ratio test*: first, check whether $R' = S'$, and then check whether $R(t_1)/S(t_1) = R(t_2)/S(t_2)$ holds for every two tuples t_1 and t_2 in $R' = S'$.

Deciding Consistency Despite the Plethora of Witnesses Let R and S denote two K-relations on the sets of attributes X and Y and consider the following computational problem:

Given two K-relations R and S, are R and S consistent?

For an infinite positive semiring K, such as the bag semiring $\mathbb{N}$, there is no immediate and a priori reason to think that this problem is algorithmically solvable. The difficulty is that in principle there are infinitely many candidate K-relations to test as witness for consistency, and the arithmetic theory of the natural numbers is highly undecidable. However, what Lemma 7 shows is that the two given K-relations R and S are consistent if and only if the single, finite and explicitly defined K-relation given by $R \bowtie S$ witnesses their consistency. Thus, if K is a semiring for which the equivalence relation $\equiv$ is decidable, this can be checked in finite time and the problem is decidable. In the next example, we show that, even for bags, the consistency of two bags may very well be witnessed by infinitely many pairwise inequivalent witnesses.

Example 4 Let a be a positive integer and let $R(AB)$, $S(BC)$ and $T_a(ABC)$ be the three bags given by the following multiplicity tables, listed alongside the two projections of T_a on AB and BC:

$R(AB)$ #	$S(BC)$ #	$T_a(ABC)$ #	$T_a[AB]$ #	$T_a[BC]$ #
$0\,0 : 1$	$0\,0 : 1$	$0\,0\,0 : a$	$0\,0 : a+1$	$0\,0 : a+1$
$1\,0 : 1$	$0\,1 : 1$	$0\,0\,1 : 1$	$1\,0 : a+1$	$0\,1 : a+1$
		$1\,0\,0 : 1$		
		$1\,0\,1 : a$		

It is evident that $T_a[AB] = (a+1)R$ and $T_a[BC] = (a+1)S$, but $T_a \not\equiv T_b$ unless $a = b$. The conclusion is that there are infinitely many different equivalence classes that witness the consistency of R and S.

Entropy Maximization Let us turn our attention again to probability distributions. The canonical representatives of the equivalence classes are the $\mathbb{R}^{\geq 0}$-relations T that satisfy (17.4). We argued already that for such canonical $\mathbb{R}^{\geq 0}$-relations we have that $\equiv$ agrees with $=$. Therefore, the set of canonical $\mathbb{R}^{\geq 0}$-relations T that witness the consistency of two given probability distributions $R(X)$ and $S(Y)$ can be identified with the set of feasible solutions of a linear program that has one real variable x_t representing $T(t)$ for each XY-tuple t in the join of the supports of R and S:

$$\begin{aligned}
&\textstyle\sum_{t:t[X]=r} x_t = R(r) && \text{for each } r \in R', \\
&\textstyle\sum_{t:t[Y]=s} x_t = S(r) && \text{for each } s \in S', \\
&\textstyle\sum_t x_t = 1 && \\
&x_t \geq 0 && \text{for each } t \in R' \bowtie S'.
\end{aligned} \tag{17.26}$$

The set of probability distributions $P(XY)$ that witness the consistency of R and S is thus a polytope $\mathrm{W}(R, S)$ which is non-empty if and only if R and S are consistent. A natural question to ask is whether there is some particular probability distribution in this polytope that is better motivated than any other such probability distribution. For example, following the principle of maximum entropy, we could ask for the probability distribution that maximizes *Shannon's Entropy* (see Sect. 2.1. in Cover & Thomas, 2006) among those that witness the consistency, i.e., we want to maximize

$$\mathrm{H}_P(XY) = - \sum_{xy \in P(XY)'} P(xy) \log_2(P(xy)) \tag{17.27}$$

subject to the constraint that P is in $\mathrm{W}(R, S)$. Since the entropy is a concave function over the probability simplex (Theorem 2.7.3 in Cover & Thomas, 2006), and since $\mathrm{W}(R, S)$ is a bounded polytope and hence a compact subset of $\mathbb{R}^n$ in appropriate dimension n (unless it is empty), the maximum of (17.27) exists and is achieved at a unique point in $\mathrm{W}(R, S)$. In our setting, writing $Z = X \cap Y$, it is perhaps more natural to maximize the *conditional entropy*, i.e.,

$$\mathrm{H}_P(XY|Z) := - \sum_{z \in P(Z)'} P(z) \sum_{xy \in P(XY)'} P(xy|z) \log_2(P(xy|z)), \tag{17.28}$$

where $P(xy|z) := 0$ if $(xy)[Z] \neq z$ or $P(z) = 0$, and $P(xy|z) := P(xy)/P(z)$ otherwise, with the added convention that $0 \log_2(0) = 0$. For being a convex combination of concave functions over the probability simplex, the conditional entropy $\mathrm{H}_P(XY|Z)$ is again a concave function of P ranging over $\mathrm{W}(R, S)$, which means that the maximum also exists and is achieved at a unique point in $\mathrm{W}(R, S)$. We write $R \bowtie_{\mathrm{H}} S$ for the unique probability distribution in $\mathrm{W}(R, S)$ that achieves the maximum of (17.27) and we write $R \bowtie_{\mathrm{CH}} S$ for the one that achieves the maximum of (17.28). Note that, a priori, due to the logarithms in the definition of entropy, the probability distributions $R \bowtie_{\mathrm{H}} S$ and $R \bowtie_{\mathrm{CH}} S$ need not even have rational components. Interestingly, as will follow from the development below, our join operation $\bowtie$ applied to consistent probability distributions coincides with both $\bowtie_{\mathrm{H}}$ and $\bowtie_{\mathrm{CH}}$, up to the equivalence, which means that both $R \bowtie_{\mathrm{H}} S$ and $R \bowtie_{\mathrm{CH}} S$ *are* indeed rational probability distributions in case R and S are themselves rational.

We argued already in (17.13) that, for probability distributions R and S, our join $R \bowtie S$ coincides with $R \bowtie_{\mathrm{P}} S$ up to equivalence. Moreover, if R and S are consistent, then we have $R[Z] = S[Z]$ for $Z := X \cap Y$, which means that if we write $r := t[X]$, $s := t[Y]$, $u := t[Z]$, and $U := R[Z] = S[Z]$, then

$$(R \bowtie_{\mathrm{P}} S)(t) = R(r)S(s)/U(u). \tag{17.29}$$

This identity implies that $R \bowtie_{\mathrm{P}} S$ is a *product extension* of R and S in the sense of Malvestuto (1988) (see the first paragraph of page 73 in Malvestuto, 1988 for the precise definition of the notion of *product extension*), hence $R \bowtie_{\mathrm{P}} S$ maximizes entropy as a consequence of Malvestuto's Theorem 8. We state this result as Lemma 8 below and reproduce its short proof in Malvestuto (1988) for completeness.

Lemma 8 (Malvestuto, 1988) *If R and S are consistent probability distributions, then $R \bowtie_{\mathrm{P}} S = R \bowtie_{\mathrm{H}} S$.*

Proof Let X and Y be the sets of attributes of R and S, write $Z = X \cap Y$, and assume that R and S are consistent. Let $U := R[Z] = S[Z]$, where the equality follows from the assumption that R and S are probability distributions that are consistent. Write $P := R \bowtie_{\mathrm{H}} S$ and $Q := R \bowtie_{\mathrm{P}} S$. By (17.13) we have $Q \equiv R \bowtie S$, so Q witnesses the consistency of R and S by Lemma 7. Moreover, by design, Q is a probability distribution, and so are R and S by assumption, so $Q[Z] = R[Z] = S[Z] = U$. Since P is also a feasible solution of (17.26), also P is a probability distribution that witnesses the consistency of R and S, so $P[Z] = R[Z] = S[Z] = U$. The conclusion of these is that

$$P[Z] = Q[Z] = U, \tag{17.30}$$
$$P[X] = Q[X] = R, \tag{17.31}$$
$$P[Y] = Q[Y] = S. \tag{17.32}$$

In particular $\mathrm{H}_P(XY) \geq \mathrm{H}_Q(XY)$ since P maximizes (17.27). We show that $\mathrm{H}_P(XY) \leq \mathrm{H}_Q(XY)$, from which it will follow that $P = Q$ since we argued already that the maximum of (17.27) is unique.

Let $\mathrm{D}(P||Q)$ denote the Kullback–Leibler divergence (see Sect. 2.3 in Cover & Thomas, 2006) between two probability distributions $P(X)$ and $Q(X)$ over the same set of attributes X, which is defined as

$$\mathrm{D}(P||Q) := \sum_{x \in P'} P(x)\log(P(x)/Q(x)), \tag{17.33}$$

with the conventions that $0\log(0/q) = 0$ and $p\log(p/0) = \infty$. The Information Inequality (Theorem 2.6.3 in Cover & Thomas, 2006) states that $\mathrm{D}(P||Q) \geq 0$. Therefore

$$\mathrm{H}_P(XY) = -\sum_{t \in P'} P(t)\log(P(t)) \leq -\sum_{t \in P'} P(t)\log(Q(t)). \tag{17.34}$$

Using (17.29), the right-hand side of (17.34) equals

$$\sum_{t \in P'} P(t)\log(U(t[Z]))) - \sum_{t \in P'} P(t)\log(R(t[X])) - \sum_{t \in P'} P(t)\log(S(t[Y])) \tag{17.35}$$

Splitting the set of tuples t in P' by $t[Z]$, the first term in (17.35) rewrites into

$$\sum_{u \in U'} \sum_{\substack{t \in P': \\ t[Z]=u}} P(t)\log(U(u)) = \sum_{u \in U'} \log(U(u))P(u) = \sum_{u \in U'} \log(U(u))Q(u) \quad (17.36)$$

where the first equality follows from (17.2), and the second follows from (17.30). Exactly the same argument for the second and third terms in (17.35), and applying (17.2) to $Q(u)$, $Q(r)$, and $Q(s)$, rewrites (17.35) into

$$\sum_{t \in Q'} Q(t)\log(U(t[Z]))) - \sum_{t \in Q'} Q(t)\log(R(t[X])) - \sum_{t \in Q'} Q(t)\log(S(t[Y])) \tag{17.37}$$

and therefore, by (17.29), into

$$-\sum_{t \in P'} Q(t)\log(Q(t)) = \mathrm{H}_Q(XY). \tag{17.38}$$

Combining (17.34), (17.35), (17.37), and (17.38) we get $\mathrm{H}_P(XY) \leq \mathrm{H}_Q(XY)$ as was to be shown. □

Next, we show that $\bowtie_{\mathrm{P}}$ also maximizes conditional entropy; it follows that $R \bowtie_{\mathrm{CH}} S = R \bowtie_{\mathrm{H}} S$.

Lemma 9 *If R and S are two consistent probability distributions, then $R \bowtie_{\mathrm{P}} S = R \bowtie_{\mathrm{CH}} S$.*

Proof Let X and Y be the sets of attributes of R and S, write $Z = X \cap Y$, and assume that R and S are consistent. Let $U := R[Z] = S[Z]$, where the equality follows from the assumption that R and S are probability distributions that are consistent. Write $P := R \bowtie_{\mathrm{CH}} S$ and $Q := R \bowtie_{\mathrm{P}} S$. By (17.13) we have $Q \equiv R \bowtie S$, so Q witnesses the consistency of R and S by Lemma 7. Moreover, by design, Q is a probability distribution, and so are R and S by assumption, hence $Q[Z] = R[Z] = S[Z] = U$. Since P is also a feasible solution of (17.26), also P is a probability distribution that witnesses the consistency of R and S, hence $P[Z] = R[Z] = S[Z] = U$. The conclusion of these is that $P[Z] = Q[Z] = U$ and both P and Q are feasible solutions of (17.26). In particular, $\mathrm{H}_P(XY|Z) \geq \mathrm{H}_Q(XY|Z)$ since P maximizes (17.28). We show that $\mathrm{H}_P(XY|Z) \leq \mathrm{H}_Q(XY|Z)$, from which it will follow that $P = Q$ since we argued already that the maximum of (17.27) is unique.

We introduce a piece of notation. Let $X_0 := X \setminus Z$ and $Y_0 := Y \setminus Z$. For each Z-tuple $u \in U'$ we write P_u and Q_u to denote the probability distributions over X_0Y_0 defined by $P_z(w) := P(wu)/P(u)$ and $Q_z(w) := Q(wu)/Q(u)$ for every X_0Y_0-tuple w. Using the obvious fact that if $D(X)$ is a probability distribution over X and $Z \subseteq Y \subseteq X$ then $\mathrm{H}_D(Z) = \mathrm{H}_{D(Y)}(Z)$, we have

$$\mathrm{H}_P(XY|Z) = \sum_{u \in U'} U(u)\mathrm{H}_{P_u(X_0Y_0)}(X_0Y_0), \tag{17.39}$$
$$\mathrm{H}_Q(XY|Z) = \sum_{u \in U'} U(u)\mathrm{H}_{Q_u(X_0Y_0)}(X_0Y_0). \tag{17.40}$$

Thus, to prove that $\mathrm{H}_P(XY|Z) \leq \mathrm{H}_Q(XY|Z)$ it suffices to show that $\mathrm{H}_{P_u}(X_0Y_0) \leq \mathrm{H}_{Q_u}(X_0Y_0)$ for each $u \in U'$. Now note that for every X_0-tuple r_0 and every Y_0-tuple s_0, and every $u \in U'$, we have

$$Q_u(r_0s_0) = Q(r_0s_0u)/Q(u) = R(r_0u)S(s_0u)/U(u)^2 = R_u(r_0)S_u(s_0), \tag{17.41}$$

where the first follows from the definition of Q_u, the second from (17.29) and $Q(u) = U(u)$, and the third follows from setting $R_u(r_0) := R(r_0u)/R(u)$ and $S_u(s_0) := S(s_0u)/S(u)$ and the fact that $R(u) = S(u) = U(u)$. Now recall that $P[X] = Q[X] = R$, so $P_u[X_0] = Q_u[X_0] = R_u$ for every $u \in U'$, and also $P[Y] = Q[Y] = S$, so $P_u[Y_0] = Q_u[Y_0] = S_u$ for every $u \in U'$. The conclusion is that the marginals of P_u and Q_u agree, and those of Q_u are independent by (17.41). It follows that

$$\mathrm{H}_{P_u}(X_0Y_0) \leq \mathrm{H}_{P_u(X_0)}(X_0) + \mathrm{H}_{P_u(Y_0)}(Y_0) \tag{17.42}$$
$$= \mathrm{H}_{Q_u(X_0)}(X_0) + \mathrm{H}_{Q_u(Y_0)}(Y_0) \tag{17.43}$$
$$= \mathrm{H}_{Q_u}(X_0Y_0) \tag{17.44}$$

where the first follows from $\mathrm{D}(P_u(X_0Y_0)||P_u(X_0)P_u(Y_0)) \geq 0$ by the Information Inequality (Theorem 2.6.3 in Cover & Thomas, 2006), the second follows from equal marginals, and the third follows from the fact that $\mathrm{D}(Q_u(X_0Y_0)||Q_u(X_0)Q_u(Y_0)) \geq 0$ holds with equality if and only if the marginals $Q_u(X_0)$ and $Q_u(Y_0)$ are independent (see again Theorem 2.6.3 in Cover & Thomas, 2006), which we argued is the case for Q_u. □

The following result is an immediate consequence of Lemmas 8 and 9.

Corollary 1 *Let $R(X)$ and $S(Y)$ be probability distributions, and let $Z = X \cap Y$. If R and S are consistent, then the probability distributions P and Q among those in $\mathrm{W}(R, S)$ that maximize entropy $\mathrm{H}_P(XY)$ and conditional entropy $\mathrm{H}_Q(XY|Z)$ are equal. Moreover, if R and S are rational, then P and Q are rational.*

Summarizing, we have proved that whenever R and S are consistent probability distributions we have $R \bowtie S \equiv R \bowtie_{\mathrm{P}} S$ and $R \bowtie_{\mathrm{P}} S = R \bowtie_{\mathrm{H}} S = R \bowtie_{\mathrm{CH}} S$, which we view as evidence that our definition of $\bowtie$ is well motivated.

Lossless Join Decompositions We provide further justification for the definition of the join of two K-relations by showing that a decomposed K-relation can be reconstructed (up to equivalence) by joining its decomposed parts, under the same hypothesis that makes it possible to reconstruct an ordinary relation by joining its decomposed parts. This justification for the $\bowtie$ operation is valid for an arbitrary positive semiring K.

Let U be a set of attributes, let P be an ordinary relation over U, and let V, W and X, Y be pairs of subsets of U. We say that P *satisfies the functional depen-*

dency $V \to W$ if whenever two tuples in P agree on all attributes in V, then they also agree on all attributes in W. The *decomposition of* P *along* X *and* Y consists of the projections $R = P[X]$ and $S = P[Y]$ of P on the sets X and Y, respectively. Such a decomposition is said to be a *lossless-join* decomposition if $P = R \bowtie S$, that is, the relation P can be reconstructed by joining the parts $R = P[X]$ and $S = P[Y]$ of the decomposition.

The following lemma gives a sufficient condition for a decomposition to be a lossless join one. Even though this lemma is standard textbook material, we include a proof for completeness and comparison with what is to follow.

Lemma 10 *Let P be an ordinary relation that is decomposed along X and Y. If P satisfies the functional dependency $X \cap Y \to X \setminus Y$ or P satisfies the functional dependency $X \cap Y \to Y \setminus X$, then this decomposition is lossless-join.*

Proof From the definitions, it follows that if $R = P[X]$ and $S = P[Y]$ are the projections of R on X and Y, respectively, then $P \subseteq R \bowtie S$. Assume that P satisfies the functional dependency $X \cap Y \to X \setminus Y$ (the other case is proved using a symmetric argument). We will show that $R \bowtie S \subseteq P$. Let t be a tuple in $R \bowtie S$. It follows that $t[X] \in R = P[X]$ and $t[Y] \in S = P[Y]$. Therefore, there are tuples t_1 and t_2 in P, such that $t[X] = t_1[X]$ and $t[Y] = t_2[Y]$. Since P satisfies the functional dependency $X \cap Y \to X \setminus Y$ and since $t_1[X \cap Y] = t[X \cap Y] = t_2[X \cap Y]$, we must have that $t_1[X \setminus Y] = t_2[X \setminus Y]$. Since $t[X] = t_1[X]$ and $t[Y] = t_2[Y]$, it follows that $t = t_2$, hence $t \in P$; this completes the proof that $R \bowtie S \subseteq P$. □

It is easy to see that there are lossless-join decompositions of relations that satisfy neither the functional dependencies $X \cap Y \to X \setminus Y$ nor the functional dependency $X \cap Y \to Y \setminus X$. Thus, Lemma 10 is a sufficient, but not necessary, condition for a decomposition to be a lossless join one. The condition, however, is necessary and sufficient for relations over a schema that satisfy a set functional dependencies. To make this statement precise, we recall a basic definition from relational databases. Let U be a set of attributes, let F be a set of functional dependencies between subsets of U, and let $V \to W$ be a functional dependency. We say that F *logically implies* $V \to W$, denoted $F \models V \to W$ if whenever a relation R satisfies every functional dependency in F, then R also satisfies $V \to W$. The following is a well known result in relational database theory (see, e.g., Theorem 7.5 in Ullman, 1988).

Theorem 1 *Let U be a set of attributes, let F be a set of functional dependencies between subsets of U, and let X and Y are two subsets of U. Then the following statements are equivalent:*

(a) $F \models X \cap Y \to X \setminus Y$ *or* $F \models X \cap Y \to Y \setminus X$.
(b) For every relation R over U that satisfies every functional dependency in F, it holds that if R is decomposed along X and Y, then this decomposition is a lossless-join one.

Our next result says that Lemma 10 extends to decompositions of K-relations, where K is a positive semiring. We first need to extend the notions appropriately. If P is a K-relation, then the *decomposition of P along X and Y* consists of the marginals $R = P[X]$ and $S = P[Y]$. We say that the decomposition is *lossless-join* if $P \equiv R \bowtie S$, where $\bowtie$ is the join operation on K-relations.

Lemma 11 *Let P be a K-relation that is decomposed along X and Y. If the support P' of P satisfies the functional dependency $X \cap Y \to X \setminus Y$ or P' satisfies the functional dependency $X \cap Y \to Y \setminus X$, then this decomposition is lossless-join.*

Proof For concreteness, let us assume that the attributes of P are ABC and that P is decomposed along $X = AB$ and $Y = BC$. The proof remains the same in the general case and with only notational changes.

We will show that $P \equiv R \bowtie S$, where $R = P[X]$ and $S = P[Y]$. In fact, we will show that $R \bowtie S = c^*_{S,X\cap Y}P$. By Part 2 of Lemma 1 we have $R' = P'[X]$ and $S' = P'[Y]$. In addition, since P' satisfies the functional dependency $X \cap Y \to X \setminus Y$ or the functional dependency $X \cap Y \to Y \setminus X$ we have $P' = R' \bowtie S' = (R \bowtie S)'$, where the first equality follows from Lemma 10 and the second from Lemma 4. Let (a, b, c) be a tuple in $(R \bowtie S)'$, so in particular $(a, b, c) \in P'$. By the definition of $R \bowtie S$, we have that $(R \bowtie S)(a, b, c) = R(a, b)S(b, c)c_S(b)$. We now examine the quantities $R(a, b)$ and $S(b, c)$ separately, and for that we distinguish by cases.

Case 1: The ordinary relation P' satisfies the functional dependency $X \cap Y \to X \setminus Y$, which, in this case, amounts to $B \to A$. For $R(a, b)$ we have

$$R(a,b) = \sum_{c':(a,b,c')\in P'} P(a,b,c') = \sum_{a',c':(a',b,c')\in P'} P(a',b,c') = P(b), \tag{17.45}$$

where the first equality follows from $R = P[X]$ and (17.2), the second follows from the fact that, since P' satisfies the functional dependency $B \to A$, we must have that $(a', b, c') \in P'$ implies $a' = a$, and the third follows from (17.2). For $S(b, c)$ we have

$$S(b,c) = \sum_{a':(a',b,c)\in P'} P(a',b,c) = P(a,b,c), \tag{17.46}$$

where the first equality follows from $S = P[Y]$ and (17.2), and the second follows from the fact that, since P' satisfies the functional dependency $B \to A$ and $(a, b, c) \in P'$, we must have that $(a', b, c) \in P'$ holds if and only if $a' = a$.

Case 2: The ordinary relation P' satisfies the functional dependency $X \cap Y \to Y \setminus X$, which, in this case, amounts to $B \to C$. Using a similar analysis as in the previous case, for $S(b, c)$ we have that

$$S(b,c) = \sum_{a':(a',b,c)\in P'} P(a',b,c) = \sum_{a',c':(a',b,c')} P(a',b,c') = P(b), \tag{17.47}$$

where the first equality follows from $S = P[Y]$ and (17.2), the second follows from the fact that, since P' satisfies the functional dependency $B \to C$, we must have that $(a', b, c') \in P'$ implies $c' = c$, and the third follows from (17.2). For $R(a, b)$ we have

$$R(a, b) = \sum_{c' : (a,b,c') \in P'} P(a, b, c') = P(a, b, c) \tag{17.48}$$

where the first equality follows from $R = P[X]$ and (17.2), and the second follows from the fact that, since P' satisfies the functional dependency $B \to C$ and $(a, b, c) \in P'$ we must have that $(a, b, c') \in P'$ holds if and only if $c' = c$.

In both cases, it follows that

$$(R \bowtie S)(a, b, c) = R(a, b)S(b, c)c_S(b) = P(b)P(a, b, c)c_S(b) = c^*_{P,B}P(a, b, c), \tag{17.49}$$

where the first equality follows from (17.11), the second follows from (17.45) and (17.46) in one case, and from (17.47) and (17.48) in the other, and the last follows from $c^*_{P,B} = c_S(b)P(b)$ by (17.10) since $b \in P[B]' = P'[B]$ given that $(a, b, c) \in P'$ and Part 2 of Lemma 1. This proves that $R \bowtie S = c^*_{P,B}P$, which was to be shown. □

The last result in this section asserts that the preceding Theorem 1 extends to decompositions of K-relations.

Proposition 1 *Let K be a positive semiring, let U be a set of attributes, let F be a set of functional dependencies between subsets of U, and let X and Y be two subsets of U. The following statements are equivalent:*

(a) $F \models X \cap Y \to X \setminus Y$ *or* $F \models X \cap Y \to Y \setminus X$.
(b) *For every K-relation R over U whose support R' satisfies every functional dependency in F, it holds that if R is decomposed along X and Y, then this decomposition is a lossless-join one.*

Proof First, assume that $F \models X \cap Y \to X \setminus Y$ or $F \models X \cap Y \to Y \setminus X$. Let R be a be a K-relation whose support R' satisfies every functional dependency in F. Then R' satisfies the functional dependency $X \cap Y \to X \setminus Y$ or R' satisfies the functional dependency $X \cap Y \to Y \setminus X$. By Lemma 11, the decomposition of R along X and Y is lossless join.

Next, assume that for every K-relation R over U whose support R' satisfies every functional dependency in F, it holds that if R is decomposed along X and Y, then this decomposition is a lossless-join one. Let P be an arbitrary ordinary relation over U that satisfies every functional dependency in F. We will show that if P is decomposed along X and Y, then the decomposition is a lossless join one, hence, by Theorem 1, we have that $F \models X \cap Y \to X \setminus Y$ or $F \models X \cap Y \to Y \setminus X$. Turn P into a K-relation R whose support is P and where all tuples in the support have value 1 in K. In other words, consider the K-relation R such that for every tuple t, we have that $R(t) = 1$ if $t \in P$, and $R(t) = 0$ if $t \notin P$. By hypothesis, the decomposition of R along X and Y is a lossless join one. Therefore, $R \equiv R[X] \bowtie R[Y]$, as K-relations. By Lemma 4 and Part 2 of Lemma 2, we have that $R' = R[X]' \bowtie R[Y]'$, i.e., the support of the join is the join of the supports as ordinary relations. From the definition of R, we have that $R' = P$; moreover, from Part 2 of Lemma 1, we have that $R[X]' = R'[X]$ and $R[Y]' = R'[Y]$. Since $R'[X] = P[X]$ and $R'[Y] = P[Y]$, we conclude that $P = P[X] \bowtie P[Y]$, thus the decomposition of P along X and Y is a lossless join one, which was to be shown. □

17.4 Consistency of Three or More *K*-Relations

While the definition of consistency of two K-relations has a straightforward generalization to the case of three or more K-relations, not all the related concepts will go through: the join of three or more K-relations will be particularly problematic. We start with the definitions.

Let K be a positive semiring and let $R_1(X_1), \dots, R_m(X_m)$ be K-relations. We say that the collection $R_1, \dots, R_m$ is *globally consistent* if there is a K-relation T over $X_1 \cup \cdots \cup X_m$ such that $R_i \equiv T[X_i]$ for all $i \in [m]$. We say that such a K-relation *witnesses* the global consistency of $R_1, \dots, R_m$. The equivalence classes $[R_1], \dots, [R_m]$ are called *globally consistent* if their representatives $R_1, \dots, R_m$ are globally consistent. As in the case of two equivalence classes, it is easy to see using transitivity of $\equiv$ that this notion is well defined in that it does not depend on the chosen representatives.

We also say that the relations $R_1, \dots, R_m$ are *pairwise consistent* if for every $i, j \in [m]$ we have that $R_i[X_i]$ and $R_j[X_j]$ are consistent. From the definitions, it follows that if $R_1, \dots, R_m$ are globally consistent, then they are also pairwise consistent. The converse, however, need not be true, in general. In fact, the converse fails even for ordinary relations, that is, for $\mathbb{B}$-relations, where $\mathbb{B}$ is the Boolean semiring. For example, it is easy to see that the ordinary relations $R(AB) = \{00, 11\}$, $S(BC) = \{01, 10\}$, $T(AC) = \{00, 11\}$ are pairwise consistent but not globally consistent.

In the context of relational databases, there has been an extensive study of global consistency for ordinary relations. We present an overview of some of the main findings next.

17.4.1 Global Consistency in the Boolean Semiring

For this section K is the Boolean semiring $\mathbb{B}$ and therefore K-relations are ordinary relations or, simply, *relations*. Let $R_1, \dots, R_m$ be a collection of relations. The *relational join* or, simply, the *join* of $R_1, \dots, R_m$ is the relation $R_1 \bowtie \cdots \bowtie R_m$ consisting of all $(X_1 \cup \cdots \cup X_m)$-tuples t such that $t[X_i]$ belongs to R_i for all $i = 1, \dots, m$. The following facts are well known and easy to prove (e.g., see Honeyman et al., 1980):

(1) If T is a relation witnessing the global consistency of $R_1, \ldots, R_m$, then $T \subseteq R_1 \bowtie \cdots \bowtie R_m$.
(2) The collection $R_1, \ldots, R_m$ is globally consistent if and only if $(R_1 \bowtie \cdots \bowtie R_m)[X_i] = R_i$ for all $i = 1, \ldots, m$. Consequently, if the collection $R_1, \ldots, R_m$ is globally consistent, then the join $R_1 \bowtie \cdots \bowtie R_m$ is the largest relation witnessing their consistency.

As seen earlier, pairwise consistency is a necessary, but not sufficient, condition for global consistency. This was exemplified by three relations R, S, T with schema AB, BC, AC, respectively. In contrast, it is not hard to see that if the schema of the three relations had been AB, BC, CD, then pairwise consistency would have been a necessary and sufficient condition for the global consistency of any three K-relations over these schema. This raises the question whether it is possible to characterize the set of schema for which pairwise consistency is a necessary and sufficient condition for global consistency. This question was investigated and answered by Beeri et al. (1983). Before describing their results, we need to introduce a number of notions from hypergraph theory.

Hypergraphs A *hypergraph* is a pair $H = (V, E)$, where V is a set of *vertices* and E is a set of *hyperedges*, each of which is a non-empty subset of V. Clearly, the undirected graphs without self-loops are precisely the hypergraphs all the hyperedges of which are two-element sets of vertices; such hyperedges are called *edges*.

Let $H = (V, E)$ be a hypergraph. The *reduction* of H, denoted by $R(H)$, is the hypergraph whose set of vertices is V itself and whose hyperedges are those hyperedges $X \in E$ that are not included in any other hyperedge of H. A hypergraph H is *reduced* if $H = R(H)$. If $W \subseteq V$, then the *hypergraph induced by W on H*, denoted by $H[W]$, is the hypergraph whose set of vertices is W and whose hyperedges are the non-empty subsets of the form $X \cap W$, where $X \in E$ is a hyperedge of H; in symbols, $H[W] = (W, \{X \cap W : X \in E\} \setminus \{\emptyset\})$.

Every collection $X_1, \ldots, X_m$ of sets of attributes can be identified with a hypergraph $H = (V, E)$, where $V = X_1 \cup \cdots \cup X_m$ and $E = \{X_1, \ldots, X_m\}$. Conversely, every hypergraph $H = (V, E)$ gives rise to a collection $X_1, \ldots, X_m$ of sets of attributes, where $X_1, \ldots, X_m$ are the hyperedges of H. For this reason, we can move from collections of sets of attributes to hypergraphs (and vice versa) in a seamless way. In what follows, we will consider several structural properties of hypergraphs that, as shown in Beeri et al. (1983), give rise to necessary and sufficient conditions for pairwise consistency to coincide with global consistency.

Acyclic Hypergraphs We begin by defining the notion of an acyclic hypergraph, which generalizes the notion of an acyclic undirected graph with no self-loops. For this, we need to introduce several auxiliary notions. If H is a hypergraph and u and v are vertices of H, then a *path from u to v* is a sequence $Y_1, \ldots, Y_k$ of hyperedges of H, for some positive integer k, such that $u \in Y_1$ and $v \in Y_k$, and $Y_i \cap Y_{i+1} \neq \emptyset$, for all $i \in [k-1]$. Using the notion of a path, one defines the notions of a *connected component* of a hypergraph and of a *connected* hypergraph in the obvious way.

Let $H = (V, E)$ be a reduced hypergraph and let X and Y be two distinct hyperedges of H with $X \cap Y \neq \emptyset$. We say that $X \cap Y$ is an *articulation set* of H if the number of connected components of the reduction $R(H[V \setminus (X \cap Y)])$ of $H[V \setminus (X \cap Y)]$ is greater than the number of the connected components of H. We say that a reduced hypergraph H is *acyclic* if for every subset W of vertices of H, if $R(H[W])$ is connected and has more than one hyperedges, then it has an articulation set. Finally, a hypergraph H is *acyclic* if its reduction $R(H)$ is acyclic. Otherwise, H is *cyclic*. We say that a collection $X_1, \ldots, X_m$ of sets of attributes is *acyclic* if the hypergraph with hyperedges $X_1, \ldots, X_m$ is acyclic; otherwise, we say that the collection $X_1, \ldots, X_m$ is *cyclic*.

To illustrate these concepts, consider the set $V = \{A_1, \ldots, A_n\}$ of vertices and the sets E_1 and E_2 of hyperedges, where $E_1 = \{\{A_i, A_{i+1}\} : 1 \leq i \leq n-1\}$ and $E_2 = E_1 \cup \{\{A_n, A_1\}\}$. It is not hard to verify that the hypergraph $H_1 = (V, E_1)$ is acyclic, whereas the hypergraph $H_2 = (V, E_2)$ is cyclic as soon as $n \geq 3$.

Conformal and Chordal Hypergraphs The *primal* graph of a hypergraph $H = (V, E)$ is the undirected graph that has V as its set of vertices and has an edge between any two distinct vertices that appear together in at least one hyperedge of H. A hypergraph H is *conformal* if the set of vertices of every clique (i.e., complete subgraph) of the primal graph of H is contained in some hyperedge of H. For example, both hypergraphs H_1 and H_2 above are conformal, whereas the hypergraph $H_3 = (V, E_3)$ with $V = \{A_1, \ldots, A_n\}$ and $E_3 = \{V \setminus \{A_i\} : 1 \leq i \leq n\}$ is not conformal as long as $n \geq 3$.

A hypergraph H is *chordal* if its primal graph is chordal, that is, if every cycle of length at least four of the primal graph of H has a chord, i.e., an edge between two non-consecutive vertices in the cycle. For example, the hypergraphs H_1 and H_3 above are chordal, whereas the hypergraph H_2 is not chordal for $n \geq 4$. Observe that, in case $n = 3$, the hypergraph H_2 is chordal but not conformal.

Running Intersection Property We say that a hypergraph H has the *running intersection property* if there is a listing $X_1, \ldots, X_m$ of all hyperedges of H such that for every $i \in [m]$ with $i \geq 2$, there exists a $j < i$ such that $X_i \cap (X_1 \cup \cdots \cup X_{i-1}) \subseteq X_j$.

For example, the hypergraph H_1 has the running intersection property with the listing being $\{A_1, A_2\}, \ldots, \{A_{n-1}, A_n\}$, whereas the hypergraphs H_2 and H_3 do not have the running intersection property as long as $n \geq 3$.

Join Trees A *join tree* for a hypergraph H is an undirected tree T with the set E of the hyperedges of H as its vertices and such that for every vertex v of H, the set of vertices of T containing v forms a subtree of T, i.e., if v belongs to two vertices X_i and X_j of T, then v belongs to every vertex of T in the unique path from X_i to X_j in T.

For example, the hypergraph H_1 has a join tree (in fact, the join tree is a path) with edges of the form $\{\{A_i, A_{i+1}\}, \{A_{i+1}, A_{i+2}\}\}$ for $i \in [n-2]$, whereas the hypergraphs H_2 and H_3 do not have a join tree for $n \geq 3$.

Graham's Algorithm Consider the following iterative algorithm on hypergraphs: given a hypergraph $H = (V, E)$, apply the following two operations repeatedly until neither of the two operations can be applied:

1. If v is a vertex that appears in only one hyperedge X_i of H, then delete v from X_i.
2. If there are two hyperedges X_i and X_j such that $i \neq j$ and $X_i \subseteq X_j$, then delete X_i from E.

It can be shown that this algorithm has the Church–Rosser property, that is, it produces the same hypergraph independently of the order in which the above two operations are applied.

We say that *Graham's algorithm succeeds* on a hypergraph $H = (V, E)$ if the algorithm, given H as input, returns the empty hypergraph $H = (V, \emptyset)$ as output. Otherwise, we say that *Graham's algorithm fails* on H. We also say that H is *accepted* by Graham's algorithm if the algorithm succeeds on H; otherwise we say that it is *rejected*.

For example, Graham's algorithm succeeds on the hypergraph H_1, whereas it fails on the hypergraphs H_2 and H_3 as long as $n \geq 3$ (in fact, it returns H_2 and H_3, respectively).

This algorithm was designed by Graham (1979). A similar algorithm was designed by Yu and Ozsoyoglu (1979); these two algorithms are often referred to as the GYO Algorithm (see Abiteboul et al., 1995).

Local-to-Global Consistency Property The notions introduced so far can be thought of as "syntactic" or "structural" properties that some hypergraphs possess and others do not, because their definitions involve only the vertices and the hyperedges of the hypergraph at hand. In contrast, the next notion is "semantic", in the sense that its definition also involves relations whose sets of attributes are the hyperedges of the hypergraph at hand.

Let H be a hypergraph and let $X_1, \ldots, X_m$ be a listing of all hyperedges of H. We say that H has the *local-to-global consistency property for ordinary relations* if every pairwise consistent collection $R_1(X_1), \ldots, R_m(X_m)$ of relations of schema $X_1, \ldots, X_m$ is globally consistent.

For example, it can be shown that the hypergraph H_1 has the local-to-global consistency property for ordinary relations, whereas the hypergraphs H_2 and H_3 do not have this property as long as $n \geq 3$.

We are now ready to state the main result in Beeri et al. (1983).

Theorem 2 (Theorem 3.4 in Beeri et al., 1983) *Let H be a hypergraph. The following statements are equivalent:*

(a) H is an acyclic hypergraph.
(b) H is a conformal and chordal hypergraph.
(c) H has the running intersection property.
(d) H has a join tree.
(e) H is accepted by Graham's algorithm.
(f) H has the local-to-global consistency property for ordinary relations.

As an illustration of Theorem 2, let us return to the hypergraphs H_1, H_2, H_3 encountered earlier. Hypergraph H_1 has all six properties in Theorem 2, whereas hypergraphs H_2 and H_3 have none of these properties when $n \geq 3$.

17.4.2 Global Consistency in Arbitrary Positive Semirings

Let K be an arbitrary, but fixed, positive semiring. In this section, we investigate some aspects of global consistency for collections of K-relations.

As discussed earlier, if $R_1, \dots, R_m$ is a globally consistent collection of ordinary relations, then the join $R_1 \bowtie \cdots \bowtie R_m$ witnesses the global consistency of $R_1, \dots, R_m$ (and, in fact, is the largest such witness). At first, one may expect that a similar result may hold for globally consistent collections $R_1, \dots, R_m$ of K-relations. It turns out, however, that the concept of the join of three or more K-relations is problematic, even for the case in which K is the bag semiring $\mathbb{N}$ of non-negative integers. Note that, using the join of two relations, the join of three K-relations R, S, T could be defined as either $R \bowtie (S \bowtie T)$ or as $(R \bowtie S) \bowtie T$. The join of ordinary relations is associative, hence these two expressions coincide for ordinary relations. In contrast, there are bags R, S, T that are globally consistent and such that

$$R \bowtie (S \bowtie T) \not\equiv (R \bowtie S) \bowtie T, \tag{17.50}$$

and neither $R \bowtie (S \bowtie T)$ nor $(R \bowtie S) \bowtie T$ witnesses the global consistency of R, S, T.

Example 5 Let $W(ABC)$ be the bag given by its table of multiplicities below, along with its three marginals $R(AB)$, $S(BC)$, $T(AC)$:

$W(ABC)$ #	$R(AB)$ #	$S(BC)$ #	$T(AC)$ #
1 1 2 : 1	1 1 : 1	1 2 : 1	1 2 : 1
1 2 3 : 2	1 2 : 2	1 4 : 4	1 3 : 2
2 1 4 : 4	2 1 : 4	2 2 : 2	2 1 : 3
2 2 2 : 2	2 2 : 2	2 3 : 2	2 2 : 2
2 3 1 : 3	2 3 : 3	3 1 : 3	2 4 : 4

By construction the collection of three bags R, S, T is globally consistent as witnessed by W. We produced $N_1 := (R \bowtie S) \bowtie T$ and $N_2 := R \bowtie (S \bowtie T)$ by computer, along with their marginals on AB, BC, AC. We display two bags $[N_1']$ and $[N_2']$ that are in the equivalence classes of N_1 and N_2, respectively, along with the two marginals $P_1 := [N_1'][BC]$ and $P_2 := [N_2'][AB]$ that suffice to verify the claim that neither N_1 nor N_2 witness the consistency of R, S, T:

$[N_1'](ABC)$ #	$[N_2'](ABC)$ #	$P_1(BC)$ #	$P_2(AB)$ #
1 1 2 : 14	1 1 2 : 1	1 2 : 22	1 1 : 1
1 2 2 : 7	1 2 2 : 5	1 4 : 48	1 2 : 10
1 2 3 : 21	1 2 3 : 5	2 2 : 35	2 1 : 20
2 1 2 : 8	2 1 2 : 4	2 3 : 21	2 2 : 5
2 1 4 : 48	2 1 4 : 16	3 1 : 42	2 3 : 15
2 2 2 : 28	2 2 2 : 5		
2 3 1 : 42	2 3 1 : 15		

The ratio test shows that $[N_1']$ and $[N_2']$ are not equivalent ($14/1 \neq 7/5$), so $\bowtie$ is not associative, not even up to equivalence. The ratio test applied to the bags $P_1(BC)$ and $S(BC)$ shows that they are not equivalent ($22/1 \neq 48/4$), and the ratio test applied to the bags $P_2(AB)$ and $R(AB)$ shows that they are not equivalent ($1/1 \neq 10/2$). Thus, neither N_1 nor N_2 witness the consistency of R, S, T.

Observe that the collection AB, BC, AC of the sets of attributes of the relations in Example 5 is cyclic. It turns out that this is no accident. Indeed, we show next that if a collection $X_1, \ldots, X_m$ of sets of attributes is acyclic and if $R_1(X_1), \ldots, R_m(X_m)$ is a globally consistent collection of K-relations of schema $X_1, \ldots, X_m$, then a witness of their global consistency can always be built iteratively through joins of two K-relations. In fact, we show something stronger, namely, that it suffices for $R_1, \ldots, R_m$ to be pairwise consistent K-relations. For stating this lemma we need the following definition. The *iterated left-join* of the K-relations $R_1, \ldots, R_m$ is the K-relation

$$((\cdots(R_1 \bowtie R_2) \bowtie \cdots \bowtie R_{m-2}) \bowtie R_{m-1}) \bowtie R_m, \tag{17.51}$$

i.e., the sequential join of $R_1, \ldots, R_m$ with the join operations associated to the left. More formally, the iterated left-join of $R_1, \ldots, R_m$ is defined by induction on m. For $m = 1$ it is R_1, and for $m \geq 2$ it is $R \bowtie R_m$ where R is the iterated left-join of $R_1, \ldots, R_{m-1}$.

Lemma 12 *Let $X_1, \ldots, X_m$ be an acyclic collection of sets of attributes. There exists a permutation $\pi : [m] \to [m]$ such that if $R_1(X_1), \ldots, R_m(X_m)$ are pairwise consistent K-relations of schema $X_1, \ldots, X_m$, then they are globally consistent, and the iterated left-join of $R_{\pi(1)}, \ldots, R_{\pi(m)}$ witnesses their global consistency. In particular, if they are globally consistent, then the iterated left-join of some permutation of them witnesses their global consistency.*

Proof Assume that $X_1, \ldots, X_m$ is an acyclic collection of sets of attributes. By Theorem 2, this collection has the running intersection property, hence there exists a permutation $\pi : [m] \to [m]$ such that for every $i \in [m]$ with $i \geq 2$, there exists $j \in [m]$ such that $\pi(j) < \pi(i)$ and $X_{\pi(i)} \cap (X_{\pi(1)} \cup \cdots \cup X_{\pi(i-1)}) \subseteq X_{\pi(j)}$. By renaming the sets, we may assume that π is the identity, so for every $i \in [m]$ with $i \geq 2$, there is a $j \in [i-1]$ such that $X_i \cap (X_1 \cup \cdots \cup X_{i-1}) \subseteq X_j$. Fix a collection of K-relations $R_1, \ldots, R_m$ for $X_1, \ldots, X_m$ and assume that they are pairwise consistent.

For each $i \in [m]$, let $T_i := ((R_1 \bowtie \cdots \bowtie R_{i-2}) \bowtie R_{i-1}) \bowtie R_i$ with the joins associated to the left. We show, by induction on $i = 1, \ldots, m$, that T_i is a K-relation over $X_1 \cup \cdots \cup X_i$ that witnesses the consistency of $R_1, \ldots, R_i$.

For $i = 1$ the claim is obvious since $T_1 = R_1$. Assume then that $i \geq 2$ and that the claim is true for smaller indices. Let $X := X_1 \cup \cdots \cup X_{i-1}$ and let $j \in [i-1]$ be such that $X_i \cap X \subseteq X_j$. By induction hypothesis, we know that T_{i-1} is a K-relation over X that witnesses the consistency of $R_1, \ldots, R_{i-1}$. First, we show that T_{i-1} and R_i are consistent. By Lemma 7 it suffices to show that $T_{i-1}[X \cap X_i] \equiv R_i[X \cap X_i]$. Let $Z = X \cap X_i$, so $Z \subseteq X_j$ by the choice of j, and indeed $Z = X_j \cap X_i$. Since $j \leq i-1$, we have $R_j \equiv T_{i-1}[X_j]$. By Part 3 of Lemma 2 and Part 3 of Lemma 1, we have $R_j[Z] \equiv T_{i-1}[X_j][Z] = T_{i-1}[Z]$. By assumption, also R_j and R_i are consistent, and $Z = X_j \cap X_i$, which by Lemma 7 implies $R_j[Z] \equiv R_i[Z]$. By transitivity, we get $T_{i-1}[Z] \equiv R_i[Z]$, hence, by $Z = X \cap X_i$ and Lemma 7, the K-relations T_{i-1} and R_i are consistent. We show that $T_i = T_{i-1} \bowtie R_i$ witnesses the consistency of $R_1, \ldots, R_i$. Since T_{i-1} and R_i are consistent, first note that $T_{i-1} \equiv T_i[X]$ and $R_i \equiv T_i[X_i]$ by Lemma 7. Now fix $k \leq i-1$ and note that

$$R_k \equiv T_{i-1}[X_k] \equiv T_i[X][X_k] = T_i[X_k], \tag{17.52}$$

where the first equivalence follows from the fact that T_{i-1} witnesses the consistency of $R_1, \ldots, R_{i-1}$ and $k \leq i-1$, the second equivalence follows from $T_{i-1} \equiv T_i[X]$ together with Part 3 of Lemma 2 applied to $X_k \subseteq X$, and the equality follows again from $X_k \subseteq X$ and this time from Part 3 of Lemma 1. Thus, T_i witnesses the consistency of $R_1, \ldots, R_i$, which was to be shown. □

In what follows, we explore the interplay between pairwise consistency and global consistency of K-relations, aiming to extend Theorem 2 to arbitrary positive semirings.

Local-to-Global Consistency Property for K-Relations We extend the notion of local-to-global consistency property from ordinary relations to K-relations. Let H be a hypergraph and let $X_1, \ldots, X_m$ be a listing of all hyperedges of H. We say that H has the *local-to-global consistency property for K-relations* if every pairwise consistent collection $R_1(X_1), \ldots, R_m(X_m)$ of K-relations of schema $X_1, \ldots, X_m$ is globally consistent.

Theorem 3 *Let K be a positive semiring and let H be a hypergraph. The following statements are equivalent:*

(a) H is an acyclic hypergraph.
(b) H has the local-to-global consistency property for K-relations.

The claim that (a) implies (b) in Theorem 3 follows directly from the preceding Lemma 12. We concentrate on proving that (b) implies (a). To do this, we need four technical lemmas. By Theorem 2, a hypergraph H is acyclic if and only if it is conformal and chordal. The first two technical lemmas state, in effect, that the "minimal" non-conformal hypergraphs, as well as the "minimal" non-chordal hypergraphs, have very simple forms.

Lemma 13 (Brault-Baron, 2016) *A hypergraph $H = (V, E)$ is not conformal if and only if there exists a subset W of V with the property that $|W| \geq 3$ and $R(H[W]) = (W, \{W \setminus \{A\} : A \in W\})$, where $R(H[W])$ is the reduction of the hypergraph $H[W]$ induced by W.*

Proof The *if* direction is immediate since, given that $|W| \geq 3$, the set W forms a clique in the primal graph that is not included in any hyperedge of H; otherwise no $W \setminus \{A\}$ with $A \in W$ would be a hyperedge in the reduced hypergraph of $H[W]$. For the *only if* direction, let W be a clique in the primal graph of H that is not included in any hyperedge of H and that is minimal with this property. Since the two vertices of every edge of the primal graph are included in some hyperedge of H we have $|W| \geq 3$. In addition, by minimality of W, each $W \setminus \{A\}$ with $A \in W$ is included in some hyperedge X of A that does not contain W, so $X \cap W = W \setminus \{A\}$. This means that each $W \setminus \{A\}$ is a hyperedge of $H[W]$, and also of its reduced hypergraph since W is not included in any hyperedge of H. Conversely, if X is a hyperedge in E, then there is some $A \in W$ such that $X \cap W \subseteq W \setminus \{A\}$. It follows that $R(H[W]) = (W, \{W \setminus \{A\} : A \in W\})$. □

Lemma 14 *A hypergraph $H = (V, E)$ is not chordal if and only if there exists a subset W of V such that $|W| \geq 4$ and $R(H[W]) = (W, \{\{A_i, A_{i+1}\} : i \in [n]\})$, where $A_1, \ldots, A_n$ is an enumeration of W and $A_{n+1} := A_1$.*

Proof The *if* direction is immediate since it implies that the primal graph of H contains a chordless cycle of length at least four. For the *only if* direction, let W be the set of vertices of a shortest chordless cycle of length at least four in the primal graph of H. □

The next two technical lemmas state that the local-to-global consistency property for K-relations is preserved under induced hypergraphs, and also under reductions.

Lemma 15 *If a hypergraph H has the local-to-global consistency property for K-relations, then for every subset W of the vertices of H the hypergraph $H[W]$ also has the local-to-global consistency property for K-relations.*

Proof Assume that the hypergraph $H = (V, E)$ has the local-to-global consistency property for K-relations. We will show that, for every vertex $A \in V$, the hypergraph $H[V \setminus \{A\}] = (V, \{X \setminus \{A\} : X \in E\})$ also has the local-to-global consistency property for K-relations. The statement of the lemma will follow from iterating this statement over all attributes A in $V \setminus W$.

Let $X_1, \ldots, X_m$ be a listing of all hyperedges of H. Fix a vertex A in V and write $Y_i := X_i \setminus \{A\}$ for all $i \in [m]$. Let $R_1, \ldots, R_m$ be a collection of pairwise consistent K-relations for $Y_1, \ldots, Y_m$. Fix an arbitrary value u_0 in the domain $\mathrm{Dom}(A)$ of the attribute A. We define a collection of K-relations $S_1, \ldots, S_m$ for $X_1, \ldots, X_m$ as follows. For each $i \in [m]$ with $A \notin X_i$, let $S_i := R_i$. For each $i \in [m]$ with $A \in X_i$, let S_i be the K-relation over X_i defined for every X_i-tuple t by $S_i(t) := 0$ if $t(A) \neq u_0$ and by $S_i(t) := R_i(t[Y_i])$ if $t(A) = u_0$. We claim that the K-relations $S_1, \ldots, S_m$ are pairwise consistent.

In order to see this, fix $i, j \in [m]$ and distinguish the two cases whether $A \notin X_i X_j$ or $A \in X_i X_j$: If $A \notin X_i X_j$, then $S_i = R_i$ and $S_j = R_j$ and therefore S_i and S_j are consistent because R_i and R_j are consistent. If $A \in X_i X_j$, then let R be a K-relation over $Y_i Y_j$ that witnesses the consistency R_i and R_j and let S be the K-relation over $X_i X_j$ defined for every $X_i X_j$-tuple t by $S(t) := 0$ if $t(A) \neq u_0$ and by $S(t) := R(t[Y_i Y_j])$ if $t(A) = u_0$. We claim that S witnesses the consistency of S_i and S_j. We show that $S_i \equiv S[X_i]$ and a symmetric argument will show that $S_j \equiv S[X_j]$. In order to see this, first we argue that $R[Y_i] = S[Y_i]$. Indeed, for every Y_i-tuple r we have

$$R(r) = \sum_{\substack{s \in R': \\ s[Y_i]=r}} R(s) = \sum_{\substack{t \in \mathrm{Tup}(X_i X_j): \\ t[Y_i]=r, t(A)=u_0}} R(t[Y_i Y_j]) = \sum_{\substack{t \in S': \\ t[Y_i]=r}} S(t) = S(r), \tag{17.53}$$

where the first equality follows from (17.2), the second follows from the fact that the map $t \mapsto t[Y_i Y_j]$ is a bijection between the set of $X_i X_j$-tuples t such that $t[Y_i] = r$ and $t(A) = u_0$ and the set of $Y_i Y_j$-tuples s such that $s[Y_i] = r$, the third follows from the definition of S, and the fourth follows from (17.2). For later use, let us note that we did not assume that $i \neq j$ for showing (17.53). In case $i = j$, the K-relation R_i can serve as R, and S equals S_i, which shows that $R_i = S_i[Y_i]$.

In case $A \notin X_i$, we have that $Y_i = X_i$, hence Eq. (17.53) already shows that $S_i = R_i \equiv R[Y_i] = S[Y_i]$, so $S_i \equiv S[X_i]$. In case $A \in X_i$, let $a, b \in K \setminus \{0\}$ be such that $aR_i = bR[Y_i]$ and we show $aS_i = bS[X_i]$. For every X_i-tuple r with $r(A) \neq u_0$, we have $S_i(r) = 0$ and also $S(r) = \sum_{t:t[X_i]=r} S(t) = 0$ since $t[X_i] = r$ and $A \in X_i$ implies $t(A) = r(A) \neq u_0$. Thus, $aS_i(r) = 0 = bS(r)$ in this case. For every X_i-tuple r with $r(A) = u_0$, we have

$$aS_i(r) = aR_i(r[Y_i]) = bR(r[Y_i]) = bS(r[Y_i]), \tag{17.54}$$

where the first equality follows from the definition of S_i and the assumption that $r(A) = u_0$, the second follows from the choices of a and b, and the third follows from (17.53). Continuing from the right-hand side of (17.54), we have

$$bS(r[Y_i]) = b \sum_{\substack{t \in S': \\ t[Y_i]=r[Y_i]}} S(t) = b \sum_{\substack{t \in S': \\ t[X_i]=r}} S(t) = bS(r), \tag{17.55}$$

where the first equality follows from (17.2), the second follows from the assumption that $A \in X_i$ and $r(A) = u_0$ together with $S(t) = 0$ in case $t(A) \neq u_0$, and the third follows from (17.2). Combining (17.54) with (17.55), we get $aS_i(r) = bS(r)$ also in this case. This proves that $S_i \equiv S[X_i]$; a completely symmetric argument proves that $S_j \equiv S[X_j]$.

Since $S_1, \ldots, S_m$ are pairwise consistent K-relations for $X_1, \ldots, X_m$, by assumption they are globally consistent. Let N be a K-relation over $X_1 \cup \cdots \cup X_m$ that witnesses their consistency. Let $M := N[Y_1 \cup \cdots \cup Y_m]$ and we argue that M witnesses the consistency of $R_1, \ldots, R_m$, which will prove the lemma. Fix $i \in [m]$ and let $a, b \in K \setminus \{0\}$ be such that $aS_i = bS[X_i]$. For every Y_i-tuple r we have

$$aR_i(r) = aS_i(r) = bN(r) = bM(r), \tag{17.56}$$

where the first equality follows $S_i = R_i$ in case $A \notin X_i$ and from (17.53) applied to $i = j$ and $R = R_i$ in case $A \in X_i$, the second follows from the choice of a and b, and the third follows from the choice of M and Part 3 of Lemma 1. $\square$

Lemma 16 *If a hypergraph H has the local-to-global consistency property for K-relations, then $R(H)$ also has the local-to-global consistency property for K-relations.*

Proof Assume that the hypergraph $(V, \{X_1, \ldots, X_m\})$ has the local-to-global consistency property for K-relations. We will show that if X_m is covered by some other hyperedge, i.e., $X_m \subseteq X_i$ for some $i \leq m-1$, then the hypergraph $(V, \{X_1, \ldots, X_{m-1}\})$ also has the local-to-global consistency property for K relations. The statement of the lemma will follow from iterating this statement over all hyperedges of H that are covered by some other hyperedge of H.

Let $R_1, \ldots, R_{m-1}$ be a collection of pairwise consistent K-relations for $X_1, \ldots, X_{m-1}$. Define $R_m := R_i[X_m]$. We claim that $R_1, \ldots, R_m$ are pairwise consistent. It suffices to check that R_j and R_m are consistent for any $j \in [m]$ with $j \neq m$. By assumption, we know that R_j and R_i are consistent, which means that there exists a K-relation T that witnesses their consistency; we have $R_j \equiv T[X_j]$ and $R_i \equiv T[X_i]$. Let $S := T[X_j X_m]$. We have

$$R_j \equiv T[X_j] = T[X_j X_m][X_j] = S[X_j], \tag{17.57}$$

where the first equality follows from the choice of T, the second follows from Part 3 of Lemma 1 and the third follows from the choice of S. Likewise,

$$R_m = R_i[X_m] \equiv T[X_i][X_m] = T[X_m] = T[X_j X_m][X_m] = S[X_m], \tag{17.58}$$

where the first equality is by the choice of R_m, the equivalence follows from the choice of T, the assumption that $X_m \subseteq X_i$, and Part 3 of Lemma 2, the next equality follows from the assumption that $X_m \subseteq X_i$ and Part 3 of Lemma 1, the one to last equality follows again from Part 3 of Lemma 1, and the last equality follows from the choice of S. Thus, S witnesses the consistency of R_j and R_m.

Since $R_1 \ldots, R_m$ are pairwise consistent K-relations for $X_1, \ldots, X_m$, by assumption they are globally consistent. The same K-relation that witnesses their global consistency also witnesses the global consistency of $R_1, \ldots, R_{m-1}$, which completes the proof. $\square$

Generalized Tseitin Construction and Proof of Theorem 3 The minimal non-conformal and minimal non-chordal hypergraphs from Lemmas 13 and 14 share the following properties: (1) all their hyperedges have the same number of vertices, and (2) all their vertices appear in the same number of hyperedges. For hypergraphs H that have these properties, we construct a collection $C(H; K)$ of K-relations that are indexed by the hyperedges of H; these relations will play a crucial role in the proof of Theorem 3.

Let $H = (V, E)$ be a hypergraph and let d and k be positive integers. The hypergraph H is called *k-uniform* if every hyperedge of H has exactly k vertices. It is called *d-regular* if any vertex of H appears in exactly d hyperedges of H. For example, the non-conformal hypergraph of Lemma 13 is k-uniform and d-regular for $k := d := |W| - 1$. Likewise, the non-chordal hypergraph of Lemma 14 is k-uniform and d-regular for $k := d := 2$. For a positive semiring K and each k-uniform and d-regular hypergraph H with $d \geq 2$ and with hyperedges $E = \{X_1, \dots, X_m\}$, we construct a collection of K-relations $C(H; K) := \{R_1(X_1), \dots, R_m(X_m)\}$, where R_i is a K-relation that has attributes X_i. The collection $C(H; K)$ of these K-relations will turn out to be pairwise consistent but not globally consistent. Note that by the characterization of acyclicity in terms of Graham's algorithm, a hypergraph that is k-uniform and d-regular for some $k \geq 2$ and $d \geq 2$ will never be acyclic: Graham's procedure will not even start to remove any hyperedge or any vertex. Hence, the existence of the K-relations $R_1, \dots, R_m$ that violates the local-to-global consistency property is compatible with Theorem 3. The construction is defined as follows.

The attributes of all the K-relations $R_1, \dots, R_m$ have domain $\mathbb{Z}/d\mathbb{Z}$, which we identify with the initial segment of the integers $\{0, \dots, d-1\}$. For each $i \in [m]$ with $i \neq m$, let R_i be the unique K-relation over X_i whose support contains all tuples $t : X_i \to \{0, \dots, d-1\}$ whose total sum $\sum_{C \in X_i} t(C)$ is congruent to 0 mod d; i.e., $R(t) := 1$ for each such X_i-tuple, and $R(t) := 0$ for any other X_i-tuple. For $i = m$, let R_i be the unique K-relation over X_i whose support contains all tuples $t : X_i \to \{0, \dots, d-1\}$ whose total sum $\sum_{C \in X_i} t(C)$ is congruent to 1 mod d; i.e., again $R(t) := 1$ for each such X_i-tuple, and $R(t) := 0$ otherwise.

To show the pairwise consistency of $R_1, \dots, R_m$, it suffices, by Lemma 7, to show that for every two distinct $i, j \in [m]$, we have $R_i[Z] \equiv R_j[Z]$, where $Z := X_i \cap X_j$. In turn, this follows from the claim that for every Z-tuple $t : Z \to \{0, \dots, d-1\}$, we have $R_i(t) = R_j(t) = N_Z 1 = 1 + \cdots + 1$, the sum of $N_Z := d^{k-|Z|-1}$ many units of the semiring K. Indeed, since by k-uniformity every hyperedge of H has exactly k vertices, for every $u \in \{0, \dots, d-1\}$, there are exactly N_Z many X_i-tuples $t_{i,u,1}, \dots, t_{i,u,N_Z}$ that extend t and have total sum congruent to u mod d. It follows then that $R_i[Z] = R_j[Z]$ regardless of whether $n \in \{i, j\}$ or $n \notin \{i, j\}$, and hence any two R_i and R_j are consistent by Lemma 7. To argue that the relations $R_1, \dots, R_m$ are not globally consistent, we proceed by contradiction. If R were a K-relation that witnesses their consistency, then it would be non-empty and its support would contain a tuple t such that the projections $t[X_i]$ belong to the supports R'_i of the R_i, for each $i \in [m]$. In turn this means that

$$\textstyle\sum_{C \in X_i} t(C) \equiv 0 \bmod d, \quad \text{for } i \neq m \tag{17.59}$$
$$\textstyle\sum_{C \in X_i} t(C) \equiv 1 \bmod d, \quad \text{for } i = m. \tag{17.60}$$

Since by d-regularity each $C \in V$ belongs to exactly d many sets X_i, adding up all the equations in (17.59) and (17.60) gives

$$\textstyle\sum_{C \in V} dt(C) \equiv 1 \bmod d, \tag{17.61}$$

which is absurd since the left-hand side is congruent to 0 mod d, the right-hand side is congruent to 1 mod d, and $d \geq 2$ by assumption.

We now have all the tools needed to present the proof of Theorem 3.

Proof of Theorem 3 As stated earlier, the direction (a) implies (b) follows from Lemma 12. For showing that (b) implies (a), let us assume the contrary and then we will derive a contradiction. Let $H = (V, E)$ be a smallest counterexample to the statement that (b) implies (a), meaning that the following three conditions hold: (i) H is not acyclic, (ii) H has the local-to-global consistency property for K-relations, and (iii) H is minimal in the sense that $n = |V|$ is smallest possible with properties (i) and (ii), and among those, $m = |E|$ is smallest possible. Since H is not acyclic, we know that H is either not conformal or not chordal. We distinguish the two cases.

Case 1: H is not conformal. By Lemmas 15 and 16, the minimality of H, and Lemma 13, we have $n \geq 3$ and $m = n$; indeed, $E = \{V_i : i \in [n]\}$ where $V_i = V \setminus \{A_i\}$ and $A_1, \ldots, A_n$ is an enumeration of V. Thus, H is k-uniform and d-regular for $k = d = n - 1 \geq 2$. The construction $C(H; K)$ gives a collection of K-relations $R_1, \ldots, R_n$, where R_i has attributes V_i, which are pairwise consistent but not globally consistent, which is a contradiction.

Case 2: H is not chordal. By Lemmas 15 and 16, the minimality of H, and Lemma 14, we have $n \geq 4$ and $m = n$, and indeed $E = \{V_i : i \in [n]\}$ where $V_i = \{A_i, A_{i+1}\}$ and $A_1, \ldots, A_n$ is an enumeration of V with $A_{n+1} := A_1$. Thus, H is k-uniform and d-regular for $k = d = 2$. Again, the construction $C(H; K)$ gives a collection of K-relations $R_1, \ldots, R_n$, where R_i has attributes V_i, which are pairwise consistent but not globally consistent, which is a contradiction. □

An inspection of the proof of Theorem 3 reveals that actually a stronger result is established. Specifically, let H be a hypergraph and let $X_1, \ldots, X_m$ be its hyperedges. The proof of Theorem 3 shows that if H is not acyclic, then there are ordinary relations $R_1(X_1), \ldots, R_m(X_m)$ such that, for every positive semiring K, the K-relations $S_1(X_1), \ldots, S_m(X_m)$ with $S_i' = R_i$ are pairwise consistent but globally inconsistent. In more informal terms, the proof of Theorem 3 actually shows that if a hypergraph is acyclic, then there is an essentially *uniform* counterexample to the local-to-global consistency property for K-relations that works for all positive semirings K.

Before establishing the main result in this paper, we bring into the picture one more notion from hypergraph theory that was introduced by Vorob'ev (1962) in his study of global consistency for probability distributions.

Vorob'ev Regular Hypergraphs A *complex* is a hypergraph $H = (V, E)$ whose set E of hyperedges is closed under taking subsets, i.e., if $X \in E$ and $Z \subseteq X$, then $Z \in E$. The *downward closure* of a hypergraph H is the hypergraph whose vertices are those of H and whose hyperedges are all the subsets of the hyperedges of H. Clearly, the downward closure of a hypergraph is a complex.

Let K be a complex. Let X and Y be two different maximal hyperedges of K, where a hyperedge is *maximal* if it is not a proper subset of any other hyperedge. We say that X yields a maximal intersection with Y if the intersection $X \cap Y$ is not a proper subset of the intersection $X \cap Z$ of some third[1] hyperedge Z of K. A maximal hyperedge X of K is called *extreme in* K if all maximal intersections of X with hyperedges of K are equal. Let X be an extreme hyperedge in K. The *proper* vertices of X are those that do not belong to any other maximal hyperedge of K; it should be emphasized that, according to this definition, a proper vertex of X is allowed to belong to some other hyperedge as long as it is a subset of X. The *normal subcomplex* of K corresponding to the extreme edge X is the subcomplex of K consisting of all hyperedges of K that do not intersect the set of proper vertices of X. A subcomplex K' of K is called a *normal subcomplex* if it is the normal subcomplex corresponding to some extreme hyperedge of K. A *normal series* of K is a sequence of subcomplexes

$$K = K_0 \supset K_1 \supset \cdots \supset K_r \tag{17.62}$$

of the complex K in which for every ℓ with $1 \leq \ell \leq r - 1$, the complex $K_{\ell+1}$ is a normal subcomplex of the complex K_ℓ, and the final complex K_r does not have any extreme hyperedges. We say that K is *regular* if there exists a normal series of K in which the last term is the complex without vertices. We say that a hypergraph H is *Vorob'ev regular* if its downward closure is a regular complex.

We will show that a hypergraph is Vorob'ev regular if and only if it is acyclic. This result will follow from the next two lemmas and Theorem 2. We have not been able to locate a published proof of this result in the literature, even though the equivalence between acyclicity and Vorob'ev regularity has been mentioned in Hill (1991) and Yannakakis (1996).

Lemma 17 *If H is a Vorob'ev regular hypergraph, then Graham's algorithm succeeds on H.*

Proof Let H be a Vorob'ev regular hypergraph. The proof that Graham's algorithm succeeds on H is by induction on the length of a normal series of the downward closure K of H. If the length of a normal series of K is zero, then K and hence H itself is the empty hypergraph and there is nothing to prove. Assume then that K has

[1] The condition "third" is missing in Vorob'ev (1962), and it is necessary since $X \cap Y$ is always a proper subset of $X \cap Z$ whenever $X = Z$ and X is maximal.

a normal series that has length at least one, let K_1 be the first subcomplex of K in the series, and let X be the extreme hyperedge of K corresponding to which K_1 is its normal subcomplex. Then K_1 consists of the hyperedges of K that do not intersect the proper vertices of X. Equivalently, K_1 is obtained from K by deleting all the proper vertices of X. Since the proper vertices of X appear in no other maximal hyperedge of K, this means that if we delete the proper vertices of X from all the hyperedges of H in which they appear, then we obtain a hypergraph whose downward closure is K_1. Moreover, such a hypergraph H_1 can be obtained from H by a applying a sequence of operations of Graham's algorithm: first delete all the hyperedges that are proper subsets of X, then delete all the proper vertices of X. Now, K_1 is also Vorob'ev's regular, and its normal series has length one less than that of K. Hence, by induction hypothesis, Graham's algorithm succeeds on H_1, which means that there is a sequence of operations of Graham's algorithm that applied to H_1 yield the empty hypergraph. By concatenating the two sequences of operations, we get a single sequence of operations of Graham's algorithm that starts at H and yields the empty hypergraph. This proves that Graham's algorithm succeeds on H. □

Lemma 18 *If H is a hypergraph that has a join tree, then H is Vorob'ev regular.*

Proof Let H be a hypergraph that has a join tree. By induction on the number of its hyperedges, we show that H is Vorob'ev regular. Let K be the downward closure of H. If H has no hyperedges, then it is Vorob'ev regular. If H has just one hyperedge X, then X is an extreme hyperedge of K since it yields no maximal intersections at all. It follows that K is regular since all the vertices of e are proper, so H is Vorob'ev regular. Assume now that H has more than one hyperedges. Let X be a leaf of the join tree of H, and let Y be the unique hyperedge of H such that $\{X, Y\}$ is an edge of the join tree. We consider the following two cases.

Case 1: If X is not a maximal hyperedge, say $X \subseteq Z$ for some other edge Z of H, then $X = X \cap Z \subseteq X \cap Y$ by the connectivity property of the join tree since the unique path from X to Z in the tree must pass through Y. Hence, $X \subseteq Y$. Now, let H' be the hypergraph that results from deleting X from H. Since $X \subseteq Y$, the tree that results from trimming the leaf X from the join tree of H is a join tree of H'. By induction hypothesis, H' is Vorob'ev regular. But X was not maximal in H, so H and H' have the same downward closure K, which shows that H is also Vorob'ev regular.

Case 2: If X is a maximal hyperedge of H, then we claim that X is an extreme hyperedge of K. First, X is a maximal hyperedge of K by assumption. Second, we show that X yields maximal intersection with Y. Indeed, if Z is a third hyperedge of H, then the unique path from X to Z in the join tree goes through Y and, by the connectivity property of the join tree, every vertex in $X \cap Z$ belongs to $Z \cap Y$. So $X \cap Y$ is not a proper subset of $X \cap Z$. Next, we show that all maximal intersections of X are equal to the maximal intersection of X with Y. Let Z be a third hyperedge of H and assume that X yields a maximal intersection with Z. In particular, $X \cap Z$ is a not a proper subset of $X \cap Y$. But the connectivity property of the join tree implies $X \cap Z \subseteq X \cap Y$ since the unique path from X to Z in the join tree goes through Y. Thus $X \cap Z = X \cap Y$.

We proved that X is an extreme hyperedge of K. Now, let H' be the hypergraph that is obtained from H by deleting X and all the vertices that appeared only in X. The normal subcomplex K' of K corresponding to X is the downward closure of H'. If we trim the leaf X from the join tree of H, we get a join tree of H' with one node less. It follows from the induction hypothesis that H' is Vorob'ev regular. Thus K' is Vorob'ev regular, and so is K since K' is a normal subcomplex of it. $\square$

Lemmas 17, 18, and Theorem 2 imply the following result.

Corollary 2 *Let H be a hypergraph. The following statements are equivalent:*

(a) H is an acyclic hypergraph.
(b) H is a Vorob'ev regular hypergraph.

Finally, by combining Theorems 2, 3, and Corollary 2, we obtain the main result of this paper.

Theorem 4 *Let K be a positive semiring and let H be a hypergraph. The following statements are equivalent:*

(a) H is an acyclic hypergraph.
(b) H is a conformal and chordal hypergraph.
(c) H has the running intersection property.
(d) H has a join tree.
(e) H is accepted by Graham's algorithm.
(f) H is a Vorob'ev regular hypergraph.
(g) H has the local-to-global consistency property for K-relations.

By applying Theorem 4 with K set to the Boolean semiring $\mathbb{B}$, we obtain the original Beeri–Fagin–Maier–Yannakakis Theorem 2. It should be noted that while our proof of Theorem 4 actually used Theorem 2 several times, all the uses of Theorem 2 made were for arguing that the various syntactic characterizations of hypergraph acyclicity are equivalent. These equivalences can be shown directly without referring to any semantic notion, and for this reason we can say that our proof does *not* rely on the semantic part of Theorem 2. In fact, the main construction that we gave in the proof of Theorem 3 appears to be new and gives an alternative proof of the semantic part of Theorem 2: If H has local-to-global consistency property for ordinary relations, then H is acyclic.

While still different, the construction we gave in the proof of Theorem 3 is closer to Vorob'ev's proof of his theorem from Vorob'ev (1962) than to the proof of the Theorem 2 from Beeri et al. (1983). We discuss this next.

17.4.3 Consistency of Distributions and Vorob'ev's Theorem

In the rest of this section, we study the consistency of K-relations when the semiring K is $\mathbb{R}^{\geq 0}$, i.e., the set of non-negative real numbers with the standard addition and multiplication operations. We place the focus on probability distributions, which, to recall, are nothing but the $\mathbb{R}^{\geq 0}$-relations T that satisfy the normalization constraint

$$T[\emptyset] = \sum_{t \in T'} T(t) = 1. \tag{17.63}$$

Our goal for the rest of this section is to show that the main result of Vorob'ev from Vorob'ev (1962) follows from our general result Theorem 4 about arbitrary positive semirings.

Consistency of Probability Distributions Following Vorob'ev (1962), we say that two probability distributions $P(X)$ and $Q(Y)$ are consistent if there exists a probability distribution $T(XY)$ such that $T[X] = P$ and $T[Y] = Q$. A collection $P_1(X_1), \ldots, P_m(X_m)$ of probability distributions is *globally consistent* if there exists a probability distribution $P(X_1 \cdots X_m)$ such that $P[X_i] = P_i$ holds for every $i \in [m]$. The collection is called *pairwise consistent* if any two distributions in the collection are consistent. We start by showing that, when the probability distributions are presented as $\mathbb{R}^{\geq 0}$-relations that satisfy (17.63), this classical notion of consistency of probability distributions coincides with the notion of consistency that we have been studying in this paper. The following basic fact was observed already in Sect. 17.2. It says that, as $\mathbb{R}^{\geq 0}$-relations, the probability distributions are the canonical representatives of their equivalence classes under $\equiv$. We give an even shorter proof in a slightly different language.

Lemma 19 *For every two probability distributions $R(X)$ and $S(X)$ over the same set of attributes X, it holds that $R \equiv S$ if and only if $R = S$.*

Proof The *if* direction is trivial. For the *only if* direction, let a and b be positive reals such that $aR(t) = bS(t)$ holds for every X-tuple t. Then $a = aR[\emptyset] = bS[\emptyset] = b$, where the first equality follows from (17.63), the second follows from (17.2) and the choice of a and b, and the third follows from (17.63). Dividing through by $a = b \neq 0$, it follows that $R(t) = S(t)$ holds for every X-tuple t. □

The next lemma states that any $\mathbb{R}^{\geq 0}$-relation that witnesses the consistency of a collection of probability distributions is itself a probability distribution.

Lemma 20 *For every collection $R_1, \ldots, R_m$ of probability distributions and every $\mathbb{R}^{\geq 0}$-relation R, if R witnesses the global consistency of $R_1, \ldots, R_m$, then R is itself a probability distribution.*

Proof For any $i \in [m]$ we have $R[X_i] = R_i$, where X_i is the set of attributes of R_i. By Part 3 of Lemma 1 we have $R[\emptyset] = R[X_i][\emptyset] = R_i[\emptyset] = 1$, for any $i \in [m]$; i.e., R satisfies (17.63) and is hence a probability distribution. □

In view of Lemmas 19 and 20, the classical notions of consistency of probability distributions coincides with the notions of consistency of K-relations that we have been studying in this paper when $K = \mathbb{R}^{\geq 0}$. We are ready to state the local-to-global consistency property for probability distributions.

Vorob'ev's Theorem Let H be a hypergraph and let $X_1, \ldots, X_m$ be a listing of all its hyperedges. We say that H has the *local-to-global consistency property for probability distributions* if every pairwise consistent collection of probability distributions $P_1(X_1), \ldots, P_m(X_m)$ is globally consistent.

One of the main motivations for writing this paper was to obtain a common generalization of the result of Beeri, Fagin, Maier, and Yannakakis stated in Theorem 2 and the result of Vorob'ev stated next.

Theorem 5 (Theorem 4.2 in Vorob'ev's, 1962) *Let H be a hypergraph. The following statements are equivalent:*

(a) H is a Vorob'ev regular hypergraph.
(b) H has the local-to-global consistency property for probability distributions.

Proof By Lemmas 19 and 20, conditions (b) in this Theorem and (g) in Theorem 4 for $K = \mathbb{R}^{\geq 0}$ are equivalent. The result now follows from Theorem 4 for $K = \mathbb{R}^{\geq 0}$. $\square$

Some words on the differences between our proof of Theorem 5 and Vorob'ev's proof of Theorem 4.2 in Vorob'ev (1962) are in order. In the direction (a) implies (b), except for the minor differences that stem from the use of Lemma 17, our proof is basically the same as Vorob'ev's. The main construct in that proof is the operation on probability distributions that we denoted $\bowtie_{\mathrm{P}}$ in Eq. (17.8), which appears with different notation as equation (21) in page 156 of Vorob'ev (1962).

In the direction (b) implies (a), again except for the minor differences that stem from the use of Lemma 18, our proof has some important similarities with Vorob'ev's, but also one important difference. The similarities lie in the structure of the argument. Vorob'ev first proves that the class of hypergraphs that have the local-to-global consistency property for probability distributions is the *unique* class of hypergraphs that contains those, satisfies certain closure properties, and excludes two concrete families of hypergraphs, that he calls $\{G_n\}$ and $\{Z_n\}$ in Theorem 2.2 in page 152 of Vorob'ev (1962). This characterization we also prove through the combination of Lemmas 15 and 16, which stand for the closure properties, Lemmas 13 and 14, whose featuring hypergraphs are precisely the hypergraphs G_n and Z_n from Vorob'ev (1962), and the construction $C(H; K)$ through Case 1 for $H = G_n$, and Case 2 for $H = Z_n$, in the proof of Theorem 3. Another similarity lies in the way we handle Z_n: both proofs build a cycle of K-relations that implement equality constraints except for one K-relation in the cycle that implements an inequality constraint. The important difference lies in the way we handle G_n. Vorob'ev's proof is a linear-algebraic argument over Euclidean space that requires some non-trivial calculations, while our argument is more combinatorial and arguably simpler as it relies on basic modular arithmetic and the totally obvious fact that $0 \not\equiv 1 \bmod d$, for $d \geq 2$.

17.5 Concluding Remarks

We conclude by discussing some open problems and directions for future research.

- Beeri et al. (1983) showed that hypergraph acyclicity is also equivalent to certain semantic conditions other than the local-to-global consistency property for ordinary relations. The existence of a *full reducer* is arguably the most well known and useful such semantic property (see also Maier, 1983; Ullman, 1988). By definition, a *full reducer* of a hypergraph H with $X_1, \ldots, X_m$ as its hyperedges is a program consisting of a finite sequence of semijoin statements of the form $R_i := R_i \ltimes R_j$ such that if this program is given a collection $R_1(X_1), \ldots, R_m(X_m)$ of ordinary relations as input, then the output is a collection of pairwise consistent ordinary relations.
  It remains an open problem to define a suitable semijoin operation for K-relations and prove (or disprove) that for every positive semiring K, a hypergraph H is acyclic if and only if H has a full reducer for K-relations. One of the technical difficulties is that the join operation on two K-relations introduced and studied here is not, in general, associative (in fact, as seen earlier, it is not associative even when K is the bag semiring of non-negative integers).
- Here, we studied the notion of consistency up to normalization, which is based on the notion of equivalence $\equiv$ of two K-relations. As mentioned in Sect. 17.3.1, one could define the notion of strict consistency, where two K-relations $R(X)$ and $S(Y)$ are *strictly consistent* if there is a K-relation $T(XY)$ such that $T[X] = R$ and $T[Y] = S$. We already pointed out that consistency up to normalization and strict consistency are different notions for bags. Furthermore, there are bags R and S that are strictly consistent, but every bag T that witnesses their strict consistency has the property that its support T' is *properly contained* in the ordinary join $R' \bowtie S'$ of the supports R' and S'. One of the simplest such examples is the pair of bags $R(AB) = \{12 : 1,\ 22 : 1\}$ and $S(BC) = \{21 : 1,\ 22 : 1\}$, whose strict consistency is witnessed by the bags $T_1(ABC) = \{122 : 1,\ 221 : 1\}$ and $T_2(ABC) = \{121 : 1,\ 222 : 1\}$, but, as one can easily verify, no other bag. In contrast, if two K-relations R and S are consistent up to normalization, then the join operation $R \bowtie S$ we introduced here witnesses the consistency up to normalization of R and S, and its support is the ordinary join of the supports of R and S (see Lemma 4).
  The aforementioned differences between consistency up to normalization and strict consistency notwithstanding, it is still possible to obtain results about strict consistency for bags that are analogous to results established here for consistency up to normalization for K-relations. Specifically, in Atserias and Kolaitis (2021), we defined the notion of the *local-to-global strict consistency property for bags*[2] and, using tools from linear programming and maximum flow problems, we showed that this property is equivalent to hypergraph acyclicity. Note that the notion of

[2] In Atserias and Kolaitis (2021), we use the term *bag consistency*, instead of the term *strict consistency for bags*, since this is the only notion of consistency considered in that paper.

local-to-global strict consistency property can be defined for K-relations over an arbitrary, but fixed, positive semiring K. It is an open problem to determine for which positive semirings K the local-to-global strict consistency property is equivalent to hypergraph acyclicity.

- Positive semirings have been used in different areas of mathematics and computer science (see, e.g., Golan, 2013; Gondran & Minoux, 2008). In particular, the min-plus semiring has been used in the analysis of dynamic programming algorithms, while the Viterbi semiring has been used in the study of statistical models. It would be interesting to investigate potential applications of the results reported here to semirings other than the Boolean semiring and the semiring of the non-negative real numbers with the standard arithmetic operations.

Acknowledgements Albert Atserias was partially funded by European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, grant agreement ERC-2014-CoG 648276 (AUTAR), and by MICCIN grant PID2019-109137GB-C22 (PROOFS). Phokion Kolaitis was partially supported by NSF Grant IIS-1814152.

References

Abramsky, S., & Brandenburger, A. (2011). A unified sheaf-theoretic account of non-locality and contextuality. arXiv:1102.0264

Atserias, A., Bulatov, A. A., & Dawar, A. (2009). Affine systems of equations and counting infinitary logic. *Theoretical Computer Science, 410*(18), 1666–1683.

Abramsky, S., Barbosa, R. S., Kishida, K., Lal, R., & Mansfield, S. (2015). Contextuality, cohomology and paradox. In S. Kreutzer (Ed.), *24th EACSL annual conference on computer science logic, CSL 2015, September 7–10, 2015, Berlin, Germany*. LIPIcs (Vol. 41, pp. 211–228). Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Abramsky, S. (2013). Relational databases and Bell's theorem. In V. Tannen, L. Wong, L. Libkin, W. Fan, W.-C. Tan, & M. P. Fourman (Eds.), *Search of elegance in the theory and practice of computation - Essays dedicated to Peter Buneman*. Lecture notes in computer science (Vol. 8000, pp. 13–35). Berlin: Springer.

Abramsky, S. (2014). Contextual semantics: From quantum mechanics to logic, databases, constraints, and complexity. *Bulletin of the EATCS, 113*.

Abiteboul, S., Hull, R., & Vianu, V. (1995). *Foundations of databases*. Boston: Addison-Wesley.

Atserias, A., & Kolaitis, P. G. (2021). Structure and complexity of bag consistency. In *Proceedings of the Fortieth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 2021)* (pp. 247–259).

Abramsky, S., Mansfield, S., & Barbosa, R. S. (2011). The cohomology of non-locality and contextuality. In B. Jacobs, P. Selinger, & B. Spitters (Eds.), *Proceedings 8th International Workshop on Quantum Physics and Logic, QPL 2011, Nijmegen, Netherlands, October 27–29, 2011. EPTCS* (Vol. 95, pp. 1–14).

Barbosa, R. S. (2015). *Contextuality in quantum mechanics and beyond*. Ph.D thesis, University of Oxford, Oxford, UK.

Bell, J. S. (1964). On the Einstein-Podolsky-Rosen paradox. *Physics Physique Fizika, 1*(3), 195.

Beeri, C., Fagin, R., Maier, D., & Yannakakis, M. (1983). On the desirability of acyclic database schemes. *Journal of the ACM, 30*(3), 479–513.

Buss, S. R., Grigoriev, D., Impagliazzo, R., & Pitassi, T. (2001). Linear gaps between degrees for the polynomial calculus modulo distinct primes. *Journal of Computer and System Sciences, 62*(2), 267–289.

Brault-Baron, J. (2016). Hypergraph acyclicity revisited. *ACM Computing Surveys, 49*(3), 54:1–54:26.

Chvátal, V., & Szemerédi, E. (1988). Many hard examples for resolution. *Journal of the ACM, 35*(4), 759–768.

Cover, T. M., & Thomas, J. A. (2006). *Elements of information theory* (2nd ed.). New York: Wiley.

Dechter, R. (2003). *Constraint processing*. Elsevier Morgan Kaufmann.

Francis, G. K. (2007). The impossible tribar. *A topological picturebook* (pp. 65–76). Berlin: Springer.

Green, T. J., Karvounarakis, G., & Tannen, V. (2007). Provenance semirings. In L. Libkin (Ed.), *Proceedings of the Twenty-Sixth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 11–13, 2007, Beijing, China* (pp. 31–40). ACM.

Gondran, M., & Minoux, M. (2008). *Graphs, dioids and semirings: New models and algorithms* (Vol. 41). Berlin: Springer Science & Business Media.

Golan, J. S. (2013). *Semirings and their applications*. Berlin: Springer Science & Business Media.

Graham, M. H. (1979). On the universal relation. Technical report, University of Toronto, Ontario, Canada.

Grädel, E., & Tannen, V. (2017). Semiring provenance for first-order model checking. arXiv:1712.01980

Hardy, L. (1992). Quantum mechanics, local realistic theories, and lorentz-invariant realistic theories. *Physical Review Letters, 68*(20), 2981.

Hill, J. R. (1991). Relational databases: A tutorial for statisticians. In *Proceedings of Symposium on the Interface between Computer Science and Statistics* (pp. 86–93).

Honeyman, P., Ladner, R. E., & Yannakakis, M. (1980). Testing the universal instance assumption. *Information Processing Letters, 10*(1), 14–19.

Maier, D. (1983). *The theory of relational databases*. Computer Science Press.

Malvestuto, F. M. (1988). Existence of extensions and product extensions for discrete probability distributions. *Discrete Mathematics, 69*(1), 61–77.

Penrose, L. S., & Penrose, R. (1958). Impossible objects: A special type of visual illusion. *British Journal of Psychology*.

Popescu, S., & Rohrlich, D. (1994). Quantum nonlocality as an axiom. *Foundations of Physics, 24*(3), 379–385.

Tseitin, G. S. (1968). On the complexity of derivation in propositional calculus. *Structures in constructive mathematics and mathematical logic*, 115–125.

Ullman, J. D. (1988). *Principles of database and knowledge-base systems, Volume I*. Principles of computer science series (Vol. 14). Computer Science Press.

Ullman, J. D., & Widom, J. (2002). *A first course in database systems* (2nd ed.). Prentice: Prentice Hall.

Vorob'ev, N. N. (1962). Consistent families of measures and their extensions. *Theory of Probability & Its Applications, 7*(2), 147–163.

Yannakakis, M. (1996). Perspectives on database theory. *SIGACT News, 27*(3), 25–49.

Yu, C. T., & Ozsoyoglu, M. Z. (1979). An algorithm for tree-query membership of a distributed query. In *COMPSAC 79. Proceedings. Computer Software and The IEEE Computer Society's Third International Applications Conference, 1979* (pp. 306–312). IEEE.

Part IV
Game Comonads and Descriptive Complexity

Chapter 18
Constraint Satisfaction, Graph Isomorphism, and the Pebbling Comonad

Anuj Dawar

Abstract The pebbling comonad introduced in (Abramsky, Dawar, Wang 2017) gives a categorical account relating natural approximations of homomorphism and isomorphism. On the one hand we have the local consistency algorithms that approximate homomorphism and on the other the Weisfeiler–Leman algorithms that approximate isomorphism. Both of these have elegant characterizations as pebble games. In this paper we give a brief tour through the background that led to the definition of the pebbling comonad and look at some prospects it offers.

Keywords Finite model theory · Constraint satisfaction problem · Graph isomorphism · Comonads · Category theory

18.1 Introduction

The field of theoretical computer science has, since its inception, been dominated by two central concerns: (1) how to ensure and verify the correctness of computing systems; and (2) how to measure the resources required for computations and ensure their efficiency. These concerns have led to the development of fields of study in formal methods and semantics on the one hand, and in algorithmics and computational complexity on the other. This division runs deep, as can be seen for instance, in the two volumes of the *Handbook of Theoretical Computer Science* published in 1990 (van Leeuwen, 1990) and the similar division of the journal *Theoretical Computer Science* into two series and the *ICALP* conference into two tracks. The application of logic in computer science is considered squarely to be within the *Volume B/Track B* side of this divide. Yet, the subject of logic in computer science itself reflects on a smaller scale the divide within theoretical computer science. It can broadly be divided into *Logic and Algorithms* on the one hand and *Logic and Semantics* on the other. The former uses formal logic primarily as a description lan-

A. Dawar (✉)
University of Cambridge, Cambridge, England
e-mail: anuj.dawar@cl.cam.ac.uk

A. Palmigiano and M. Sadrzadeh (eds.), *Samson Abramsky on Logic and Structure in Computer Science and Beyond*, Outstanding Contributions to Logic 25,
https://doi.org/10.1007/978-3-031-24117-8_18

guage. It is concerned with algorithmic questions about deduction, but just as often about questions of the expressive power of a logic to make statements about computational structures such as automata, databases, graphs, etc. On the other hand, Logic and Semantics, using tools from type theory and category theory, focusses on an abstract description of programming. The two also tend to draw on different traditions within logic, one being more model-theoretic and the other more proof-theoretic. In particular, the field of *finite model theory* has developed largely through the interaction of Logic and Algorithms.

In 2016, I, along with Samson Abramsky, Phokion Kolaitis and Prakash Panangaden organized a semester-long programme at the Simons Institute for the Theory of Computing whose explicit aim was to bring together researchers from the two strands of work in logic in computer science. The programme was very successful in stimulating new interactions between these areas. One outcome worth highlighting is the paper (Abramsky et al., 2017) which I co-authored with Samson Abramsky and my former student Pengming Wang, which gives a novel category-theoretic account of some central constructions in finite model theory. In this article, I aim to give an exposition of the background in finite model theory and the questions that motivated this work. This will take us on a tour through work on two major algorithmic problems that have played a significant role in complexity theory: the *graph isomorphism problem* and the *constraint satisfaction problem*. We will look, in particular, to approaches based on logic that have been deployed in the study of these two problems.

The graph isomorphism problem has a rare status in complexity theory as one of the few natural problems in **NP** that is not known to be either in **P** or **NP**-complete. It has been intensively studied and a relatively recent result by Babai (2016) places it in quasi-polynomial time. Over the years, many special cases of the graph isomorphism problem, defined by restricting the classes of input graphs by some structural parameter such as the degree or the tree-width, have been shown to be decidable in polynomial time (see Grohe, 2017 and references therein). The algorithms deployed for this fall into two broad categories: combinatorial algorithms like the Weisfeiler–Leman method (more details on this can be found below) and algebraic algorithms based on permutation groups. Babai's algorithm relies on combinations of both of these.

The constraint satisfaction problem can be seen as the problem of determining the existence of a homomorphism between finite relational structures, as first noted by Feder and Vardi (1998). Feder and Vardi conjectured that for every fixed such structure $\mathbb{D}$, the class of structures $\mathsf{CSP}(\mathbb{D})$ which admit homomorphisms to $\mathbb{D}$ is either in **P** or **NP**-complete. The conjecture has now been confirmed by results of Bulatov (2017) and Zhuk (2017). It was already observed in the seminal paper of Feder and Vardi that those structures $\mathbb{D}$ for which $\mathsf{CSP}(\mathbb{D})$ is known to be in polynomial time fall into two classes: those where *logical* methods work and those where *group theoretic* methods work. The logical methods in question are based on definability in Datalog. A key insight in the years of research that followed was that the "group-theoretic" methods were better understood in terms of *clone theory*. Thus analysed using methods from universal algebra, these problems were tackled by generalizations of the Gaussian elimination algorithm. The proofs of Bulatov and Zhuk extend the scope of these methods and establish their limits, yielding a complete classification of problems of the form $\mathsf{CSP}(\mathbb{D})$.

Years before the breakthrough results of Bulatov, Zhuk and Babai, it had been remarked that there was a striking resemblance between the tractability landscapes of the graph isomorphism problem and of the constraint satisfaction problem. For both problems, there is a standard logical or combinatorial algorithm that establishes tractability in a large number of special cases. This is the local consistency test for constraint satisfaction problems and the Weisfeiler–Leman test for graph isomorphism. In particular, they both are successful on graphs of bounded treewidth. Indeed, it is known that the constraint satisfaction problems solvable by local consistency methods are *exactly* those with bounded treewidth duality (Bulatov et al., 2008). In addition, for both problems, there are a number of sporadic tractable cases that are dealt with by algebraic methods. For example, isomorphism for bounded-degree graphs is solved by group-theoretic methods due to Luks (1982), and solving systems of linear equations over finite fields is a constraint satisfaction problem solved by Gaussian elimination. And then, there are apparently intractable cases: **NP**-complete in the case of constraint satisfaction and **GI**-complete in the case of graph isomorphism.

This analogy between the research landscapes on the two algorithmic problems becomes a bit more mathematically meaningful when we note that both the local consistency algorithm for constraint satisfaction and the Weisfeiler–Leman algorithm for graph isomorphism are parameterized by a positive integer k. Seeing k-local consistency as an approximation to homomorphism, one can define a natural notion of composition. This leads to the observation that the k-Weisfeiler–Leman test is looking exactly for an invertible k-local consistent map. This observation calls out for a category-theoretic treatment and it was the germ of the idea that led to development of the k-pebbling co-monad in Abramsky et al. (2017). It yields, among other insights, an elegant co-algebraic account of tree-width.

18.2 Finite Model Theory

Finite Model Theory is a term that has come to be used, since the 1980s, to describe a distinct subject of study concerned with the expressive power of logical languages when their interpretation is restricted to finite structures. While model theory has involved studying the expressive power of logic, it would be wrong to see finite model theory as merely the model theory of finite structures. It has evolved a distinctive set of questions, methods and results that are rather different from those that exercise the minds of model theorists. The subject has grown out of questions raised in theoretical computer science, particularly through its connections with computational complexity and database theory. One distinguishing feature is that first-order logic, which played an important role in the early development of model theory, has a somewhat lesser role in finite model theory.

In the following, we assume the reader is familiar with the basic notions of first-order logic (FO), as can be found, for example, in Enderton (1972). We work with finite *relational* vocabularies. That is, each vocabulary σ is a sequence of relation symbols each with an associated arity. A structure $\mathbb{A}$ is a *finite* set A along with an interpretation $R^{\mathbb{A}} \subseteq A^a$ for each relation symbol R in σ of arity a.

One reason why first-order logic is not central to finite model theory is that the relation of elementary equivalence between structures is trivial on finite structures. A large part of classical model theory can arguably be described as the study of this equivalence relation and the structure of its equivalence classes. Two structures $\mathbb{A}$ and $\mathbb{B}$ are elementarily equivalent if, for every first-order sentence φ,

$$\mathbb{A} \models \varphi \text{ if, and only if, } \mathbb{B} \models \varphi.$$

This is crucial in establishing inexpressibility results. For instance, by proving that all dense linear orders without endpoints are elementarily equivalent, we establish that other properties that might distinguish such orders (such as Dedekind completeness) are not first-order axiomatizable.

On finite structures, the elementary equivalence relation is trivial, in that any two elementarily equivalent structures are isomorphic. Indeed, any finite structure is described up to isomorphism by a single sentence. Given a structure $\mathbb{A} = (A, R_1, \ldots, R_m)$ in a vocabulary $\sigma = \{R_1, \ldots, R_m\}$, where A is a set of n elements, we can construct a sentence

$$\delta_{\mathbb{A}} = \exists x_1 \ldots \exists x_n \left(\psi \wedge \forall y \bigvee_{1 \leq i \leq n} y = x_i \right)$$

where, $\psi(x_1, \ldots, x_n)$ is the conjunction of all atomic and negated atomic formulas that hold in $\mathbb{A}$ under some fixed bijection between A and the variables $x_1, \ldots, x_n$. Now, for any structure $\mathbb{B}$, $\mathbb{B} \models \delta_{\mathbb{A}}$ if, and only if, $\mathbb{A} \cong \mathbb{B}$.

This means that first-order logic can make all the distinctions that are to be made between finite structures. Still, the expressive power of first-order logic on finite structures is weak: for any first-order sentence φ, $\mathrm{Mod}(\varphi)$ can be decided by a deterministic Turing machine with logarithmic work space, where $\mathrm{Mod}(\varphi)$ is the collection of finite models of φ. What accounts for this disparity?

Essentially, in the model theory of infinite structures, as we classify structures by elementary equivalence, we are looking at the expressive power of theories, i.e., possibly infinite sets of sentences. Two structures that are elementarily equivalent cannot be distinguished by any first-order theory. In contrast, any isomorphism closed class S of finite structures is defined by the set of negations of the sentences $\delta_{\mathbb{A}}$, as above, for finite structures $\mathbb{A}$ not in S. Certainly, this theory may have infinite models, but the collection of its finite models is exactly S. Thus, the expressive power of first-order theories is uninteresting. On the other hand, to establish inexpressibility results for first-order *sentences*, we need to consider weaker equivalence relations than elementary equivalence. These are usually obtained by stratifying the elementary equivalence relation according to parameters such as the quantifier depth or the number of variables in a formula.

A second reason why first-order logic does not hold a central place when considering finite structures is the fact, mentioned above, that the expressive power of single sentences of first-order logic is very weak. A large part of the motivation and the early impetus for the study of finite model theory came from its connections with computational complexity. There is a close connection between the *descriptive complexity* of a property of finite structures—i.e., the complexity of the logical constructs required to define the property—and its *computational complexity*, the resources required on a machine to decide the property. The paradigmatic result relating these two kinds of complexity is the theorem of Fagin (1974) which states that a property is definable in existential second-order logic if, and only if, it is decidable in NP. Much work in the subject has been generated by the question first posed by Chandra and Harel (1982) asking whether one can give a similar logical characterization for the complexity class P (see Gurevich, 1988 for a detailed discussion and a precise formulation of this question). The logics that have been studied in this context are intermediate in expressive power between first-order logic and second-order logic. Examples are *fixed-point logic* (FP) and *fixed-point logic with counting* (FPC). We briefly introduce these next. It should be noted that single sentences of these extended logics can always be seen as theories in first-order logic of a particular, uniform, sort. They can then be again analysed through stratifications of the notion of elementary equivalence.

18.2.1 Fixed-Point Logics

Let $\varphi(R, x_1, \ldots, x_l)$ be a first-order formula in the vocabulary $\sigma \cup \{R\}$, where R is a relation symbol of arity l and $x_1, \ldots, x_l$ are variables that appear free in φ. On a given σ-structure $\mathbb{A}$, this formula defines an operator $\Phi : \wp(A^l) \to \wp(A^l)$[1] that maps a relation $S \subset A^l$ interpreting the symbol R to the relation $\Phi(S) = \{\bar{a} \mid (\mathbb{A}, S, \bar{a}) \models \varphi(R, \bar{x})\} \subset A^l$. This allows us to define an *increasing* sequence of relations on $\mathbb{A}$:

- $\Phi^0 = \emptyset$
- $\Phi^{m+1} = \Phi^m \cup \Phi(\Phi^m)$

The *inflationary fixed point* of Φ is the limit of this sequence. It is easy to see that on a structure with n elements, the limit is reached after at most n^l stages.

The logic FP is obtained by closing first-order logic under the following formula-formation rule:

[1] Here, $\wp(X)$ denotes the powerset of X.

- If φ is a formula, R an l-ary relational variable, $\bar{x}$ an l-tuple of elementary variables, and $\bar{t}$ an l-tuple of terms, then $\mathbf{ifp}_{R,\bar{x}}\varphi(\bar{t})$ is also a formula, where all occurrences in φ of R and of variables in $\bar{x}$ are bound, while all occurrences in $\bar{t}$ of variables in $\bar{x}$ remain free.

The intended meaning of the formula is that the tuple denoted by $\bar{t}$ is in the relation that is the inflationary fixed point of the operator defined by $\varphi(R, \bar{x})$.

Example 18.2.1 The following formula:

$$\forall u \forall v[\mathbf{ifp}_{T,xy}(x = y \vee \exists z(E(x, z) \wedge T(z, y)))](u, v)$$

is satisfied in a graph (V, E) if, and only if, the graph is connected. Indeed, the least relation T that satisfies the equivalence $T(x, y) \equiv x = y \vee \exists z(E(x, z) \wedge T(z, y)))$ is the reflexive and transitive closure of E. This is, therefore the fixed point defined by the operator **ifp**. The formula can now be read as saying that every pair u, v is in this reflexive-transitive closure.

The importance that is attached to FP in the realm of finite model theory is explained in part by the Immerman-Vardi theorem. This result was established independently by Immerman (1986) and Vardi (1982) in 1982 (a similar result is shown by Livchak, 1983) and demonstrates a close relationship between inductive definitions and feasible computation. For this theorem, we consider classes of finite *ordered* structures, which are structures with a distinguished relation $<$ that is interpreted as a linear order of the universe.

Theorem 18.2.2 (Immerman-Vardi) *A class of finite ordered structures is definable by a sentence of* FP *if, and only if, membership in the class is decidable by a deterministic Turing machine in* polynomial time.

If we do not restrict ourselves to structures which interpret a linear order, it is still the case that every class of structures definable in FP is decidable in polynomial time. However, the converse fails, as there are properties that are easily computable (such as the property of a set having an even number of elements) that are not expressible in the logic (see Example 18.2.10 below).

Inflationary fixed-point logic with counting (FPC) extends FP with the ability to express the cardinality of definable sets. The logic is built up from the usual set of symbols, except that we have two sorts of variables: $v_1, v_2, \ldots$ ranging over the domain elements of the structure, and $\nu_1, \nu_2, \ldots$ ranging over the non-negative integers. In order to avoid undecidability, quantification of number variables has to be bounded by some definable number term, as we describe in more detail below. In addition, we also have second order variables $X_1, X_2, \ldots$, each of which has a type which is a finite string in {element, number}*. Thus, if X is a variable of type (element,number), it is to be interpreted by a binary relation relating elements to numbers. The logic allows us to build up *counting terms* according to the following rule:

if φ is a formula and x is a variable of the first sort, then $\#x\varphi$ is a term.

The intended semantics is that $\#x\varphi$ denotes the number of elements that satisfy the formula φ. The formulas of FPC are now described by the following set of rules:

- all atomic formulas of first-order logic are formulas of FPC;
- if τ_1 and τ_2 are terms of numeric sort (that is each one is either a number variable or a counting term) then each of $\tau_1 < \tau_2$ and $\tau_1 = \tau_2$ is a formula;
- if φ and ψ are formulas then so are $\varphi \wedge \psi$, $\varphi \vee \psi$ and $\neg\psi$;
- if φ is a formula, x is an element variable, ν is a number variable, and η is a term of numeric sort, then $\exists x\, \varphi$ and $\exists \nu \leq \eta\, \varphi$ are formulas; and
- if X is a relation symbol of type σ, $\mathbf{x}$ is a tuple of variables whose sorts match the type σ and $\mathbf{t}$ is a tuple of terms of type σ, then $[\mathbf{ifp}_{X,\mathbf{x},}\, \varphi](\mathbf{t})$ is a formula.

It can now be proved (see Otto, 1997) that any formula of FPC which has no free number variables can be interpreted in a unique way in a structure $\mathbb{A}$. Thus, a sentence of FPC does define a class of structures and we can legitimately ask what the complexity of this class is. In particular, this logic can define the class of structures that have an even number of elements, which makes it strictly more expressive than FP.

Example 18.2.3 The following sentence is satisfied in a structure $\mathbb{A}$ if, and only if, the number of elements of $\mathbb{A}$ that satisfy the formula $\varphi(x)$ is even.

$$\exists \nu_1 \leq [\#x\varphi] \exists \nu_2 \leq \nu_1 (\nu_1 = [\#x\varphi] \wedge (\nu_2 + \nu_2 = \nu_1))$$

In particular, taking φ to be a universally true formula such as $x = x$, we get a sentence that defines evenness. Here we have used the addition symbol in the subformula $\nu_2 + \nu_2 = \nu_1$. It should be noted that this denotes a relation that is easily definable by induction on the domain of numbers.

To analyze the expressive power of logics such as FP and FPC, it is useful to think of the individual sentences of these logics as defining first-order theories. As we noted, any isomorphism-closed class of finite structures is defined by a first-order theory. Those that are definable in FP or FPC are given by first-order theories of a particular regular form. This enables us to analyze them using restricted notions of elementary equivalence. We introduce these next.

18.2.2 Equivalences

The *quantifier rank* of a first-order formula φ is the maximal depth of nesting of quantifiers (irrespective of alternations) inside φ. Write $\mathrm{FO}[p]$ for those formulas of first-order logic with quantifier rank at most p. Also, for a pair of structures $\mathbb{A}$ and $\mathbb{B}$, we write $\mathbb{A} \equiv_{\mathrm{FO}[p]} \mathbb{B}$ to denote that a sentence $\varphi \in \mathrm{FO}[p]$ is true in $\mathbb{A}$ if, and only if, it is true in $\mathbb{B}$. It is not difficult to show that, on structures in a fixed finite relational vocabulary, the equivalence relation $\equiv_{\mathrm{FO}[p]}$ has finite index. It follows that a class of structures is definable in first-order logic if, and only if, it is invariant under $\equiv_{\mathrm{FO}[p]}$, for some p. This makes the family of equivalence relations $\equiv_{\mathrm{FO}[p]}$ central to analysing the expressive power of FO on finite structures (see Ebbinghaus & Flum, 1999, Chap. 2). For extensions of first-order logic, we look to other stratifications of the relation of elementary equivalence.

Write L^k for the collection of formulas of first-order logic that involve just the variables $x_1, \ldots, x_k$. Each variable may appear bound more than once in a formula φ but the total number of distinct variables is no more than k. A sentence φ of first-order logic is equivalent to one in L^k if no sub-formula of φ contains more than k free variables. We refer to the maximum number of free-variables in a sub-formula of φ as the *width* of φ. The interest in fragments of first-order logic of bounded width comes from the connection with fixed-point logic. To be precise, for a pair of structures $\mathbb{A}$ and $\mathbb{B}$, write $\mathbb{A} \equiv^{L^k} \mathbb{B}$ to denote that any sentence $\varphi \in L^k$ is true in $\mathbb{A}$ if, and only if, it is true in $\mathbb{B}$. Kolaitis and Vardi (1992) noted that the collection of finite models of any sentence of the fixed-point logic FP is invariant under the relation $\equiv^{L^k}$ for some fixed k. Indeed, we can say more: for every sentence φ of FP there is a k such that φ is equivalent (on finite structures) to the conjunction of a countable set of sentences of L^k (Dawar et al., 1995). In other words, its collection of finite models is defined by a first-order theory axiomatized by a set of L^k sentences. Thus, to prove that some property is not definable in FP, it suffices to show that it is not invariant under $\equiv^{L^k}$ for any fixed k.

The key tool in analysing the expressive power of fixed-point logic with counting FPC is the extension of first-order logic with *counting quantifiers*. For each natural number i, we have a quantifier $\exists^i$ where $\mathbb{A} \models \exists^i x\, \varphi$ if, and only if, there are at least i distinct elements $a \in A$ such that $\mathbb{A} \models \varphi[a/x]$. While the extension of first-order logic with counting quantifiers is no more expressive than FO itself (in contrast to the situation with counting terms used in the definition of FPC), the presence of these quantifiers does affect the number of variables that are necessary to express a property. Let C^k denote the k-variable fragment of first-order logic with counting quantifiers. That is, C^k consists of those formulas in which only the variables $x_1, \ldots, x_k$ appear, free or bound. For two structures $\mathbb{A}$ and $\mathbb{B}$, we write $\mathbb{A} \equiv^k \mathbb{B}$ to denote that the two structures are not distinguished by any sentence of C^k. The link between this and FPC is the following fact, established by Immerman and Lander (1990):

Theorem 18.2.4 *For every sentence φ of* FPC*, there is a k such that if $\mathbb{A} \equiv^k \mathbb{B}$, then $\mathbb{A} \models \varphi$ if, and only if, $\mathbb{B} \models \varphi$.*

Indeed, this theorem follows from the fact that for any formula φ of FPC, there is a k so that on structures with at most n elements, φ is equivalent to a formula θ_n of C^k. And then, again, it can be shown that any collection of finite structures defined by a sentence of FPC is axiomatized by a collection of sentences of C^k, for some fixed k. This makes the equivalence relations $\equiv^k$ objects of great interest in finite model theory. It turns out that they are essentially the same as a well-studied class of equivalence relations in the context of graph isomorphism, a topic we turn to in Sect. 18.3.

Example 18.2.5 In Example 18.2.3 we saw a formula of FPC, indeed one without fixed-point operators, that defined the class of structures $\mathcal{E}_\varphi$ in which the number of elements satisfying a formula $\varphi(x)$ is even. Indeed, $\mathcal{E}_\varphi$ is necessarily invariant under $\equiv^k$ where k is the number of variables in φ. For, suppose that $\mathbb{A}$ is a structure in which there are exactly i elements satisfying φ. Then, the C^k formula $\exists^i x\varphi(x) \wedge \neg\exists^{i+1} x\varphi(x)$ is true in $\mathbb{A}$. Hence, any structure $\equiv^k$-equivalent to $\mathbb{A}$ also has exactly i elements satisfying φ. In particular, taking φ to be the formula $x = x$, we see that the collection of structures of even cardinality is $\equiv^1$-closed.

On the other hand, there is no k such that the collection of structures of even cardinality is $\equiv^{L^k}$-closed. The translation of the formula $\exists^i x(x = x)$ into one without counting quantifiers requires at least i variables.

18.2.3 Pebble Games

As we have noted, a central concern in finite model theory is to establish limits on the expressive power of logics, often strengthenings of first-order logic, by studying *weakenings* of the relation of elementary equivalence (which is, in the finite, the same as isomorphism). An important methodology for doing so is based on *Spoiler-Duplicator games*. These are two-player games played on a pair of relational structures $\mathbb{A}$ and $\mathbb{B}$ of the same vocabulary between two players we call Spoiler and Duplicator. The former aims to show, using limited resources, that the two structures are different while the latter pretends they are the same.

Consider again two stratifications of the relation of elementary equivalence that we introduced in Sect. 18.2.2: $\equiv_{\mathrm{FO}[p]}$ and $\equiv^{L^k}$. These are usually characterized by two versions of games: the *Ehrenfeucht-Fraïssé* game and the *pebble* game respectively. We combine these here, for ease of presentation, into a double stratification of elementary equivalence. We write $\mathbb{A} \equiv_p^{L^k} \mathbb{B}$ to denote that $\mathbb{A}$ and $\mathbb{B}$ are not distinguished by any sentence of L^k with quantifier rank at most p.

The characterization of $\equiv_p^{L^k}$ in terms of Spoiler-Duplicator games is essentially given by Barwise (1977) though versions were also independently presented by Immerman (1986) and Poizat (1982). The game board consists of two structures $\mathbb{A}$ and $\mathbb{B}$ and k pairs of pebbles (α_i, β_i), $1 \le i \le k$. The pebbles $\alpha_1, \ldots, \alpha_l$ are initially placed on the elements of an l-tuple $\bar{s}$ of elements in $\mathbb{A}$, and the pebbles $\beta_1, \ldots, \beta_l$ on an l-tuple $\bar{t}$ in $\mathbb{B}$, $l \le k$. At each round of the game, Spoiler picks up a pebble (either an unused pebble or one that is already on the board) and places it on an element of the corresponding structure. For instance, it might take pebble β_i and place it on an element, say b_i of $\mathbb{B}$. Duplicator must respond by placing the matching pebble in the opposite structure. In the above example, it must place α_i on an element, say a_i, of $\mathbb{A}$. If at the end of the round the partial map $f : \mathbb{A} \to \mathbb{B}$ given by $a_i \mapsto b_i$ is not a partial isomorphism, then Spoiler has won the game; otherwise it can continue for another round.

The result that links this game with the equivalence relation $\equiv_p^{L^k}$ is the following:

Theorem 18.2.6 *Duplicator has a strategy for playing the k-pebble game for p rounds starting with the position* $(\mathbb{A}, \bar{s})$, $(\mathbb{B}, \bar{t})$ *if, and only if,* $(\mathbb{A}, \bar{s}) \equiv_p^{L^k} (\mathbb{B}, \bar{t})$.

We do not give a proof of this here as it can be found in standard reference works such as Ebbinghaus and Flum (1999). To relate the game to the two separate stratifications by quantifier rank and by number of variables considered above in Sect. 18.2.2, we note that if k exceeds p, then the restriction on the number of pebbles is vacuous. On the other hand, if p exceeds $|\mathbb{A}|^k \cdot |\mathbb{B}|^k$, then in a p-move game some game position must repeat. Hence, the existence of a Duplicator winning strategy for p moves implies that it has a strategy to play forever. We can summarise these observations in the following corollary.

Corollary 18.2.7 $\mathbb{A} \equiv_{\mathrm{FO}[p]} \mathbb{B}$ *if, and only if, Duplicator has a winning strategy in the p-pebble, p-move game played on* $\mathbb{A}, \mathbb{B}$. $\mathbb{A} \equiv^{L^k} \mathbb{B}$ *if, and only if, Duplicator has a winning strategy to play forever in the k-pebble game played on* $\mathbb{A}, \mathbb{B}$.

The first game characterizing the equivalence relations $\equiv^k$ was given by Immerman and Lander (1990). We are interested in an alternative game given by Hella (1996). As before the game is played on structures $\mathbb{A}$ and $\mathbb{B}$ by Spoiler and Duplicator using pebbles $\alpha_1, \dots, \alpha_k$ on $\mathbb{A}$ and $\beta_1, \dots, \beta_k$ on $\mathbb{B}$. If $|A| \neq |B|$, Spoiler wins the game immediately. Otherwise, each round starts by Spoiler picking a pair of corresponding pebbles, say α_i and β_i. Duplicator has to respond by choosing a bijection $h : A \to B$ which respects the partial map defined by the currently pebbled elements *excluding* the pebbles picked by Spoiler. In the above example, the mapping h given by Duplicator must satisfy $h(a_j) = b_j$ for all pebbles α_j and β_j $(j \neq i)$, where a_j denotes the element of A on which α_j is placed and b_j the element of B on which β_j is placed. Spoiler then chooses $a \in A$ and places α_i on a and β_i on $h(a)$. This completes one round in the game. If, after this round, the partial map from $\mathbb{A}$ to $\mathbb{B}$ defined by the pebbled positions is not a partial isomorphism, then Spoiler has won the game. Otherwise it can continue for another round. Again, writing $\mathbb{A} \equiv_p^k \mathbb{B}$ to denote that $\mathbb{A}$ and $\mathbb{B}$ are not distinguished by any formula of C^k of quantifier rank at most p, we have:

Theorem 18.2.8 *Duplicator has a strategy for playing the k-pebble bijection game for p rounds starting with the position* $(\mathbb{A}, \bar{s})$, $(\mathbb{B}, \bar{t})$ *if, and only if,* $(\mathbb{A}, \bar{s}) \equiv_p^k (\mathbb{B}, \bar{t})$.

By the same arguments as before, we can extract from this a characterization of the equivalence relation $\equiv^k$.

Corollary 18.2.9 $\mathbb{A} \equiv^k \mathbb{B}$ *if, and only if, Duplicator has a winning strategy to play forever in the k-pebble bijection game played on* $\mathbb{A}, \mathbb{B}$.

Example 18.2.10 In Example 18.2.1 we saw that the class of connected graphs is definable in FP. We can prove that this class is not definable in first-order logic by presenting, for each positive integer p a pair of graphs G_p and H_p such that G_p is connected and H_p is not, but $G_p \equiv_{\mathrm{FO}[p]} H_p$. Taking G_p to be a simple cycle of length greater than 2^p and H_p to be the union of two disjoint cycles, each of length greater than 2^p suffices. The details of the Duplicator winning strategy are left as an exercise.

The property of a structure having an even number of elements is not even definable in FP. This can be proved by exhibiting, for each positive integer k, a pair of structures $\mathbb{A}_k$ and $\mathbb{B}_k$ such that $\mathbb{A}_k$ has an even number of elements, $\mathbb{B}_k$ has an odd number of elements and yet $\mathbb{A}_k \equiv^{L^k} \mathbb{B}_k$. Letting both structures be complete graphs on more than k elements suffices for this purpose.

18.2.4 One-Sided Games

In proving one direction of Theorem 18.2.6, one shows that any sentence of L^k with quantifier depth p which distinguishes $\mathbb{A}$ from $\mathbb{B}$ yields a winning strategy for Spoiler in the k-pebble p-round game played on this pair of structures. We construct the strategy by induction on the sentence. Inductively, we have a formula $\theta(\bar{x})$ with free variables $\bar{x}$ and tuples $\bar{a}$ and $\bar{b}$ in the two structures $\mathbb{A}$ and $\mathbb{B}$ (corresponding to the elements pebbled so far) such that $\mathbb{A} \models \theta[\bar{a}]$ and $\mathbb{B} \not\models \theta[\bar{b}]$. Assume θ begins with a quantifier (otherwise it is a Boolean combination of formulas and we can choose one of them that has this property). If it is an existential quantifier, Spoiler's strategy is to choose a witness for it in $\mathbb{A}$ and if it a universal quantifier then choose a counter-example for it in $\mathbb{B}$. If the variable that is quantified is x_i, it is pebble i that Spoiler moves.

It is worth noting that in the bijection game, Spoiler does not need to choose which structure to play in. This is because the bijection that Duplicator is required to play essentially gives a response to any element that Spoiler might play in either of the two structures. In logical terms, this is reflected in the fact that when we have counting quantifiers, we do not need to have separate existential and universal quantifiers, as they are subsumed as special cases.

The observation that a Spoiler strategy on a pair of structures $(\mathbb{A}, \mathbb{B})$ essentially plays existential quantifiers on the left-hand side and universal quantifiers on the right-hand side allows us to define games corresponding to restricted logics. Assume we have a formula of L^k in negation normal form (i.e., with negations only in front of atomic formulas). We say such a formula is *existential* if it contains no universal quantifiers. Write $\mathbb{A} \Rrightarrow_p^{\exists L^k} \mathbb{B}$ to indicate that every existential sentence of L^k of quantifier rank p true in $\mathbb{A}$ is also true in $\mathbb{B}$. This relation is characterized by the k-pebble p-round game in which Spoiler is confined to play pebbles on elements of $\mathbb{A}$.

A final variation we are interested in is *existential positive* logic. A formula of L^k is existential positive if it is existential and contains no negations. We write $\exists^+ L^k$

for the collection of existential positive formulas of L^k and $\mathbb{A} \Rrightarrow_p^{\exists^+ L^k} \mathbb{B}$ to indicate that every sentence of $\exists^+ L^k$ of quantifier rank p true in $\mathbb{A}$ is also true in $\mathbb{B}$. This is characterized by a k-pebble p-round game in which Spoiler always plays on $\mathbb{A}$ and we also relax the winning condition for Duplicator. Instead of being required to ensure that the partial map $f : \mathbb{A} \to \mathbb{B}$ given by $a_i \mapsto b_i$ is a partial isomorphism, Duplicator wins as long as it is a partial *homomorphism*. In other words, Duplicator is not required to ensure that if $a_i \neq a_j$ then $b_i \neq b_j$ and that if $R(b_{i_1}, \ldots, b_{i_r})$ for some r-ary relation R then $R(a_{i_1}, \ldots, a_{i_r})$. Once again, by requiring Duplicator to win an infinite game, we characterize the relation $\mathbb{A} \Rrightarrow^{\exists^+ L^k} \mathbb{B}$ with no bound on the quantifier rank. This existential positive k-pebble game is an important starting point of our exploration. It was introduced by Kolaitis and Vardi (1995).

Example 18.2.11 If K_2 denotes the graph consisting of two vertices and an edge between them, then we have for any graph G, $G \longrightarrow K_2$ if, and only if, G is 2-colourable. We can also show that $G \longrightarrow K_2$ if, and only if, $G \Rrightarrow^{\exists^+ L^3} K_2$. To see this, it suffices to show that if G is not 2-colourable, then Spoiler has a winning strategy in the 3-pebble one-sided positive game played on G and K_2. Spoiler's strategy is to place one pebble on a vertex in an odd cycle of G and use the other two pebbles to "walk" around the cycle.

On the other hand, if K_3 is the 3-vertex triangle, then we have for any graph G, $G \longrightarrow K_3$ if, and only if, G is 3-colourable. In this case, we can find for every k, a graph G which is *not* 3-colourable but such that $G \Rrightarrow^{\exists^+ L^k} K_3$. For example, consider T_n, the triangulated $n \times n$ grid. This is a graph on the vertex set $\{v_{i,j} \mid 0 \leq i, j, \leq n-1\}$ with an edge between $v_{i,j}$ and $v_{i',j'}$ whenever: $i = i'$ and $j' = j+1$; $j = j'$ and $i' = i+1$; or $i' = i+1$ and $j' = j+1$. All additions here are modulo n. It is easily seen T_n is 3-colourable if, and only if, n is a multiple of 3. Yet, for all n sufficiently larger than k we have $T_n \Rrightarrow^{\exists^+ L^k} K_3$.

18.2.5 Infinitary Logics

The pebble games introduced in Sects. 18.2.3 and 18.2.4 above are often formulated for infinitary logics. I give a brief account of the connection here, though in general in this paper we eschew the formalism of infinitary logics and refer instead to first-order theories. The logics $L^k_{\infty\omega}$, $C^k_{\infty\omega}$ and $\exists^+ L^k_{\infty\omega}$ are the closures under infinitary conjunctions and disjunctions of the logics L^k, C^k and $\exists^+ L^k$ respectively. Also, we write $L^\omega_{\infty\omega}$, $C^\omega_{\infty\omega}$ and $\exists^+ L^\omega_{\infty\omega}$ for the infinitary logics that are obtained by taking the unions of $L^k_{\infty\omega}$, $C^k_{\infty\omega}$ and $\exists^+ L^k_{\infty\omega}$ respectively, for all k.

The games we introduced are easily modified (and this is the usual presentation) so that Duplicator is required to play forever without losing. Now, Duplicator has a winning strategy on two structures $\mathbb{A}$ and $\mathbb{B}$ in the k-pebble existential game (for instance) if, and only if, any formula of $\exists^+ L^k_{\infty\omega}$ true in $\mathbb{A}$ is true in $\mathbb{B}$. Similarly, this winning condition applied to the k-pebble game and the k-pebble bijection game characterize equivalence in the logics $L^k_{\infty\omega}$ and $C^k_{\infty\omega}$, respectively.

When $\mathbb{A}$ and $\mathbb{B}$ are *finite* structures, then Duplicator has a strategy to play forever in one of these games as soon as it has a winning strategy for p rounds of the game for sufficiently large p. Indeed, if p is larger than the number of positions in the game on $(\mathbb{A}, \mathbb{B})$ then in any play of a p-round game, some position must be repeated. Thus, if Duplicator can survive p rounds of the game, it can play forever. We can conclude from this that if every formula of one of the logics $\exists^+ L^k$, L^k or C^k that is true in $\mathbb{A}$ is true in $\mathbb{B}$, then the same is true for the corresponding infinitary logic. In short, the infinitary logics do not have any greater *distinguishing power* on finite structures than their finitary counterparts, though they do have much greater *expressive power*. So, the finite models of any sentence of $L^k_{\infty\omega}$ must be invariant under the equivalence relation $\equiv^{L^k}$, however the models of a sentence of L^k must actually be invariant under $\equiv^{L^k}_p$ for some fixed p.

The restriction to finite structures is essential for this argument. Indeed, on infinite structures, it is not the case that any sentence of $L^k_{\infty\omega}$ is invariant under $\equiv^{L^k}$. Instead, we can perform a more refined analysis where the games may have transfinite length. The details are not relevant here and can be found in Dawar (1993). From the point of view of finite model theory, the main interest in the infinitary logics is that their expressive power subsumes the fixed-point extensions of first-order logic that are of great interest. In particular, $L^k_{\infty\omega}$ subsumes fixed-point logic (FP), $C^k_{\infty\omega}$ subsumes fixed-point logic with counting (FPC) and $\exists^+ L^k_{\infty\omega}$ subsumes Datalog (again the restriction to finite structures, or at least to structures of some bounded cardinality, is essential for all of these). This is useful for proving inexpressibility results for the fixed-point logic. To show that some class of structures is undefinable in one of these fixed-point logics it is sufficient to show that it is not definable in the corresponding infinitary logic. This is the same as showing the class is not closed under the corresponding relation: $\equiv^{L^k}$, $\equiv^k$ or $\Rrightarrow^{\exists^+ L^k}$. We stick to the latter formulation and dispense with dealing with infinitary logics directly.

Before we move on from infinitary logics, one further remark is in order. It is known that for any finite structure $\mathbb{A}$ we can construct a sentence φ of L^k such that $\mathbb{B} \models \varphi$ if, and only if, $\mathbb{A} \equiv^{L^k} \mathbb{B}$ (Dawar et al., 1995). In other words, the $\equiv^{L^k}$ class of $\mathbb{A}$ is characterized by a single finitary sentence. The analogous fact also holds for $\equiv^k$ and the logic C^k (Otto, 1997). However, it is not the case for the existential positive fragment. In this case, for any finite $\mathbb{A}$, we can construct a φ in $\exists^+ L^k_{\infty\omega}$ such that $\mathbb{B} \models \varphi$ if, and only if, $\mathbb{A} \Rrightarrow^{\exists^+ L^k} \mathbb{B}$, but this sentence cannot necessarily be made finitary (see Atserias et al., 2006, Proposition 7.9).

18.2.6 Datalog

`Datalog` is a database query language which can be seen as an extension of existential positive first-order logic with a recursion mechanism. Equivalently, it can be seen as the existential positive fragment of the logic of least fixed points `LFP` (see Libkin, 2004, Chap. 10).

A *Datalog program* is a finite set of rules of the form $T_0 \leftarrow T_1, \ldots, T_m$, where each T_i is an atomic formula. T_0 is called the *head* of the rule, while the right-hand side is called the *body*. These atomic formulas use relational symbols from a vocabulary $\sigma \cup \tau$, where the symbols in σ are called *extensional* predicates and those in τ are *intensional* predicates. Every symbol that occurs in the head of a rule is an intensional predicate, while both intensional and extensional predicates can occur in the body of a rule. The semantics of such a program is defined with respect to a σ-structure $\mathbb{A}$. Say that a rule $T_0 \leftarrow T_1, \ldots, T_m$ is satisfied in a $\sigma \cup \tau$ expansion $\mathbb{A}'$ of $\mathbb{A}$ if $\mathbb{A}' \models \forall \bar{x}(\bigwedge_{1 \leq i \leq m} T_i \rightarrow T_0)$, where $\bar{x}$ enumerates all the variables occurring in the rule. The interpretation of a `Datalog` program in $\mathbb{A}$ is the smallest expansion of $\mathbb{A}$ (when ordered by pointwise inclusion of the relations interpreting τ) satisfying all the rules in the program. This is uniquely defined as it is obtained as the simultaneous least fixed-point of the existential closure of the right-hand side of the rules. We distinguish one intensional predicate G and call it the *goal predicate*. Then, the query computed by a program π is the interpretation of G in the interpretation of π in $\mathbb{A}$. In particular, if G is a 0-ary predicate symbol (i.e., a Boolean variable), π defines a Boolean query, i.e., a class of structures. In this case, we write $\mathbb{A} \models \pi$ to indicate that $\mathbb{A}$ is in the class of structures defined by π.

Every query definable in Datalog is definable in FP. This is simply because the query can be defined by taking the simultaneous fixed-point of the existential closure of the right-hand side of the rules and such fixed-points are easily expressed using the fixed-point operator in FP (see Ebbinghaus & Flum, 1999). Indeed, Datalog can be understood as the existential positive fragment of FP. Just as with the existential positive fragment of first-order logic, every query expressible in Datalog is closed under homomorphisms.

Kolaitis and Vardi (1995) showed that the any query π definable in Datalog can be expressed in $\exists^+ L^k_{\infty\omega}$ for some fixed k. In other words, there is a k such if $\mathbb{A} \models \pi$ and $\mathbb{A} \Rrightarrow^{\exists^+ L^k} \mathbb{B}$, then $\mathbb{B} \models \pi$, and this proves a very useful tool for analysing its expressive power.

Example 18.2.12 In Example 18.2.1, we showed that connectedness of graphs is definable in FP. This property is not closed under homomorphisms so cannot be definable in `Datalog`. However, we can define the reflexive and transitive closure of a binary relation E with the following `Datalog` program.

$$
\begin{aligned}
T(x, x) &\leftarrow \\
T(x, y) &\leftarrow E(x, z), T(z, y).
\end{aligned}
$$

The collection of graphs that are *not* 2-colourable is defined by the following `Datalog` program, with goal predicate Odd-Cycle which detects the presence of an odd length cycle.

$$
\begin{aligned}
&\text{Odd-path}(x, y) &&\leftarrow E(x, y)\\
&\text{Odd-path}(x, y) &&\leftarrow E(x, w), E(w, z), \text{Odd-path}(z, y)\\
&\text{Odd-cycle} &&\leftarrow \text{Odd-path}(x, x).
\end{aligned}
$$

On the other hand, the collection of graphs that are not 3-colourable is not definable in `Datalog` since it is not definable in $\exists^+ L^k_{\infty\omega}$ for any fixed k, as noted in Example 18.2.11.

18.3 Graph Isomorphism

The graph isomorphism problem is that of determining whether two graphs G and H are really the same, in the sense that there is a bijection ι from the vertices of G to the vertices of H such that (u, v) is an edge of G if, and only if, $(\iota(u), \iota(v))$ is an edge of H. More generally, we might be interested in testing two relational structures of the same vocabulary σ for isomorphism. The isomorphism problem for σ-structures reduces easily (in the sense of computational complexity) to graph isomorphism. As the graph isomorphism problem is very well studied in the literature, in this brief overview we restrict our attention to graphs, often with additional colours on the vertices. Everything generalizes quite naturally to finite relational structures of a fixed vocabulary σ.

Computationally, the graph isomorphism problem is equivalent to the problem of determining the orbits of the automorphism group of a graph G. That is, given a graph G, we wish to partition its vertices into the minimum number of classes so that for any pair of vertices u, v in the same class there is an automorphism of G that takes u to v. More generally, we may wish to partition the k-tuples (for some k) of vertices of G into equivalence classes based on the automorphisms. These problems are all equivalent (as long as k is bounded by a constant), in the sense that there are easy polynomial-time reductions between them. None of them is known to be solvable in polynomial-time nor known to be **NP**-complete. By Babai's groundbreaking result (Babai, 2016), we know that they can be solved in quasi-polynomial time, i.e., by an algorithm running in time $\exp(O(\log n)^c)$ for some constant $c > 1$.

Many algorithmic approaches to the graph isomorphism problem rely on approximating the automorphism partition. That is to say, we aim at partitioning the vertices (or tuples of vertices) of G into a partition that may be coarser than the partition into automorphism classes. Such approximations also yield, by standard reductions, approximations of the isomorphism relation.

18.3.1 Weisfeiler–Leman Refinement

Consider undirected, loopless, simple graphs. That is, a graph G is a set of vertices $V(G)$ along with a set of edges $E(G)$ where each edge $e \in E(G)$ is a two-element set $e = \{u, v\} \subseteq V(G)$. Where G is clear from context, we simply write V and E for the vertex and edge set respectively. For a set C, a C-coloured graph is a a graph G together with a function $\chi : V \to C$. For a C-coloured graph (G, χ), we refer to the partition of V given by $\{\{v \mid \chi(v) = c\} \mid c \in C\}$ as the χ-partition of V.

A very commonly used procedure for testing isomorphism is *vertex refinement*. This is an algorithm that produces, given a C-coloured graph (G, χ), the coarsest partition $\mathcal{P}$ of V refining the χ-partition such that if $u, v \in P \in \mathcal{P}$, then for every $Q \in \mathcal{P}$, u and v have the same number of neighbours in Q. Note that the partition of V produced by the vertex refinement procedure is no finer than the orbit partition. For many graphs G it is, in fact, the orbit partition but for others, for example regular graphs with no colouring, it can be properly coarser (see Babai et al., 1980). In practice, this provides a very efficient test for isomorphism that works on almost all graphs.

Weisfeiler and Leman (1976) defined a stronger refinement procedure. Their definition gives the coarsest *coherent partition* of V^2. The notion of coherence is given in terms of an algebraic condition. It was shown through the work of Babai, Mathon and others that this can be interpreted as a combinatorial procedure and generalized to give a partition of V^k satisfying a certain stability condition (see Cai et al., 1992 for a brief history of this method). This has come to be known as the *k-dimensional Weisfeiler–Leman* method and we define it formally next.

For each integer $k \geq 2$, the k-dimensional Weisfeiler–Leman algorithm gives a partition of V^k that is the coarsest partition $\mathcal{P}$ satisfying the following stability condition: if $\bar{u}, \bar{v} \in V^k$ are tuples in the same part of $\mathcal{P}$ and $(P_1, \ldots, P_k)$ is a k-tuple of parts of $\mathcal{P}$, then the order-preserving map from $\bar{u}$ to $\bar{v}$ is a partial isomorphism from G to itself and $|\{x \mid \bar{u}[x/i] \in P_i \text{ for } 1 \leq i \leq k\}| = |\{x \mid \bar{v}[x/i] \in P_i \text{ for } 1 \leq i \leq k\}|$. Here, $\bar{u}[x/i]$ denotes the tuple obtained by substituting x for the ith element of $\bar{u}$ and $|S|$ denotes the cardinality of a set S. In other words, we can get the partition $\mathcal{P}$ by the following iterated process. Define the equivalence relation $\sim_0$ on V^k by saying that two tuples $\bar{u}, \bar{v} \in V^k$ are equivalent if the order-preserving map from $\bar{u}$ to $\bar{v}$ is a partial isomorphism from G to itself. Say we have defined the equivalence relation $\sim_i$ and for each $\bar{u} \in V^k$ and each k-tuple $p = (P_1, \ldots, P_k)$ of equivalence classes of $\sim_i$, let $t_p^{\bar{u}}$ denote the number of vertices x such that for all i the tuple we get by substituting x in the ith position of $\bar{u}$ is in P_i. Then, let $\bar{u} \sim_{i+1} \bar{v}$ if, and only if, $\bar{u} \sim_i \bar{v}$ *and* $t_p^{\bar{u}} = t_p^{\bar{v}}$. The relation $\sim_{i+1}$ is a refinement (at least as fine) of $\sim_i$. This iterative process must converge to a stable partition $\mathcal{P}$ and that is the partition obtained by the k-dimensional Weisfeiler–Leman algorithm. This partition is again no finer than the partition of V^k into orbits under the action of the automorphism group of G, since the orbit partition clearly satisfies the stability condition. Also, for sufficiently large values of k, in particular for $k \geq n-1$, it *is* the orbit partition.

The k-dimensional Weisfeiler–Leman algorithm gives rise to an equivalence relation on graphs whereby two graphs G and H are equivalent if they are not distinguished by the corresponding isomorphism test. We denote this by $G \equiv_{\mathrm{WL}}^k H$. It is not hard to show that this relation can be decided in time $O(n^{ck})$ for some constant c, with n being the maximum of the sizes of the graphs G and H. It is also easy to see that as we let k grow, the equivalences cannot get coarser. That is to say, for all k, $G \equiv_{\mathrm{WL}}^{k+1} H$ implies $G \equiv_{\mathrm{WL}}^k H$. If there were a constant k such that $\equiv_{\mathrm{WL}}^k$ were the same as isomorphism on all finite graphs, we would have a polynomial-time test for graph isomorphism. It was this question that was answered by a now-famous construction due to Cai et al. (1992). They construct a sequence of pairs of graphs (G_k, H_k), one for each $k \in \mathbb{N}$ such that:

- for each k, $G_k \equiv^k_{\mathrm{WL}} H_k$;
- for each k, $G_k \not\cong H_k$, i.e., the graphs are not isomorphic; and
- G_k and H_k have $O(k)$ vertices.

It is a consequence that not only does a constant value of k not suffice to distinguish all graphs, but any value of k that is $o(n)$, where n is the number of vertices is not sufficient.

18.3.2 Logical and Other Characterizations

The Weisfeiler–Leman test for isomorphism is connected with logic in the work of Immerman and Lander (1990) who show, essentially, that the k-dimensional Weisfeiler–Leman equivalence is the same as indistinguishability in the logic C^{k+1}. In our notation, $\equiv^k_{\mathrm{WL}}$ and $\equiv^{k+1}$ are the same equivalence relation on graphs. This also provides a natural way to extend the Weisfeiler–Leman test beyond graphs to general relational structures.

We thus have three strikingly different characterizations of the family of equivalence relations $\equiv^k$: the combinatorial one in terms of stable partitions; the logical one in terms of definability in C^k; and the characterization through bijection games. Over the years a number of other characterizations have been added. We noted above that the original work of Weisfeiler and Leman defined what we now call the 2-dimensional Weisfeiler–Leman test in terms of coherent configurations. This can be also be generalized to higher dimensions giving a purely algebraic view of the family of relations (see Holm, 2010; Evdokimov & Ponomarenko, 1999). It also turns out that the equivalence relations can be characterized in terms of linear programming relaxations of the graph isomorphism problem (Grohe & Otto, 2015). Indeed, the equivalence relation given by the vertex refinement procedure which, in retrospect, is also called the 1-dimensional Weisfeiler–Leman equivalence, is the same as the relation of *fractional isomorphism* (Ramana et al., 1994) which is the natural relaxation of an integer program corresponding to graph isomorphism. The higher dimensions of Weisfeiler–Leman equivalence are then obtained by considering suitable lift-and-project hierarchies over this.

The variety of independently arising characterizations suggests that we have hit upon a natural and robust notion of equivalence. Moreover, it gives rise to a natural measure of complexity of classes of structures that we call *counting width*.

Definition 18.3.1 For any isomorphism-closed class of finite structures $\mathcal{C}$, let $\mathcal{C}_n$ denote the collection of structures in $\mathcal{C}$ with at most n elements. We write $\nu_{\mathcal{C}} : \mathbb{N} \to \mathbb{N}$ for the function such that $\nu_{\mathcal{C}}(n)$ is the least k for which $\mathcal{C}_n$ is closed under $\equiv^k$. We call $\nu_{\mathcal{C}}$ the *counting width* of $\mathcal{C}$.

It is clear that $\nu_{\mathcal{C}} = O(n)$ for any class $\mathcal{C}$. Moreover, for any class $\mathcal{C}$ that is $\equiv^k$-closed for a fixed k, in particular any class definable in FPC, we have $\nu_{\mathcal{C}} = O(1)$. The construction of Cai et al. (1992) yields a class of graphs $\mathcal{C}$ with $\nu_{\mathcal{C}} = \Omega(n)$. Since then, many other natural problems have been shown to have unbounded counting width. These include **NP**-complete problems such as graph 3-colourability and Hamiltonicity but also natural problems in **P** such as that of deciding if a system of equations over a finite field is solvable.

18.3.3 Isomorphism on Restricted Classes

The question of whether there is a general polynomial-time algorithm for graph isomorphism remains open. A sub-exponential algorithm for it was first established in 1983 (Babai & Luks, 1983) and this was improved to a quasi-polynomial time algorithm in 2016 (Babai, 2016). In the intervening years, while there was little progress on the general case, it was shown that on many restricted classes of graphs, the problem does admit polynomial-time solutions. These include classes of graphs:

- of bounded degree (Luks, 1982);
- of bounded colour-class size (Furst et al., 1980);
- of bounded eigenvalue multiplicity (Babai et al., 1982);
- of bounded tree-width (Bodlaender, 1990);
- of bounded genus (Filotti & Maye, 1980);
- which exclude a fixed graph as minor (Ponomarenko, 1988); and
- which exclude a fixed graph as a topological minor (Grohe & Marx, 2015).

Many of the algorithms that established these cases are broadly classified as *group-theoretic*. That is to say, they aim to identify the automorphism group of the graph by a suitable decomposition. The structural restrictions placed on the graph by assumption are shown to make the decomposition tractable. This group-theoretic method was pioneered in the work of Luks in the early eighties.

Another line of work has sought to show that on specific classes of graphs, the Weisfeiler–Leman algorithms of fixed dimension are sufficient to establish the polynomial-time decidability of graph isomorphism. Immerman and Lander (1990) already showed that $\equiv^2$ is the same as isomorphism on trees. The work of Grohe and Mariño (1999) established that for each w there is a k such on the class of graphs of treewidth at most w, $\equiv^k$ coincides with isomorphism, and Grohe strengthened this result by showing the same for classes of graphs that exclude a w-clique as a minor (Grohe, 2017). It's worth noting that in each of these cases, it was already known that the isomorphism problem is polynomial-time computable by other means. What the Weisfeiler–Leman algorithms provide is a systematic and unifying account.

On the other hand, there are classes of graphs with tractable isomorphism problem where we know that the Weisfeiler–Leman algorithms fail. In particular the graphs constructed by Cai, Fürer and Immerman which show that $\equiv^k$ does not capture isomorphism for any fixed k can actually be chosen to be of degree 3 and colour-class size 4. Thus, for the first two examples on our list above, the combinatorial Wesfeiler-Leman algorithm is insufficient and it seems that the group-theoretic techniques are essential.

The current frontier of graph isomorphism algorithms combines the algebraic and the combinatorial techniques. Thus, Babai's celebrated quasi-polynomial time algorithm for the general problem builds on Luks' group-theoretic method and also combines it with a Weisfeiler–Leman refinement of poly-logarithmic dimension. The result of Grohe and Marx (2015) for classes of graphs excluding a fixed topological minor also combines the two kinds of methods. This is the last result on the list above, and subsumes others above it.

18.4 Constraint Satisfication

Constraint Satisfaction Problems (CSP) is a general term used to describe algorithmic problems where we have a collection of variables V, which can take values over a domain D subject to some constraints. A constraint is specified by saying that the tuple of values taken by a given tuple $\bar{v}$ of variables must lie in some relation R on D. The problem then is to determine whether one can assign a value from D to each variable in V in a way that all constraints are satisfied. This is a generalization of problems such as solving systems of equations. Many natural algorithmic problems can be formulated as constraint satisfaction problems. These include well-known **NP**-complete problems such as Boolean satisfiability and graph 3-colourability and a wide variety of others. Despite the richness of the framework, it has been noted that in terms of computational complexity, all known constraint satisfaction problems over finite domains fall into one of two classes. They are either polynomial-time solvable or **NP**-complete. This led Feder and Vardi (1998) to conjecture that these were the only two possibilities. This famous *dichotomy conjecture* spurred a large research effort over two decades (see Bulatov et al., 2009; Krokhin & Zivný, 2017). The conjecture was finally confirmed independently in the work of Bulatov (2017) and Zhuk (2017).

We obtain a specific *constraint language* by fixing a finite relational structure $\mathbb{D}$: the domain D along with a collection of relations $R_1, \ldots, R_m$ on it. An instance to be solved is then specified by a similar relational structure $\mathbb{V}$: the set V of variables and for each relation R_i, a set of tuples from V whose interpretation must be in the relation. A solution is just a homomorphism from $\mathbb{V}$ to $\mathbb{D}$. For a fixed $\mathbb{D}$, we write $\mathsf{CSP}(\mathbb{D})$ to denote the class of structures $\mathbb{V}$ which map homomorphically to $\mathbb{D}$. The dichotomy theorem of Bulatov and Zhuk then says that for every $\mathbb{D}$, either $\mathsf{CSP}(\mathbb{D})$ is in **P** or it is **NP**-complete.

It turns out that the computational complexity of $\mathsf{CSP}(\mathbb{D})$ is completely determined by the algebraic structure of the so-called *clone of polymorphisms* of $\mathbb{D}$ (Bulatov et al., 2005). A polymorphism of $\mathbb{D}$ is a homomorphism from $\mathbb{D}^m$ to $\mathbb{D}$ for some power m. The collection of all polymorphisms form a clone over the finite universe D and the equational theory of this clone completely determines the complexity of $\mathbb{D}$. The Bulatov-Zhuk theorem tells us that essentially $\mathsf{CSP}(\mathbb{D})$ is **NP**-complete if, and only if, the clone is generated by projections alone.

However, it is not just the complexity, but also the *definability* of the class of structures $\mathsf{CSP}(\mathbb{D})$ that can be classified by its polymorphisms. Some CSPs can be solved by means of a simple algorithm, known as *local consistency* (Barto & Kozik, 2014). This is parameterized by a natural number k. It aims at checking, for a fixed set of k variables, whether there is way of assigning values to them while satisfying all constraints involving them. These are then propagated to other variables, but only keeping track of k at a time. If k is large enough (e.g. bigger than the number of variables) this decides satisfiability of the instance. But, for some particular structures $\mathbb{D}$, there is a fixed value of k, independent of the instance, such that k-local consistency suffices to determine membership in $\mathsf{CSP}(\mathbb{D})$. These constraint satisfaction problems are said to have *bounded width*.

18.4.1 Bounded Width CSP

A CSP of bounded width is necessarily tractable, but not all tractable CSPs have bounded width. This distinction was already noted in the seminal work of Feder and Vardi (1998). They classify efficient methods for solving CSPs into those that are expressible in Datalog and those using *group-theoretic* methods. A canonical example of the latter being the problem of solving a system of equations over the two-element field.

Since $\mathsf{CSP}(\mathbb{D})$ is defined as the class of structures that map homomorphically to $\mathbb{D}$, it follows that the complement of this class is closed under homomorphisms: if $\mathbb{A} \notin \mathsf{CSP}(\mathbb{D})$ and $\mathbb{A} \longrightarrow \mathbb{B}$ then $\mathbb{B} \notin \mathsf{CSP}(\mathbb{D})$. We write $\overline{\mathsf{CSP}(\mathbb{D})}$ to denote the complement of $\mathsf{CSP}(\mathbb{D})$. When we speak of the definability of a constraint satisfaction problem in Datalog, we mean that $\overline{\mathsf{CSP}(\mathbb{D})}$ is defined by a Datalog query. Similarly, since $\overline{\mathsf{CSP}(\mathbb{D})}$ is closed under homomorphisms and $\mathbb{A} \longrightarrow \mathbb{B}$ implies that $\mathbb{A} \Rrightarrow^{\exists^+ L^k} \mathbb{B}$ for all values of k, it is reasonable to ask for which D is it the case that $\overline{\mathsf{CSP}(\mathbb{D})}$ is closed under $\Rrightarrow^{\exists^+ L^k}$ for some k. The answer is that it is exactly those constraint satisfaction problems of bounded width (see Kolaitis & Vardi, 2008):

Theorem 18.4.1 *The following are equivalent for any finite relational structure* $\mathbb{D}$*:*

- $\overline{\mathsf{CSP}(\mathbb{D})}$ *is of bounded width;*
- $\overline{\mathsf{CSP}(\mathbb{D})}$ *is definable in* Datalog*; and*
- $\overline{\mathsf{CSP}(\mathbb{D})}$ *is closed under* $\Rrightarrow^{\exists^+ L^k}$ *for some* k*.*

Whether or not $\mathsf{CSP}(\mathbb{D})$ is of bounded width is completely determined by the algebraic variety of the clone of polymorphisms of $\mathbb{D}$. In particular, Barto and Kozik

(2014) show that $\mathsf{CSP}(\mathbb{D})$ is of bounded width if, and only if, this clone contains *weak near unanimity* polymorphisms of all but finitely many arities. Combining this with a result of Atserias et al. (2009) we get the following further characterization of the bounded width constraint satisfaction problems, formulated in Dawar and Wang (2015).

Theorem 18.4.2 *The following are equivalent for any finite relational structure* $\mathbb{D}$*:*

- $\mathsf{CSP}(\mathbb{D})$ *is of bounded width; and*
- $\mathsf{CSP}(\mathbb{D})$ *is closed under* $\equiv^k$ *for some* k*.*

This is sometimes formulated as a *definability dichotomy* theorem: each $\mathsf{CSP}(\mathbb{D})$ is either definable in Datalog (a language of very weak expressive power) or not even in $C^{\omega}_{\infty\omega}$ (a language of considerable expressive power), which precedes the complexity dichotomy theorem.

Bounded width constraint satisfaction problems are intimately tied to bounds on the treewidth of structures. Recall that the treewidth of a graph, or more generally a relational structure (Feder & Vardi, 1998), is a measure of its connectivity that plays a pervasive role in algorithmic graph theory. It is linked to Datalog-definable constraint satisfaction problems in two significant ways. Let a class of structures $\mathcal{C}$ be an *obstruction set* for $\overline{\mathsf{CSP}(\mathbb{D})}$ if $\mathbb{A} \notin \mathsf{CSP}(\mathbb{D})$ if, and only if, $\mathbb{C} \longrightarrow \mathbb{A}$ for some $\mathbb{C} \in \mathcal{C}$. In other words, $\overline{\mathsf{CSP}(\mathbb{D})}$ is generated by the structures in $\mathcal{C}$ through homomorphisms. We then have that $\mathsf{CSP}(\mathbb{D})$ has bounded width if, and only if, it has an obstruction set of bounded treewidth (see Bulatov et al., 2008). A second noteworthy connection between treewidth and the Datalog definability of constraint satisfaction problems is the result of Dalmau et al. (2002) to the effect that for *any* $\mathbb{D}$ and any value of k, there is a Datalog program that defines the structures of treewidth at most k which are in $\mathsf{CSP}(\mathbb{D})$.

18.4.2 Algebraic Methods

The canonical example of a CSP which is tractable but not of bounded width is the problem of solving systems of linear equations over a finite field. For example, consider the structure $\mathbb{Z}_2$ with universe $\{0, 1\}$ and two ternary relations R_0 and R_1 where for each i, R_i contains the four triples $(a, b, c) \in \{0, 1\}^3$ such that $a + b + c = i$ (mod 2). We can then see an arbitrary structure $\mathbb{A}$ over the vocabulary R_0, R_1 as a system of equations with three variables per equation, and there is a homomorphism $\mathbb{A} \longrightarrow \mathbb{Z}_2$ if, and only if, the system is solvable.

That $\mathsf{CSP}(\mathbb{Z}_2)$ is not definable in Datalog is known from Feder and Vardi (1998) and that it is not invariant under $\equiv^k$ is known from Atserias et al. (2009). The fact that it is tractable follows from a simple Gaussian elimination algorithm. It is noteworthy that much of the progress towards solving the CSP dichotomy conjecture of Feder and Vardi over the years can be understood as extending and generalising the reach of linear algebraic methods such as Gaussian elimination to an ever wider class of problems. Major milestones in this direction include the algorithm for CSPs with Maltsev polymorphisms (Bulatov & Dalmau, 2006), the tractability of constraint systems with few subpowers (Idziak et al., 2010) and, most significantly the algorithms of Bulatov (2017) and Zhuk (2017) for alls CSPs with a weak near unanimity polymorphism.

18.5 The Pebbling Comonad

We can understand the relation $\mathbb{A} \Rrightarrow^{\exists^+ L^k} \mathbb{B}$ as implying that the k-local consistency algorithm cannot tell us that there is not a homomorphism from $\mathbb{A}$ to $\mathbb{B}$. It is an efficient algorithm, for fixed values of k, and in many cases it is complete, that is to say it correctly determines whether or not $\mathbb{A} \longrightarrow \mathbb{B}$. In particular, this is so when $\mathsf{CSP}(\mathbb{B})$ is of bounded width or when the treewidth of $\mathbb{A}$ is small (with respect to k). Thus, we can see the family of relations $\Rrightarrow^{\exists^+ L^k}$ indexed by k as *tractable approximations* of the homomorphism relation.

In a similar way, we understand the family of relations $\mathbb{A} \equiv^k \mathbb{B}$ as tractable approximations of the isomorphism relation. Indeed, this is exactly how the k-dimensional Weisfeiler–Leman equivalences have been widely studied in the graph isomorphism literature. And, it turns out that they have many equivalent characterizations in terms of logic, combinatorics, algebra and others. Again, these tractable approximations are exact on some interesting restricted instances, such as when the treewidth of $\mathbb{A}$ and $\mathbb{B}$ is bounded. This suggests an analogy with the situation for homomorphism, which turns out to be more than just an analogy.

The relation $\mathbb{A} \Rrightarrow^{\exists^+ L^k} \mathbb{B}$ is witnessed by a winning strategy for Duplicator in the existential positive k-pebble game. Moreover, we can compose strategies in the sense that if α is a winning strategy in the game played on $(\mathbb{A}, \mathbb{B})$ (we write $\alpha : \mathbb{A} \Rrightarrow^{\exists^+ L^k} \mathbb{B}$) and similarly $\beta : \mathbb{B} \Rrightarrow^{\exists^+ L^k} \mathbb{C}$ then these can be combined to give a Duplicator winning strategy in the game played on $\mathbb{A}$ and $\mathbb{C}$. Though we have not yet defined strategies formally as objects intuitively we can think of the composed strategy $\beta\alpha$ as one where Duplicator's response to a Spoiler move on $a \in \mathbb{A}$ is obtained by taking the response given by β to Spoiler playing $\alpha(a)$ in $\mathbb{B}$. The identity strategy on $(\mathbb{A}, \mathbb{A})$ is one where Duplicator's response to any Spoiler move on an element a is to play on a itself. We thus have a category which captures the k-local consistency approximation of homomorphism. And, it turns out that isomorphism in this category is exactly the relation $\equiv^k$.

It was this insight that is the germ of the definition of the *pebbling comonad*, $\mathbb{P}_k$ (Abramsky et al., 2017). Before defining this, I give a brief introduction to comonads. I assume that the reader has some familiarity with basic definitions from category theory, in particular the notions of category, functor and natural transformation. An introduction may be found in Abramsky and Tzevelekos (2010). For a finite signature σ, we are interested in the category $\mathcal{R}(\sigma)$ of relational structures over σ. The objects of the category are such structures and the maps are homomorphisms between structures.

A *comonad* $\mathbb{T}$ on a category $\mathcal{C}$ is a triple $(\mathbb{T}, \epsilon, \delta)$ where $\mathbb{T}$ is an endofunctor of $\mathcal{C}$, and ϵ and δ are natural transformations, giving for each object $A \in \mathcal{C}$, morphisms $\epsilon_A : \mathbb{T}A \longrightarrow A$ and $\delta_A : \mathbb{T}A \longrightarrow \mathbb{T}\mathbb{T}A$ so that the following diagrams commute.

$$\begin{tikzcd}
\mathbb{T}A \arrow[r, "\delta_A"] \arrow[d, "\delta_A"'] & \mathbb{T}\mathbb{T}A \arrow[d, "\mathbb{T}\delta_A"] \\
\mathbb{T}\mathbb{T}A \arrow[r, "\delta_{\mathbb{T}A}"'] & \mathbb{T}\mathbb{T}\mathbb{T}A
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbb{T}A \arrow[r, "\delta_A"] \arrow[d, "\delta_{\mathbb{A}}"'] \arrow[dr, equal] & \mathbb{T}\mathbb{T}A \arrow[d, "\mathbb{T}\epsilon_{\mathbb{A}}"] \\
\mathbb{T}\mathbb{T}A \arrow[r, "\epsilon_T"'] & \mathbb{T}A
\end{tikzcd}$$

We call ϵ the *counit* and δ the comultiplication of the comonad $(\mathbb{T}, \epsilon, \delta)$.

Associated with any comonad $(\mathbb{T}, \epsilon, \delta)$ is a *Kleisli category* we denote $\mathcal{K}(\mathbb{T})$. The objects are the objects of the underlying category $\mathcal{C}$ and the maps $A \xrightarrow[\mathcal{K}(\mathbb{T})]{} B$ are morphisms $\mathbb{T}A \longrightarrow B$ in $\mathcal{C}$. Composition is given by the comultiplication:

$$\mathbb{T}\mathbb{A} \xrightarrow{\delta_{\mathbb{A}}} \mathbb{T}\mathbb{T}\mathbb{A} \xrightarrow{\mathbb{T}f} \mathbb{T}\mathbb{B} \xrightarrow{g} \mathbb{C}.$$

The identity morphisms are given by the counit: $\epsilon_{\mathbb{A}} : \mathbb{T}\mathbb{A} \longrightarrow \mathbb{A}$.

A coalgebra for the comonad is a map $\alpha : A \to \mathbb{T}A$ such that the following diagrams commute.

$$\begin{tikzcd}
A \arrow[r, "\alpha"] \arrow[d, "\alpha"'] & \mathbb{T}A \arrow[d, "\delta_{\mathbb{A}}"] \\
\mathbb{T}A \arrow[r, "\mathbb{T}\alpha"'] & \mathbb{T}\mathbb{T}A
\end{tikzcd}
\qquad
\begin{tikzcd}
A \arrow[r, "\alpha"] \arrow[dr, "id_{\mathbb{A}}"'] & \mathbb{T}A \arrow[d, "\epsilon_{\mathbb{A}}"] \\
 & A
\end{tikzcd}$$

To get the comonad $\mathbb{P}_k$, we define for each structure $\mathbb{A}$ an infinite structure $\mathbb{P}_k\mathbb{A}$ with universe $(A \times [k])^+$. To get a functor, we lift a morphism $f : \mathbb{A} \longrightarrow \mathbb{B}$ to the map $\mathbb{P}_k\mathbb{A} \longrightarrow \mathbb{P}_k\mathbb{B}$ given by

$$[(a_1, p_1), \dots, (a_m, p_m)] \mapsto [(f(a_1), p_1), \dots, (f(a_m), p_m)].$$

The counit $\epsilon_{\mathbb{A}}$ takes a sequence $[(a_1, p_1), \dots, (a_m, p_m)]$ to a_m, i.e., the first component of the last element of the sequence. The comultiplication $\delta_{\mathbb{A}}$ takes a sequence $[(a_1, p_1), \dots, (a_m, p_m)]$ to the sequence $[(s_1, p_1), \dots, (s_m, p_m)]$ where $s_i = [(a_1, p_1), \dots, (a_i, p_i)]$. The relations are defined so that $(s_1, \dots, s_r) \in R^{\mathbb{P}_k\mathbb{A}}$ if, and only if, the s_i are all comparable in the prefix order of sequences (and hence form a chain), $R^{\mathbb{A}}(\epsilon_{\mathbb{A}}(s_1), \dots, \epsilon_{\mathbb{A}}(s_r))$ and whenever s_i is a prefix of s_j and ends with the pair (a, p), there is no prefix of s_j properly extending s_i which ends with (a', p)

for any $a' \in A$. It can then be verified by diagram chasing that all conditions in the definition of a comonad are satisfied.

We can think of $\mathbb{P}_k\mathbb{A}$ as an infinite unfolding of $\mathbb{A}$ into a tree-like structure where each branch codes an infinite play of Spoiler moves in a pebble game. At any point of the tree, there is a tuple of at most k elements of $\mathbb{A}$ which are "live" in the game. A homomorphism $\mathbb{P}_k\mathbb{A} \longrightarrow \mathbb{B}$ then provides a Duplicator response to the last Spoiler move in any such history. Thus, the morphisms in the Kleisli category of the comonad $\mathbb{P}_k$ are exactly Duplicator winning strategies. And, one can show that isomorphisms are exactly winning strategies in the k-pebble bijection game.

The comonad exposes a number of interesting features of these k-pebble games. It links them to the definition of treewidth. Indeed, we are able to show that a structure $\mathbb{A}$ has treewidth less than k if, and only if, it admits a coalgebra $\alpha : \mathbb{A} \to \mathbb{P}_k\mathbb{A}$. The comonad also gives an elegant account of the construction of Cai, Fürer and Immerman, which can be seen as a means of lifting the proof that systems of linear equations over the two-element field GF_2 is not solvable by k-local consistency to a proof that solvability of such systems is not invariant under $\equiv^k$. Also, the observation we made above that we cannot characterise the class of structures $\mathbb{B}$ for which $\mathbb{A} \Rrightarrow^{\exists^+ L^k} \mathbb{B}$ by means of a finitary formula of $\exists^+ L^k$ can be turned into a no-go theorem that shows that any comonad like $\mathbb{P}_k$ must be infinitary.

18.6 Preservation Theorems

In early work on finite model theory, classical preservation theorems of model theory were a topic of great interest. It was shown that many such theorems fail when we restrict ourselves to finite structures. For instance the Łoś-Tarski preservation theorem (Tarski, 1954) tells us that every sentence of first-order logic whose models are closed under extensions is equivalent to an existential sentence. This was shown by Tait (1959) to fail in the finite. That is to say, there is a sentence of first-order logic whose finite models are closed under extensions but which is not equivalent *over finite structures* to an existential sentence. Similarly, Lyndon's preservation theorem (Lyndon, 1959) which tells us that monotone sentences are equivalent to positive ones also fails in the finite. Rossman (2008) showed, however, that the *homomorphism preservation* theorem holds in the finite: every sentence of first-order logic whose finite models are closed under homomorphisms is equivalent over finite structures to an existential positive sentence.

The homomorphism preservation theorem has been of considerable interest to work in constraint satisfaction problems. Among other things, it classifies the first-order definable CSPs as exactly those with a finite obstruction set. There has also been interest in investigating whether such preservation theorems might hold for richer logics. For instance, Datalog is the existential positive fragment of the fixed-point logic FP. Could it be that every sentence of FP whose models are closed under homomorphisms is definable in Datalog? This question was answered negatively in Dawar and Kreutzer (2008), where a homomorphism-closed class $\mathcal{P}$ of finite structures is constructed which is definable in FP but not in Datalog. It is noteworthy that while most proofs that show some property is not Datalog definable work by showing that the property is not invariant under $\Rrightarrow^{\exists^+ L^k}$ for any k, this is not the case for the class $\mathcal{P}$. Indeed, it is $\Rrightarrow^{\exists^+ L^3}$ invariant, and the proof relies rather on a pumping lemma argument. What this shows is that the reason that $\mathcal{P}$ is not definable in Datalog does not have to do with the closure properties of classes that are Datalog definable but with the weakness of the recursion mechanism incorporated in Datalog. It could still be the case that every homomorphism-closed property definable in FP is invariant under $\Rrightarrow^{\exists^+ L^k}$ for some k. Indeed, the following is an open question.

Question 1 If $\mathcal{C}$ is a class of finite structures, closed under homomorphisms and invariant under $\equiv^{L^k}$, is it also invariant under $\Rrightarrow^{\exists^+ L^{k'}}$ for some k'?

A positive answer to this question would provide a preservation theorem for infinitary logic. It says that any homomorphism-closed property definable in the infinitary logic $L^{\omega}_{\infty\omega}$ is definable in its existential positive fragment. This is, indeed, an open question even without the restriction to finite structures. Some partial results in this direction were obtained by Feder and Vardi (2003) who show the *existential* fragment of $L^{\omega}_{\infty\omega}$ is no more expressive than its existential positive fragment in defining homomorphism-closed properties.

One can ask the same question for logics with counting. Since cardinalities of definable sets are not preserved along homomorphisms, it seems conceivable that counting quantifiers are of no use in defining homomorphism-closed properties. Moreover, as the construction of the pebbling comonad has shown us, there is a natural relationship between the relations $\equiv^k$ and $\Rrightarrow^{\exists^+ L^k}$. Thus, it seems natural to ask the following, stronger, variant of Question 1.

Question 2 If $\mathcal{C}$ is a class of finite structures, closed under homomorphisms and invariant under $\equiv^k$ is it also invariant under $\Rrightarrow^{\exists^+ L^{k'}}$ for some k'?

A positive answer to this would be a homomorphism preservation theorem for the infinitary logic with counting: $C^{\omega}_{\infty\omega}$. Note, that we do have a positive answer to the question in the special case where the class of structures $\mathcal{C}$ is of the form $\overline{\mathsf{CSP}(\mathbb{D})}$ for some fixed finite structure $\mathbb{D}$. This follows from the definability dichotomy theorem (Theorem 18.4.2). As the pebbling comonad gives us a category in which the relation $\Rrightarrow^{\exists^+ L^k}$ corresponds to the existence of morphisms and $\equiv^k$ to isomorphism, there is a purely categorical formulation of Question 2.

Another variation of homomorphism preservation is obtained by considering preservation along the relations $\Rrightarrow^{\exists^+ L^k}$. The following question, asked in Atserias et al. (2006) remains open.

Question 3 Is it the case that every first-order definable class of structures which is closed under the relation $\Rrightarrow^{\exists^+ L^k}$ is definable in $\exists^+ L^k$?

18.7 Concluding Remarks

Determining whether there is a homomorphism or an isomorphism between a pair of finite structures, and finding one if there is, are very natural computational problems. Indeed, many other well-studied problems can be recast as special cases of these. Moreover, both homomorphism and isomorphism are computationally difficult problems in their own ways. Neither is known to be solvable in polynomial time and the first is known to be **NP**-complete. In both cases, the study has given rise to algorithms that approximate the problems in various ways. In the case of homomorphism problems, we are usually interested in tractable approximations for practical reasons as general-purpose constraint solvers are often inefficient. In the case of isomorphism, the interest is more theoretical. In practice, we have very efficient general-purpose solvers but the approximations give us a greater structural understanding of the problem.

The local consistency algorithms on the one hand and Weisfeiler–Leman algorithms on the other provide a well-behaved and much studied family of approximations of homomorphism and isomorphism respectively. They have arisen independently in many different contexts and in many different guises. It turns out that they are also intimately related to each other. Our work has shown that they can be understood as deciding homomorphism and isomorphism respectively in the same natural category: the Kleisli category of the pebbling comonad. This categorical view opens up new avenues of exploration.

One such avenue is to study other methods of approximating homomorphism and isomorphism that go beyond local consistency. For instance, the Bulatov-Zhuk dichotomy results give us a classification of all $\mathbb{D}$ for which $\mathsf{CSP}(\mathbb{D})$ is tractable. Besides the ones of bounded width, they are solvable by a combination of algebraic techniques based on a generalized Gaussian elimination. Does this give a structural approximation of homomorphism on all structures in the same way as k-local consistency does? If so, can it be captured in a comonadic, or other categorical structure? What notions of isomorphism does this give rise to? In the other direction, one could explore other approximations of isomorphism given by group-theoretic algorithms. Can they be graded and classified in the same way? If so, what notions of morphism arise from them?

References

Abramsky, S., Dawar, A., & Wang, P. (2017). The pebbling comonad in finite model theory. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)* (pp. 1–12).

Abramsky, S., & Tzevelekos, N. (2010). Introduction to categories and categorical logic. In *New structures for physics* (pp. 3–94). Springer.

Atserias, A., Bulatov, A., & Dawar, A. (2009). Affine systems of equations and counting infinitary logic. *Theoretical Computer Science, 410*(18), 1666–1683.

Atserias, A., Dawar, A., & Kolaitis, P. G. (2006). On preservation under homomorphisms and unions of conjunctive queries. *Journal of the ACM, 53*, 208–237.

Babai, L. (2016). Graph isomorphism in quasipolynomial time [extended abstract]. In *Proceeding of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC* (pp. 684–697).
Babai, L., Erdős, P., & Selkow, S. M. (1980). Random graph isomorphism. *SIAM Journal on Computing, 9*, 628–635.
Babai, L., Grigoryev, D. Yu., & Mount, D. M. (1982). Isomorphism of graphs with bounded eigenvalue multiplicity. In *Proceedings of the 14th Annual ACM Symposium on Theory of Computing (STOC)* (pp. 310–324).
Babai, L., & Luks, E. M. (1983). Canonical labeling of graphs. In *Proceedings of the 15th ACM Symposium on the Theory of Computing* (pp. 171–183).
Barto, L., & Kozik, M. (2014). Constraint satisfaction problems solvable by local consistency methods. *Journal of the ACM, 61*.
Barwise, J. (1977). On Moschovakis closure ordinals. *Journal of Symbolic Logic, 42*, 292–296.
Bodlaender, H. L. (1990). Polynomial algorithms for graph isomorphism and chromatic index on partial k-trees. *Journal of Algorithms, 11*, 631–643.
Bulatov, A. A. (2017). A dichotomy theorem for nonuniform csps. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS* (pp. 319–330).
Bulatov, A. A., & Dalmau, V. (2006). A simple algorithm for mal'tsev constraints. *SIAM Journal on Computing, 36*, 16–27.
Bulatov, A. A., Grohe, M., Kolaitis, P. G., & Krokhin, A. A. (Eds.), (2009). The constraint satisfaction problem: Complexity and approximability. In *Dagstuhl Seminar Proceedings* (Vol. 09441)
Bulatov, A. A., Jeavons, P., & Krokhin, A. A. (2005). Classifying the complexity of constraints using finite algebras. *SIAM Journal on Computing, 34*(3), 720–742.
Bulatov, A. A. Krokhin, A. A., & Larose, B. (2008). Dualities for constraint satisfaction problems. In *Complexity of Constraints - An Overview of Current Research Themes* (pp. 93–124).
Cai, J.-Y., Fürer, M., & Immerman, N. (1992). An optimal lower bound on the number of variables for graph identification. *Combinatorica, 12*(4), 389–410.
Chandra, A., & Harel, D. (1982). Structure and complexity of relational queries. *Journal of Computer and System Sciences, 25*, 99–128.
Dalmau, V., Kolaitis, Ph. G., & Vardi, M. Y. (2002). Constraint satisfaction, bounded treewidth, and finite-variable logics. In *International Conference on Principles and Practice of Constraint Programming* (pp. 310–326). Springer.
Dawar, A. (1993). *Feasible computation through model theory*. Ph.D thesis, University of Pennsylvania.
Dawar, A., & Kreutzer, S. (2008). On Datalog vs. LFP. In *International Colloquium on Automata, Languages, and Programming (ICALP)* (pp. 160–171). Springer.
Dawar, A., Lindell, S., & Weinstein, S. (1995). Infinitary logic and inductive definability over finite structures. *Information and Computation, 119*(2), 160–175.
Dawar, A., & Wang, P. (2015). A definability dichotomy for finite valued CSPs. In *24th EACSL Annual Conference on Computer Science Logic, CSL 2015* (pp. 60–77).
Ebbinghaus, H.-D., & Flum, J. (1999). *Finite model theory* (2nd ed.). Berlin: Springer.
Enderton, H. B. (1972). *A mathematical introduction to logic*. Cambridge: Academic.
Evdokimov, S., & Ponomarenko, I. (1999). On highly closed cellular algebras and highly closed isomorphisms. *Electronic Journal of Combinatorics, 6*(1), Research paper 18, 31 pages.
Fagin, R. (1974). Generalized first-order spectra and polynomial-time recognizable sets. In R. M. Karp (Ed.), *Complexity of Computation, SIAM-AMS Proceedings* (Vol. 7, pp. 43–73).
Feder, T., & Vardi, M. Y. (1998). Computational structure of monotone monadic SNP and constraint satisfaction: A study through Datalog and group theory. *SIAM Journal of Computing, 28*, 57–104.
Feder, T., & Vardi, M. Y. (2003). Homomorphism closed vs existential positive. In *Proceedings of the 18th IEEE Symposium on Logic in Computer Science* (pp. 311–320).
Filotti, I. S., & Mayer, J. N. (1980). A polynomial-time algorithm for determining the isomorphism of graphs of fixed genus (working paper). In *Proceedings of the 12th Annual ACM Symposium on Theory of Computing (STOC)* (pp. 236–243).

Furst, M. L., Hopcroft, J. E., & Luks, E. M. (1980). Polynomial-time algorithms for permutation groups. In *21st Annual Symposium on Foundations of Computer Science (FOCS)* (pp. 36–41).

Grohe, M., & Mariño, J. (1999). Definability and descriptive complexity on databases of bounded tree-width. In *Proceedings of the 7th International Conference on Database Theory*. LNCS (Vol. 1540, pp. 70–82). Springer.

Grohe, M., & Marx, D. (2015). Structure theorem and isomorphism test for graphs with excluded topological subgraphs. *SIAM Journal on Computing, 44*, 114–159.

Grohe, M., & Otto, M. (2015). Pebble games and linear equations. *The Journal of Symbolic Logic, 80*, 797–844.

Grohe, M. (2017). *Descriptive complexity, canonisation, and definable graph structure theory*. Lecture notes in logic (Vol. 47). Cambridge: Cambridge University Press.

Gurevich, Y. (1988). Logic and the challenge of computer science. In E. Börger (Ed.), *Current trends in theoretical computer science* (pp. 1–57). Computer Science Press.

Hella, L. (1996). Logical hierarchies in PTIME. *Information and Computation, 129*, 1–19.

Holm, B. (2010). *Descriptive complexity of linear algebra*. Ph.D thesis, University of Cambridge.

Idziak, P. M., Markovic, P., McKenzie, R., Valeriote, M., & Willard, R. (2010). Tractability and learnability arising from algebras with few subpowers. *SIAM Journal on Computing, 39*, 3023–3037.

Immerman, N. (1986). Relational queries computable in polynomial time. *Information and Control, 68*, 86–104.

Immerman, N., & Lander, E. S. (1990). Describing graphs: A first-order approach to graph canonization. In A. Selman (Ed.), *Complexity theory retrospective.* Berlin: Springer.

Kolaitis, P. G., & Vardi, M. Y. (1992). Infinitary logics and 0–1 laws. *Information and Computation, 98*(2), 258–294.

Kolaitis, P. G., & Vardi, M. Y. (1995). On the expressive power of Datalog: Tools and a case study. *Journal of Computer and System Sciences, 51*, 110–134.

Kolaitis, Ph. G., & Vardi, M. Y. (2008). A logical approach to constraint satisfaction. In *Complexity of constraints - An overview of current research themes* (pp. 125–155).

Krokhin, A. A., & Zivný, S. (Eds.) (2017). *The constraint satisfaction problem: Complexity and approximability*. Dagstuhl Follow-Ups (Vol. 7). Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

van Leeuwen, J. (Ed.) (1990). *Handbook of theoretical computer science*. MIT Press.

Libkin, L. (2004). *Elements of finite model theory*. Berlin: Springer.

Livchak, A. (1983). The relational model for process control. *Automated Documentation and Mathematical Linguistics, 4*, 27–29.

Luks, E. M. (1982). Isomorphism of graphs of bounded valence can be tested in polynomial time. *Journal of Computer and System Sciences, 25*, 42–65.

Lyndon, R. C. (1959). An interpolation theorem in the predicate calculus. *Pacific Journal of Mathematics, 9*, 129–42.

Otto, M. (1997). *Bounded variable logics and counting—A study in finite models*. Lecture notes in logic (Vol. 9). Berlin: Springer.

Poizat, B. (1982). Deux ou trois choses que je sais de L_n. *Journal of Symbolic Logic, 47*(3), 641–658.

Ponomarenko, I. (1988). The isomorphism problem for classes of graphs that are invariant with respect to contraction (Russian). *Zap. Nauchn. Sem. Leningrad. Otdel. Mat. Inst. Steklov. (LOMI), 174*, 147–177.

Ramana, M. V., Scheinerman, E. R., & Ullman, D. (1994). Fractional isomorphism of graphs. *Discrete Mathematics, 132*, 247–265.

Rossman, B. (2008). Homomorphism preservation theorems. *Journal of the ACM, 55*.

Tait, W. W. (1959). A counterexample to a conjecture of Scott and Suppes. *Journal of Symbolic Logic, 24*, 15–16.

Tarski, A. (1954). Contributions to the theory of models I. *II. Indagationes Mathematicae, 16*, 572–588.

Vardi, M. Y. (1982). The complexity of relational query languages. In *Proceedings of the 14th ACM Symposium on Theory of Computing* (pp. 137–146).

Weisfeiler, B., & Leman, A. (1976). *On construction and identification of graphs*. Lecture notes in mathematicsPrinceton: Citeseer.

Zhuk, D. (2017). A proof of CSP dichotomy conjecture. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS* (pp. 331–342).

Chapter 19
Monadic Monadic Second Order Logic

Mikołaj Bojańczyk, Bartek Klin, and Julian Salamanca

Abstract One of the main reasons for the correspondence of regular languages and monadic second-order logic is that the class of regular languages is closed under images of surjective letter-to-letter homomorphisms. This closure property holds for structures such as finite words, finite trees, infinite words, infinite trees, elements of the free group, etc. Such structures can be modelled using monads. In this paper, we study which structures (understood via monads in the category of sets) are such that the class of regular languages (i.e. languages recognized by finite algebras) are closed under direct images of surjective letter-to-letter homomorphisms. We provide diverse sufficient conditions for a monad to satisfy this property. We also present numerous examples of monads, including positive examples that do not satisfy our sufficient conditions, and counterexamples where the closure property fails.

Keywords Monad · Monadic second order logic · Recognizable language distributive law

Research supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (ERC consolidator grant LIPA, agreement no. 683080).

M. Bojańczyk · J. Salamanca
University of Warsaw, Warsaw, Poland
e-mail: bojan@mimuw.edu.pl

J. Salamanca
e-mail: jsalamanca@mimuw.edu.pl

B. Klin (✉)
University of Oxford, Oxford, UK
e-mail: bartek.klin@cs.ox.ac.uk

A. Palmigiano and M. Sadrzadeh (eds.), *Samson Abramsky on Logic and Structure in Computer Science and Beyond*, Outstanding Contributions to Logic 25,
https://doi.org/10.1007/978-3-031-24117-8_19

19.1 Introduction

A seminal result in automata theory is that recognizable languages (i.e. languages recognized by finite state devices) are exactly those that can be defined in monadic second-order logic (MSO). This result was originally shown for finite words by (independently) Büchi (1990b, Corollary 4), Elgot (1961, Theorem 5.3) and Trakhtenbrot (1961), but it is also true for:

- ω-words, shown by Büchi (1990a, Theorem 1);
- finite trees, shown by Doner (1970, Corollary 3.8) and Thatcher and Wright (1968, Theorem 14);
- infinite trees, shown by Rabin (1969, Theorem 1.7);
- countable linear orders, shown by Carton, Colcombet and Puppis (2011, Theorem 3) building on Shelah (1975, Theorem 6.2);
- graphs of bounded treewidth, which follows from Courcelle's Theorem (Courcelle, 1990, Theorem 4.4) and its converse (Bojańczyk & Pilipczuk, 2016, Theorem 2.10).

For more about these results, see the survey by Thomas (1997).

The large number of examples calls for a more systematic framework, where the notion of a composable structure (be it a finite word, ω-word, finite tree etc.) would be a parameter, and a characterization of the expressive power of MSO, a result. In our view, the lack of such a framework is a manifestation of what Samson Abramsky has recently called

> a remarkable divide in the field of logic in Computer Science, between two distinct strands: one focussing on semantics and compositionality ("Structure"), the other on expressiveness and complexity ("Power"). It is remarkable because these two fundamental aspects of our field are studied using almost disjoint technical languages and methods, by almost disjoint research communities. (Abramsky & Shah, 2018, p. 1).

A generic approach to MSO was proposed in Bojańczyk (2015), using monads. Monads are a standard categorical tool to study compositionality, and they are firmly rooted in the "Structure" strand of Theoretical Computer Science. Applying them to study the expressive power of MSO is a step towards building a bridge over the divide.

Monads are general enough to capture structures such as words, trees, graphs, etc. but specific enough to describe concepts such as recognizable languages (as observed by Eilenberg & Wright, 1967, Sect. 11), syntactic algebras (Bojańczyk, 2015, Sect. 3) or pseudovarieties (Bojańczyk, 2015, Sect. 4). Another advantage of monads is that they can model infinite objects (such as infinite words or trees, which are central topics in automata theory), which is not the case for some alternative approaches, such as Steinby's approach via universal algebra (Steinby, 1998).

When discussing the relationship of MSO and recognizable languages, there are two implications to consider.

- *Recognizable ⇒ definable in* MSO. This implication is easy for structures such as finite words or trees, where MSO can be used to define some canonical decomposition. In other cases, the proof can be much harder. Indeed, the proofs are relatively recent in the cases of countable linear orders (Carton et al., 2011, Theorem 3), graphs of bounded treewidth (Bojańczyk & Pilipczuk, 2016, Theorem 2.10), or graphs of bounded linear cliquewidth (Bojańczyk et al., 2018, Theorem 3.5). The implication is known to fail for graphs of unbounded treewidth or cliquewidth (Courcelle & Engelfriet, 2012, Proposition 4.36), and it also fails for infinite trees under a naive definition of finite algebras (Bojańczyk & Klin, 2019, Sect. 4). Sometimes, e.g. for graphs of bounded cliquewidth, the implication remains an open question (Bojańczyk et al., 2018, Sect. 8). A general understanding of this implication seems to be a hard problem, and we do not make any attempts in that direction in this paper.
- *Definable in* MSO *⇒ recognizable.* In all known cases, when recognizable languages are defined in terms of finite algebras, this implication is relatively straightforward.[1] Roughly speaking, languages recognized by finite algebras are automatically closed under Boolean combinations, and closure under existential quantification is proved using some kind of powerset construction. This argument is so deceptively simple that (Bojańczyk, 2015, Lemma 6.2) wrongly claimed that it works for every monad. One purpose of our paper is to correct this mistake, show counterexamples to the claim, and study conditions that make the implication hold; the resulting landscape of monads turns out to be quite interesting.

Motivating example: finite words To explain how MSO can be defined for an arbitrary monad, we begin by describing MSO for finite words in a manner which can be translated to a more generic setting.

A word w over an alphabet Σ can be understood as a relational structure (Thomas, 1997, Sect. 2.1), denoted by $\underline{w}$, where the universe is the positions of the word, there is a binary predicate $x < y$ for the order on positions, and for every $a \in \Sigma$ there is a unary predicate Q_a which selects positions that have label a. To define properties of a word w, we can use sentences of first-order logic or monadic second-order logic over the vocabulary of $\underline{w}$. For example, if $w \in \{a, b\}^*$ then the sentence

$$\underbrace{\exists x\big[Q_a(x)}_{\substack{\text{there is}\\ \text{a position}\\ \text{with label } a}} \wedge \underbrace{\forall y\ (y > x \Rightarrow Q_a(x))\big]}_{\substack{\text{such that all later positions}\\ \text{also have label } a}}$$

is true in $\underline{w}$ if and only if w belongs to the regular language $(a + b)^* a^+$. This sentence uses only first-order logic (quantification over positions), but some regular languages need monadic second-order logic (quantification over sets of positions). A famous

[1] For this it is important that algebras, and not automata, are used as the notion of recognizability. For example, the hard part of Rabin's theorem is showing that languages recognized by nondeterministic automata are closed under complementation, see Rabin (1969, Theorem 1.5) or Thomas (1997, Theorem 6.2). This difficulty disappears when considering algebras, where complementation is achieved simply by flipping the accepting set.

example is the regular language of words of even length, which is defined by the sentence

$$\underbrace{\exists X}_{\substack{\text{there is}\\ \text{a set of}\\ \text{positions}}} \bigwedge \begin{cases} \underbrace{\forall x\ \exists y\ y \leq x \wedge y \in X}_{\text{the first position is in } X} \\ \underbrace{\forall x\ \exists y\ y \geq x \wedge y \notin X}_{\text{the last position is not in } X} \\ \underbrace{\forall x\ \forall y\ (x < y \wedge \neg(\exists z\ x < z < y)) \Rightarrow (x \in X \Leftrightarrow y \notin X)}_{\text{for every two consecutive positions, exactly one is in } X}. \end{cases}$$

The theorem of Büchi, Elgot and Trakhtenbrot that we mentioned above says that a language of finite words is regular if and only if it can be defined by a sentence of MSO. The "regular $\Rightarrow$ definable in MSO" implication can be proved by using the logic to formalise the semantics of a nondeterministic finite automaton (or a regular expression). For the converse implication, one shows that the class of regular languages has all the closure properties that are used in MSO. This observation is formalised in the following result. (A letter-to-letter homomorphism is defined to be a function of the form $f^* : \Sigma^* \to \Gamma^*$ for some $f : \Sigma \to \Gamma$.)

Proposition 1 (cf. Bojańczyk, 2015, Lemma 6.1) *For finite words, the class of* MSO *definable languages is the least class of languages over finite alphabets that contains the languages* $0^* \subseteq \{0, 1\}^*$ *and* $0^*1^* \subseteq \{0, 1\}^*$ *and is closed under union, intersection, complement, inverse images and direct images along surjective letter-to-letter homomorphisms.*

Proof (*Sketch*) Following Thomas (1997, Sect. 2.3), one can eliminate first-order variables from MSO, and keep only the monadic second-order variables. Instead of the usual atomic predicates $x < y$ and $Q_a(x)$, which use first-order variables, one uses atomic predicates

$$\underbrace{X \subseteq Y}_{\text{set inclusion}} \qquad \underbrace{X < Y}_{\substack{\text{every position in } X \text{ is}\\ \text{before every position in } Y}} \qquad \underbrace{X \subseteq Q_a}_{\text{all positions in } X \text{ have label } a}$$

A formula with free variables $\phi(X_1, \ldots, X_n)$ over an alphabet Σ can be seen as a language over an extended alphabet $\Sigma \times 2^n$. The idea is to interpret a word over the extended alphabet as a word together with a valuation of the sets $X_1, \ldots, X_n$, as indicated by the n bits stored by every position. For instance, if $\phi(X, Y) = X \subseteq Y$ and $\psi(X, Y) = Y \subseteq X$, then the word

$$w = (a, 0, 0)(a, 1, 0)(b, 1, 1),$$

does not satisfy ϕ, but satisfies ψ.

The set of words that satisfy $\exists X\, \phi(X, Y_1, \dots, Y_n)$ is exactly the direct image of the set of words that satisfy $\phi(X, Y_1, \dots, Y_n)$ under the letter-to-letter homomorphism π^* which is obtained by lifting to words the projection

$$\pi : \Sigma \times 2^{n+1} \to \Sigma \times 2^n \qquad (a, x, y_1, \dots, y_n) \mapsto (a, y_1, \dots, y_n)$$

In this sense, the second-order existential quantifier is abstractly captured by direct images under (surjective) letter-to-letter homomorphims.

Given the above observations, one can show the theorem by induction of the structure of MSO formulas. The language $0^* \subseteq \{0, 1\}^*$ corresponds to the predicates $X \subseteq Y$ and $X \subseteq Q_a$ by means of an inverse image (the idea is that the letter 0 represents the positions which satisfy $X \subseteq Y$ and 1 represents the other positions, likewise for $X \subseteq Q_a$). The language $0^*1^* \subseteq \{0, 1\}^*$ corresponds to the predicate $X < Y$, again by an inverse image. Direct images of surjective letter-to-letter homomorphisms correspond to the quantifier $\exists$. Closure under union, intersection and complement come from the logical connectives $\vee$, $\wedge$ and $\neg$, respectively. $\square$

Motivated by Proposition 1, one can define an abstract version of MSO in any monad, at least over the category of sets, as the least class of languages with the closure properties stated in the proposition. This definition was proposed in Bojańczyk (2015, Sect. 6.1), and is described in more detail in Sect. 19.3. A parameter of the definition is a choice of atomic languages (such as 0^* and 0^*1^* in Proposition 1). If the atomic languages are recognizable, and the class of recognizable languages has the required closure properties, then all MSO definable languages are going to be recognizable. As mentioned previously, most closure properties for recognizable languages are straightforward: Boolean combinations and inverse images under surjective letter-to-letter homomorphisms (or, indeed, under arbitrary homomorphisms). There is one exception: direct images under surjective letter-to-letter homomorphisms. The main topic of this paper is understanding this last closure property.

As we will see, there are examples of monads for which the closure property fails. Also, when the property holds, it can hold for various reasons. We identify three rather diverse sufficient conditions:

1. Some monads admit a powerset construction for algebras, which entails closure under direct images. In Sect. 19.5, we identify a sufficient condition on the monad (we call such monads weakly epi-cartesian) which guarantees that a powerset construction works. Many monads are weakly epi-cartesian, including monads for all structures that have been traditionally considered in automata theory (words, trees, etc.).

   As a by-product, we prove that weakly cartesian monads admit distributive laws over the powerset monad (Theorem 22). This result is of independent interest: it extends previous results by Jacobs (2004), and it contrasts with recent negative results of Klin and Salamanca (2018) and Zwart and Marsden (2019).

2. Another sufficient condition is having a Mal'cev term (see Sect. 19.4.2). For example, the free group monad has a Mal'cev term, but it does not admit a powerset construction.
3. Yet another reason is that the monad preserves finiteness, which implies that all languages over finite alphabets are trivially recognizable. Although seemingly trivial, this condition can cover interesting examples, such as the monad of idempotent monoids, where preservation of finiteness is true but not obvious.

We will also show several examples of monads where the closure property holds, but which do not fall into any of the three classes described above. The project of understanding MSO for monads—even in the restricted setting of the category of sets—is still far from complete. We hope, however, that the reader will be intrigued by the rich variety of examples—we show Monad 27 of them altogether.

Structure of the paper. In Sect. 19.2 we recall basic notions and results about monads, followed by an abstract monadic definition of MSO in Sect. 19.3. In Sects. 19.4 and 19.5 we present three sufficient conditions which guarantee that recognizable languages are closed under direct images of surjective letter-to-letter homomorphisms. We show numerous examples there. Other positive examples, which do not satisfy any of the three conditions, are shown in Sect. 19.6. Section 19.7 lists a few counterexamples where the closure property fails. In Sect. 19.8 we illustrate what would happen if somewhat stronger closure properties were required of a monad. We conclude in Sect. 19.9 by sketching main directions of future work.

19.2 Monads

We now introduce some basic concepts and intuitions related to monads. Everything in this section is completely standard (see e.g. Awodey, 2010; MacLane, 1971).

One of the several possible intuitive understandings of monads is that they formalize "ways to collect things". Given a set Σ (understood as an alphabet), a monad returns a set $T\Sigma$ of collections, or structures, built out of letters from Σ. The kind and shape of these structures depends on the monad: for example, the finite list monad sets $T\Sigma$ to be the set of all finite sequences of elements of Σ, and the powerset monad returns the set of all subsets of Σ. A monad must provide further structure:

- A way to apply functions to structures element-wise; formally, a function $f : \Sigma \to \Gamma$ should yield a function $Tf : T\Sigma \to T\Gamma$ so that identity functions and function composition are preserved (formally, this makes T a functor);
- A way to build "singleton" structures, formalized as a function $\eta_\Sigma : \Sigma \to T\Sigma$, called the *unit*, for each alphabet Σ;
- A way to "flatten" structures of structures to single-layer structures, formalized as a function $\mu_\Sigma : T(T\Sigma) \to T\Sigma$, called the *multiplication*, for each alphabet Σ.

These ingredients are subject to a few axioms. First, both the unit and the multiplication must be *natural*, i.e., they must be invariant under arbitrary renamings of elements. Formally, the "naturality squares"

$$
\begin{tikzcd}
\Sigma \arrow[r, "\eta_\Sigma"] \arrow[d, "f"'] & T\Sigma \arrow[d, "Tf"] \\
\Gamma \arrow[r, "\eta_\Gamma"'] & T\Gamma
\end{tikzcd}
\qquad
\begin{tikzcd}
TT\Sigma \arrow[r, "\mu_\Sigma"] \arrow[d, "TTf"'] & T\Sigma \arrow[d, "Tf"] \\
TT\Gamma \arrow[r, "\mu_\Gamma"'] & T\Gamma
\end{tikzcd}
\tag{19.1}
$$

must commute for every function $f : \Sigma \to \Gamma$. Furthermore, the multiplication operation must be associative and the unit must be the actual two-sided unit for it, in the sense made formal in the following definition:

Definition 2 A *monad* (T, η, μ) (on the category $\texttt{Set}$ of sets and functions) is a functor $T : \texttt{Set} \to \texttt{Set}$ together with natural transformations $\eta : Id \Rightarrow T$ and $\mu : TT \Rightarrow T$, subject to axioms:

$$
\begin{tikzcd}
T\Sigma \arrow[r, "\eta_{T\Sigma}"] \arrow[dr, equal] & TT\Sigma \arrow[d, "\mu_\Sigma"] & T\Sigma \arrow[l, "T\eta_\Sigma"'] \arrow[dl, equal] \\
& T\Sigma &
\end{tikzcd}
\qquad
\begin{tikzcd}
TTT\Sigma \arrow[r, "T\mu_\Sigma"] \arrow[d, "\mu_{T\Sigma}"'] & TT\Sigma \arrow[d, "\mu_\Sigma"] \\
TT\Sigma \arrow[r, "\mu_\Sigma"'] & T\Sigma
\end{tikzcd}
\tag{19.2}
$$

for all sets Σ.

We will usually denote a monad (T, η, μ) simply by T. If a risk of confusion arises (i.e. in the presence of more than one monad), its unit and multiplication will then be called η^T and μ^T.

Here are a few standard examples of monads.

Monad 1 The *list monad*, also known as the *free monoid monad*, is defined by $T\Sigma = \Sigma^*$, the set of finite words over Σ, with $Tf(x_1 \cdots x_n) = f(x_1) \cdots f(x_n)$ for $f : \Sigma \to \Gamma$. The unit is defined by singleton words: $\eta_\Sigma(x) = x$, and multiplication by words concatenation: $\mu_\Sigma((w_1)(w_2) \cdots (w_n)) = w_1 w_2 \cdots w_n$.

Monad 2 The same definitions restricted to non-empty lists form the *non-empty list monad* $T\Sigma = \Sigma^+$, also called the *free semigroup monad*.

Monad 3 The *powerset monad* is defined by $T\Sigma = \mathcal{P}\Sigma$, the set of all subsets of Σ, with the action on functions defined by direct image: $\mathcal{P}f(\Delta) = \overrightarrow{f}(\Delta)$ for $f : \Sigma \to \Gamma$ and $\Delta \subseteq \Sigma$. The unit is defined by singletons: $\eta_\Sigma(x) = \{x\}$, and multiplication by set union: $\mu_\Sigma(\Phi) = \bigcup \Phi$ for $\Phi \subseteq \mathcal{P}\Sigma$.

Monad 4 The *nonempty powerset monad* $\mathcal{P}^+$ is defined exactly the same but with $T\Sigma$ restricted to nonempty subsets of Σ.

Monad 5 The *finite powerset monad* is also defined the same but with $T\Sigma$ restricted to finite subsets of Σ. This is also called the *free semilattice monad*.

Monad 6 The *bag monad* $\mathcal{B}$, also known as the *multiset monad* or the *free commutative monoid monad*, is defined so that $\mathcal{B}\Sigma$ is the set of functions from Σ to $\mathbb{N}$ that are zero almost everywhere, and the action on functions is defined by

$$\mathcal{B}f(\beta)(y) = \sum_{x \in f^{-1}(y)} \beta(x);$$

the sum is well-defined since β returns zero almost everywhere. The unit of $\mathcal{B}$ maps an element $x \in \Sigma$ to the function $[x \mapsto 1]$ that maps x to 1 and is zero everywhere else. Multiplication is defined by:

$$\mu_\Sigma(\beta)(x) = \sum_{\phi \in \mathcal{B}\Sigma} \beta(\phi) \times \phi(x).$$

We shall be looking at numerous examples of monads in the following. Many of them are best presented in terms of operations and equations. A useful recipe for defining a (finitary) monad on `Set` begins by considering an algebraic *signature*, i.e., a set of operation symbols, each with an associated finite arity. With a signature fixed, the notion of a *term* over a set X of variables is defined as usual: a variable $x \in X$ is a term, and $\sigma(t_1, \ldots, t_n)$ is a term whenever σ is a symbol of arity n and $t_1, \ldots, t_n$ are terms. An *equation* is a pair of terms (over the same set of variables). A set of equations defines, for any alphabet Σ, a congruence relation on the set of terms over Σ in the expected way: it is the least equivalence relation that is compatible with the operations and satisfies all the equations. One then defines $T\Sigma$ to be the set of equivalence classes of terms over Σ, under that congruence relation. For the unit, $\eta_\Sigma(x)$ is (the equivalence class of) the variable x, and multiplication is defined by term substitution. It is standard to check that these ingredients form a monad on `Set`, and the original set of equations is then called an *equational presentation* of that monad.

Example 3 Let the signature contain a binary symbol $\cdot$ and a constant (i.e., a symbol of arity zero) e. The unit and associativity equations:

$$e \cdot x = x \cdot e = x \qquad (x \cdot y) \cdot z = x \cdot (y \cdot z)$$

form a presentation of the free monoid monad (Monad 1). Adding a further equation for commutativity:

$$x \cdot y = y \cdot x$$

one obtains the free commutative monoid monad (Monad 6). Adding yet another equation:

$$x \bullet x = x$$

yields a presentation of the finite powerset monad (Monad 5).

Definition 4 Given a monad T, an (Eilenberg–Moore) *algebra* for T (shortly, a *T-algebra*) is a set A together with a function $\alpha : TA \to A$ subject to two axioms:

$$\begin{tikzcd} A \arrow[r, "\eta_A"] \arrow[dr, equal] & TA \arrow[d, "\alpha"] \\ & A \end{tikzcd} \qquad \begin{tikzcd} TTA \arrow[r, "\mu_A"] \arrow[d, "T\alpha"'] & TA \arrow[d, "\alpha"] \\ TA \arrow[r, "\alpha"'] & A. \end{tikzcd}$$

A T-algebra *homomorphism* from $\mathbf{A} = (A, \alpha)$ to $\mathbf{B} = (B, \beta)$ is a function $h : A \to B$ such that the diagram

$$\begin{tikzcd} TA \arrow[r, "Th"] \arrow[d, "\alpha"'] & TB \arrow[d, "\beta"] \\ A \arrow[r, "h"'] & B \end{tikzcd}$$

commutes. T-algebras and their homomorphisms form a category, denoted $\mathtt{Alg}(T)$.

It is easy to check that for any set Σ, the set $T\Sigma$ with $\mu_\Sigma : TT\Sigma \to T\Sigma$ is a T-algebra. This is the *free T-algebra over* Σ; its fundamental property is that for any T-algebra $\mathbf{A}$, homomorphisms from $T\Sigma$ to $\mathbf{A}$ are in bijective correspondence with functions from Σ to A.

Example 5 For T the list monad (Monad 1), a T-algebra on A is a function that interprets arbitrary finite sequences over A as elements of A. The two axioms of T-algebras mean that the function is associative in the obvious sense; indeed, it easy to check that T-algebras correspond to monoids. Moreover, T-algebra homomorphisms are monoid homomorphisms. $T\Sigma = \Sigma^*$ is the free monoid over Σ.

The same schema applies to any equationally definable class of algebras. We can therefore consider "free X monad" where X can stand for semigroup, commutative semigroup, group, abelian group, lattice, distributive lattice, Boolean algebra and so on, with algebras for the "free X monad" being exactly X's.

19.3 Monadic Monadic Second Order Logic

We now define our abstract monadic MSO, inspired by Proposition 1. First we define a general notion of a recognizable language (Eilenberg & Wright, 1967).

Definition 6 (*Recognizable T-languages*) Let T be a monad on `Set` and Σ a finite set of symbols, called the *alphabet* in this context. A *T-language* over Σ is a subset of $T\Sigma$. A T-algebra $\mathbf{A} = (A, \alpha)$ *recognizes* a T-language L over Σ iff there is a homomorphism $h : T\Sigma \to \mathbf{A}$, and a subset $S \subseteq A$, such that L is the inverse image of S along h. A T-language is *recognizable* if it is recognized by some finite T-algebra.

For a fixed T, let Rec_T denote the class of all recognizable T-languages.

Example 7 For T the free monoid monad (Monad 1), Rec_T is the class of regular languages (or, equivalently, MSO-definable languages). □

Recognizable languages are closed under Boolean operations.

Proposition 8 *For any monad T, the class Rec_T is closed under binary unions, binary intersections and complement, for a fixed alphabet.*

Proof Let L_i be a language that is recognized by a finite T-algebra $\mathbf{A}_i$ with $S_i \subseteq A_i$ through homomorphisms $h_i : T\Sigma \to \mathbf{A}_i$, $i = 1, 2$. Then:

- $L_1 \cap L_2$ is recognized by $\mathbf{A}_1 \times \mathbf{A}_2$ through the homomorphism $\langle h_1, h_2 \rangle$ and the subset
$$\{(a_1, a_2) \mid a_1 \in S_1 \text{ and } a_2 \in S_2\};$$
- $L_1 \cup L_2$ is recognized by $\mathbf{A}_1 \times \mathbf{A}_2$ through the homomorphism $\langle h_1, h_2 \rangle$ and the subset
$$\{(a_1, a_2) \mid a_1 \in S_1 \text{ or } a_2 \in S_2\};$$
- $T\Sigma \setminus L_1$ is recognized by $\mathbf{A}_1$ through h_1 and $A_1 \setminus S_1$. □

Let $h : T\Sigma \to T\Gamma$ be a T-algebra homomorphism. If a language $L \subseteq T\Gamma$ is recognizable then the inverse image $\overleftarrow{h}(L) \subseteq T\Sigma$ is recognizable as well: it is recognized by the same T-algebra and subset as L. So recognizable languages are closed under inverse images of T-algebra homomorphisms. One may ask the same question about direct images: if $L \subseteq T\Sigma$ is recognizable, is $\overrightarrow{h}(L) \subseteq T\Gamma$ recognizable as well? This question will be the our main technical focus in the following.

We call an algebra homomorphism $h : T\Sigma \to T\Gamma$ a *letter-to-letter homomorphism* if $h = Tf$ for some function $f : \Sigma \to \Gamma$. We call it *surjective* if f is surjective.[2] Proposition 1 motivates the following definition (Bojańczyk, 2015, Sect. 6).

[2] All functors on `Set` preserve surjective functions since every surjective function f has a right inverse g in the sense that $f \circ g = \mathrm{id}$, then we apply T to the last equation to obtain that Tf is also

Definition 9 Let T be a monad on $\texttt{Set}$. Let $\mathcal{L}$ be a family of T-languages over finite alphabets. We define $\mathsf{MSO}(\mathcal{L})$ as the least class of T-languages over finite alphabets that contains $\mathcal{L}$ and is closed under Boolean operations, inverse images of homomorphisms, and direct images of surjective letter-to-letter homomorphisms.

Example 10 Proposition 1 means that:

$$\mathsf{MSO}(\{0^*, 0^*10^*\}) = \mathsf{Rec}_{(-)^*}$$

where both 0^* and 0^*1^* are considered as $(-)^*$-languages over $\{0, 1\}$. □

Now we are ready to state the main technical question of this paper, which is a necessary step towards a result similar to Proposition 1 for other monads, namely:

- When does $\mathcal{L} \subseteq \mathsf{Rec}_T$ imply $\mathsf{MSO}(\mathcal{L}) \subseteq \mathsf{Rec}_T$?

By Proposition 8, Rec_T is closed under Boolean operations. Trivially, inverse images of recognizable languages are also recognizable. Thus the question is reduced to:

- When is Rec_T closed under direct images of surjective letter-to-letter homomorphisms?

Asking this specific question is, to some extent, a design decision. In the classical setting of finite words, recognizable languages are closed under direct images along arbitrary homomorphisms, so one may reasonably ask for a stronger closure property: under arbitrary homomorphisms, or perhaps under arbitrary (i.e. possible non-surjective) letter-to-letter homomorphisms. We will look at these variants of the question in Sect. 19.8.

In any case, one may wonder whether perhaps the question (say, in the weak version as stated above) has an affirmative answer for *every* monad. Indeed, this was mistakenly claimed as a fact in (Bojańczyk, 2015, Lemma 6.2). However, as the following counterexample shows, the situation is not so simple.

Monad 7 Let T be the free monoid monad quotiented by the additional equation

$$x \bullet x \bullet x = x \bullet x. \tag{19.3}$$

The congruence (which we denote $\sim$) induced on Σ^* by (19.3) has been the subject of some research: Brzozowski (1980) asked whether all equivalence classes of this congruence are regular languages, and the question has remained open ever since (see Pin, 2017). We do not need to answer that question for our purposes.

---

surjective. Conversely, under the assumption that each component of η is injective, we have that T is faithful and faithful functors reflect epimorphisms (=surjections in the category $\texttt{Set}$). To show that T is faithful assume $Tf = Tg$ for $f, g : X \to Y$ then

$$\eta_Y \circ g = Tg \circ \eta_X = Tf \circ \eta_X = \eta_Y \circ f$$

which implies $g = f$ since η_Y is injective. There will be no confusion between Tf being surjective and f being surjective since all the monads we consider are such that η_Σ is injective for each Σ.

Elements of $T\Sigma$ are $\sim$-equivalence classes of finite Σ-words, and so a T-language over Σ can be identified with a $\sim$-closed language of Σ-words. It is easy to see that a T-language is recognizable if and only if its associated language of Σ-words is regular.

Let $\Gamma = \{a, b, c\}$ and $\Sigma = \Gamma \cup \{0, 1\}$. Consider the language of finite Σ-words:

$$L = \Gamma^* 0 \Gamma^* 1.$$

It is obviously regular. It is also $\sim$-closed. Indeed, assume that

$$u0v1 = w$$

for some $u, v \in \Gamma^*$ and $w \in \Sigma^*$. This means that there is a finite sequence of applications of (19.3) that transforms $u0v1$ into w. If this sequence is nonempty then its first step must detect a square, that is, a word of the form xx, in $u0v1$. (The other option is that a cube xxx must be present, which implies a square anyway). Since 0 and 1 occur in that word only once each, it is clear that the square subword must occur entirely within u or within v. This means that the result of applying (19.3) to $u0v1$ still belongs to L. By induction, also w must belong to L.

We have thus proved that L, considered as a subset of $T\Sigma$, is T-recognizable.

Now consider $\Sigma' = \Gamma \cup \{0\}$ and let $h : \Sigma \to \Sigma'$ map 1 to 0 and act identically on all other letters. The direct image of L along h, construed (as it should be) as a subset of $T\Sigma'$, corresponds to a language K of Σ'-words which arises as a $\sim$-closure of the language $\Gamma^* 0 \Gamma^* 0$. (Note that the language itself is not $\sim$-closed, as e.g. $a0a0 \sim a0a0a0$.)

We shall show that K is not regular, which will imply that the direct image of L under h is not recognizable.

First, it is a well-known fact that since Γ has 3 elements, the language Γ^* contains infinitely many square-free words. Moreover, two distinct square-free words cannot be $\sim$-equivalent, since (19.3) cannot be applied to a square-free word.

Were the language K regular, its Myhill–Nerode congruence would have a finite index, and so there would be two distinct square-free words $v, w \in \Gamma^*$ which are Myhill–Nerode equivalent with respect to K.

Thanks to (19.3) we have $v0v0v0 \in K$ so, since v and w are Myhill–Nerode equivalent, also

$$v0w0v0 \in K.$$

But it is easy to check that if v and w are square-free and $v \neq w$ then $v0w0v0$ is also square-free, and so it is not $\sim$-equivalent to any word other than itself. We arrive at a contradiction, and so K cannot be regular.

More counterexamples of this kind are presented in Sect. 19.7. But first, now knowing that our main technical question is non-trivial, let us study some conditions on the monad T that guarantee an affirmative answer to it.

19.4 Simple Sufficient Conditions

In this section, we present two conditions on a monad T that guarantee that the class Rec_T is closed under direct images of surjective letter-to-letter homomorphisms. Another, more elaborate sufficient condition will be presented in Sect. 19.5.

19.4.1 Monads That Preserve Finiteness

One straightforward case where direct images of recognizable languages are also recognizable, is the case where the functor part of the monad preserves finiteness, i.e., maps finite sets to finite sets. Trivially, then, any language over Σ is recognized by the identity homomorphisms into the finite algebra $T\Sigma$, so in particular any direct image is recognizable.

Examples of monads with this property include: the powerset monad and its variants, the double contravariant powerset monad 2^{2^Σ}, monads for idempotent semigroups, idempotent monoids, distributive lattices, Boolean algebras, semimodules over a finite semiring, and, in general, any *locally finite variety*. A variety is *locally finite* if every finitely generated algebra is finite.

For a given equational theory, checking if a finitely generated algebra is finite could be a challenging problem. For instance, the fact that a finitely generated idempotent semigroup is finite is a nontrivial fact, with multiple proofs published in different papers, (see e.g. Green & Rees, 1952; Brown & Lazerson, 2009). Another interesting case are distributive lattices, where the exact size of the free algebra on n generators is still unknown for values of $n > 8$.

19.4.2 Monads with a Mal'cev Term

Equational theories over finite signatures that have a Mal'cev term induce monads for which direct images of recognizable languages along surjective letter-to-letter homomorphisms are also recognizable. A Mal'cev term is defined as follows (Burris & Sankappanavar, 1981, II.12).

Definition 11 A ternary term $p(x, y, z)$ on an equational class V is called a *Mal'cev term* if the following identities hold in V:

$$p(x, x, y) = y = p(y, x, x). \tag{19.4}$$

Equational classes with a Mal'cev term are *congruence-permutable* (Burris & Sankappanavar, 1981, Theorem II.12.2), i.e., any two congruences commute in them. Examples of such classes include:

Monad 8 The *free group monad*, presented by a binary operation $\bullet$, a unary operation $(-)^{-1}$ and a constant operation 1, subject to the usual group axioms. A Mal'cev term p is given by $p(x, y, z) = x \bullet y^{-1} \bullet z$.

Similarly, Mal'cev terms can be found for any equational class that has a group reduct, including abelian groups, rings, vector spaces over a field and algebras over a field. In this case, if the group reduct is given in additive notation, then the term p is given by $p(x, y, z) = x - y + z$.

Monad 9 The *free quasigroup monad*, presented by three binary operations $/$, $\bullet$ and $\backslash$ subject to equations:

$$x\backslash(x \bullet y) = x \qquad (x \bullet y)/y = x \qquad x \bullet (x\backslash y) = y \qquad (x/y) \bullet y = x.$$

A Mal'cev term is $p(x, y, z) = (x/(y\backslash y)) \cdot (y\backslash z)$.

Monad 10 The *free Boolean algebra monad*, presented by binary operations $\vee$ and $\wedge$, a unary operation $\neg$ and constants 0 and 1, subject to familiar equations. A Mal'cev term is

$$p(x, y, z) = (x \wedge z) \vee (x \wedge \neg y \wedge \neg z) \vee (\neg x \wedge \neg y \wedge z).$$

Monad 11 The *free Heyting algebra monad*, presented by binary operations $\vee$, $\wedge$ and $\to$, and constants 0 and 1, subject to the standard equations of Heyting algebras. A Mal'cev term is

$$p(x, y, z) = ((x \to y) \to z) \wedge ((z \to y) \to z) \wedge (x \vee z).$$

Note that, apart from the case of Boolean algebras, in all the above examples finitely generated free algebras are infinite.

Proposition 12 *For a monad T presented by a set of equations that admits a Mal'cev term, the class Rec_T is closed under direct images of surjective letter-to-letter homomorphisms.*

Before proceeding with the proof, we recall the standard notion of a congruence. Let $\mathcal{F} = \{f_i\}_{i \in I}$ be a type of algebras, where each operation symbol f_i has arity n_i, and let A be an $\mathcal{F}$–algebra. A binary relation θ on A is a *congruence on* A if it is an equivalence relation and has the compatibility property in the following sense:

- For every $i \in I$, $(a_1, b_1), \ldots, (a_{n_i}, b_{n_i}) \in \theta$ implies $(f_i(a_1, \ldots, a_{n_i}), f_i(b_1, \ldots, b_{n_i})) \in \theta$.

Given such a congruence, the quotient A/θ has an $\mathcal{F}$-algebra structure (see e.g. Burris & Sankappanavar, 1981, II.5 for more properties of congruences and quotient algebras).

Proof Let $L \subseteq T\Sigma$ be recognized by a homomorphism $h : T\Sigma \to A$ through a subset $S \subseteq A$ as $L = \overleftarrow{h}(S)$, where A is a finite T-algebra. Without loss of generality, we may assume that h is a surjective homomorphism. Let $g : \Sigma \to \Gamma$ be a surjective function. Define the relation $\approx$ on $T\Gamma$ as

$$\approx \; = \{(Tg(u), Tg(v)) \mid u, v \in T\Sigma,\; h(u) = h(v)\}.$$

Then $\approx$ is a congruence on $T\Gamma$, which implies that $T\Gamma/{\approx}$ is a T-algebra. Indeed, the relation $\approx$ is clearly reflexive, symmetric and has the compatibility property (the latter because h is a homomorphism). To prove transitivity, let $p(x, y, z)$ be a Mal'cev term and assume $a \approx b$ and $b \approx c$, then

$$a = p(a, b, b) \approx p(b, b, c) = c$$

since $\approx$ is reflexive and has the compatibility property.

Let $\tau : T\Gamma \to T\Gamma/{\approx}$ be the canonical quotient map. Define a function $e : A \to T\Gamma/{\approx}$ by:

$$e(h(u)) = \tau(Tg(u)) \qquad \text{for } u \in T\Sigma;$$

this is well defined since h is surjective and by the definition of $\approx$ and τ. Moreover, e is surjective because both Tg and τ are; as a result, the T-algebra $T\Gamma/{\approx}$ is finite.

We show that the direct image of L along Tg is recognized by the homomorphism τ to $T\Gamma/{\approx}$ through the subset $\overrightarrow{e}(S) \subseteq T\Gamma/{\approx}$. In other words, we will show that

$$\overrightarrow{Tg}\left(\overleftarrow{h}(S)\right) = \overleftarrow{\tau}\left(\overrightarrow{e}(S)\right).$$

To this end, for any $w \in T\Gamma$ calculate:

$$\begin{aligned} w \in \overrightarrow{Tg}\left(\overleftarrow{h}(S)\right) &\iff \exists v \in T\Sigma.\; Tg(v) = w,\; h(v) \in S \\ &\iff \exists u \in T\Sigma.\; Tg(u) \approx w,\; h(u) \in S \\ &\iff \exists u \in T\Sigma.\; \tau(Tg(u)) = \tau(w),\; h(u) \in S \\ &\iff \exists u \in T\Sigma.\; e(h(u)) = \tau(w),\; h(u) \in S \\ &\iff \exists a \in S.\; e(a) = \tau(w) \iff w \in \overleftarrow{\tau}\left(\overrightarrow{e}(S)\right) \end{aligned}$$

which finishes the proof. □

Remark The key step in the above proof is the general observation that in a congruence-permutable variety, every reflexive and symmetric relation with the compatibility property is a congruence (Hutchinson, 1994, Proposition 3.8.). Reflexive, symmetric relations with the compatibility property are called *tolerances*.

In Sect. 19.7.3 we shall see that relaxing the Mal'cev condition even slightly does not guarantee recognizable languages to be preserved under direct images along surjective letter-to-letter homomorphisms.

19.5 Weakly Epi-Cartesian Monads

In this section, we present a class of monads—called weakly epi-cartesian monads—where a powerset construction can be used to prove that recognizable languages are closed under direct images of letter-to-letter homomorphisms. This class includes the classical case of finite words (Monad 1).

We begin with an intuitive description of the powerset construction. Consider a monad T and a language $L \subseteq T\Sigma$ which is recognized by a homomorphism $h : T\Sigma \to \mathbf{A}$ for some T-algebra $\mathbf{A} = (A, \alpha)$. For a surjective function on the alphabet $f : \Sigma \to \Gamma$, we want to show that the direct image of L under the letter-to-letter homomorphism $Tf : T\Sigma \to T\Gamma$ is also recognizable.

A natural idea is to consider a powerset algebra, where the universe is the family of nonempty subsets of A, and the product operation is defined using:

$$t \in T\mathcal{P}^{+}A \quad \mapsto \quad \{\alpha(s) : s \in TA \underbrace{\text{ can be obtained from } t \text{ by choosing an element from each set}}_{\text{can be made precise, see (19.8) in Definition (19)}}\}.$$

There are two issues that need to be addressed here. First, one must check that the powerset algebra is indeed a T-algebra, i.e., that it satisfies the axioms of Definition 4. The second issue is finding a homomorphism into the powerset algebra that recognizes the direct image of L under Tf. The natural idea is to consider the function

$$t \in T\Gamma \qquad \mapsto \qquad \{h(s) : s \in T\Sigma \;\text{ satisfies } Tf(s) = t\};$$

however, it is not immediately clear that this function is a homomorphism. In fact this does not happen in every monad, contrary to what was claimed in (Bojańczyk, 2015, Lemma 6.2). For example, in the monad of groups (Monad 8) the powerset algebra does not satisfy, in general, the axiom $x \cdot x^{-1} = e$.

The goal of this section is to establish a condition on the monad T which ensures that the above powerset construction works as expected.

19.5.1 Definitions

First, a few standard categorical definitions. A commuting diagram of sets and functions:

$$\begin{tikzcd} P \arrow[r, "h"] \arrow[d, "k"'] & X \arrow[d, "f"] \\ Y \arrow[r, "g"'] & Z \end{tikzcd} \tag{19.5}$$

is called a *weak pullback* if, for each $x \in X$ and $y \in Y$ such that $f(x) = g(y)$, there is some $p \in P$ such that $h(p) = x$ and $k(p) = y$. If p is unique for each such x and

y then the square is a *pullback*. For every two functions f and g as above a canonical pullback exists and is defined by

$$P = \{(x, y) \mid f(x) = g(y)\}, \tag{19.6}$$

with $h = \pi_1$ and $k = \pi_2$, projections from P into X and Y.

A functor T *preserves weak pullbacks* iff applying T to everything in a weak pullback as in (19.5) results in a weak pullback again. A routine categorical argument shows that one may equivalently require only that (canonical) pullbacks are mapped to weak pullbacks; preservation of all other weak pullbacks follows. So, T preserving weak pullbacks means that for all functions $f : X \to Z$ and $g : Y \to Z$, and for any $t \in TX$ and $r \in TY$ such that $Tf(t) = Tg(r)$, there exists some $p \in TP$ (for P as in (19.6)) such that $T\pi_1 = t$ and $T\pi_2 = r$.

Intuitively, this means that whenever two terms t and r are sufficiently similar to be equated by equating some variables, then there is some term p, built of compatible pairs of variables, that projects to t and r. Many monads, such as the free monoid monad and the powerset monad and its variants, have this property. The following examples show how the property may fail.

Example 13 The free group monad (Monad 8) does not preserve weak pullbacks: the square

$$\begin{tikzcd} \emptyset \arrow[r, "h"] \arrow[d, "k"'] & \emptyset \arrow[d, "f"] \\ \{a, b\} \arrow[r, "g"'] & \{c\} \end{tikzcd}$$

(with f, g, h, k the unique functions of their type) is a pullback, and terms $t = 1 \in T\emptyset$ and $r = a \cdot b^{-1} \in T\{a, b\}$ are equated by Tf and Tg, but there is no term $p \in T\emptyset$ that would be mapped to r by Tk. □

Example 14 Monad 7, i.e., the free monoid monad subject to the additional axiom $x \cdot x \cdot x = x \cdot x$, does not preserve weak pullbacks. Indeed, the square

$$\begin{tikzcd} \{a, b\} \times \{c, d\} \arrow[r, "\pi_1"] \arrow[d, "\pi_2"'] & \{a, b\} \arrow[d, "f"] \\ \{c, d\} \arrow[r, "g"'] & \{e\} \end{tikzcd}$$

(with f, g the unique functions of their type) is a pullback, and the terms $t = a \cdot b \in T\{a, b\}$ and $r = c \cdot d \cdot c \in T\{c, d\}$ are equated by Tf and Tg, but there is no term $p \in T(\{a, b\} \times \{c, d\})$ that would project to t and r. □

Given `Set`-functors F and G, a natural transformation $\alpha : F \Rightarrow G$ is *weakly cartesian* if for every function $f : X \to Y$ the naturality square

$$\begin{tikzcd} FX \arrow[r, "\alpha_X"] \arrow[d, "Ff"'] & GX \arrow[d, "Gf"] \\ FY \arrow[r, "\alpha_Y"'] & GY \end{tikzcd}$$

is a weak pullback. A weaker property is being *weakly epi-cartesian*, where only naturality squares for surjective f are required to be weak pullbacks.

We can now formulate our main property of interest:

Definition 15 A monad (T, η, μ) is *weakly cartesian* if T preserves weak pullbacks and η and μ are weakly cartesian. It is *weakly epi-cartesian* if T preserves weak pullbacks and η and μ are weakly epi-cartesian.

Weakly cartesian monads have been studied in the literature (Weber, 2004; Clementino et al., 2014; Fritz & Perrone, 2018). The notion of a weakly epi-cartesian monad seems to be new.

The requirement that the naturality square

$$\begin{tikzcd} X \arrow[r, "\eta_X"] \arrow[d, "f"'] & TX \arrow[d, "Tf"] \\ Y \arrow[r, "\eta_Y"'] & TY \end{tikzcd}$$

is a weak pullback amounts to saying that whenever a unit term $\eta_Y(y)$ is obtained as a value of Tf applied to some $t \in TX$, then TX must itself be a unit term $\eta_X(x)$ for some x such that $f(x) = y$. So weak (epi-)cartesianness of η intuitively means that one cannot obtain a unit term by equating variables in a non-unit term. This holds for many monads, including the free monoid monad. Notably, the property fails for the powerset monad, where the non-unit term $\{a, b\} \in \mathcal{P}\{a, b\}$ is mapped to the unit $\{c\} \in \mathcal{P}\{c\}$ by $\mathcal{P}f$, for the unique function $f : \{a, b\} \to \{c\}$.

Let us now look at the multiplication μ being weakly (epi)-cartesian. To say that the naturality square

$$\begin{tikzcd} TTX \arrow[r, "\mu_X"] \arrow[d, "TTf"'] & TX \arrow[d, "Tf"] \\ TTY \arrow[r, "\mu_Y"'] & TY \end{tikzcd}$$

is a weak pullback means that, for any term $t \in TX$, if a term $Tf(t)$ can be decomposed as "term of terms" $\theta \in TTY$ (so that $Tf(t)$ is the flattening of θ), then the term t itself has a similar decomposition. So, intuitively, weak (epi)-cartesianness of μ means that equating variables in a term does not introduce essentially new ways of decomposing the term.

Theorem 16 *If T is a weakly epi-cartesian monad, then* Rec_T *is closed under direct images along surjective letter-to-letter homomorphisms.*

Proof See Sect. 19.5.3. □

19.5.2 Examples

The three conditions of weak (epi-)cartesianness may fail in various configurations. For example, the powerset monad and its variants (Monads 3, 4 and 5) preserve weak pullbacks and have weakly cartesian multiplication, but their unit is not weakly epi-cartesian. On the other hand, Monad 7 does not preserve weak pullbacks, its unit is weakly cartesian and its multiplication is not weakly epi-cartesian. Further examples include:

Monad 12 Consider the monad (T, η, μ) associated to the equational theory of a single ternary function symbol p and a single unary symbol s, with axioms $p(x, x, y) = s(y) = p(y, x, x)$. Then T does not preserve weak pullbacks, η is weakly cartesian, and μ is weakly cartesian.

We shall study this monad in more detail in Sect. 19.7.3.

Monad 13 Consider the monad (T, η, μ) associated to the equational theory on the signature $\{f, e\}$, where f is a unary function symbol and e is a constant, with the only axiom $f(f(x)) = e$. Then T preserves weak pullbacks, η is weakly cartesian, μ is weakly epi-cartesian, but μ is not weakly cartesian, cf. Clementino et al. (2014, Example 3.4).

This example distinguishes between the notions of weakly cartesian and weakly epi-cartesian monads.

However, many monads are weakly epi-cartesian, as the following examples show.

A rich source of weakly cartesian monads (and therefore of weakly epi-cartesian monads) are equational theories whose axioms are given by regular linear equations. A term is called *linear* if no variable appears in it more than once. An equation is called *regular* if each variable appears on the left-hand side if and only if it appears on the right-hand side. An equation is *regular linear* if it is regular and both sides of it are linear. For instance, the equations $x \bullet (y \bullet z) = (x \bullet y) \bullet z$, $x \bullet e = x$, and $e \bullet x = x$, which axiomatize the variety of monoids, are regular linear. The commutativity axiom $x \bullet y = y \bullet x$ is also regular linear. On the other hand, the group axiom $x \bullet x^{-1} = e$ and the idempotence axiom $x \bullet x = x$ are not regular linear.

It is not difficult to check that a monad presented by a set of regular linear equations is weakly cartesian. In fact, every such monad is *analytic*: in addition to being weakly cartesian, its functor weakly preserves *wide* pullbacks. This is a part of a more general framework, see Szawiel and Zawadowski (2014).

In particular, the monads of semigroups, commutative semigroups, monoids, commutative monoids and the variety of all algebras for any given signature, are all weakly cartesian, hence also weakly epi-cartesian.

Example 17 Rec_T is closed under direct images of surjective letter-to-letter homomorphisms for the monads of semigroups, commutative semigroups, monoids, commutative monoids and the variety of all algebras for any given signature. □

The next four example monads are not finitary, so they do not have finitary equational presentations. Nevertheless, they are all weakly cartesian.

Monad 14 (*A monad for ω-words*) Consider the following extension of the list monad to infinite words. The monad maps an alphabet Σ to the set $\Sigma^+ \cup \Sigma^\omega$ of nonempty words of length at most ω. The unit is the same as in the list monad, while the multiplication operation is defined by

$$\mu_X(w_1 w_2 \cdots) = \begin{cases} w_1 w_2 \cdots & \text{if all words } w_1, w_2, \ldots \text{ are finite} \\ w_1 w_2 \cdots w_k & \text{if } w_k \text{ is the first infinite word among } w_1, w_2, \ldots \end{cases}$$

Algebras for this monad are essentially the same as ω-semigroups (Perrin & Pin, 2004, Sect. II.4), which are known to recognize the same languages as Büchi automata on ω-words. This monad is weakly epi-cartesian, so by Theorem 16 the recognizable languages are closed under direct images along surjective letter-to-letter homomorphisms, thus showing that all MSO definable languages are recognized by ω-semigroups.

Monad 15 (*Countable linear orders*) Monad 14 can be generalised from ω-words to other labelled linear orders, where the order type of the positions is not necessarily ω. Consider the monad where $T\Sigma$ is the set of countable linear orders labelled by Σ, up to isomorphism. The unit and multiplication are defined in the natural way (contrary to Monad 14, there is no need to truncate). For example, $T\{a, b\}$ contains the following element: the linear order of the rational numbers, labelled so that the positions with label a are dense and the same is true for the positions with label b. Using a back-and-forth argument, one can show that element described above is unique up to isomorphism. Shelah showed that satisfiability is decidable for MSO over this monad (Shelah, 1975, Theorem 6.2), while Carton, Colcombet and Puppis (Carton et al., 2011, Proposition 3 and Theorem 3) showed that the recognizable languages are exactly the ones that are definable in MSO. This monad is weakly epi-cartesian, and therefore Theorem 16 implies that recognizable languages are closed under direct images of surjective letter-to-letter homomorphisms, which in turn implies that MSO definable languages are recognizable (Carton et al., 2011, Proposition 3).

In fact, as far as closure under direct images is concerned, there is nothing special about countable linear orders, as shown by the following monad.

Monad 16 One can also consider a variant of the previous monad, but for labelled linear orderings of cardinality at most continuum. This monad is also weakly epi-cartesian, and therefore MSO definable languages are recognizable. There is, however, a price to pay for considering uncountable orders: the satisfiability problem for MSO is undecidable for this monad, as proved by Shelah (1975, Theorem 7).

There are also interesting monads that lie between ω-word and all countable linear orders.

Monad 17 Consider the countable linear orders which are scattered, i.e. do not contain any rational sub-ordering. This monad was studied implicitly by Carton and Rispal, where the Eilenberg–Moore algebras for the monad were called $\diamond$-algebras in Rispal and Carton (2005, Definition 6). This monad is weakly epi-cartesian, and therefore Theorem 16 implies that MSO definable languages are necessarily recognizable (this was already known in Rispal & Carton, 2005).

Some monads do not admit any presentation by regular linear equations, yet they still are weakly (epi-)cartesian. For example:

Monad 18 (*A left-idempotent operation*) For a more unusual example of a weakly cartesian monad, consider a monad T presented by an equational theory with a single binary symbol $\bullet$ and with a single (non-linear) equation:

$$\begin{tikzcd}[column sep=small, row sep=small] & \bullet \arrow[dl, dash] \arrow[dr, dash] & & \\ x & & \bullet \arrow[dl, dash] \arrow[dr, dash] & \\ & x & & y \end{tikzcd} \quad = \quad \begin{tikzcd}[column sep=small, row sep=small] & \bullet \arrow[dl, dash] \arrow[dr, dash] & \\ x & & y \end{tikzcd} . \tag{19.7}$$

To prove that T preserves weak pullbacks, consider any pullback diagram

$$\begin{tikzcd} P \arrow[r, "\pi_1"] \arrow[d, "\pi_2"'] & X \arrow[d, "f"] \\ Y \arrow[r, "g"'] & Z, \end{tikzcd}$$

where $P = \{(x, y) \mid f(x) = g(y)\}$ and π_1 and π_2 are the projections from P. Then, for any $r \in TX$, $s \in TY$ and $t \in TZ$ such that

$$Tf(r) = t = Tg(s),$$

we need to find $p \in TP$ such that $T\pi_1(p) = r$ and $T\pi_2(p) = s$.

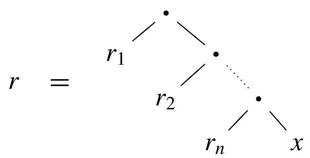
Formally, elements of TZ are equivalence classes of binary trees with elements of Z in leaves; each equivalence class can be represented by its unique smallest element: one where the pattern on the left-hand side of (19.7) does not appear. We therefore only consider r, s and t of this form, and we proceed by induction on the size of t.

For the base case, if t is a single letter (i.e. $t \in Z$) then obviously $r \in X$ and $s \in Y$, therefore we can put $p = (r, s)$.

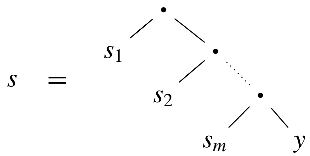
For the inductive step, let

$$t \quad = \quad \begin{tikzcd}[column sep=small, row sep=small] & \bullet \arrow[dl, dash] \arrow[dr, dash] & \\ t' & & z \end{tikzcd}$$

for some $t', z \in TZ$. Since $Tf(r) = t$, it follows that



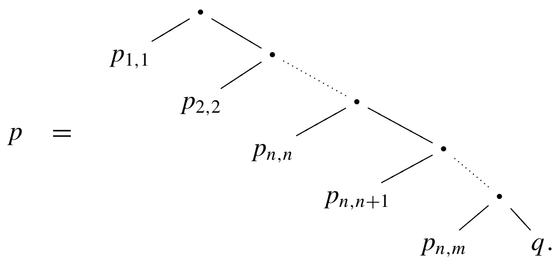
for some $r_1, \ldots, r_n, x \in TX$ such that $Tf(r_i) = t'$ and $Tf(x) = z$. Similarly,



for some $s_1, \ldots, s_m, y \in TY$ such that $Tg(s_i) = t'$ and $Tg(y) = z$. Assume, without loss of generality, that $n \leq m$.

By the inductive assumption, for each $1 \leq i \leq n$ and $1 \leq j \leq m$ there is a $p_{i,j} \in TP$ such that $T\pi_1(p_{i,j}) = r_i$ and $T\pi_2(p_{i,j}) = s_j$. Also by the inductive assumption, there is some $q \in TP$ such that $T\pi_1(q) = x$ and $T\pi_2(q) = y$.

Now consider $p \in TP$ defined by:



It is easy to check that $T\pi_1(p) = r$ and $T\pi_2(p) = s$ as required. This completes the proof that T preserves weak pullbacks.

It is easy to see that the unit of T is cartesian, i.e., that the square

$$\begin{tikzcd} X \arrow[r, "\eta_X"] \arrow[d, "f"'] & TX \arrow[d, "Tf"] \\ Y \arrow[r, "\eta_Y"'] & TY \end{tikzcd}$$

is a pullback for every $f : X \to Y$. Indeed, if $Tf(t) = \eta_Y(y)$ for $t \in TX$ and $y \in Y$ then there must be $t = \eta_X(x)$ for some (necessarily, unique) $x \in X$ such that $f(x) = y$.

It remains to be proved that the multiplication of T is weakly cartesian, i.e., that

$$\begin{tikzcd} TTX \arrow[r, "\mu_X"] \arrow[d, "TTf"'] & TX \arrow[d, "Tf"] \\ TTY \arrow[r, "\mu_Y"'] & TY \end{tikzcd}$$

is a weak pullback for every $f : X \to Y$. The proof is similar to the proof of the fact that T preserves weak pullbacks. For any $r \in TTY$, $s \in TX$ and $t \in TY$ such that

$$\mu_X(r) = t = Tf(s),$$

we need to find $p \in TTX$ such that $TT(p) = r$ and $\mu_X(p) = s$, and the construction proceeds by induction on the size of t.

Monad 19 (*A guarded-idempotent operation*) Consider the monad T defined by the signature $\{\bullet, o\}$, where $\bullet$ is a binary operation symbol and o is a unary operation symbol, and equations $x \bullet (y \bullet z) = (x \bullet y) \bullet z$ and $o(x \bullet x) = o(x)$.

To prove that T preserves weak pullbacks, consider any pullback diagram

$$\begin{tikzcd} P \arrow[r, "\pi_1"] \arrow[d, "\pi_2"'] & X \arrow[d, "f"] \\ Y \arrow[r, "g"'] & Z, \end{tikzcd}$$

where $P = \{(x, y) \mid f(x) = g(y)\}$ and π_1 and π_2 are the projections from P. Let $r \in TX, s \in TY$ and $t \in TZ$ such that $Tf(r) = t = Tg(s)$, we need to find $p \in TP$ such that $T\pi_1(p) = r$ and $T\pi_2(p) = s$.

Elements of TZ are equivalence classes of terms on Z; each equivalence class can be represented by its unique smallest element, up to associativity: one where the pattern $o(x \cdot x)$ does not appear. We therefore assume r, s and t of this form, and we proceed by induction on the size of t.

For the base case, if t is a single letter (i.e. $t \in Z$) then obviously $r \in X$ and $s \in Y$, therefore we can put $p = (r, s)$.

For the inductive case, let $t \in TZ$. We have the following cases:

(i) t does not contain the function symbol o. In this case, r and s do not contain o either. Therefore, t is of the form $t = z_1 \cdot \cdots \cdot z_n$ which implies that r and s are of the form $r = x_1 \cdot \cdots \cdot x_n$ and $s = y_1 \cdot \cdots \cdot y_n$. Hence, we can take $p = (x_1, y_1) \cdot \cdots \cdot (x_n, y_n)$.

(ii) t contains the function symbol o. Let $t = t_1 \cdot o(t_2) \cdot t_3$ where $t_i \in TX$. Then, r and s are of the form $r = r_1 \cdot o(r_2) \cdot r_3$ and $s = s_1 \cdot o(s_2) \cdot s_3$, $r_i \in TX$ and $s_i \in TY$, with $Tf(r_1) = Tg(s_1), Tf(o(r_2)) = Tg(o(s_2))$ and $Tf(r_3) = Tg(s_3)$. By the induction hypothesis, there exists $p_1, p_3 \in TP$ such that:

$$T\pi_1(p_1) = r_1, \quad T\pi_2(p_1) = s_1, \quad T\pi_1(p_3) = r_3, \quad \text{and} \quad T\pi_2(p_3) = s_3,$$

Now, since $Tf(o(r_2)) = Tg(o(s_2))$ we have the following cases:

(a) $Tf(r_2) = Tg(s_2)$. Then, by the induction hypothesis, there exists $p_2 \in TP$ such that $T\pi_1(p_2) = r_2$ and $T\pi_2(p_2) = s_2$.
(b) $Tf(r_2) = Tg((s_2)^{2^n})$ for some $n \geq 0$. Then, by the induction hypothesis, there exists $p_2 \in TP$ such that $T\pi_1(p_2) = r_2$ and $T\pi_2(p_2) = (s_2)^{2^n}$.
(c) $Tf((r_2)^{2^n}) = Tg(s_2)$ for some $n \geq 0$. Then, by the induction hypothesis, there exists $p_2 \in TP$ such that $T\pi_1(p_2) = (r_2)^{2^n}$ and $T\pi_2(p_2) = s_2$.

In each case, put $p = p_1 \cdot o(p_2) \cdot p_3$.

It is easy to check that, in each case, $T\pi_1(p) = r$ and $T\pi_2(p) = s$ as required. This completes the proof that T preserves weak pullbacks.

To see that the unit of T is cartesian, consider a function $f : X \to Y$, $t \in TX$ and $y \in Y$ such that $Tf(t) = \eta_Y(y)$. Then there must be $t = \eta_X(x)$ for some (necessarily, unique) $x \in X$ such that $f(x) = y$.

Finally, the proof that the multiplication μ is weakly cartesian is done by using a similar argument as the one that T preserves weak pullbacks.

Monad 20 (*Weakly epi–cartesian monad that is not weakly cartesian*) Consider the monad T presented by an equational theory with two unary operation symbols f and g, and a constant symbol e subject to the equations:

$$f(f(x)) = g(g(y)) = e$$

Note that TX is infinite for every non-empty X ($\{fg(x), fgfg(x), fgfgfg(x), \ldots\} \subseteq TX$ for $x \in X$). Also, the multiplication μ is not weakly cartesian since for the inclusion $\iota : \{x\} \to \{x, y\}$ we have that $T\iota(e) = \mu_{\{x,y\}}\Big(f\big(f(y)\big)\Big)$ but $f\big(f(y)\big)$ is not in the image of $TT\iota$. Nevertheless, T is a weakly epi-cartesian monad, so Rec_T is closed under direct images along surjective letter-to-letter homomorphisms.

19.5.3 Proof of Theorem 16

We start by recalling the standard concept of a distributive law between monads (Beck, 1969). Distributive laws are a standard tool for composing monads; we will use them to lift the powerset functor to the category of algebras for a monad.

Definition 18 Let (T, η^T, μ^T) and (S, η^S, μ^S) be monads. A *distributive law* of (T, η^T, μ^T) over (S, η^S, μ^S) is a natural transformation $\lambda : TS \Rightarrow ST$ that satisfies the following axioms:

$$
\begin{tikzcd}
T \arrow[r, "T\eta^S"] \arrow[dr, bend right, "\eta^S T"'] \arrow[dr, phantom, "(a)"] & TS \arrow[d, "\lambda"] & S \arrow[l, "\eta^T S"'] \arrow[dl, bend left, "S\eta^T"] \arrow[dl, phantom, "(b)"] \\
& ST &
\end{tikzcd}
\qquad
\begin{tikzcd}
TTS \arrow[rr, "\mu^T S"] \arrow[d, "T\lambda"'] & \arrow[d, phantom, "(d)"] & TS \arrow[d, "\lambda"] & \arrow[d, phantom, "(c)"] & TSS \arrow[ll, "T\mu^S"'] \arrow[d, "\lambda S"] \\
TST \arrow[r, "\lambda T"'] & STT \arrow[r, "S\mu^T"'] & ST & SST \arrow[l, "\mu^S T"] & STS \arrow[l, "S\lambda"]
\end{tikzcd}
$$

Sometimes one is interested in laws that satisfy only some of these axioms. In particular, a distributive law of the functor T over the monad (S, η^S, μ^S) only satisfies the axioms (a) and (c). We will be interested in distributive laws of monads over the powerset or the nonempty powerset, i.e., $S = \mathcal{P}$ or $S = \mathcal{P}^+$.

Definition 19 (*The Jacobs Law*) Let T be a functor. For every set X define a function $\lambda_X : T\mathcal{P}X \to \mathcal{P}TX$ by:

$$\lambda_X(t) = \{s \in TX \mid \exists r \in T(\in_X) \text{ s.t. } T\pi_1(r) = s,\ T\pi_2(r) = t\}, \tag{19.8}$$

where $\in_X$ is the membership relation on X:

$$\in_X = \{(x, S) \mid x \in S \subseteq X\}$$

and $\pi_1 : \in_X \to X$ and $\pi_2 : \in_X \to \mathcal{P}^+X$ are the canonical projections.

It is easy to see that if $t \in T\mathcal{P}^+X$ then $\lambda_X(t)$ is not empty. Indeed, pick any function $k : \mathcal{P}^+X \to X$ such that $k(S) \in S$ for all $S \in \mathcal{P}^+X$. Then

$$\langle k, \mathrm{id}\rangle : \mathcal{P}^+X \to \in_X$$

and the element $r = T\langle k, \mathrm{id}\rangle(t) \in T(\in_X)$ witnesses the fact that $\lambda_X(t)$ is not empty. As a result, λ_X restricts to a function $\lambda_X : T\mathcal{P}^+X \to \mathcal{P}^+TX$.

The following result is due to Jacobs (2004, Sect. 4).

Proposition 20 *If a functor T preserves weak pullbacks, then λ as in Definition 19 is a natural transformation and it is a distributive law of the functor T over the powerset monad $\mathcal{P}$ (and over the non-empty powerset monad $\mathcal{P}^+$).* $\square$

For any sets X and Y, there is a canonical partial order on the set of functions from X to $\mathcal{P}Y$: we say that $f \leq g : X \to \mathcal{P}Y$ iff $f(x) \subseteq g(x)$ for all $x \in X$.

The following property of λ will be useful later:

Proposition 21 *If T preserves weak pullbacks then λ as in Definition 19 is* monotone*: whenever $f \leq g : X \to \mathcal{P}Y$ then*

$$\lambda_X \circ Tf \leq \lambda_X \circ Tg : TX \to \mathcal{P}TY.$$

Proof Assume $f \leq g$, and pick some $t \in TX$ and $s \in \lambda_Y(Tf(t))$. This means that there is some $r \in T{\in_Y}$ such that

$$T\pi_1(r) = s \qquad \text{and} \qquad T\pi_2(r) = Tf(t).$$

Consider the pullback

$$\begin{tikzcd} Q \arrow[r, "\pi_2"] \arrow[d, "\mathrm{id}\times f"'] & X \arrow[d, "f"] \\ \in_Y \arrow[r, "\pi_2"'] & \mathcal{P}Y \end{tikzcd}$$

where

$$Q = \{(y, x) \mid x \in X,\ \ y \in f(x)\}.$$

Since T preserves weak pullbacks, the square

$$\begin{tikzcd} TQ \arrow[r, "T\pi_2"] \arrow[d, "T(\mathrm{id}\times f)"'] & TX \arrow[d, "Tf"] \\ T\in_Y \arrow[r, "T\pi_2"'] & T\mathcal{P}Y \end{tikzcd}$$

is a weak pullback. By the properties of r, and by the definition of a weak pullback, there is some $q \in TQ$ such that

$$T(\mathrm{id} \times f)(q) = r \quad (\text{hence } T\pi_1(q) = T\pi_1(r)) \qquad \text{and} \qquad T\pi_2(q) = t.$$

Since $f \le g$, the function $\mathrm{id} \times g : Q \to \in_Y$ is well defined. Put $u = T(\mathrm{id} \times g)(q) \in T\in_Y$ and calculate:

$$T\pi_1(u) = T\pi_1(q) = T\pi_1(r) = s$$

and

$$T\pi_2(u) = Tg(T\pi_2(q)) = Tg(t),$$

hence u is a witness to the fact that $s \in \lambda_Y(Tg(t))$ as required. □

We now extend Jacobs's Proposition 20 from functors to monads (cf. Definition 15).

Theorem 22 *If a monad (T, η, μ) is weakly cartesian, then λ as in Definition 19 is a distributive law of the monad over the powerset monad $\mathcal{P}$. If it is weakly epi-cartesian, then λ is a distributive law of the monad over the non-empty powerset monad $\mathcal{P}^+$.*

Proof The proofs of both statements are almost identical, so we shall present only the proof of the first statement (the one for $\mathcal{P}$), and mark with $(\star)$ the two places where replacing $\mathcal{P}$ with $\mathcal{P}^+$ makes a difference.

Since weakly (epi)-cartesian monads preserve weak pullbacks by definition, the naturality of λ and axioms (a) and (c) in Definition 18 follow from Proposition 20. What remains to be proved is that λ satisfies axioms (b) and (d) in Definition 18. For the axiom (b), we need to prove that

$$\lambda_X(\eta_{\mathcal{P}X}(Z)) = \mathcal{P}\eta_X(Z) \qquad \text{for } Z \subseteq X. \tag{19.9}$$

For the left-hand side, unfold from (19.8):

$$s \in \lambda_X(\eta_{\mathcal{P}X}(Z)) \iff \exists r \in T(\in_X) \text{ s.t. } T\pi_1(r) = s,\ T\pi_2(r) = \eta_{\mathcal{P}X}(Z). \tag{19.10}$$

For the right-hand side:

$$s \in \mathcal{P}\eta_X(Z) \iff \exists z \in Z \text{ s.t. } s = \eta_X(z). \tag{19.11}$$

For the right-to-left inclusion of (19.9), given z as in (19.11), it is enough to put $r = \eta_{\in_X}(z, Z)$ and use naturality of η.

For the left-to-right inclusion of (19.9), assume r as in (19.10). Since T is weakly cartesian, the square

$$\begin{tikzcd}
\in_X \arrow[r, "\pi_2"] \arrow[d, "\eta_{\in_X}"'] & \mathcal{P}X \arrow[d, "\eta_{\mathcal{P}X}"] \\
T\in_X \arrow[r, "T\pi_2"'] & T\mathcal{P}X
\end{tikzcd}$$

is a weak pullback. ($\star$) If $\mathcal{P}$ is replaced by $\mathcal{P}^+$, it is enough to assume that T is weakly epi-cartesian. Indeed, the function $\pi_2 : \in_X \to \mathcal{P}^+X$ is surjective, so weak-epi-cartesianness is enough to conclude that the square is a weak pullback.

By the properties of r stated in (19.10), and by the definition of weak pullback, there is some $z \in Z$ such that $\eta_{\in_X}(z, Z) = r$. By naturality of η on $\pi_1 : \in_X \to X$, we obtain

$$\eta_X(z) = \eta_X(\pi_1(z, Z)) = T\pi_1(\eta_{\in_X}(z, Z)) = T\pi_1(r) = s$$

as required. This completes the proof of (19.9).

For the axiom (d), we need to prove that

$$\lambda_X(\mu_{\mathcal{P}X}(t)) = \mathcal{P}\mu_X(\lambda_{TX}(T\lambda_X(t))) \qquad \text{for } t \in TT\mathcal{P}X. \tag{19.12}$$

For the left-hand side, unfold from (19.8):

$$s \in \lambda_X(\mu_{\mathcal{P}X}(t)) \iff \exists r \in T(\in_X) \text{ s.t. } T\pi_1(r) = s,\ T\pi_2(r) = \mu_{\mathcal{P}X}(t). \tag{19.13}$$

For the right-hand side:

$$
\begin{aligned}
s \in \mathcal{P}\mu_X(\lambda_{TX}(T\lambda_X(t))) &\iff \exists u \in \lambda_{TX}(T\lambda_X(t)) \text{ s.t. } \mu_X(u) = s \\
&\iff \exists v \in T(\in_{TX}) \text{ s.t. } \mu_X(T\pi_1(v)) = s,\ T\pi_2(v) = T\lambda_X(t).
\end{aligned}
\tag{19.14}
$$

For the left-to-right inclusion of (19.12), assume r as in (19.13). Since T is weakly cartesian, the square

$$
\begin{tikzcd}
TT\in_X \arrow[r, "TT\pi_2"] \arrow[d, "\mu_{\in_X}"'] & TT\mathcal{P}X \arrow[d, "\mu_{\mathcal{P}X}"] \\
T\in_X \arrow[r, "T\pi_2"'] & T\mathcal{P}X
\end{tikzcd}
$$

is a weak pullback. ($\star$) As before, if $\mathcal{P}$ is replaced by $\mathcal{P}^+$, it is enough to assume that T is weakly epi-cartesian since the function $\pi_2 : \in_X \to \mathcal{P}^+X$ is surjective.

By the properties of r stated in (19.13), and by the definition of weak pullback, there is some $w \in TT\in_X$ such that

$$\mu_{\in_X}(w) = r \qquad \text{and} \qquad TT\pi_2(w) = t.$$

Consider a function $\kappa : T\in_X \to \in_{TX}$ defined by:

$$\kappa(u) = (T\pi_1(u), \lambda_X(T\pi_2(u))).$$

This is well defined: indeed, $u \in T\in_X$ itself witnesses that $T\pi_1(u) \in \lambda_X(T\pi_2(u))$.

Put $v = T\kappa(w) \in T(\in_{TX})$. Then, using naturality of μ, calculate:

$$\mu_X(T\pi_1(v)) = \mu_X(T\pi_1(T\kappa(w))) = \mu_X(TT\pi_1(w)) = T\pi_1(\mu_{\in_X}(w)) = T\pi_1(r) = s$$

and

$$T\pi_2(v) = T\pi_2(T\kappa(w)) = T\lambda_X(TT\pi_2(w)) = T\lambda_X(t)$$

as required in (19.14). For the right-to-left inclusion of (19.12), assume v as in (19.14). Consider the pullback

$$
\begin{tikzcd}
Q \arrow[r, "\pi_2"] \arrow[d, "\mathrm{id}\times\lambda_X"'] & TPX \arrow[d, "\lambda_X"] \\
\in_{TX} \arrow[r, "\pi_2"'] & PTX
\end{tikzcd}
$$

where

$$Q = \{(p, z) \mid p \in TX,\ z \in TPX,\ p \in \lambda_X(z)\}. \tag{19.15}$$

Unfolding the definition of λ_X in this, we obtain:

$$Q = \{(p, z) \mid p \in TX,\ z \in TPX,\ \exists m \in T{\in_X} \text{ s.t. } T\pi_1(m) = p,\ T\pi_2(m) = z\}.$$

This means that we can select a function, call it $\rho : Q \to T{\in_X}$, such that

$$T\pi_1(\rho(p, z)) = p \qquad \text{and} \qquad T\pi_2(\rho(p, z)) = z \qquad \text{for } (p, z) \in Q.$$

Since T preserves weak pullbacks, the square

$$\begin{tikzcd} TQ \arrow[r, "T\pi_2"] \arrow[d, "T(\mathrm{id}\times\lambda_X)"'] & TTPX \arrow[d, "T\lambda_X"] \\ T{\in_{TX}} \arrow[r, "T\pi_2"'] & TPTX \end{tikzcd}$$

is a weak pullback. By the properties of v stated in (19.14), and by the definition of weak pullback, there is some $q \in TQ$ such that

$$T(\mathrm{id} \times \lambda_X)(q) = v \quad (\text{hence } T\pi_1(q) = T\pi_1(v)) \qquad \text{and} \qquad T\pi_2(q) = t.$$

Put $r = \mu_{\in_X}(T\rho(q)) \in T{\in_X}$. Then, using naturality of μ, calculate:

$$T\pi_1(r) = T\pi_1(\mu_{\in_X}(T\rho(q))) = \mu_X(TT\pi_1(T\rho(q))) = \mu_X(T\pi_1(q)) = \mu_X(T\pi_1(v)) = s$$

and

$$T\pi_2(r) = T\pi_2(\mu_{\in_X}(T\rho(q))) = \mu_{\mathcal{P}X}(TT\pi_2(T\rho(q))) = \mu_{\mathcal{P}X}(T\pi_2(q)) = \mu_{\mathcal{P}X}(t)$$

as required in (19.13). This completes the proof of (19.12) and the entire theorem. □

Armed with λ, we are ready to prove Theorem 16. Let $L \subseteq TX$ be a language recognized by a finite T-algebra $\mathbf{A} = (A, \alpha)$ as $L = \overleftarrow{h}(C)$, where $h : TX \to \mathbf{A}$ is a homomorphism and and $C \subseteq A$.

Seeing λ as a distributive law of the monad T over the functor $\mathcal{P}^+$, we can lift the functor $\mathcal{P}^+ : \mathtt{Set} \to \mathtt{Set}$ to $\widehat{\mathcal{P}^+} : \mathtt{Alg}(T) \to \mathtt{Alg}(T)$, see Beck (1969). In particular, as is easy to check using the axioms from Definition 18,

$$\widehat{\mathcal{P}^+}\mathbf{A} = (\mathcal{P}^+ A, \mathcal{P}^+\alpha \circ \lambda_A)$$

is a legal T-algebra.

For a surjective function $f : X \to Y$, define $\overline{f} : Y \to \mathcal{P}^+ X$ as the inverse image function on Y:

$$\overline{f}(y) = \{x \in X \mid f(x) = y\}.$$

Let $g : X \to Y$ be a surjective function. Since all functors on $\mathtt{Set}$ preserve surjectivity, $Tg : TX \to TY$ is also surjective. We will show that the direct image of L along Tg is recognized by the algebra $\widehat{\mathcal{P}^+}\mathbf{A}$. To this end, consider the function $k : TY \to \mathcal{P}^+ A$ defined by the composition:

$$k \quad = \quad TY \xrightarrow{\overline{Tg}} \mathcal{P}^+ TX \xrightarrow{\mathcal{P}^+ h} \mathcal{P}^+ A$$

and the subset $\hat{C} \subseteq \mathcal{P}^+ A$ defined by

$$\hat{C} = \{D \subseteq A \mid D \cap C \neq \emptyset\}.$$

A straightforward calculation shows that the direct image of L along Tg coincides with the inverse image $\overleftarrow{k}(\hat{C})$.

To finish the proof it is enough to show that k is a T-algebra morphism. That is, we need to show that the outer shape of the following diagram commutes:

$$\begin{tikzcd}
TTY \arrow[r, "T\overline{Tg}"] \arrow[dr, "\overline{TTg}"'] \arrow[dd, "\mu_Y"'] & T\mathcal{P}^+TX \arrow[r, "T\mathcal{P}^+h"] \arrow[d, "\lambda_{TX}"] \arrow[dr, phantom, "(3)"] & T\mathcal{P}^+A \arrow[d, "\lambda_A"] \\
 & \mathcal{P}^+TTX \arrow[r, "\mathcal{P}^+Th"] \arrow[d, "\mathcal{P}^+\mu_X"] \arrow[dr, phantom, "(4)"] & \mathcal{P}^+TA \arrow[d, "\mathcal{P}^+\alpha"] \\
TY \arrow[r, "\overline{Tg}"'] \arrow[ur, phantom, "(2)"] & \mathcal{P}^+TX \arrow[r, "\mathcal{P}^+h"'] & \mathcal{P}^+A
\end{tikzcd}$$

Part (3) commutes by naturality of λ and part (4) commutes since h is a T-algebra homomorphism. We now show the commutativity of (1) and (2).

For (2), given $s \in TTY$, we have:

$$\mathcal{P}^+\mu_X(\overline{TTg}(s)) = \{\mu_X(r) \mid r \in TTX \text{ and } TTg(r) = s\},$$
$$\overline{Tg}(\mu_Y(s)) = \{t \in TX \mid \mu_Y(s) = Tg(t)\}.$$

To prove $\mathcal{P}^+\mu_X(\overline{TTg}(s)) = \overline{Tg}(\mu_Y(s))$ we consider both inclusions.

For the left-to-right inclusion, put $t := \mu_X(r)$ and use naturality of μ. For the right-to-left inclusion, use the fact that, since T is weakly epi-cartesian,

$$\begin{tikzcd} TTX \arrow[r, "TTg"] \arrow[d, "\mu_X"'] & TTY \arrow[d, "\mu_Y"] \\ TX \arrow[r, "Tg"'] & TY \end{tikzcd}$$

is a weak pullback.

To prove that (1) commutes, we will use the following simple fact:

Lemma 23 *Let $f : X \to Y$ be a surjective function. Then:*

(i) $\eta^{\mathcal{P}}_X \subseteq \bar{f} \circ f$ and $\bar{f}$ is the minimum function from Y to $\mathcal{P}^+X$ with this property,
(ii) $\mathcal{P}^+ f \circ \bar{f} \subseteq \eta^{\mathcal{P}}_Y$ and $\bar{f}$ is the maximum function from Y to $\mathcal{P}^+X$ with this property. □

Using this together with Proposition 21 and the axiom (a) from Definition 18 for λ, we obtain:

$$\begin{aligned} \eta^{\mathcal{P}}_{TX} \subseteq \overline{Tg} \circ Tg \;&\Rightarrow\; \eta^{\mathcal{P}}_{TTX} = \lambda_{TX} \circ T\eta^{\mathcal{P}}_{TX} \subseteq \lambda_{TX} \circ T\overline{Tg} \circ TTg \\ &\Rightarrow\; \overline{TTg} \subseteq \lambda_{TX} \circ T\overline{Tg} \end{aligned}$$

Similarly, we obtain that:

$$\begin{aligned} \mathcal{P}^+Tg \circ \overline{Tg} \subseteq \eta^{\mathcal{P}}_{TY} \;&\Rightarrow\; \lambda_{TY} \circ T\mathcal{P}^+Tg \circ T\overline{Tg} \subseteq \lambda_{TY} \circ T\eta^{\mathcal{P}}_{TY} = \eta^{\mathcal{P}}_{TTY} \\ &\Rightarrow\; \mathcal{P}^+TTg \circ \lambda_{TX} \circ T\overline{Tg} \subseteq \eta^{\mathcal{P}}_{TTY} \\ &\Rightarrow\; \lambda_{TX} \circ T\overline{Tg} \subseteq \overline{TTg} \end{aligned}$$

which completes the proof that (1) commutes, and the proof of the entire theorem. □

19.6 Other Examples

In this section, we show a few more cases where Rec_T is closed under direct images along surjective letter-to-letter homomorphisms. These cases do not satisfy any of the sufficient conditions presented so far. They are also quite varied, with no clear pattern emerging:

- the reader monad in Sect. 19.6.1 is infinitary, but its finite algebras are rather restricted, and preservation under direct images is proved by a compactness argument,

- for the free lattice monad (Sect. 19.6.2), preservation follows from a convexity-based argument,
- the monad in Sect. 19.6.3 is rather peculiar in that almost no language is recognizable for it, so preservation of recognizable languages under direct images holds for trivial reasons,
- for the monad in Sect. 19.6.4, a construction similar to that described in Sect. 19.5 works, but with the powerset replaced by a more complex powerset-squared construction.

19.6.1 The Reader Monad

Monad 21 Let Σ^ω denote the set of ω-sequences of letters from Σ. The functor $T = (-)^\omega$ acts on functions in the expected way:

$$f^+(a_0 a_1 a_2 \cdots) = (f(a_0))(f(a_1))(f(a_2)) \cdots .$$

This is an infinitary monad, with the unit mapping a letter $a \in \Sigma$ to the constant sequence $aaa \cdots$, and the multiplication defined by the diagonal function:

$$\mu_\Sigma(w_0 w_1 w_2 \cdots) = w_{0,0} w_{1,1} w_{2,2} \cdots$$

This is a particular instance of the Haskell "reader" monad.

We will prove that recognizable T-languages are closed under taking direct images along surjective letter-to-letter homomorphisms. To fix the notation, we choose an arbitrary surjective function $f : \Sigma \to \Gamma$ for Σ, Γ finite, and any language $L \subseteq T\Sigma$ recognizable by a homorphism $h : T\Sigma \to \mathbf{A}$ to a finite T-algebra $\mathbf{A} = (A, \alpha)$. This means that for some subset $S \subseteq A$:

$$t \in L \iff h(t) \in S.$$

Assuming all this, we shall prove that the direct image of L along Tf is recognizable by a homomorphism from $T\Gamma$ to a finite T-algebra.

Without loss of generality we may assume that L is recognizable by a point in A, i.e. that $S = \{a\}$ for some $a \in A$. This is because both taking inverse images and taking direct images commutes with unions, and recognizable languages are closed under finite unions. Such languages have a rather rigid "rectangular" structure:

Lemma 24 *If $L \subseteq T\Sigma$ is recognizable by a point then*

$$L = Z_1 \times Z_2 \times Z_3 \times \cdots \tag{19.16}$$

for some sequence of subsets $Z_1, Z_2, \ldots \subseteq \Sigma$.

Proof Given a language L recognizable by $a \in A$ for an algebra $\alpha : TA \to A$, define

$$Z_n = \{x \in \Sigma \mid \exists v \in L. v_n = x\}.$$

for each $n \in \mathbb{N}$. Then the left-to-right inclusion in (19.16) is obvious. For the right-to-left inclusion, consider a word $w \in Z_1 \times Z_2 \times \cdots$; this means that for every $n \in \mathbb{N}$ there is a word $v_n \in L$ such that $v_{n,n} = w_n$. Then $w = \mu_\Sigma(v_1 v_2 \cdots)$, so:

$$h(w) = h(\mu_\Sigma(v_1 v_2 \cdots)) = \alpha(Th(v_1 v_2 \cdots)) = \alpha(aaa \cdots) = a$$

so $w \in L$. □

For any Σ, the set $T\Sigma$ is equipped with the product topology, whose basis is the family of all *open balls* defined by

$$\mathcal{B}_I(w) = \{v \in T\Sigma \mid \forall n \in I.\ v_n = w_n\}$$

for $w \in T\Sigma$ and finite $I \subseteq \mathbb{N}$. With respect to this topology:

Lemma 25 *Every L recognizable by a point is closed.*

Proof From Lemma 24 it easily follows that $T\Sigma \setminus L$ is open. Indeed, for L as in (19.16), take any $w \notin L$. Then there is an $n \in \mathbb{N}$ such that $w_n \notin Z_n$, and then $\mathcal{B}_{\{n\}}(w)$ is disjoint from L. □

Lemma 26 *Every L recognizable by a point in a finite algebra A is open.*

Proof Let $L = \overleftarrow{f}\{a\}$ for some $a \in A$. Then

$$T\Sigma \setminus L = \bigcup_{b \in A \setminus \{a\}} \overleftarrow{f}\{b\},$$

each $\overleftarrow{f}\{b\}$ is closed by Lemma 25, and their union is closed since a finite union of closed sets is closed. □

We can now prove a stronger version of Lemma 24. Call a language $L \in T\Sigma$ *finitely rectangular* if

$$L = Z_1 \times Z_2 \times Z_3 \times \cdots \tag{19.17}$$

for some sequence of subsets $Z_1, Z_2, \ldots \subseteq X$ such that $Z_n = \Sigma$ for all except finitely many n.

Lemma 27 *Every nonempty L recognizable by a point in a finite algebra is finitely rectangular.*

Proof By Lemma 24, L is of the form

$$L = Z_1 \times Z_2 \times Z_3 \times \cdots ;$$

we need to prove that $Z_n = \Sigma$ for almost all n.

Since Σ is a finite set, the space $T\Sigma$ is compact. Its subspace L is closed by Lemma 25, so it is also compact. Consider the family of all those open balls $\mathcal{B}_I(w)$ that are contained in L. Since L is open by Lemma 26, for each $w \in L$ there is some I such that $\mathcal{B}_I(w)$ belongs to this family, so the family is an open cover of L. By compactness, it contains a finite subcover

$$\mathcal{B}_{I_1}(w_1), \ldots, \mathcal{B}_{I_k}(w_k). \tag{19.18}$$

Let $I = I_1 \cup \cdots \cup I_k$. Fix any $w \in L$ and take any $v \in T\Sigma$ such that $w_n = v_n$ for all $n \in I$. Since L is covered by (19.18) we have $w \in \mathcal{B}_{I_i}(w_i)$ for some i. Since $I_i \subseteq I$ also $v \in \mathcal{B}_{I_i}(w_i)$, so $v \in L$. Since we did not constrain v on the coordinates outside of I, we have that $Z_n = \Sigma$ for $n \notin I$. □

Finitely rectangular languages are closed under direct images of surjective homomorphisms:

Lemma 28 *For a surjective function $f : \Sigma \to \Gamma$, if $L \subseteq T\Sigma$ is finitely rectangular then so is the direct image $\overrightarrow{Tf}(L) \subseteq T\Gamma$.*

Proof Obviously

$$\overrightarrow{Tf}(Z_1 \times Z_2 \times \cdots) = \overrightarrow{f}(Z_1) \times \overrightarrow{f}(Z_2) \times \cdots ,$$

and $\overrightarrow{f}(\Sigma) = \Gamma$ since f is surjective. □

The final piece of the puzzle is:

Lemma 29 *Every finitely rectangular language is recognizable by a finite algebra.*

Proof Let $L \subseteq T\Sigma$ be as in (19.17), and let $I \subseteq \mathbb{N}$ be the (finite) set of those n where $Z_n \neq \Sigma$. Define $A = \Sigma^I$, and let $\alpha : TA \to A$ be defined by:

$$\alpha(a_1 a_2 \cdots)(n) = a_n(n) \qquad \text{for } n \in I.$$

We need to check that α is a T-algebra. For the unit axiom, given $a \in A$ and $n \in I$, calculate:

$$\alpha(\eta_A(a))(n) = \alpha(aaa\cdots)(n) = a(n).$$

For the multiplication axiom, given $\mathbf{v} = v_1 v_2 v_3 \cdots \in TTA$ and $n \in I$, calculate:

$$\alpha(\mu_A(\mathbf{v}))(n) = \alpha(v_{1,1} v_{2,2} \cdots)(n) = v_{n,n}(n)$$

and

$$\alpha(T\alpha(\mathbf{v}))(n) = \alpha(\alpha(v_1), \alpha(v_2), \ldots)(n) = \alpha(v_n)(n) = v_{n,n}(n)$$

so both expressions are equal as required, hence α is a T-algebra structure on A.

Define a homomorphism $h : T\Sigma \to A$ as a unique homomorphic extension of the function $h_0 : \Sigma \to A$ that maps every $x \in \Sigma$ to the function in Σ^I constant at x. Explicitly, this is defined by:

$$h(w)(n) = \alpha(Th_0(w))(n) = w_n \qquad \text{for } w \in T\Sigma, n \in I.$$

Recalling (19.17), define $S \subseteq A$ by:

$$S = \{g \in \Sigma^I \mid \forall n \in I.\ g(n) \in Z_n\}.$$

Then, for $w \in T\Sigma$, calculate:

$$w \in L \iff \forall n \in I. w_n \in Z_n \iff h(w) \in S,$$

hence L is recognized by $S \subseteq A$ along the homomorphism h. □

19.6.2 The Free Lattice Monad

Monad 22 The free lattice monad **FL**, whose algebras are lattices, has a well-known equational presentation over a signature with two binary symbols $\vee$ and $\wedge$. It consists of equations:

$$\begin{array}{lll} x \vee y = y \vee x & x \vee (y \vee z) = (x \vee y) \vee z & x \vee (x \wedge y) = x \\ x \wedge y = y \wedge x & x \wedge (y \wedge z) = (x \wedge y) \wedge z & x \wedge (x \vee y) = x \end{array}$$

Lattices can be usefully regarded as partial orders, defined by:

$$x \leq y \qquad \text{iff} \qquad x \vee y = y;$$

$\vee$ and $\wedge$ then become respectively join and meet operations.

We will prove that recognizable **FL**-languages are closed under taking direct images along surjective letter-to-letter homomorphisms. So, to fix the notation, we choose an arbitrary surjective function $f : \Sigma \to \Gamma$ for Σ, Γ finite, and any language $L \subseteq \mathbf{FL}(\Sigma)$ recognizable by a homorphism $h : \mathbf{FL}(\Sigma) \to A$ to a finite lattice A. This means that for some subset $S \subseteq A$:

$$t \in L \iff h(t) \in S.$$

Assuming all this, we shall prove that $\overrightarrow{Tf}(L)$, the direct image of L along Tf, is recognizable by a homomorphism from $\mathbf{FL}(\Gamma)$ to a finite lattice.

First let us consider the case where L is upwards-closed, i.e., where

$$s \in L \quad \text{and} \quad s \leq t \quad \text{implies} \quad t \in L.$$

Consider a homomorphism $k : \mathbf{FL}(\Gamma) \to \mathbf{FL}(\Sigma)$ defined as the unique extension of the map $k_0 : \Gamma \to \mathbf{FL}(\Sigma)$ defined by:

$$k_0(y) = \bigvee \{x \mid f(x) = y\}.$$

Note that the above join is nonempty since f is surjective. In particular, we have $f(k_0(y)) = y$ for all $y \in \Gamma$ and, by definition, $k_0(f(x)) \geq x$ for all $x \in \Sigma$. Since Tf and k preserve meets and joins by definition, this extends to:

$$Tf(k(t)) = t \qquad \text{and} \qquad k(Tf(s)) \geq s \tag{19.19}$$

for all $t \in \mathbf{FL}(\Gamma)$ and $s \in \mathbf{FL}(\Sigma)$.

Then it is easy to see that

$$\overrightarrow{Tf}(L) = \overleftarrow{k}(L). \tag{19.20}$$

Indeed, for the right-to-left inclusion, for $k(t) \in L$ one has

$$t \overset{(19)}{=} Tf(k(t)) \in \overrightarrow{Tf}(L).$$

For the left-to-right inclusion, take any $t \in \mathbf{FL}(\Gamma)$ such that $t = Tf(s)$ for some $s \in L$. Then

$$k(t) = k(Tf(s)) \overset{(19)}{\geq} s$$

and, since L is upwards-closed, $k(t) \in L$.

Equation (19.20) implies that $\overrightarrow{Tf}(L)$ is the inverse image of $S \subseteq A$ along the composite homomorphism $h \circ k : \mathbf{FL}(Y) \to A$, and so it is recognizable.

An analogous argument works if L is downwards-closed, with k defined using a meet rather than a join.

Now consider an arbitrary (i.e. not necessarily upwards- or downwards-closed) language L recognizable by a homomorphism $h : \mathbf{FL}(\Sigma) \to A$ to a finite lattice A. Without loss of generality we may assume that L is recognizable by a point in A, i.e., that

$$L = \overleftarrow{h}\{a\}$$

for some $a \in A$. This is because both taking inverse images and taking direct images commutes with unions, and recognizable languages are closed under finite unions.

Let $L{\uparrow}$ and $L{\downarrow}$ denote the upwards-closure and the downwards-closure of L, respectively. It is not difficult to check that upwards closure commutes with inverse images, in particular:

$$\overleftarrow{h}\,(a{\uparrow}) = L{\uparrow}.$$

Indeed, this amounts to requiring that, for every $t \in \mathbf{FL}(\Sigma)$,

$$h(t) \geq a \iff \exists s \leq t. h(s) = a.$$

The right-to-left implication is immediate since h is monotone. For the left-to-right implication, since $h : \mathbf{FL}(\Sigma) \to A$ is surjective, there is some $r \in \mathbf{FL}(\Sigma)$ such that $h(r) = a$. Put $s = t \wedge r$. Then obviously $s \leq t$, and

$$h(s) = h(t) \wedge h(r) = h(t) \wedge a = a$$

as required.

An analogous argument works for downward closures, i.e.:

$$\overleftarrow{h}\,(a{\downarrow}) = L{\downarrow}.$$

This means that both $L{\uparrow}$ and $L{\downarrow}$ are recognizable (by h), hence by the previous argument, their direct images along Tf are also recognizable. Since recognizable languages are closed under intersection, it is now enough to prove that

$$\overrightarrow{Tf}(L{\uparrow}) \cap \overrightarrow{Tf}(L{\downarrow}) = \overrightarrow{Tf}(L). \tag{19.21}$$

To this end, first notice that, since L is recognized by a point, it is *convex*, i.e.,

$$s \leq t \leq r \quad \text{and} \quad s, r \in L \quad \text{implies} \quad t \in L.$$

This implies that

$$L{\uparrow} \cap L{\downarrow} = L;$$

indeed the right-to-left inclusion is trivial, and the left-to-right inclusion easily follows from the convexity of L. This does not immediately imply (19.21), as taking direct images does not commute with intersections in general. In this case, however, it does. To see this, first notice that $L{\uparrow}$ is a *filter*, i.e., it is upward-closed and closed under finite intersections. Similarly, $L{\downarrow}$ is an *ideal*, i.e., it is downward-closed and closed under finite unions. Now all we need is the following lemma, which holds for arbitrary lattice homomorphisms:

Lemma 30 *For any lattice homomorphism $g : H \to K$ and any filter $U \subseteq H$ and ideal $D \subseteq H$ such that $U \cap D \neq \emptyset$:*

$$\overrightarrow{g}\,(U \cap D) = \overrightarrow{g}\,(U) \cap \overrightarrow{g}\,(D).$$

Proof The left-to-right inclusion is obvious. For the right-to-left inclusion, take any $t \in K$ such that

$$\exists u \in U.h(u) = t \quad \text{and} \quad \exists d \in D.h(d) = t.$$

Pick any $x \in U \cap D$ (it exists by our assumptions), and put:

$$y = d \vee (x \wedge u).$$

Since $x \in D$ and D is downward-closed, also $x \wedge u \in D$ and (since D is closed under unions) $y \in D$. On the other hand, since $x \in U$ and U is closed under intersections, also $x \wedge u \in U$ and (since U is upward-closed) $y \in U$. So $y \in D \cap U$, and it is enough to calculate:

$$h(y) = h(d) \vee (h(x) \wedge h(u)) = t \vee (h(x) \wedge t) = t.$$

□

Using this lemma for $g = Tf$, $U = L\uparrow$ and $D = L\downarrow$, we obtain (19.21) which directly implies that L is recognizable.

19.6.3 A Non-trivial Monad Whose Finite Algebras are Trivial

Monad 23 Let T be the monad associated to the signature $\{f, g\}$, where f and g are unary operations, subject to axioms:

$$fgfgg(x) = x \quad \text{and} \quad fgffgg(x) = fgffgg(y).$$

It is shown that this equational theory has no non-trivial finite models, that is, every non–empty finite model has only one element Burris (1971, Sect. 2). Indeed, if (A, f, g) is a finite algebra, from the equation $fgfgg(x) = x$ we obtain that f is a surjection and g is a injection, which implies that f and g are bijections since A is finite and $f : A \to A$ and $g : A \to A$. Hence, from the equation $fgffgg(x) = fgffgg(y)$, we conclude that $x = y$ for every $x, y \in A$. Therefore, Rec_T is trivially closed under direct images along surjective letter–to–letter homomorphisms.

Remark Even though the monad has the trivial algebra as the only finite model, the monad has infinite models. One such model is the set of the positive integers with $g(n) = 2^n$, $f(2^{2^n}) = 3^n$, $f(2^{3^n}) = n$, and $f(m) = 1$ otherwise, as shown in Burris (1971, Sect. 2). Also, it is worth mentioning that given a finite set of identities over a given finitary signature, it is undecidable if there exists a non-trivial finite model that satisfies the given identities (Mckenzie, 1975). That is, it is undecidable in general if the class Rec_T of recognizable languages is non-trivial. This means that describing the class Rec_T or even finding a non–trivial element in Rec_T could be undecidable.

19.6.4 The $x(y(yz)) = x(yz)$ Monad

Monad 24 Consider a monad T presented by an equational theory with a single binary symbol $\bullet$ and with a single equation:

$$\begin{array}{ccccc} & \bullet & & & \\ \diagup & & \diagdown & & \\ x & & \bullet & & \\ & \diagup & & \diagdown & \\ & y & & \bullet & \\ & & \diagup & & \diagdown \\ & & y & & z \end{array} \quad = \quad \begin{array}{ccccc} & \bullet & & & \\ \diagup & & \diagdown & & \\ x & & \bullet & & \\ & \diagup & & \diagdown & \\ & y & & z & \end{array} . \tag{19.22}$$

This equation is similar to the one that defined Monad 18, but the resulting monad has rather different properties. For one, its multiplication is not weakly epi-cartesian. To see this, consider

$$\Sigma = \{a_1, a_2, b, c\} \qquad \Gamma = \{a, b, c\}$$

and $f : \Sigma \to \Gamma$ defined by $f(a_1) = f(a_2) = a$, $f(b) = b$ and $f(c) = c$. Define $t \in TT\Gamma$ by:

$$t = \begin{array}{ccc} & \bullet & \\ \diagup & & \diagdown \\ b & & t' \end{array} \qquad \text{where} \qquad t' = \begin{array}{ccc} & \bullet & \\ \diagup & & \diagdown \\ a & & c \end{array} \in T\Gamma \quad \text{and } b \text{ is seen as } \eta_\Gamma(b) \in T\Gamma.$$

Then, obviously:

$$\mu_Y(t) = \begin{array}{cccc} & \bullet & & \\ \diagup & & \diagdown & \\ b & & \bullet & \\ & \diagup & & \diagdown \\ & a & & c. \end{array}$$

Now let $r \in T\Sigma$ be defined by:

$$r = \begin{array}{ccccc} & \bullet & & & \\ \diagup & & \diagdown & & \\ b & & \bullet & & \\ & \diagup & & \diagdown & \\ & a_1 & & \bullet & \\ & & \diagup & & \diagdown \\ & & a_2 & & c. \end{array}$$

It is easy to see that $Tf(r) = \mu_Y(t)$. However, there is no $p \in TT\Sigma$ such that $TTf(p) = t$ and $\mu_\Sigma(p) = r$. As a result, the naturality square

$$\begin{tikzcd} TT\Sigma \arrow[r, "\mu_\Sigma"] \arrow[d, "TTf"'] & T\Sigma \arrow[d, "Tf"] \\ TT\Gamma \arrow[r, "\mu_\Gamma"'] & T\Gamma \end{tikzcd}$$

is not a weak pullback.

In spite of this, recognizable T-languages are closed under taking direct images along surjective letter-to-letter homomorphisms.

A T-algebra is simply a set equipped with a binary operation which satisfies (19.22). Assume finite alphabets Σ and Γ, a surjective function $f : \Sigma \to \Gamma$, and a language $L \subseteq T\Sigma$ that is recognized by a subset $S \subseteq A$ of a finite T-algebra A; in other words, $L = \overleftarrow{h}(S)$ for some T-algebra homomorphism $h : T\Sigma \to A$. We will show that $\overrightarrow{Tf}(L)$ is recognized by a finite T-algebra.

To this end, define a T-algebra

$$\bar{A} = \mathcal{P}^+ A \times \mathcal{P}^+ A.$$

The intuition is that a tree $t \in T\Gamma$ will be mapped to a value $(\alpha_L, \alpha_R) \in \bar{A}$ such that α_L contains those values in A that can be attained by trees in $T\Sigma$ which are mapped to t and which are left sons of their parents. Similarly, α_R will stores values for trees that are right sons of their parents.

Formally, a binary operation $\cdot$ is defined by:

$$\begin{aligned}(\alpha_L, \alpha_R) \cdot (\beta_L, \beta_R) = \big(&\{a \cdot b \mid a \in \alpha_L, b \in \beta_R\},\\ &\{a_1 \cdot (a_2 \cdot (\cdots (a_n \cdot b) \cdot\cdot)) \mid n \geq 1, a_i \in \alpha_L, b \in \beta_R\}\big),\end{aligned}$$

where $\cdot$ on the right-hand side of the definition denotes the operation in A. It is not difficult to check that this defines a T-algebra, i.e., that the equation

$$(\alpha_L, \alpha_R) \cdot \big((\beta_L, \beta_R) \cdot ((\beta_L, \beta_R) \cdot (\gamma_L, \gamma_R))\big) = (\alpha_L, \alpha_R) \cdot \big((\beta_L, \beta_R) \cdot (\gamma_L, \gamma_R)\big) \tag{19.23}$$

holds for all $\alpha_L, \alpha_R, \beta_L, \beta_R, \gamma_L, \gamma_R \subseteq A$. The (slightly) more involved case is the right-to-left inclusion, where one needs to use the Eq. (19.22) for A. For the left-to-right inclusion even that is not needed.

A homomorphism $\bar{h} : T\Gamma \to \bar{A}$ is the unique extension of the function that maps every letter $a \in \Gamma$ to

$$\Big(\overrightarrow{h \circ \eta_\Sigma}(\overleftarrow{f}(a)), \overrightarrow{h \circ \eta_\Sigma}(\overleftarrow{f}(a))\Big)$$

(note that $h \circ \eta_\Sigma : \Sigma \to A$ is the interpretation of single letters from Σ in the algebra A).

Now define a subset $\bar{S} \subseteq \bar{A}$ by:

$$\bar{S} = \{(\alpha_L, \alpha_R) \mid \alpha_L \cap S \neq \emptyset\}.$$

This subset recognizes the direct image $\overrightarrow{Tf}(L)$. More explicitly, for any tree $t \in T\Gamma$, one has $\bar{h}(t) \in \bar{S}$ if and only if $t = Tf(r)$ for some $r \in T\Sigma$ such that $h(r) \in S$. Both implications are proved by a straightforward induction on the size of t.

19.7 Counterexamples

In this section, we illustrate cases where direct images of recognizable languages along surjective letter-to-letter homomorphisms are not recognizable. One such case, Monad 7, was shown in Sect. 19.3. Here we provide three more.

19.7.1 The Marked Words Monad

Recall Monad 2, i.e. the free semigroup monad $(-)^+$, and Monad 6, i.e., the bag monad $\mathcal{B}$. For $w \in \Sigma^+$ and $\beta \in \mathcal{B}\Sigma$, we write $w \sqsupseteq \beta$ if each $x \in \Sigma$ occurs in w at least $\beta(x)$ times.

Monad 25 Define a functor T on `Set` by:

$$T\Sigma = \{(w, \beta) \in \Sigma^+ \times \mathcal{B}\Sigma \mid w \sqsupseteq \beta\}$$

with the action on fuctions defined componentwise. It is easy to check that this is well-defined, i.e., that

$$w \sqsupseteq \beta \text{ implies } f^+(w) \sqsupseteq \mathcal{B}f(\beta).$$

T carries a monad structure where both unit and multiplication are componentwise inherited from the monads $(-)^+$ and $\mathcal{B}$. Again, it is straightforward to check that this is well defined.

To get some intuition, an element of $T\Sigma$ can be understood as a finite, nonempty word over the alphabet Σ with some positions marked, except that it is not specified exactly which positions those are; we only know how many positions labeled with every letter are marked. So, for example, markings

$$\underline{aa}bca\underline{b} \qquad \underline{a}a\underline{b}c\underline{a}b \qquad \text{and} \qquad a\underline{ab}c\underline{a}b$$

denote the same element of $T\{a, b, c\}$. This intuition is useful because it allows for a simple presentation of the monad structure: the unit of $a \in \Sigma$ is unambiguosly presented as $\underline{a}$ and multiplication can be illustrated as, for example:

$$\underline{(\underline{a}bac)}(a\underline{b}\underline{b})(ab\underline{a}c)\underline{(b\underline{a}c)} \quad \mapsto \quad \underline{a}bacabbabacb\underline{a}c.$$

This monad has a simple equational presentation over an algebraic signature that consists of one binary symbol $\bullet$ and one unary symbol $\circ$. Using the terminology of marked words, the intuition will be that $\bullet$ is an operation that concatenates two words, and $\circ$ *erases* the marking from every letter in a word. The equations are as follows:

$$(x \bullet y) \bullet z = x \bullet (y \bullet z), \qquad \circ(x \bullet y) = \circ x \bullet \circ y,$$

$$\circ\circ x = \circ x, \qquad (x \bullet y) \bullet \circ x = \circ x \bullet (y \bullet x)$$

The first three equations ensure that every element of $T\Sigma$ can be presented as a marked word over Σ, and the last equation characterizes the equivalence of marked words described above.

This completes our exhibition of the monad T.

Now, for the alphabet $\Sigma = \{a, b\}$, consider the language $L \subseteq T\Sigma$ defined by:

$$L = \{\big((ab)^n, [a \mapsto n, b \mapsto 0]\big) \mid n \in \mathbb{N}\}.$$

Using the presentation by marked words, L consists of all words of the form

$$\underline{a}b\underline{a}b \cdots \underline{a}b.$$

We claim that L is recognizable by a finite T-algebra. Intuitively, this is because a word $w \in L$ cannot be obtained by concatenating or erasing marks from any other marked words, except by concatenating subwords of w or perhaps erasing a mark from a single letter $\underline{b}$.

The algebra to recognize L has seven values:

$$A = \{\mathsf{AA}, \mathsf{AB}, \mathsf{BA}, \mathsf{BB}, \mathsf{B}, \underline{\mathsf{B}}, \bot\}.$$

Intuitively, the value BA represents those sub-words of words from L that begin with b and end with $\underline{a}$; analogously for AA, AB and BB, except that the latter value represents only words other than the singleton b. That word is represented by a separate value B. The value $\underline{\mathsf{B}}$ represents a singleton marked letter $\underline{b}$. Finally, $\bot$ is an error value, representing those words that cannot be completed to a word in L by erasing marks from all letters or by concatenating with other words.

Relying on the equational presentation of T, to define a T-algebra on A it is enough to define operations $\bullet$ and $\circ$ on it. The multiplication table for $x \bullet y$ is:

$x \bullet y$	AA	AB	BA	BB	B	$\underline{\mathsf{B}}$	$\perp$
AA	$\perp$	$\perp$	AA	AB	AB	$\perp$	$\perp$
AB	AA	AB	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$
BA	$\perp$	$\perp$	BA	BB	BB	$\perp$	$\perp$
BB	BA	BB	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$
B	BA	BB	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$
$\underline{\mathsf{B}}$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$
$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$

Note that B and BB behave in the same way in this table, and similarly for $\underline{\mathsf{B}}$ and $\perp$. However, these pairs of values are distinguished by the operation $\circ$, which is defined by:

$$\circ(\mathsf{B}) = \circ(\underline{\mathsf{B}}) = \mathsf{B} \qquad \text{and} \qquad \circ(x) = \perp \quad \text{for } x \notin \{\mathsf{B}, \underline{\mathsf{B}}\}.$$

It is easy to check that the operations defined this way satisfy the equational theory of T given above, therefore they make A a T-algebra.

Since $T\Sigma$ is a free T-algebra on Σ, any function from Σ to the set A extends uniquely to a homomorphism from $T\Sigma$ to the algebra A. Let h be the homomorphism that extends the function mapping a to AA and b to $\underline{\mathsf{B}}$. Then we have, for every $w \in T\Sigma$:

$$w \in L \iff h(w) = \mathsf{AB}$$

so L is recognized by A.

Now consider an alphabet $Y = \{c\}$ and the unique function $f : \Sigma \to \Gamma$, which maps both a and b to c. The direct image of L under Tf is:

$$\left\{\left(c^{2n}, [c \mapsto n]\right) \mid n \in \mathbb{N}\right\}. \tag{19.24}$$

Using the presentation by marked words, this language consists of all words of the form

$$\underline{c}c\underline{c}c \cdots \underline{c}c = \overbrace{\underline{cc \cdots cc}}^{n \text{ times}}\underbrace{cc \cdots cc}_{n \text{ times}}.$$

This language is not recognizable by any finite T-algebra. For assume any homomorphism h from $T\Gamma$ to some finite algebra A, and consider $w_1, w_2, w_3, \ldots \in T\Gamma$ defined by:

$$w_n = (c^n, [c \mapsto n]) = \underline{c}^n.$$

Since A is finite, there are some $n \neq m$ such that $h(w_n) = h(w_m)$. Now consider

$$v = (c^n, [c \mapsto 0]) = c^n \in T\Gamma.$$

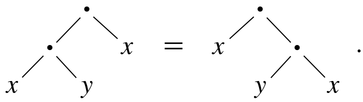
We have that

$$h(\underline{w_n}v) = h(\underline{w_m}v)$$

but $\underline{w_n}v$ belongs to the direct image (19.24) and $\underline{w_m}v$ does not, so h does not recognize the direct image. □

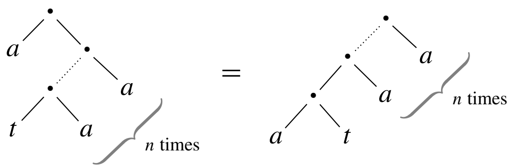
19.7.2 The Balanced Associativity Monad

Monad 26 Consider a monad T presented by a single binary symbol • and a single equation:



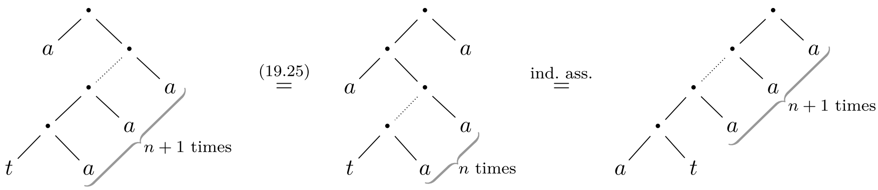
(19.25)

Elements of $T\Sigma$ do not have a canonical form as simple and intuitive as for the marked words monad, but we shall need only a limited understanding of them. First, notice that for any $a \in \Sigma$, $t \in T\Sigma$ and $n \geq 1$ there is:

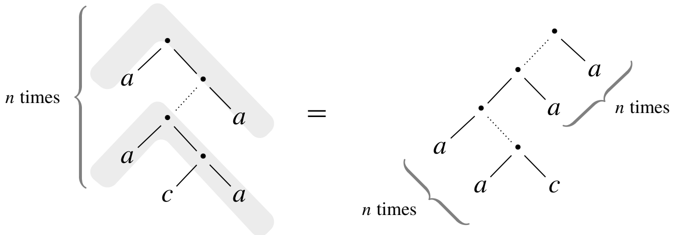


(19.26)

This is proved by simple induction on n. The base case $n = 1$ is simply (19.25) with $x = a$ and $y = t$, and for the induction step calculate:

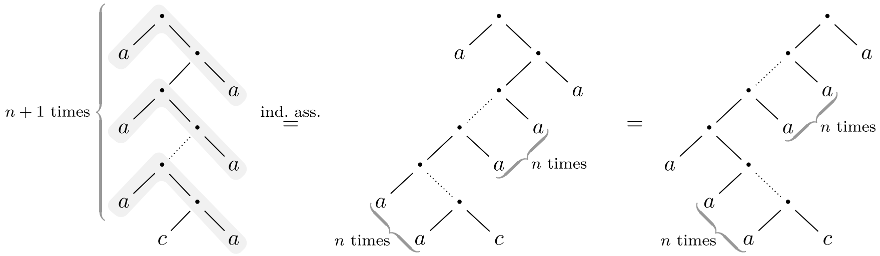


Furthermore, for any $a, c \in \Sigma$ and $n \geq 1$ we have:

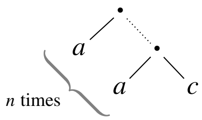


(19.27)

(on the left, the shaded pattern is repeated n times). This is again proved by induction. The base case $n = 1$ is (19.25) with $x = a$ and $y = c$, and for the induction step calculate:

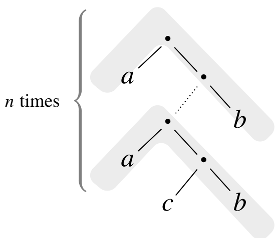


where the last equality follows from (19.26) with



$$= \quad t.$$

Now, for the alphabet $\Sigma = \{a, b, c\}$, consider the language $L \subseteq T\Sigma$ that consists of all trees of the form



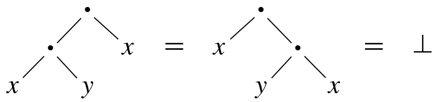
for $n \geq 1$. Note that the equation (19.25) does not apply anywhere in such a tree, so every tree of this shape forms a singleton equivalence class with respect to the congruence induced by (19.25). It is easy to recognize L with an algebra of six values:

$$A = \{\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{L}, \mathsf{R}, \perp\},$$

with the operation $\cdot$ interpreted by:

$$\mathsf{c} \cdot \mathsf{b} = \mathsf{R} \qquad \mathsf{a} \cdot \mathsf{R} = \mathsf{L} \qquad \mathsf{L} \cdot \mathsf{b} = \mathsf{R}$$

and $x \cdot y = \perp$ for all other combinations of $x, y \in A$. It is easy to check that equation (19.25) holds in this algebra, indeed



for all $x, y \in A$.

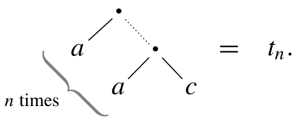
Then the unique homomorphism from $h : T\Sigma \to A$ that maps a to a, b to b and c to c recognizes L:

$$t \in L \iff h(t) = \mathsf{L}.$$

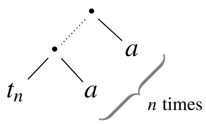
Now consider an alphabet $\Gamma = \{a, c\}$ and a function $f : \Sigma \to \Gamma$:

$$f(a) = f(b) = a, \qquad f(c) = c.$$

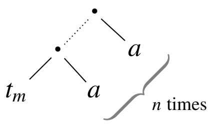
The direct image of L under Tf consists of words as in the Eq. (19.27). This language is not recognizable by any finite T-algebra. For assume any homomorphism from $T\Gamma$ to some finite algebra A, and consider $t_1, t_2, t_3, \ldots \in T\Gamma$ defined by:



Since A is finite, there are some $n \neq m$ such that $h(t_n) = h(t_m)$. Then the terms



and



have the same value under h, but the one on the left is in the direct image of L along Tf (due to (19.27)) and the one on the right is not, so h does not recognize the direct image.

19.7.3 The Not-Quite-Mal'cev Monad

Recall Monad 12, presented by a ternary operation symbol p and a unary operation symbol s subject to the equations

$$p(x, x, y) = s(y) = p(y, x, x). \tag{19.28}$$

If, additionally, there were $s(y) = y$, then the monad would immediately have a Mal'cev term and the machinery of Sect. 19.4.2 would apply.[3]

Note that any algebra for a monad that has a Mal'cev term induces a T-algebra on the same set. For instance, if $(G, \cdot, e)$ is a group, then (G, p, s), where $p(x, y, z) := x \cdot y^{-1} \cdot z$ and s is the identity function, is a T-algebra. A finite T-algebra that does not have a Mal'cev term is $P = (\{0, 1, 2, 3, 4, 5, 6\}, p, s)$ where p is defined as:

$$p(x, y, z) = \begin{cases} 1 & \text{if } x \neq 5,\ y = 5 \text{ and } z \neq 5 \\ 2 & \text{if } x \neq 6,\ y = 6 \text{ and } z \neq 6 \\ 3 & \text{if } (x, y, z) = (0, 1, 2), \\ 4 & \text{otherwise,} \end{cases}$$

and with $s(x) = 4$ for all x. The fact that (19.28) hold in P can be easily checked by hand.

Now, consider the alphabet $\Sigma = \{a, b, c\}$, the homomorphism $h : T\Sigma \to P$ such that $h(a) = 0$, $h(b) = 5$ and $h(c) = 6$, and the subset $S = \{3\} \subseteq P$. The recognizable language $L = \overleftarrow{h}(S)$ is the infinite language:

$$L = \{p(a, p(x_1, b, x_2), p(x_3, c, x_4)) \mid x_i \in T\Sigma, x_1, x_2 \neq b, x_3, x_4 \neq c\}.$$

Now consider $\Gamma = \{a, d\}$ and $f : \Sigma \to \Gamma$ defined by $f(b) = f(c) = d$ and $f(a) = a$. We show that that the direct image of L under the surjective letter-to-letter homomorphism Tf is not recognizable. To see this, note that the direct image contains (the equivalence class of) the term $s(a)$, because

$$s(a) = p(a, p(d, d, d), p(d, d, d)) = Tg(p(a, p(c, b, c), p(b, c, b))).$$

On the other hand, if $t_1 \neq t_2 \in T\Gamma$ then the direct image of L does not contain the term

$$p(s(t_1), s(t_2), a).$$

Since $T\Gamma$ is infinite, for any homomorphism g from $T\Gamma$ into a finite T-algebra A there are some terms $t_1 \neq t_2$ such that $g(t_1) = g(t_2)$. Then the terms

[3] We are grateful to an anonymous reviewer whose insightful question led to a significant improvement of this example.

$$s(a) = p(s(t_1), s(t_1), a) \qquad \text{and} \qquad p(s(t_1), s(t_2), a)$$

are mapped to the same value by g, but only the former one belongs to the direct image of L along Tf.

19.8 Related Problems

Our main focus has been on recognizable languages being preserved under taking direct images of surjective letter-to-letter homomorphisms. This is because, in the proof of Proposition 1, which concerns the particular example of the monad of finite words, direct images are taken only along projection homomorphisms that arise from the logical existential quantifier; those homomorphisms are indeed surjective and letter-to-letter. It therefore makes sense to incorporate this particular closure property in an abstract definition of MSO for other monads.

However, it should be noted that Proposition 1 would remain true if surjective letter-to-letter homomorphisms were replaced in its statement by arbitrary letter-to-letter, or indeed by arbitrary homomorphisms. The corresponding closure property would become stronger, so one may expect some monads to become non-examples for these more restrictive variants of monadic MSO. In this section we identify some of these monads.

19.8.1 Homomorphisms That are Not Letter-to-Letter

First, let us consider preserving recognizable languages under taking direct images of *arbitrary* T-algebra homomorphisms, i.e. ones that arise from functions $f : \Sigma \to T\Gamma$ via Kleisli lifting (as opposed to letter-to-letter homomorphims, which are of the form $Tf : T\Sigma \to T\Gamma$ for a map $f : \Sigma \to \Gamma$). This preservation property fails even for very basic monads, such as commutative monoids and seminearrings:

Example 31 Recall Monad 6, i.e., the free commutative monoid monad $\mathcal{B}$. The language $a^* \subseteq \mathcal{B}\{a, b\}$ is recognized by the commutative monoid $(\{0, 1\}, \max, 0)$ via the homomorphism $h : \mathcal{B}\{a, b\} \to \{0, 1\}$ such that $h(a) = 0$ and $h(b) = 1$. Now, consider the homomorphism $h : \mathcal{B}\{a, b\} \to \mathcal{B}\{a, b, c\}$ such that $h(a) = ab$ and $h(b) = c$. Then, the direct image of a^* is the language $\{w \in \mathcal{B}\{a, b, c\} \mid w(a) = w(b)\}$ which is not recognizable. □

Another counterexample is:

Monad 27 A *seminearring* is a set equipped with two monoids, i.e., associative binary operations $+$ and $\cdot$ (called *horizontal* and *vertical* composition; the latter is usually denoted simply by juxtaposition) with units respectively 0 and 1, subject to additional axioms

$$(x + y)z = xz + yz \qquad \text{and} \qquad 0x = 0.$$

As expected, the free seminearring monad T is presented by the equational theory of seminearrings. This is a rather fundamental monad from the perspective of logic: in (Bojańczyk, 2023, Sect. 6) it is explained how seminearrings are a reasonable choice for an algebra of trees. Indeed, the free seminearring over a set Σ is the set of all unranked forests with holes (called *multicontexts* in Bojańczyk, 2023).

Consider the language

$$L = ab^* = \{ab^n \mid n \in \mathbb{N}\} \subseteq T\{a, b\}.$$

This language is recognized by a finite seminearring (with 5 elements). Now consider the homomorphism $h : T\{a, b\} \to T\{a, b\}$ with $h(a) = a + a$ and $h(b) = b$. The direct image of L along h is the language

$$\{ab^n + ab^n \mid n \in \mathbb{N}\}$$

which is not recognizable. Indeed, for any seminearring homomorphism $f : T\{a, b\} \to S$ for a finite S, there must be some $n \neq m$ such that $f(b^n) = f(b^m)$; then $f(ab^n) = f(ab^m)$ and

$$f(ab^n + ab^n) = f(ab^n + ab^m).$$

19.8.2 Letter-to-Letter Homomorphisms That are Not Surjective

One may also consider the preservation of recognizable languages under direct images along arbitrary (i.e. not necessarily surjective) letter-to-letter homomorphisms. In all our counterexamples in Sect. 19.7, the failure of preservation seems to be caused by different letters being identified by a homomorphism; in other words, non-injectivity of homomorphisms seems to be the key issue. One may wonder if additionally requiring preservation under non-surjective homomorphisms changes the picture at all. As it turns out, it does.

Example 32 Recall Monad 21, the reader monad from Sect. 19.6.1. Note that $T\{1\} = \{1^\omega\} = \{111111\cdots\}$. Consider the "full" language $L = T\{1\} \subseteq T\{1\}$; it is clearly recognizable by the unique homomorphism to the one-element T-algebra. However, its direct image along the inclusion $\iota : T\{1\} \to T\{0, 1\}$:

$$\overrightarrow{T\iota}(L) = \{1^\omega\} \subseteq T\{0, 1\}$$

is not recognizable. Indeed, it is not finitely rectangular according to the terminology of Sect. 19.6.1 (and, being a singleton, it is not a union of finitely rectangular sets either), so its recognizability would contradict Lemma 27. □

Example 33 Recall Monad 23, the monad whose finite models are trivial from Sect. 19.6.3. If we consider the inclusion $\iota : \{x\} \to \{x, y\}$ then the direct image of the recognizable language $L = T(\{x\})$ under $T\iota$ is not recognizable. Indeed, the only recognizable languages are $\emptyset$ and TX for any finite X since the only finite model is the trivial model, but L is not one of those languages. □

Example 34 Recall Monad 22, the free lattice monad **FL** from Sect. 19.6.2. We shall consult the first chapter of Freese et al. (1995) for basic facts about free lattices. First, the following theorem due to Whitman (1941) provides an effective procedure for comparing two elements of a free lattice:

Theorem 35 (Freese et al. (1995), Theorem 1.8) *For any elements $a, b, c, d \in$ $\mathbf{FL}(\Sigma)$, $a \wedge b \leq c \vee d$ if and only if:*

$$a \leq c \vee d, \quad b \leq c \vee d, \quad a \wedge b \leq c, \quad \textit{or} \quad a \wedge b \leq d.$$

Writing down terms to denote elements of free lattices, one usually implicitly applies the associativity laws and writes e.g. $t_1 \wedge t_2 \wedge t_3$ instead of $(t_1 \wedge t_2) \wedge t_3$. With this convention in mind, the following canonical form of terms is considered:

Definition 36 In a free lattice $\mathbf{FL}(\Sigma)$, a formal join $t = t_1 \vee \cdots \vee t_n$ with $n > 1$ is in canonical form if:

- each t_i is in Σ or a formal meet in canonical form,
- $t_i \not\leq t_j$ for all $i \neq j$,
- if $t_i = \bigwedge t_{ij}$ then $t_{ij} \not\leq t$ for all j.

Dual conditions define the canonical form of formal meets.

By Freese et al. (1995, Theorem 1.17–18), every element in $\mathbf{FL}(\Sigma)$ can be presented by a term in canonical form, and moreover this presentation is unique up to commutativity laws.

It will be important to us that Definition 36 provides a procedure for transforming a term t into canonical form, and that this procedure relies only on removing some redundant subterms from t. As a corollary, if an element in $a \in \mathbf{FL}(\Sigma)$ is presented by a term t in canonical form such that some $w \in \Sigma$ appears in t, then a cannot be presented by any term where w does not appear. Indeed, if such a presentation t' existed then the canonical form of t' would not contain w either, which would contradict the uniqueness of canonical presentation.

The last basic fact about free lattices that we shall need is that for $|\Sigma| \geq 3$ the lattice $\mathbf{FL}(\Sigma)$ is infinite and it contains a strictly increasing infinite chain (see Freese et al., 1995, Ex. 1.24).

Now let

$$\Sigma = \{p, q, r\} \qquad \Gamma = \{p, q, r, s\}$$

with $\iota : \Sigma \to \Gamma$ the inclusion function. The language $L = T\Sigma \subseteq T\Sigma$ is recognizable by the unique homomorphism to the one-element lattice. We shall show that its direct image $\overrightarrow{T\iota}(L) \subseteq T\Gamma$ is not recognizable.

To this end, for any pair of terms $t_1 < t_2 \in \overrightarrow{T\iota}(L)$, consider the term

$$t = (t_1 \vee s) \wedge t_2.$$

Using Theorem 35, it is easy to check that s is incomparable with both t_1 and t_2, and that $t_1 < t < t_2$. Moreover, if t_1 and t_2 are in canonical form then so is t; to prove this use again Theorem 35 and the fact that s is incomparable with t_1, t_2 and all their subterms. As a result, t cannot be presented by any term where the generator s does not appear; in other words, $t \notin \overrightarrow{T\iota}(L)$.

We have just proved that between any two distinct but ordered elements of $\overrightarrow{T\iota}(L)$ there is some element not in $\overrightarrow{T\iota}(L)$. Recall that $\mathbf{FL}(\Sigma)$ contains a strictly increasing infinite chain, and since $T\iota$ is an injective function, the set $\overrightarrow{T\iota}(L)$ also contains a strictly increasing infinite chain. Inserting an element out of $\overrightarrow{T\iota}(L)$ between each two neighbouring elements in that chain, we obtain an infinite, strictly increasing chain that alternates between elements that are in and out of $\overrightarrow{T\iota}(L)$. This implies that the set $\overrightarrow{T\iota}(L)$ is not recognizable.

It should be mentioned that for a weakly cartesian monad T, the class Rec_T is in fact closed under direct images of all (not necessarily surjective) letter-to-letter homomorphisms. The proof of this is the same as in Sect. 19.5.3 but with the full powerset functor $\mathcal{P}$ used instead of $\mathcal{P}^+$. Only two points are worth making in this case: (1) the "projection" $\pi_2 : \in_X \to \mathcal{P}X$ is no longer surjective (the empty set is not in the image) and (2) the empty set plays the role of an "error state" in a powerset algebra (every letter that is not in the image of the function g is mapped to the empty set and any "operation" that involves the empty set has the empty set as a result).

19.9 Future Work

Technically, our main object of study in this paper was a notion: a monad for which languages recognizable by finite algebras are closed under taking direct images along surjective letter-to-letter homomorphisms. For such monads the abstract definition of monadic MSO makes sense, in that it only describes recognizable languages. Our various sufficient conditions, examples and counterexamples show that the notion seems rather subtle, but they do not provide a full characterization of it. The search for such a characterization is an obvious direction of further study.

It should also be said that the notion itself is a result of a few design decisions. It could be that its variants, such as the ones described in Sect. 19.8, lead to simpler characterizations while still covering essentially the same class of practically relevant examples. Some other design decisions we have not even mentioned: for example, is an arbitrary finite algebra the right notion of a recognizing device? Sometimes, in particular for infinitary monads, the answer might not be obvious, and one may want to consider a restricted class of finite algebras instead. (Note that, for an infinitary monad, a finite algebra may not admit a finite description.) Similarly, it is not clear whether one should not restrict even further the class of homomorphisms to take direct images along. This design space deserves careful exploration.

We focussed our attention on monads on the category of sets, but this leaves several natural examples out of scope. For example, the category of ranked sets is a convenient setting to study various algebras of trees and graphs. Other interesting base categories include that of vector spaces, or that of nominal sets. Another interesting idea is to employ the topo-algebraic technology of Gehrke et al. (2016, 2017), where recognizability of word languages is studied beyond the regular setting. In general, any category where a meaningful notion of direct image of a language can be formulated, may be a territory worth exploring.

The most ambitious goal is to go beyond the "definable $\Rightarrow$ recognizable" implication of the correspondence between MSO definability and recognizability of languages. Is there a generic version of the converse implication? How far can Proposition 1 be generalized? What about decidability questions about MSO? The paper Bojańczyk (2015) makes some initial steps in these and other related directions. However, much remains to be done, all in the spirit of bridging the divide between Structure and Power.

References

Abramsky, S., & Shah, N. (2018). Relating structure and power: Comonadic semantics for computational resources. In *27th EACSL Annual Conference on Computer Science Logic (CSL 2018)*. Leibniz International Proceedings in Informatics (LIPIcs) (Vol. 119, pp. 2:1–2:17).

Awodey, S. (2010). *Category Theory* (2nd ed.). New York: Oxford University Press Inc.

Beck, J. (1969). Distributive laws. In B. Eckmann (Ed.), *Seminar on triples and categorical homology theory* (pp. 119–140). Berlin: Springer.

Bojańczyk, M. (2015). Recognisable languages over monads. arXiv:1502.04898.

Bojańczyk, M., & Klin, B. (2019). A non-regular language of infinite trees that is recognizable by a sort-wise finite algebra. *Logical Methods in Computer Science,15*(4).

Bojańczyk, M., & Pilipczuk, M. (2016). Definability equals recognizability for graphs of bounded treewidth. In M. Grohe, E. Koskinen, & N. Shankar (Eds.), *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5–8, 2016* (pp. 407–416). ACM.

Bojańczyk, M., Grohe, M., & Pilipczuk, M. (2018). Definable decompositions for graphs of bounded linear cliquewidth. In A. Dawar, & E. Grädel (Eds.), *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, Oxford, UK, July 09–12, 2018* (pp. 135–144). ACM.

Bojańczyk, M. (2021) .*Algebra for trees*. In J.-É Pin (Eds.), *Handbook of automata theory* (pp. 801–838). Zürich, Switzerland: European Mathematical Society Publishing House. https://doi.org/10.4171/Automata-1/22

Brown, T., & Lazerson, E. (2009). On finitely generated idempotent semigroups. *Semigroup Forum, 78*(1), 183–186.

Brzozowski, J. (1980). Open problems about regular languages. In *Formal Language Theory* (pp. 23–47). Elsevier.

Buchi, J. R. (1990a). On a Decision Method in Restricted Second Order Arithmetic. In S. Mac Lane & D. Siefkes (Eds.), *The Collected Works of J. Richard Buchi* (pp. 425–435). New York: Springer New York.

Buchi, J. R. (1990b). Weak second-order arithmetic and finite automata. In S. Mac Lane & D. Siefkes (Eds.), *The collected works of J. Richard Buchi* (pp. 398–424). New York: Springer.

Burris, S., & Sankappanavar, H. P. (1981). *A course in universal algebra*. Graduate texts in mathematics. Berlin: Springer.

Burris, S. (1971). Models in equational theories of unary algebras. *Algebra Universalis, 1*(1), 386–392.

Carton, O., Colcombet, T., & Puppis, G. (2011). Regular languages of words over countable linear orderings. In L. Aceto, M. Henzinger, & J. Sgall (Eds.), *Automata, languages and programming*. Lecture notes in computer science (pp. 125–136). Berlin: Springer.

Clementino, M. M., Hofmann, D., & Janelidze, G. (2014). The monads of classical algebra are seldom weakly cartesian. *Journal of Homotopy and Related Structures, 9*(1), 175–197.

Courcelle, B. (1990). The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and Computation,85*(1), 12–75.

Courcelle, B., & Engelfriet, J. (2012). *Graph structure and monadic second-order logic - A language-theoretic approach*. Encyclopedia of mathematics and its applications (Vol. 138). Cambridge: Cambridge University Press.

Doner, J. (1970). Tree acceptors and some of their applications. *Journal of Computer and System Sciences, 4*(5), 406–451.

Eilenberg, S., & Wright, J. B. (1967). Automata in general algebras. *Information and Control, 11*(4), 452–470.

Elgot, C. C. (1961). Decision problems of finite automata design and related arithmetics. *Transactions of the American Mathematical Society, 98*(1), 21–51.

Freese, R. S., Ježek, J., & Nation, J. B. (1995). Free Lattices. *Advances in the Mathematical Sciences. American Mathematical Society*.

Fritz, T., & Perrone, P. (2018). Monads, partial evaluations, and rewriting. arXiv:1810.06037.

Gehrke, M., Petrisan, D., & Reggio, L. (2016). The Schützenberger product for syntactic spaces. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11–15, 2016, Rome, Italy*. LIPIcs (Vol. 55, pp. 112:1–112:14).

Gehrke, M., Petrisan, D., & Reggio, L. (2017). Quantifiers on languages and codensity monads. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20–23, 2017* (pp. 1–12). IEEE Computer Society.

Green, J. A., & Rees, D. (1952). On semi-groups in which xr = x. *Mathematical Proceedings of the Cambridge Philosophical Society, 48*(1), 35–40.

Hutchinson, G. (1994). Relation categories and coproduct congruence categories in universal algebra. *Algebra Universalis, 32*(4), 609–647.

Jacobs, B. (2004). Trace semantics for coalgebras. *Electronic Notes in Theoretical Computer Science,106*, 167–184. Proceedings of the Workshop on Coalgebraic Methods in Computer Science (CMCS).

Klin, B., & Salamanca, J. (2018). Iterated covariant powerset is not a monad. *Electronic Notes in Theoretical Computer Science,341*, 261–276. Proceedings of the MFPS XXXIV.

MacLane, S. (1971). *Categories for the working mathematician*. Graduate texts in mathematics (Vol. 5). New York: Springer.

Mckenzie, R. (1975). On spectra, and the negative solution of the decision problem for identities having a finite nontrivial model. *The Journal of Symbolic Logic, 40*(2), 186–196.

Perrin, D., & Pin, J.-É. (2004). *Infinite words: Automata, semigroups, logic and games* (Vol. 141). Academic.

Pin, J.-É. (2017). *Open problems about regular languages, 35 years later*, Chap. 7 (pp. 153–175).

Rabin, M. O. (1969). Decidability of second-order theories and automata on infinite trees. *Transactions of the American Mathematical Society, 141,* 1–35.

Rispal, C., & Carton, O. (2005). Complementation of rational sets on countable scattered linear orderings. *International Journal of Foundations of Computer Science, 16*(04), 767–786.

Shelah, S. (1975). The monadic theory of order. *Annals of Mathematics,* 379–419.

Steinby, M. (1998). General varieties of tree languages. *Theoretical Computer Science, 205*(1), 1–43.

Szawiel, S., & Zawadowski, M. (2014). Theories of analytic monads. *Mathematical Structures in Computer Science,24*(6).

Thatcher, J. W., & Wright, J. B. (1968). Generalized finite automata theory with an application to a decision problem of second-order logic. *Mathematical Systems Theory, 2*(1), 57–81.

Thomas, W. (1997). *Languages, automata, and logic* (pp. 389–455). Berlin: Springer.

Trakhtenbrot, B. A. (1961). Finite automata and logic of monadic predicates. *Doklady Akademii Nauk SSSR, 140*(326–329), 122–123.

Weber, M. (2004). Generic morphisms, parametric representations and weakly cartesian monads. *Theory and Applications of Categories, 13*(14), 191–234.

Whitman, P. M. (1941). Free lattices. *Annals of Mathematics, 42*(1), 325–330.

Zwart, M., & Marsden, D. (2019). No-go theorems for distributive laws. In *2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)* (pp. 1–13).

Chapter 20
The Strategic Balance of Games in Logic

Jouko Väänänen

Abstract Truth, consistency and elementary equivalence can all be characterised in terms of games, namely the so-called evaluation game, the model-existence game, and the Ehrenfeucht–Fraisse game. We point out the great affinity of these games to each other and call this phenomenon the *strategic balance in logic*. In particular, we give explicit translations of strategies from one game to another.

Keywords Evaluation game · Model-existence game · Ehrenfeucht–Fraisse game · Game theoretic semantics

20.1 The Three Games of Logic

The Game Theoretical Semantics of first order logic, and many other logics too, is a combination of three games: The Evaluation Game, the Model Existence Game and the EF game (short for Ehrenfeucht–Fraïssé Game). These games are closely related to each other and cover the main semantic concepts of logic. We use the term "Game Theoretical Semantics" for the general approach that emphasises and utilizes these games in semantics, rather than some other approach, e.g. set-theoretical semantics in the Tarskian sense.

This paper is written in response to a question by Samson Abramsky, about where in my book Väänänen (2011) are the promised translations between strategies of the various games that I present. To my surprise I had not given the translations explicitly in the book, perhaps because in the book I move freely from games to inductive definitions and back. In this paper the translations are given explicitly and I thank Samson for pointing out the omission in my book.

This project has received funding from the Academy of Finland (grant No 322795) and from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 101020762).

J. Väänänen (✉)
University of Helsinki, Helsinki, Finland
e-mail: jouko.vaananen@helsinki.fi

University of Amsterdam, Amsterdam, The Netherlands

A. Palmigiano and M. Sadrzadeh (eds.), *Samson Abramsky on Logic and Structure in Computer Science and Beyond*, Outstanding Contributions to Logic 25,
https://doi.org/10.1007/978-3-031-24117-8_20

In the Evaluation Game we have a structure A and a sentence ϕ. The game gives meaning to the proposition that ϕ is true in A. The meaning is equivalent to the one given by Tarski's Turth Definition. In the Model Existence Game the structure A is missing and we have just the sentence ϕ. The game gives meaning to the proposition that ϕ is consistent, or alternatively, that there is a structure in which ϕ is true. This game appears in different incarnations throughout logic, all the way from Gentzen's Natural Deduction, to Beth Tableaux, to Dialogical Games, to Hintikka's model sets, to semantic trees, to Smullyan's consistency properties. Finally, in the EF game the sentence ϕ is missing and we just have two models A and B. The game gives meaning to the proposition that some sentence is true in A but false in B. Alternatively we can say that the EF game gives meaning to, or equivalently a definition of the elementary equivalence of A and B.

The meanings that these games give to truth, consistency, and elementary equivalence are intertranslatable in the sense that a strategy in one can be converted to a strategy in the other (see Fig. 20.1). If the former is a winning strategy, so is the latter. We need not traverse via Tarski's Truth Definition. In this sense Game Theoretical Semantics stands autonomously on its own feet. This is important for those who find Tarski's approach too set-theoretical. However, the set-theoretical treatment is arguably the best representation of even the Game Theoretical Semantics, despite the intuitive appeal of the latter, when one goes into the gritty details of complicated proofs.

The three games, Evaluation Game, Model Existence Game, and EF game, penetrate into the heart of logic where we find the concepts of truth, consistency, provability, definability and isomorphism. Their mathematical interpretations and elaborations lead us into the deepest questions of mathematical logic, especially model theory and set theory. They provide tools for understanding central questions of

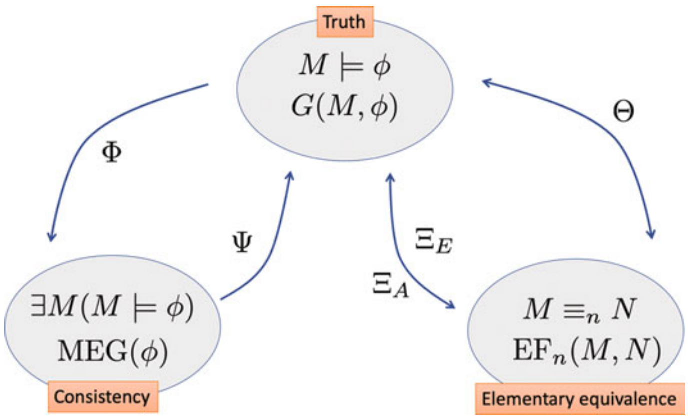


Fig. 20.1 The translations of strategies

philosophical logic such as intuitionistic, modal and other non-classical logic. They have become indispensable in theoretical computer science modelling interaction and complexity.

For more details and for references to the original sources we refer to Väänänen (2011). Everything in this paper was essentially known already in the fifties and sixties. What is perhaps new is the emphasis on the mutual relationships between the three games, taking the form of a collection of translations between strategies. This aspect has been forcefully emphasised in van Benthem (2014). The author feels that despite the ripe age of the main results, this topic has more to offer than has been discovered so far. An interesting recent development is Abramsky et al. (2017).

Notation: If X is a set, an X-assignment is a mapping with a set of variables as its domain and values in X. If s is an X-assignment and $a \in X$, then $s(a/x)$ is the X-assignment which agrees with s except that it gives x the value a. If X is clear from the context, we drop it and talk about assignments only.

20.2 Given Both a Sentence and a Model: The Evaluation Game

We describe a game which gives meaning to the *truth* of a given sentence in a given model. In its simplest form this game, the Evaluation Game, is the following: Suppose M is a structure for a vocabulary L, ϕ is a first order (or, more generally, infinitary) L-sentence and s is an assignment. We allow identity. Let $\Gamma(s)$ denote the set of all literals i.e. atomic and negated atomic formulas that s satisfies in M. As the logical operations allowed in ϕ we include $\neg$, $\wedge$, $\vee$, $\forall$ and $\exists$. To make the situation as clear as possible, we assume that ϕ is in Negation Normal Form, meaning that negation occurs only in front of atomic formulas. The game $G(M, \phi)$ has two players Abelard and Eloise. Intuitively, Eloise defends the proposition that ϕ is (informally) true in M and Abelard doubts it. All through the game the players are inspecting an L-sentence and an assignment s. We call the pair (ψ, s) a *position* of the game. In the beginning of the game the position is $(\phi, \emptyset)$. The rules are as follows: Suppose the position is (ψ, s).

(1) If ψ is a literal, the game ends and Eloise wins if $\psi \in \Gamma(s)$. Otherwise Abelard wins.
(2) If ψ is $\psi_0 \wedge \psi_1$, then Abelard chooses whether the next position is (ψ_0, s) or (ψ_1, s).
(3) If ψ is $\psi_0 \vee \psi_1$, then Eloise chooses whether the next position is (ψ_0, s) or (ψ_1, s).
(4) If ψ is $\forall x\theta$, then Abelard chooses $a \in M$ and the next position is $(\theta, s(a/x))$.
(5) If ψ is $\exists x\theta$, then Eloise chooses $a \in M$ and the next position is $(\theta, s(a/x))$.

A *strategy* for a player is a set of rules (functions) that describe exactly how that player should choose, depending on how the two players have played at earlier moves. It is a *winning strategy* if the player wins whichever way the opponent plays. The game $G(M, \phi)$ is clearly a determined game i.e. always one of the players has a winning strategy.

We say that ϕ is *true in M* if Eloise has a winning strategy in $G(M, \phi)$. This is the game-theoretical meaning of truth in a model. We can go further and say that the *game* $G(M, \phi)$ is the *meaning* of ϕ in M. Here meaning would be a broader concept than the mere truth or falsity of ϕ.

The first to formulate the Evaluation Game explicitly was probably Hintikka (1968) who later advocated the importance and usefulness of the game forcefully. Hintikka mentions Wittgenstein (1953) as an inspiration. Earlier Henkin (1961) formulated a game theoretic approach to quantifiers pointing out generalizations to very general infinitary and partially ordered quantifiers. Nowadays the Evaluation Game is a standard tool in mathematical, philosophical and computer science logic.

The game $G(M, \phi)$ reflects the entire syntactical structure of ϕ in the sense that the game $G(M, \phi \wedge \psi)$ is intimately related to the two games $G(M, \phi)$ and $G(M, \psi)$, the same with $G(M, \phi \vee \psi)$, $G(M, \exists x\phi)$ and $G(M, \forall x\phi)$. This phenomenon is a manifestation of the broader concept of *compositionality*.

If ϕ is propositional i.e. has only zero-place relation symbols and no constant or function symbols, and no quantifiers, then only moves (1)–(3) occur in $G(M, \phi)$, and the assignments can be forgotten. If ϕ is universal, the game $G(M, \phi)$ has no moves of type (5). If it is existential, the game has no moves of type (4). If universal-existential, then all type (5) moves come before type (4) moves. Furthermore, if we add new logical operations to our logic, such as infinite conjunctions and disjunctions, generalized quantifiers or higher order quantifiers, it is clear how to modify the game $G(M, \phi)$ to accommodate the new logical operations. For example, for ϕ in $L_{\infty\omega}$, we modify above (2) and (3) as follows:

(2') If ψ is $\bigwedge_{i\in I} \psi_i$, then Abelard chooses $i \in I$ and the next position is (ψ_i, s).
(3') If ψ is $\bigvee_{i\in I} \psi_i$, then Eloise chooses $i \in I$ and the next position is (ψ_i, s).

This extends also to so-called team semantics (Väänänen, 2007), in which case a position in the game $G(M, \phi)$ is a pair (ψ, X), where ψ is a subformula of ϕ and X is a team. Finally, if M is a Kripke-model and ϕ a sentence of modal logic, the game $G(M, \phi)$ is entirely similar. The assignments have a singleton domain $\{x_0\}$ and values in the frame of M. The moves corresponding to $\Diamond$ and $\Box$ are like (4) and (5):

(4') If ψ is $\Box\theta$, then Abelard chooses a node b accessible from $s(x_0)$ and the next position is $(\theta, s(b/x_0))$.
(5') If ψ is $\Diamond x\theta$, then Eloise chooses a node b accessible from $s(x_0)$ and the next position is $(\theta, s(b/x_0))$.

The game $G(M, \phi)$ can also reflect the structure of M. In fact, the games $G(M \times N, \phi)$, $G(M + N, \phi)$, and $G(\Pi_i M_i/F, \phi)$ are intimately related to the games $G(M, \phi)$, $G(N, \phi)$ and $G(M_i, \phi)$ (Feferman, 1972). The game $G(M, \phi)$ is also helpful in finding a countable submodel N of M with desired properties. For any strategy τ of Eloise in $G(M, \phi)$ let $T(M, \tau)$ be the set of countable submodels N of M such that N is closed under the functions of M and under τ i.e.

if Abelard plays in (4) always $a \in N$, then also Eloise plays in (5) always $b \in N$. Note that if $N \in T(M, \tau)$, then τ is necessarily a strategy of Eloise in $G(N, \phi)$. Moreover, if τ is a winning strategy in $G(M, \phi)$, then it is also a winning strategy in $G(N, \phi)$. The **Löwenheim–Skolem Theorem** of $L_{\omega_1\omega}$ is essentially the statement that $T(M, \phi) \neq \emptyset$, when $\phi \in L_{\omega_1\omega}$. In the *Cub Game* due to Kueker (1977) there are two players Abelard and Eloise, a set X and a set T of countable subsets of X. During the game, which we denote G_T, the players choose alternatingly elements $a_n \in X$, Abelard being the first to move. After ω moves a set $\{a_0, a_1, \ldots\}$ has been produced. We say that Eloise is the winner if this set is in T, otherwise Abelard is the winner. This game need not be determined. If Eloise has a winning strategy in G_T, then T contains a so-called cub (closed unbounded) set of countable subsets of X.

Below is a strong form of a Löwenheim–Skolem Theorem for $L_{\omega_1\omega}$, based on the Evaluation Game. It shows that the countable submodels for which the given strategy works for Eloise are "everywhere" (i.e. cub) inside M. The theorem is due to Kueker (1977).

Theorem 1 *Suppose M is a model in a countable vocabulary and ϕ is a sentence of $L_{\omega_1\omega}$. There is a mapping Υ such that if τ is a strategy of Eloise in $G(M, \phi)$, then $\Upsilon(\tau)$ is a strategy of Eloise in $G_{T(M,\phi)}$. If τ is a winning strategy, then so is $\Upsilon(\tau)$.*

Proof (Sketch) Using bookkeeping Eloise makes sure during the game $G_{T(M,\phi)}$ that her moves $a_{2n+1} \in M$ render the final set $\{a_0, a_1, \ldots\}$ such that it is both the domain of a submodel of M and if Abelard plays his moves of type (4) in $\{a_0, a_1, \ldots\}$ then so does Eloise, using τ, in her moves of type (5). $\square$

In conclusion, the game $G(M, \phi)$ is a versatile tool for understanding the meaning of a logical sentence ϕ in a mathematical structure M. Alternative tools, yielding equivalent results, are inductive methods such as the so-called Tarski's Truth Definition and the theory of inductive definitions.

20.3 The Model is Missing: The Model Existence Game

Here we have a sentence and we ask whether the sentence has a model. Thus this is about *consistency* and its opposite, *contradiction*. In particular, the issue is whether there is some model M such that Eloise can win $G(M, \phi)$? This question is governed by the Model Existence Game MEG(ϕ).

Suppose ϕ is a first order sentence. As the logical operations allowed in ϕ we include $\neg$, $\wedge$, $\vee$, $\forall$ and $\exists$. To make this as simple as possible, we assume again that ϕ is in Negation Normal Form. The game MEG(ϕ) has two players Abelard and Eloise. Intuitively, Eloise defends the proposition that ϕ has a model and Abelard doubts it. Abelard expresses his doubt by asking questions. It may be the case that Eloise has a model at her disposal but she does not want to reveal it, apart from using it to give answers to Abelard's questions, but she may also just pretend that she does.

For all we know there may not be any model. Conceivably ϕ is a contradiction, but the game should settle this. We let $C = \{c_0, c_1, \ldots, c_n, \ldots\}$ be a set of new distinct constant symbols. Intuitively these are names of elements of the supposed model. A position of the game is a finite collection S of pairs (ψ, s), where ψ is a subformula of ϕ and s is a C-assignment. In infinitary logic we may consider also infinite S. In the beginning of the game the position is $\{(\phi, \emptyset)\}$. The rules are as follows: Suppose the position is S.

1. If $(\psi_0 \wedge \psi_1, s) \in S$, then Abelard may decide that the next position is $S \cup \{(\psi_0, s)\}$ or $S \cup \{(\psi_1, s)\}$.
2. If $(\psi_0 \vee \psi_1, s) \in S$, then Abelard may decide that Eloise has to choose whether the next position is $S \cup \{(\psi_0, s)\}$ or $S \cup \{(\psi_1, s)\}$.
3. If $(\forall x\theta, s) \in S$, then Abelard may choose $n \in \mathbb{N}$ and decide that the next position is $S \cup \{(\theta, s(c_n/x))\}$.
4. If $(\exists x\theta, s) \in S$, then Abelard may decide that Eloise has to choose c_n and the next position is $S \cup \{(\theta, s(c_n/x))\}$.

Abelard wins if at some point the position S contains both (ψ, s) and $(\neg\psi, s')$, where ψ is an atomic formula and $s(x) = s'(x)$ for all variables x in ψ. If Abelard does not win, then the game may continue for infinitely many moves and then Eloise is declared the winner. Note that Abelard has a lot of control on how the game proceeds as he may focus attention on any formula in the finite set s. This is clearly a determined game.

The Model Existence Game is a game-theoretic rendering of the method of Beth Tableaux, with background in Gentzen's natural deduction. It was presented roughly as above, but without explicit mention of a game, for first order logic at about the same time by Beth (1955) and Hintikka (1955), later in a more abstract form by Smullyan (1963), and for infinitary logic by Makkai (1969).

The following theorem, essentially due to Beth (1955), demonstrates how the Model Existence Game and the Evaluation Game can be combined in the somewhat trivial case that a model is not missing but actually known (see Fig. 20.2).

Theorem 2 *Suppose the vocabulary of M is countable. There is a mapping Φ so that if τ is a strategy of Eloise in $G(M, \phi)$, then $\Phi(\tau)$ is a strategy of Eloise in MEG(ϕ). If τ is a winning strategy, then so is $\Phi(\tau)$.*

Proof By Theorem 1 there is a countable submodel N of M such that τ is a strategy of Eloise in $G(N, \phi)$ and if τ is a winning strategy in $G(M, \phi)$, then it is also a winning strategy in $G(N, \phi)$. Let $\pi : C \to N$ be an onto map. Let us say that a pair (ψ, s) is a τ-position if there is there is some sequence of positions in $G(N, \phi)$, following

$$\begin{array}{cccccc} & C & \text{——} & N \subseteq & M & \\ \Phi(\tau) & \Big| & & & \Big| & \tau \\ & \phi & & & \phi & \end{array}$$

Fig. 20.2 From model to model existence the rules of $G(N, \phi)$ starting with $(\phi, \emptyset)$, Eloise using τ, which ends at (ψ, s). A C-*translation* of the τ-position (ψ, s) is a pair (ψ, s') where s' is a C-assignment with $\pi(s'(x)) = s(x)$. The strategy $\Phi(\tau)$ of Eloise in MEG(ϕ) is to make sure that at all times the position S consists only of C-translations of τ-positions. Let us now check that Eloise can actually follow this strategy: Suppose the position is S.

1. If $(\psi_0 \wedge \psi_1, s') \in S$, then Abelard may decide that the next position is $S \cup \{\psi_0, s'\}$ or $S \cup \{\psi_1, s'\}$. Let us assume he chooses $S \cup \{\psi_0, s')\}$ as the next position. There is a sequence α of moves in $G(N, \phi)$ in which Eloise uses τ, which ends at $(\psi_0 \wedge \psi_1, s)$, a C-translation of which is $(\psi_0 \wedge \psi_1, s')$. We continue α by letting Abelard move ψ_0. Now $S \cup \{(\psi_0, s')\}$ consists of C-translations of τ-positions.
2. If $(\psi_0 \vee \psi_1, s') \in S$, then Eloise can decide that the next position is $S \cup \{(\psi_0, s')\}$ or $S \cup \{(\psi_1, s')\}$. There is a sequence α of moves in $G(N, \phi)$ in which Eloise uses τ, which ends at $(\psi_0 \vee \psi_1, s)$, a C-translation of which is $(\psi_0 \vee \psi_1, s')$. We continue α by letting Eloise move whatever τ tells her to move, say ψ_0. Now $S \cup \{(\psi_0, s')\}$ consists of C-translations of τ-moves. So we let Eloise choose $S \cup \{(\psi_0, s')\}$ as the next position.
3. If $(\forall x\theta, s') \in S$, then Abelard may choose $c \in C$ and the next position is $S \cup \{(\theta, s'(c/x))\}$. There is a sequence α of moves in $G(N, \phi)$ in which Eloise uses τ, which ends at $(\forall x\theta, s)$, a C-translation of which is $(\forall x\theta, s')$. We continue α by letting Abelard move $(\theta, s(\pi(c)/x))$. Now $S \cup \{(\theta, s'(c/x))\}$ consists of C-translations of τ-moves.
4. If $(\exists x\theta, s) \in S$, then Eloise may choose $a \in M$ and the next position is $S \cup \{\theta\langle s(a/x)\rangle\}$. There is a sequence α of moves in $G(N, \phi)$ in which Eloise uses τ, which ends at $(\exists x\theta, s)$, a C-translation of which is $(\exists x\theta, s')$. We continue α by letting Eloise use τ to move $(\theta, s(a/x))$. Let $a = \pi(c)$. Now $S \cup \{\theta\langle s'(c/x)\rangle\}$ consists of C-translations of τ-moves. We let Eloise choose $S \cup \{\theta\langle s'(c/x)\rangle\}$ as the next position.

If Eloise follows this strategy and τ happens to be a winning strategy, then she wins because if at some point there are (ψ, s') and $(\neg\psi, t')$ in the set S, such that ψ is atomic and $s'(x) = t'(x)$ for variables x in ψ, then these would be C-translations of (ψ, s) and $(\neg\psi, t)$, where $s(x) = t(x)$ for x in ψ, respectively, and s (and t) would satisfy both ψ and $\neg\psi$ in N because τ is a winning strategy, a contradiction. $\square$

The following theorem is more interesting because here we do not have a model to start with but we have to construct a model. In this sense this is reminiscent of Gödel's Completeness Theorem. The theorem demonstrates how the Model Existence Game can yield a model which then has a good fit with the Evaluation Game (see Fig. 20.3). The theorem goes back to Beth (1955).

Theorem 3 *There are a mapping Ψ and an "enumeration strategy" σ_0 of Abelard in MEG(ϕ) such that if τ is any strategy of Eloise in MEG(ϕ), playing τ against σ_0 determines a unique model $M(\tau)$ and a strategy $\Psi(\tau)$ of Eloise in $G(M(\tau), \phi)$. If τ is winning, then so is $\Psi(\tau)$.*

Fig. 20.3 From model existence to a model

$$\begin{tikzcd} C \arrow[r, dash] \arrow[d, dash, "\tau"'] & M(\tau) \arrow[d, dash, "\Psi(\tau)"] \\ \phi & \phi \end{tikzcd}$$

Proof To make the presentation a little easier, we assume ϕ has a relational vocabulary and does not contain the identity symbol. Let σ_0 be the following "enumeration strategy" of Abelard in MEG(ϕ): During the game Abelard makes sure by means of careful bookkeeping that if S is the position, then:

1. If $(\psi_0 \wedge \psi_1, s) \in S$, then during the game he will at some position $S' \supseteq S$ decide that the next position is $S' \cup \{(\psi_0, s)\}$ and at some further position $S'' \supseteq S'$ he will decide that the next position is $S'' \cup \{(\psi_1, s)\}$.
2. If $(\psi_0 \vee \psi_1, s) \in S$, then at some position $S' \supseteq S$ Abelard asks Eloise to choose whether the next position is $S' \cup \{(\psi_0, s)\}$ or $S' \cup \{(\psi_1, s)\}$.
3. If $(\forall x\theta, s) \in S$, then for all n during the game he will at some position $S' \supseteq S$ decide that the next position is $S' \cup \{(\theta, s(c_n/x)\}$.
4. If $(\exists x\theta, s) \in S$, then at some position $S' \supseteq S$ Abelard will ask Eloise to choose n after which the next position is $S' \cup \{(\theta, s(c_n/x))\}$.

Let us play MEG(ϕ) while Abelard uses this strategy and Eloise plays τ. Let

$$\mathcal{S} = \langle S_n : n < \omega \rangle$$

be the infinite sequence of positions during this play. Since the strategies of the players have been fixed, the sequence $\mathcal{S}$ is uniquely determined. Note that $S_n \subseteq S_{n+1}$ for all n. Let Γ be the union of all the positions in $\mathcal{S}$. We build a model $M = M(\tau)$ as follows: The domain of the model is $\{c_n : n \in \mathbb{N}\}$. If R is a relation symbol, then we let $R(c_{n_0}, \ldots, c_{n_k})$ hold in M if $(R(x_{n_0}, \ldots, x_{n_k}), s) \in \Gamma$ for some s such that $s(x_i) = c_i$ for $i = n_0, \ldots, n_k$. The strategy $\Psi(\tau)$ of Eloise in $G(M, \phi)$ is the following: She makes sure that if the position in $G(M, \phi)$ is (ψ, s), then $(\psi, s) \in \Gamma$. Let us see that she can follow the strategy throughout the game:

1. If ψ is $\psi_0 \wedge \psi_1$, then Abelard chooses whether the next position is (ψ_0, s) or (ψ_1, s). Suppose he chooses (ψ_1, s). Now, $(\psi_0 \wedge \psi_1, s)$ occurs in a position S during the game, because $(\psi_0 \wedge \psi_1, s) \in \Gamma$. According to how σ_0 is defined, Abelard has at some later position $S' \supseteq S$ of the game decided that the next position is $S' \cup \{(\psi_1, s)\}$. Hence $(\psi_1, s) \in \Gamma$.
2. If ψ is $\psi_0 \vee \psi_1$, then Eloise should choose whether the next position is (ψ_0, s) or (ψ_1, s). We know $(\psi_0 \vee \psi_1, s) \in S$ for some position S during the game, because $(\psi_0 \vee \psi_1, s) \in \Gamma$. By how σ_0 was defined, Abelard has at some later position $S' \supseteq S$ asked Eloise to choose between (ψ_0, s) and (ψ_1, s). The strategy τ has directed Eloise to choose, say (ψ_0, s). Thus $(\psi_0, s) \in \Gamma$ and she can safely play (ψ_0, s) in $G(M, \phi)$.

3. If ψ is $\forall x\theta$, then Abelard chooses $a \in M$ and the next position is $(\theta, s(a/x))$. Suppose he chooses $a = c_n$. Now, $(\forall x\theta, s)$ occurs in a position S during the game, because $(\forall x\theta, s) \in \Gamma$. According to how σ_0 is defined, Abelard has at some later position $S' \supseteq S$ of the game decided that the next position is $S' \cup \{(\theta, s(c_n/x))\}$. Hence $(\theta, s(a/x)) \in \Gamma$.
4. If ψ is $\exists x\theta$, then Eloise should choose for which n the next position is $(\theta, s(c_n/x))$. We know $(\exists x\theta, s) \in S$ for some position S during the game, because $(\exists x\theta, s) \in \Gamma$. By how σ_0 was defined, Abelard has at some later position $S' \supseteq S$ asked Eloise to choose n for which the next position would be $S' \cup \{(\theta, s(c_n/x))\}$. The strategy τ has directed Eloise to indeed choose an n leading to the new position $S' \cup \{(\theta, s(c_n/x))\}$. Thus $(\theta, s(c_n/x)) \in \Gamma$ and she can safely play $(\theta, s(c_n/x))$ in $G(M, \phi)$.

Suppose now that τ is a winning strategy of Eloise. We show that then $\Psi(\tau)$ is a winning strategy as well. Suppose the game $G(M, \phi)$ ends at a position (ψ, s) where ψ is atomic. Then (ψ, s) is in Γ whence s satisfies ψ in M. Suppose, on the other hand, the game $G(M, \phi)$ ends at a position $(\neg\psi, s)$ where ψ is atomic. Then $(\neg\psi, s)$ is in S_n for some n. It suffices to show that s does not satisfy ψ in M. Suppose it does. Then there is $(\psi, s') \in S_m$ for some m and some s' such that $s(x) = s'(x)$ for variables x in ψ. Then S_{n+m} has both $(\neg\psi, s)$ and (ψ, s'), contrary to the assumption that τ is a winning strategy of Eloise. □

Let us suppose we investigate ϕ using the Model Existence Game and find out that Abelard has a winning strategy in MEG(ϕ). We can form a tree, a Beth Tableaux, of all the positions when Abelard plays his winning strategy and we stop playing as soon as Abelard has won. Every branch of the tree is finite and ends in a position which includes a contradiction. We can make the tree finite by looking at the game more carefully. We can then view this tree as a kind of proof of $\neg\phi$. In this sense the Model Existence Game builds a bridge between proof theory and model theory. Strategies of Abelard direct us to proof theory, while strategies of Eloise direct us to model theory.

A winning strategy of Eloise in MEG(ϕ) can be conveniently given in the form of a so-called *consistency property*, which is just a set of finite sets of sentences satisfying conditions which essentially code a winning strategy for Eloise in MEG(ϕ). Sometimes it is more convenient to use a consistency property than Model Existence Game. But as far as strategies of Eloise are concerned, the two are one and the same thing. Consistency properties have been successfully used to prove interpolation and preservations results in model theory, especially infinitary model theory (Makkai, 1969).

Apart from first order and infinitary logic, the Model Existence Game can be used in the proof theory and model theory of logic with generalized quantifiers and higher order logic. In the infinitary logic $L_{\kappa\lambda}$, $\lambda > \omega$, the Model Existence Game yields so-called *chain models*, rather than real models. In the case of generalized quantifiers, such as the quantifier "there exists uncountably many", the Model Existence Game yields so-called weak models, which have to be transformed to real models by a model theoretic argument (Keisler, 1970). In the case of higher order logics the Model

Existence Game yields so-called *general models*, where the higher order variables range over a set of subsets and relations rather than over all subsets and relations (Henkin, 1950). The Model Existence Game can be used also in propositional and modal logic.

20.4 The Sentence is Missing: The EF Game

In the EF game we have a model but no sentence. To indicate what kind of sentence we are looking for we have two models. The sentence is supposed to be true in one but false in the other. It may be that no such sentence can be found. Then the models are *elementarily equivalent*. We describe a game in which strategies of one player track possibilities for elementary equivalence and the strategies of the other player track possibilities for a separating sentence. In its simplest form this game, the EF-game (a.k.a. Ehrenfeucht–Fraïssé game, the back-and-forth game or the model comparison game), introduced by Ehrenfeucht (1960), is the following: Suppose M and N are two structures for the same vocabulary L.

The game $\mathrm{EF}_m(M, N)$ has two players Abelard and Eloise and m moves. A position of the game is a set

$$s = \{(a_0, b_0), \ldots, (a_{n-1}, b_{n-1})\} \tag{20.1}$$

of pairs of elements such that the a_i are from A and the b_i are from B, and $n \leq m$. In the beginning of the game the position is $\emptyset$. The rules are as follows: Suppose the position is s.

1. Abelard may choose some $a_n \in A$. Then Eloise chooses $b_n \in B$ and the next position is $s \cup \{(a_n, b_n)\}$.
2. Abelard may choose some $b_n \in B$. Then Eloise chooses $a_n \in A$ and the next position is $s \cup \{(a_n, b_n)\}$.

Abelard wins if during the game the position (20.1) is such that $(a_0, \ldots, a_{n-1})$ satisfies some literal in M but $(b_0, \ldots, b_{n-1})$ does not satisfy the corresponding literal in N. If Abelard does not win, then Eloise is declared the winner.

Intuitively, Eloise defends the proposition that M and N are very similar in the sense that isomorphic structures are similar (but M and N need not be actually isomorphic). Abelard doubts this similarity. Abelard expresses his doubt by asking where would the purported isomorphism map this or that element.

An extremely simple example is two models of size $\geq m$ in the empty vocabulary. Then Eloise's winning strategy is based on her playing different elements when Abelard does the same. A slightly less trivial example is two finite linear orders of size $\geq 2^m$.

Note that this game is determined. Moreover, if Eloise knows an isomorphism $f : M \to N$ she can respond by playing always so that $b_n = f(a_n)$.

Fig. 20.4 Interaction of the EF game and the Evaluation Game

$$\begin{tikzcd} M \arrow[r, dash, "\tau"] \arrow[d, dash, "\sigma"'] & N \arrow[d, dash, "\Theta(\sigma,\tau)"] \\ \phi & \phi \end{tikzcd}$$

The following theorem demonstrates how the EF game and the Evaluation Game interact when we actually do have a sentence (see Fig. 20.4). The theorem is due to Ehrenfeucht (1960).

Theorem 4 *There is a function Θ such that if τ is a strategy of Eloise in $EF_m(M, N)$, ϕ is an $L_{\infty\omega}$-sentence of quantifier rank $\leq m$, and σ is a strategy of Eloise in $G(M, \phi)$, then $\Theta(\sigma, \tau)$ is a strategy of Eloise in $G(N, \phi)$. If τ and σ are winning strategies, then so is $\Theta(\sigma, \tau)$.*

Proof We call a position of the game $\mathrm{EF}_m(M, N)$ a τ-position if it arises while Eloise is playing τ. Suppose ϕ is a sentence of quantifier rank $\leq m$ and Eloise has a strategy σ in $G(M, \phi)$. We call a position of the game $G(M, \phi)$ a σ-position, if it arises while Eloise is playing σ. We use τ and σ to construct a strategy of Eloise in $G(N, \phi)$. If the position of the game $G(N, \phi)$ is (ψ, s), the strategy $\eta = \Theta(\sigma, \tau)$ of Eloise is to play simultaneously with $G(N, \phi)$ the game $\mathrm{EF}_m(M, N)$ and the game $G(M, \phi)$ and make sure that if

$$\pi = \{(a_0, b_0), \ldots, (a_{n-1}, b_{n-1})\}$$

is the current τ-position in $\mathrm{EF}_m(M, N)$ and $s(x) = \pi(s'(x))$ for all x in the domain of s, then (ψ, s') is the current σ-position in $G(M, \phi)$ (see Fig. 20.5).

Let us check that it is possible for Eloise to play this strategy:

1. If the position is (ψ, s) where ψ is a literal, the game ends.
2. If the position is (ψ, s) where is ψ is $\bigwedge_i \phi_i$, then Abelard chooses i and the next position is (ψ_i, s). Whichever he chooses, we let Abelard make the respective move (ψ_i, s') in $G(M, \phi)$.
3. If the position is (ψ, s) where is ψ is $\bigvee_i \phi_i$, then Eloise chooses i as follows. Since (ψ, s') is a σ-position, the strategy σ tells Eloise which of (ψ_i, s') to play in $G(M, \phi)$. Then Eloise plays the respective (ψ_i, s) in $G(N, \phi)$.

Fig. 20.5 The strategy $\Theta(\sigma, \tau)$

$$\begin{tikzcd} M \arrow[r, dash, "\pi"] \arrow[d, dash, "s'"'] & N \arrow[d, dash, "s"] \\ \phi & \phi \end{tikzcd}$$

4. If the position (ψ, s) is $(\forall x\theta, s)$, then Abelard chooses $b_n \in N$ and the next position is (θ, t), $t = s(b_n/x)$. We continue the game $\mathrm{EF}_m(M, N)$ from the τ-position $\{(a_0, b_0), \ldots, (a_{n-1}, b_{n-1})\}$ letting Abelard play b_n. The strategy τ tells Eloise to choose $a_n \in M$ so that

$$\pi' = \{(a_0, b_0), \ldots, (a_n, b_n)\} \tag{20.2}$$

is again a τ-position. Now we continue the game $G(M, \phi)$ from position $(\forall x\theta, s')$ by letting Abelard play a_n. We reach the position (θ, t'), $t' = s'(a_n/x)$, which is still a σ-position, and we have $\pi'(t'(y)) = t(y)$ for all y in the domain of t'.
5. The position (ψ, s) is $(\exists x\theta, s)$. Now we continue the game $G(M, \phi)$ from position $(\exists x\theta, s')$ by letting Eloise play, according to σ, an element a_n and we reach a new σ-position (θ, t'), $t' = s'(a_n/x)$. We continue the game $\mathrm{EF}_m(M, N)$ from the τ-position $\{(a_0, b_0), \ldots, (a_{n-1}, b_{n-1})\}$ letting Abelard play a_n. The strategy τ tells Eloise to choose $b_n \in N$ so that (20.2) is again a τ-position. We reach the position (θ, t), $t = s(b_n/x)$, and we have $t(y) = \pi(t'(y))$ for all y in the domain of t.

If σ is a winning strategy and the game $G(N, \phi)$ ends in the position (ψ, s), where ψ is a literal, then Eloise wins because then (ψ, s') is a σ-position meaning that s' satisfies the literal ψ in M, and τ being a winning strategy this means that s satisfies the literal ψ in N. $\square$

Just as the Model Existence Game can yield a model, the EF game can produce a sentence. The next theorem shows how this arises from co-operation of the EF game and the Evaluation Game (see Fig. 20.6). The theorem is due to Ehrenfeucht (1960). The emphasis of this kind of translations of strategies occurs already in van Benthem (2014).

Theorem 5 *There is a sentence $\phi \in L_{\infty\omega}$ of quantifier rank $\leq m$ and mappings Ξ_E and Ξ_A such that if τ is a strategy of Abelard in $EF_m(M, N)$, then $\Xi_E(\tau)$ is a strategy of Eloise in $G(M, \phi)$, and $\Xi_A(\tau)$ is a strategy of Abelard in $G(N, \phi)$. If τ is a winning strategy, then $\Xi_E(\tau)$ and $\Xi_A(\tau)$ are winning strategies. If L is finite and relational, the sentence ϕ is logically equivalent to a first order sentence of quantifier rank $\leq m$.*

Fig. 20.6 Interaction of the EF game and the Evaluation Game

$$\begin{tikzcd}
M \arrow[r, dash, "\tau"] \arrow[d, dash, "\Xi_E(\tau)"'] & N \arrow[d, dash, "\Xi_A(\tau)"] \\
\phi & \neg\phi
\end{tikzcd}$$

Proof Suppose s is an assignment into M with domain $\{x_0, \ldots, x_{n-1}\}$. Let

$$\psi^{0,n}_{M,s} = \bigwedge_i \psi_i$$

$$\psi^{m+1,n}_{M,s} = \left(\forall x_n \bigvee_{a\in M} \psi^{m,n+1}_{M,s(a/x_n)}\right) \wedge \left(\bigwedge_{a\in M} \exists x_n \psi^{m,n+1}_{M,s(a/x_n)}\right),$$

where ψ_i lists all the literals in the variables $x_0, \ldots, x_{n-1}$ satisfied by s in M.

The sentence ϕ we need to prove the theorem is $\psi^{m,0}_{M,\emptyset}$. Note that this only depends on M. Clearly Eloise has a trivial strategy $\Xi_E(\tau)$ in $G(M, \phi)$ (independently of τ), and this strategy is always a winning strategy: after Abelard has chosen a value for x_n, she uses that value as her choice of $a \in M$, and after Abelard has chosen $a \in M$, she uses that element as her value for x_n. Note that the quantifier-rank of $\psi^{m,n}_{M,s}$ is always m.

We now describe the strategy $\Xi_A(\tau)$ of Abelard in $G(N, \phi)$. We call a position of the EF-game a τ *-position* if it arises while Abelard is playing τ. Suppose s is an assignment into M and s' an assignment into N, both with domain $\{x_0, \ldots, x_{n-1}\}$. We use $s \cdot s'$ to denote the set of pairs $(s(x_i), s'(x_i))$, $i = 0, \ldots, n-1$. The strategy of Abelard is to play $G(N, \phi)$ in such a way that if the position at any point is $(\psi^{i,m-i}_{M,s}, s')$, then $s \cdot s'$ is a τ-position. In the beginning the position is $(\phi, \emptyset)$.

1. Suppose the position in $G(N, \phi)$ is $(\psi^{i,m-i}_{M,s}, s')$, $i > 0$, and the next move for Abelard in $\mathrm{EF}_m(M, N)$ according to τ is $a \in M$. The strategy of Abelard is to choose the latter conjunct of $\psi^{i,m-i}_{M,s}$. Then Abelard chooses the element $a \in M$ in the big conjunction move. Now it is the turn of Eloise to choose some $b \in N$ as the value of x_{m-i} and that will be the next move of Eloise in $\mathrm{EF}_m(M, N)$. The new position $s(a/x_{m-i}) \cdot s'(b/x_{m-i})$ is still a τ-position in $\mathrm{EF}_m(M, N)$. The next position in $G(N, \phi)$ is

$$(\psi^{i-1,m-i+1}_{M,s(a/x_{m-i})}, s'(b/x_{m-i})). \tag{20.3}$$

2. Suppose the position in $G(N, \phi)$ is $(\psi^{i,m-i}_{M,s}, s')$, $i > 0$, and the next move for Abelard in $\mathrm{EF}_m(M, N)$ according to τ is $b \in N$. The strategy of Abelard is to choose the former conjunct where he plays b as x_{m-i}. Now it is the turn of Eloise to choose some $a \in M$ in $G(N, \phi)$. The new position $s(a/x_{m-i}) \cdot s'(b/x_{m-i})$ is still a τ-position in $\mathrm{EF}_m(M, N)$. The next position in $G(N, \phi)$ is (20.3).
3. Finally the position is $(\psi^{0,m}_{M,s}, s')$. Note that $s \cdot s'$ is still a τ-position in $\mathrm{EF}_m(M, N)$. The game $\mathrm{EF}_m(M, N)$ has now ended. Abelard now chooses the first (in some fixed enumeration) literal conjunct of the formula $\psi^{0,m}_{M,s}$ that is not satisfied by s' in N, if any exist, otherwise he simply chooses the first conjunct.

Suppose now τ was a winning strategy of Abelard. Then at the end of the game $s \cdot s'$ is a winning position for Abelard and therefore he is indeed able to choose a conjunct of the formula $\psi^{0,m}_{M,s}$ that is not satisfied by s' in N. He has won $G(N, \phi)$.

If L is finite and relational, all the conjunctions and disjunctions are essentially finite because there are only finitely many non-equivalent formulas of a fixed quantifier rank. Thus, although ϕ is, a priori, a formula of the infinitary logic $L_{\infty\omega}$, it is logically equivalent to a first order formula. □

There is a tight connection between σ, τ and $\Theta(\sigma, \tau)$. This is reflected in a connection between ϕ and $\mathrm{EF}_m(M, N)$ which goes deeper into the structure of ϕ than the mere condition that its quantifier rank is at most m. If the non-logical symbols of ϕ are in $L' \subset L$, then it suffices that τ is a strategy of Eloise in the game $\mathrm{EF}_m(M \restriction L', N \restriction L')$ between the reducts $M \restriction L'$ and $N \restriction L'$. If we know more about the syntax of ϕ, for example that it is existential, universal or positive, we can modify $\mathrm{EF}_M(M, N)$ accordingly by stipulating that Abelard only moves in M, only moves in N, or that he has to win by finding an atomic (rather than literal) relation which holds in M but not in N.

Likewise, many useful observations can be made about the sentence ϕ and the strategy τ. Here are some. We already observed that the quantifier rank of the separating sentence ϕ is the same as the length of the EF game for which Abelard has a winning strategy. If τ is a winning strategy of Abelard even in the game $\mathrm{EF}_m(M \restriction L', N \restriction L')$ for some $L' \subset L$, then the separating sentence ϕ can be chosen so that its non-logical symbols are all in L'. If τ is such that Abelard plays only in M, we can make ϕ existential. If τ is such that Abelard plays only in N, we can make ϕ universal. If Abelard wins with τ even the harder game in which he has to win by finding an atomic (rather than literal) relation which holds in M but not in N, then we can take ϕ to be a positive sentence.

Strategies in $\mathrm{EF}_m(M, N)$ reflect structural properties of M and N. If we know a strategy of Eloise in $\mathrm{EF}_m(M_i, N_i)$ for $i \in I$, we can construct strategies of Eloise for EF games between products and sums of the models M_i and the respective products and sums of the models N_i. This can be extended to so-called κ-local functors (Feferman, 1972). For an example of the use of tree-decompositions, see e.g. Grohe (2007).

Just as in the Evaluation Game, the EF game permits modifications which extend the above two theorems to other logics. Accordingly, EF games are known for infinitary logics, generalized quantifiers, and higher order logics. In modal logic the corresponding game is called a bisimulation game. Even propositional logic has an EF game although there are no quantifiers (Hella & Väänänen, 2015). In the EF game for team semantics the players move teams, not elements (Väänänen, 2007).

The sentences arising from Theorem 5 are behind the Distributive Normal Forms of Hintikka (1953) and are known in infinitary logic as building blocks of Scott Sentences (Scott, 1965).

In an important variant called the Pebble Game, the players can pick only a fixed finite number of elements from the models but they can give up some elements in order to make room for new ones. This game is closely related to first order (or infinitary) logic with a fixed finite number of variables (free or bound). It was introduced independently by N. Immerman and B. Poizat in 1982. A good reference is Kolaitis and Vardi (1992).

20.5 Further Work

The Evaluation Game, the Model Existence Game and the EF game go so deep into the essential concepts of logic such as truth, consistency, and separating models by sentences, that a lot of research in logic can be represented in terms of these games. This may not be very interesting in itself. However, the translations of the strategies between the games suggest a coherent uniform approach to syntax and semantics and at the same time to model theory and proof theory. Moreover, both the Evaluation Game and the EF game are indifferent as to whether the models are finite or infinite, which gives them a useful role in computer science logic. Despite the vast literature on each of the three games separately, there seems to be a lot of potential for the study of their interaction as a manifestation of the Strategic Balance of Logic.

References

Abramsky, S., Dawar, A., & Wang, P. (2017). The pebbling comonad in finite model theory. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20–23, 2017* (pp. 1–12). IEEE Computer Society.

Beth, E. W. (1955). *Semantic entailment and formal derivability*. Mededelingen der koninklijke Nederlandse Akademie van Wetenschappen, afd. Letterkunde. Nieuwe Reeks, Deel 18, No. 13. N. V. Noord-Hollandsche Uitgevers Maatschappij, Amsterdam.

Ehrenfeucht, A. (1960). An application of games to the completeness problem for formalized theories. *Fundamenta Mathematicae, 49*, 129–141 (1961).

Feferman, S. (1972). Infinitary properties, local functors, and systems of ordinal functions. In *Conference in Mathematical Logic—London '70 (Proceedings of the Conference, Bedford College, London, 1970)*. Lecture Notes in Mathematics (Vol. 255, pp. 63–97). Berlin: Springer.

Grohe, M. (2007). The complexity of homomorphism and constraint satisfaction problems seen from the other side. *Journal of the ACM, 54*(1), Art. 1, 24.

Hella, L., & Väänänen, J. (2015). The size of a formula as a measure of complexity. In Å. Hirvonen, J. Kontinen, R. Kossak, & A. Villaveces (Eds.), *Logic without borders - Essays on set theory, model theory, philosophical logic and philosophy of mathematics*. Ontos mathematical logic (Vol. 5, pp. 193–214). De Gruyter.

Henkin, L. (1950). Completeness in the theory of types. *Journal of Symbolic Logic, 15*, 81–91.

Henkin, L. (1961). Some remarks on infinitely long formulas. In *Infinitistic Methods (Proceedings of the Symposium on the Foundations of Mathematics, Warsaw, 1959)* (pp. 167–183). Pergamon, Oxford.

Hintikka, J. (1953). Distributive normal forms in the calculus of predicates. *Acta Philosophica Fennica, 6*, 71.

Hintikka, J. (1955). Form and content in quantification theory. *Acta Philosophica Fennica, 8*, 7–55.

Hintikka, J. (1968). Language-games for quantifiers. In N. Rescher (Ed.), *Studies in logical theory* (pp. 46–72). Blackwell.

Keisler, H. J. (1970). Logic with the quantifier "there exist uncountably many". *Annals of Mathematical Logic, 1*, 1–93.

Kolaitis, P. G., & Vardi, M. Y. (1992). Infinitary logics and 0-1 laws. In *Selections from the 1990 IEEE Symposium on Logic in Computer Science* (Vol. 98, pp. 258–294).

Kueker, D. W. (1977). Countable approximations and Löwenheim-Skolem theorems. *Ann. Math. Logic, 11*(1), 57–103.

Makkai, M. (1969). On the model theory of denumerably long formulas with finite strings of quantifiers. *Journal of Symbolic Logic, 34*, 437–459.

Scott, D. (1965). Logic with denumerably long formulas and finite strings of quantifiers. In *Theory of Models (Proceedings of the 1963 International Symposium Berkeley)* (pp. 329–341). Amsterdam: North-Holland.

Smullyan, R. M. (1963). A unifying principal in quantification theory. *Proceedings of the National Academy of Sciences of the United States of America, 49*, 828–832.

Väänänen, J. (2007). *Dependence logic*. London mathematical society student texts (Vol. 70). Cambridge: Cambridge University Press. A new approach to independence friendly logic.

Väänänen, J. (2011). *Models and games*. Cambridge studies in advanced mathematics (Vol. 132). Cambridge: Cambridge University Press.

van Benthem, J. (2014). *Logic in games*. Cambridge, MA: MIT Press.

Wittgenstein, L. (1953). *Philosophical investigations*. New York: The Macmillan Co. Translated by G. E. M. Anscombe.

Part V
Categorical and Logical Semantics

Chapter 21
Compositionality in Context

Alexandru Baltag, Johan van Benthem, and Dag Westerståhl

Abstract Compositionality is a principle used in logic, philosophy, mathematics, linguistics, and computer science for assigning meanings to language expressions in a systematic manner following syntactic construction, thereby allowing for a perspicuous algebraic view of the syntax-semantics interface. Yet the status of the principle remains under debate, with positions ranging from compositionality always being achievable to its having genuine empirical content. This paper attempts to sort out some major issues in all this from a logical perspective. First, we stress the fundamental harmony between Compositionality and its apparent antipode of Contextuality that locates meaning in interaction with other linguistic expressions and in other settings than the actual one. Next, we discuss basic further desiderata in designing and adjudicating a compositional semantics for a given language in harmony with relevant contextual syntactic and semantic cues. In particular, in a series of concrete examples in the realm of logic, we point out the dangers of over-interpreting compositional solutions, the ubiquitous entanglement of assigning meanings and the key task of explaining given target inferences, and the dynamics of new language design, illustrating how even established compositional semantics can be rethought in a fruitful manner. Finally, we discuss some fresh perspectives from the realm of game semantics for natural and formal languages, the general setting for Samson Abramsky's influential work on programming languages and process logics. We highlight outside-in coalgebraic perspectives on meanings as finite or infinitely unfolding behavior that might challenge and enrich current discussions of compositionality.

Keywords Compositionality · Contextuality · Currying · Inference · Language design · Dependence · Game semantics · Coalgebra

A. Baltag · J. van Benthem (✉)
ILLC, University of Amsterdam, Science Park 107, 1098 Amsterdam, The Netherlands
e-mail: j.vanbenthem@uva.nl

A. Baltag
e-mail: thealexandrubaltag@gmail.com

D. Westerståhl
Department of Philosophy, Stockholm University, 10691 Stockholm, Sweden
e-mail: dag.westerstahl@philosophy.su.se

A. Palmigiano and M. Sadrzadeh (eds.), *Samson Abramsky on Logic and Structure in Computer Science and Beyond*, Outstanding Contributions to Logic 25,
https://doi.org/10.1007/978-3-031-24117-8_21

21.1 The Makings of a Semantics: Compositionality, Contextuality, and Criteria for Judging Design

The three intertwined topics to be addressed in this paper are introduced in the following characteristically clear passage from Samson Abramsky's paper (Abramsky, 2006):

> It is, or should be, an aphorism of semantics that *the key to compositionality is parameterization*. Choosing the parameters aright allows the meaning of expressions to be made sensitive to their contexts, and hence defined compositionally. While this principle could, in theory, be carried to the point of trivialization, in practice the identification of the right form of parameterization does usually represent some genuine insight into the structure at hand, (o.c., p. 19).

Samson Abramsky is a champion of the methodology of compositionality in the semantics of programming languages, Abramsky and McCusker (1999), Abramsky (2003), game semantics of logical systems, Abramsky and Jagadeesan (1994), Abramsky (2006), Abramsky and Väänänen (2009), and more generally the category-theoretic foundations of logic and computation. In this survey and discussion paper, we look at compositionality in its broadest sense, from philosophy and linguistics to logic and computer science. We discuss what the principle says, first proceeding abstractly in broad mathematical terms, starting from the account in Hodges (2001, 2011) and adding some general observations of our own. Here and throughout this article, we will highlight the strategy of achieving compositionality by adding new parameters of evaluation, reflecting the spirit of the above quote and linking up with Frege's celebrated principle of Contextuality. Following the abstract analysis, we consider how compositionality fares in a number of case studies, and what further basic features are entangled with compositionality and contextuality when giving a logical semantics for a language. Finally, we will discuss what we can learn from them about further characteristics of well-behaved semantics. But before we do all this, an introduction is in order to the main themes as we see them.

Compositionality. The principle of Compositionality says that the meaning of a linguistic expression is a function of the meanings of its parts, plus the syntactic mode of composition of these parts. The most striking feature of this formulation is its extreme generality: it says nothing about what meanings are or what syntax (the notion of 'parts') should look like. It should be seen primarily, we suggest, as a way of organizing one's ideas about the syntax-semantics interface of a language in a perspicuous algebraic format. The principle occurs widely in logic, linguistics, philosophy, computer science, and cognitive science. But its status remains under debate. Is compositionality a refutable empirical claim, a methodological recommendation, or a bit of both? Can it always be satisfied in setting up a semantics, or is it a design constraint with real empirical bite? Camps keep forming around these issues, books keep getting written. One purpose of this paper is to add some concrete considerations to these debates stemming from analyzing a number of case studies in logical semantics.

A good thing? Here is a question of motivation that should come first. Why is Compositionality so important? A number of different virtues have been touted in the literature: it facilitates learnability, it underlies our ability to produce infinitely many meaningful or even true new sentences, it explains successful linguistic communication, it is a sufficient condition for computationally efficient interpretability of natural and formal languages; see Pagin and Westerståhl (2010b) for a survey and evaluation of these claims. Indeed, the inductive, or rather recursive character, of the usual compositional semantics, makes a language computably learnable. And from another computational standpoint, compositionality facilitates a divide and conquer strategy for the analysis and synthesis of programs, Janssen (2011), a virtue also emphasized in Abramsky's work. Finally, compositionality is often seen as guaranteeing the existence of an algebraic semantics validating perspicuous algebraic laws of reasoning with the key notions of a language, Andréka et al. (2001). Our purpose in this paper is not to adjudicate these claimed virtues, or even to survey all results and approaches. We refer to Janssen (2011) in the "Handbook of Logic and Language" for a survey of compositionality in logic, linguistics, and computer science which is up to date until 2000. For the period after that, the reader may consult Pagin and Westerståhl (2010a, b), and the "Oxford Handbook of Compositionality", (Hinzen et al., 2012).

In this article, we concentrate on two methodological virtues. Compositionality is a method for laying out meanings in a perspicuous manner, or if you prefer a mentalistic picture: for organizing our thoughts perspicuously, as far as expressed in language. Moreover, following this method facilitates the design of logical proof systems that govern reasoning with these meaningful notions.

Simple and complex. It is also good to realize what Compositionality does not say. It is often thought that it is all about constructing *complex meanings* out of *simple ones*. This view is reinforced by the syntax of logical languages, where one tends to view atomic formulas as expressing simple facts about concrete objects, while further logical operators add complexity. However, we see all of this as wrong, or at least confused. Compositional design makes no assumptions about the complexity of basic meanings, and it can equally well lead from complex to complex meanings, perhaps even from complex to simple ones with expressions like "ignore" or "delete". The neutrality of Compositionality with respect to complexity of meanings will show in particular with the substitutability versions to be discussed in Sect. 21.2. Finally, on the issue of simplicity, there may also be a confusion at work. In a relative sense, composition of meanings may indeed start from simpler atomic parts, where the simplicity just means that these parts are not analyzed any further. This is the simplicity of abstraction. But in concrete instances, atomic sentences can have very complex meanings, dwarfing the marginal complexity added by the compositional construction.[1]

[1] For instance, an atomic predicate like 'friendly person' may have a very complex meaning such as 'likely to be more pleasant and helpful than the average person in the reference group', and then the additional complexity of understanding the meaning of, say, 'two friendly persons' seems marginal. A similar point is made by Dummett (1973) about Russell's 'logical atomism' versus Frege's view

Perhaps the simplest way of illustrating all these issues is with the concrete case of Boolean algebra, whose language has an algebraic semantics that is a showcase of compositionality. In a Boolean set algebra, all algebraic terms denote sets: there is absolutely no way of saying that a complex term denotes a more complex set than a Boolean variable. Moreover, while some Boolean algebras indeed consist of simple objects, such as the two numbers 0, 1, another perfectly legitimate semantic structure for this language is the Lindenbaum algebra of propositional logic, whose objects are defined as equivalence classes of the relation of provable equality from the principles of Boolean Algebra. Clearly, the latter objects are defined by reference to the whole language, and much more complex than the truth table operations involved in computing their compositions. But Compositionality does not favor either model as an interpretation for the Boolean language.[2] This observation brings us to our next theme.

Contextuality. The principle of compositionality is usually ascribed to Frege, who introduced recursive syntax and matching meanings in modern logic and studies of linguistic meaning generally, (Dummett, 1973; Pagin and Westerståhl, 2010a), though this attribution has also been questioned, (Janssen, 2001). However this may be, Frege also stated a different influential insight in Frege (1884, x), namely, his well-known *Context Principle*:

> never …ask for the meaning of a word in isolation, but only in the context of a sentence.

This view of meaning as interaction with the rest of the language seems very close to certain modern views. One is reminded of Zellig Harris's dictum

> To know the meaning of a word, look at the company it keeps,

but also—with a wider notion of context to which we will return—of Abramsky's insistence on finding the right contextual parameters. This external view of meaning may seem at odds with the intuitively more internal perspective of compositionality. But this is a misconception. The preceding discussion of simplicity versus complexity of meanings pointed at a peaceful co-existence: basic unanalyzed parts could actually have meanings dependent on their role in the context of the whole language. And as we explain in the next section, Hodges has provided an elegant more precise mathematical way of resolving the tension between compositionality and fregean contextuality.

The natural, and perhaps even inevitable, interplay of compositionality and contextuality is taken for granted in this paper. Many of our concrete examples are about the search for the right contextual parameters beyond the immediate syntax of the sentence and the actual occasion of its use, that make compositionality possible. Still, an important clarification needs to be made here, since the term "context" tends

where basic expressions could have very complex meanings. In the background, Dummett also distinguishes two complementary directions of 'recognition' and 'analysis' of meanings that are relevant to our topic of compositionality and contextuality.

[2] Similar points apply to model-theoretic semantics, the vehicle for our later examples. This is related to algebraic semantics via a representation theory that is beyond our scope here.

to be overused, and carries at least two different senses. In the *semantic sense of contextuality*, expressions may get their meanings in rich models with 'indices of evaluation (points, worlds, situations, etc.)' that pack all the necessary information for a compositional modus operandi. A simple example is modal logic, where interpreting expressions at one point or world may require information about the truth values of their parts at all other points. Abramsky's quote also holds for the whole history of logical semantics: indices of evaluation develop as required by the needs of compositionality.

However, there is also a stronger sense of what may be called *syntactic contextuality*, where the meaning of an expression at some point may depend on the meanings of arbitrary expressions of the language at this, or other points. The latter version seems more in line with Frege's Context Principle. This stronger version, too, occurs naturally in logical semantics: especially when we consider meaning assignments that have to respect the intuitive validity of given inference patterns: conclusions from sentences do not need to be syntactic parts. Also, in our later discussion of game semantics, meanings in language games may be strategies that record what an agent would say under various circumstances, which can be syntactically quite different alternatives to what is actually said. This paper keeps an open eye to both senses of contextuality.

The content of this paper. We will start by presenting a general mathematical perspective on compositionality due to Hodges (Sect. 21.2). Next, we present some classical compositional semantics in Sect. 21.3, as material for a supplementary analysis in terms of what we call "currying" (Sect. 21.4). After that, we turn to concrete case studies of semantics for logical systems in order to enrich the picture of what compositional semantics involves. Considering instances from modal and first-order logic, we draw some general heuristic lines in Sect. 21.5, including the dangers of overinterpreting received 'solutions' and the role of impossibility results. In Sect. 21.6, we discuss a particularly pervasive feature of compositional semantics: its entanglement with other desiderata, in particular, prior intuitions of valid inference, and in Sect. 21.7, the design of new languages based on compositional analysis of some initial given language. In both cases, we shake received wisdom a bit, using case studies of generalized assignment semantics and a logic of dependence to reopen discussion on semantic choices that may have seemed settled historically once and for all. This concludes what might be called the classical part of our presentation.

Coda: game perspectives. While most themes in this article fit well with classical philosophical and linguistic discussions of compositionality, we add one more twist. In Sect. 21.8, we discuss a number of themes emanating from the long-standing use of *games* in logic, linguistics and computer science. In particular, in Abramsky's semantics of programming languages and concurrent processes, computation has come to mean producing, not just transitions from input to outputs, but complex ongoing *behavior*, that can be infinite just as well as finite. We present some basic ideas from game models for computing, including more radical co-algebraic versions where behavior is not built up from inside in terms of basic building blocks, but is only observable from the outside. This might well affect received understandings of semantics more broadly, also in philosophy and linguistics, as it suggests that meanings are associated with long-term use, a very radical form of 'dynamic semantics'. In particular, sentence structure and its compositional meaning might just be the 'tip of an iceberg', representing a certain top level of formulating a slice of potentially much more complex behavior, or in a more mentalist picture: of thoughts. While our aim here is not to endorse these radical consequences, we do feel they are an exciting complement to the more traditional literature on compositionality.

Finally, since even the radicalism of game semantics is well within the mind set of logicians and philosophers, in Sect. 21.9, we briefly remind the reader of the larger world around us, including perspectives on compositionality from such paradigms as distributional semantics and empirical cognitive science.

21.2 The Minimal Machinery of Compositionality

Compositionality has invited general formulations that admit of mathematical definitions and results. Approaches include universal algebra (cf. Blok and Pigozzi (1989) on algebraizability of logics), abstract recursion theory (compositionality as computability, Pagin (2012)) and in particular, category theory, with Samson Abramsky's long-standing work as a prominent instance. Bits and pieces of relevant formal theory can also be found elsewhere, for instance in the extensive body of work on translations between logical systems, Carnielli and d'Ottaviano (1997), van Benthem (2019), especially, when viewing giving a semantics as a form of translation between object and meta-language.

Probably the most careful available general abstract analysis of compositionality is the one spelled out by Hodges, (2001, 2011). To fix ideas, and for later reference and discussion, we give a quick overview in this section.

21.2.1 Compositionality Defined

A set E of syntactically well-formed expressions is given, construed as *terms* in a partial term algebra $\mathbf{S} = (E, A, \alpha)_{\alpha \in \Sigma}$, generated from the atoms (lexical items) in a subset A of E by the partial operations in Σ.[3] Adding a set of variables we get the usual notion of a *polynomial* $\pi[x_1, \ldots, x_n]$ in the term algebra.[4]

[3] The partial term algebra approach is taken from Hodges (2001); the analysis presented in Hodges (2011) uses a more abstract notion of a *constituent structure*.

[4] In Hodges' original set-up there is also a homomorphism from E to surface strings, needed when the syntax allows (structural or lexical) ambiguities. Since our examples will mainly come from logic, we ignore the distinction between terms and strings here.

A *semantics* is simply a map μ from E to some non-empty set M of values ('μ' for 'meaning').[5] No constraints are placed on M; it could contain truth values, possible worlds, individual objects or assignments, sets of these, elements in some given algebra, sets of formulas, sets of proofs, sets of strategies, etc. Importantly, μ can be partial: if its domain X consists of expressions whose meaning is not in doubt, the issue may be how to extend μ from X to all of E.[6]

Next, given the function μ, expressions/terms can be classified in terms of meaningfulness and of sameness of meaning: for $e, f \in E$,

(1) $e \sim_\mu f$ iff for every polynomial $\pi[x]$ we have $\pi[e] \in X \Leftrightarrow \pi[f] \in X$

(2) $e \equiv_\mu f$ iff $e, f \in X$ and $\mu(e) = \mu(f)$

Think of the equivalence classes of $\sim_\mu$ as semantic categories: sameness of category is preservation of μ-meaningfulness under substitutions. When $X = E$ they are also syntactic categories: reflecting the analysis of grammaticality in categorial grammars in terms of preservation under substitutions.

Here $\equiv_\mu$ is a partial equivalence relation (a *synonymy*) on E with domain X. This is all we need to define the substitution version of compositionality:

Definition 1 (*Substitutional compositionality*) A partial equivalence relation $\equiv$ on E with domain X (so $X = \{e : e \equiv e\}$) is *compositional* if whenever $e_i \equiv f_i$ for $i = 1, \ldots, n$, and $\pi[e_1, \ldots, e_n]$, $\pi[f_1, \ldots, f_n]$ are both in X, we have $\pi[e_1, \ldots, e_n] \equiv \pi[f_1, \ldots, f_n]$. If $\equiv$ is $\equiv_\mu$ for some semantics μ with domain X, we say that μ is *s-compositional*.[7]

The more familiar functional version of compositionality is as follows.

Definition 2 (*functional compositionality*) A function μ is *f-compositional* if for every operation $\alpha \in \Sigma$ there is a corresponding operation r_α on M such that if $\alpha(e_1, \ldots, e_k) \in X$, then $\mu(\alpha(e_1, \ldots, e_k)) = r_\alpha(\mu(e_1), \ldots, \mu(e_k))$.

F-compositionality of μ entails s-compositionality but presupposes, in contrast with the latter, that $X = dom(\mu)$ is closed under subterms. But when that holds, the two are equivalent. In this case—for instance, when μ is total—we can drop the s- and f- prefixes.

---

[5] The 'switcher semantics' of Kathrin Glüer and Peter Pagin generalizes the Hodges set-up to a *set* of semantic functions, with switching between them governed by linguistic context. Pagin and Westerståhl (2010c) is an application to the semantics of quotation; a full presentation is Glüer and Pagin (2021). We will not use this generalization here.

[6] Hodges' set-up generalizes classical algebraic accounts of syntax and semantics, such as Montague (1970), in the following ways: (1) the syntax algebra is partial rather than many-sorted; (2) the meaning function μ can be partial as well; (3) there is no given semantic algebra (although *if* μ is compositional, such an algebra is induced on M); (4) meaning is not compositional by definition but a property that a semantics can have or not have.

[7] This is of course similar to $\equiv$ being a *congruence relation*; indeed, when X is closed under subterms the two notions coincide (see Westerståhl, 2004). But in important applications, the domain of μ is not closed under subterms, and s-compositionality as defined here is then the most general notion of compositionality.

These classical formulations of compositionality assign meanings directly to linguistic expressions. No input from the linguistic or extra-linguistic context is made explicit. But taking contextual factors into account is in no way contrary to compositionality. The next subsection reviews how Hodges combines Frege's Context Principle with compositionality. In Sect. 21.4 we discuss compositionality with explicit contextual parameters.

Digression (dependence). Definitions 1 and 2 can be seen as expressing notions of *dependence*. In particular, s-compositionality resembles a semantic intuition of dependence, as fixing values for an expression by values for its components. And f-compositionality resembles a widespread alternative intuition of dependence as definability by some explicit function. Notions of dependence in logic will be discussed in Sects. 21.6 and 21.7 below. On the other hand, compositionality also involves *independence*, since meanings of expressions do not need meanings of syntactic material not occurring in the expression. But our later discussion of Contextuality in this section will show how the two extremes meet.

21.2.2 Contextuality and the Lifting Lemma

With the preceding in place, Hodges defines a third equivalence relation among terms, which is total even when $\equiv_\mu$ is partial. (We use 'F' for 'Frege'.)

Definition 3 (*fregean values*)

(3) $\quad e \equiv^F_\mu f$ iff $e \sim_\mu f$ and for all $\pi[x]$, if $\pi[e] \in X$ then $\mu(\pi[e]) = \mu(\pi[f])$

The equivalence class $|e|^F_\mu$ of e under $\equiv^F_\mu$ is called the *fregean value* of e.

Essentially, the fregean value of an expression e outside X is given by its contribution to the meanings (given by μ) of complex expressions in X containing e. This can be seen as a precise implementation of fregean Contextuality.

Say that μ is *cofinal* if every term in E is a subterm of some term in X.

Lemma 4 (Hodges' Lifting Lemma) *Suppose that μ is cofinal, $\pi[e_1, \ldots, e_n] \in E$, and also $e_i \equiv^F_\mu f_i$ for $i = 1, \ldots, n$. Then $\pi[f_1, \ldots, f_n] \in E$ and $\pi[e_1, \ldots, e_n] \equiv^F_\mu \pi[f_1, \ldots, f_n]$.*

In particular, $\equiv^F_\mu$ is always compositional when μ is cofinal, so the (total) semantics $|\cdot|^F_\mu$ is compositional. And if two expressions of the same category (related by $\sim_\mu$) have *different* fregean values, this is witnessed by a corresponding difference of μ-values of complex expressions containing them. Hodges calls the combination of these two properties *full abstraction*. Moreover, the fregean values are *unique* in the sense that, if another total semantics ν has the corresponding properties with respect to μ, then $\equiv_\nu\ =\ \equiv^F_\mu$.[8]

The relation $\equiv^F_\mu$ refines $\equiv_\mu$ in the sense that if $e, f \in X$, then $e \equiv^F_\mu f$ implies $e \equiv_\mu f$ (use the unit polynomial). If μ was s-compositional to begin with, and is also *husserlian* in the sense that $e \equiv_\mu f$ implies $e \sim_\mu f$,[9] then the converse holds as well. In that case one can take (by choosing suitable representatives) the total fregean semantics to *extend* the partial μ to all terms.

Example 5 (*First-order logic*) Let E be the set of first-order formulas (in some given signature) and $X = dom(\mu)$ the subset of sentences, for which μ provides the usual values: with a fixed model $\mathcal{M} = (M, I)$, $\mu(\varphi) = 1$ iff $\mathcal{M} \models \varphi$.[10] μ is cofinal, s-compositional (replacing a true (false) sub*sentence* of a sentence with another true (false) sentence doesn't change the truth value in $\mathcal{M}$), and, for $\varphi, \psi \in E$, $\varphi \sim_\mu \psi$ iff $FV(\varphi) = FV(\psi)$, so μ is trivially husserlian. By the above, the fregean semantics can be taken to extend μ to all formulas.

But the fregean values of formulas with free variables, i.e. their equivalence classes under $\equiv^F_\mu$, aren't the usual tarskian values. Finding interesting semantic values is "where semantic theory takes off" (Hodges, 2011, p. 270). In the case of FOL, we already know what these values should look like: the function ν defined by

$$\nu(\varphi) = \langle FV(\varphi), [\![\varphi]\!]_{\mathcal{M}}\rangle$$

where $[\![\varphi]\!]_{\mathcal{M}} = \{s \in M^{FV(\varphi)} : \mathcal{M}, s \models \varphi\}$, has the desired properties (listed in footnote 8), and is essentially the usual compositional tarskian semantics.[11]

Example 6 (*Independence-friendly logic*) The language of Independence-Friendly Logic (IF) (in one of its versions, see Hintikka and Sandu (2011) or Mann et al. (2011)) is like that of FOL but now with additional quantifiers

---

[8] More precisely, consider these properties of a total semantics ν for E: (i) If $e \equiv_\nu f$ and $\pi[e] \in X$, then $\pi[f] \in X$; (ii) If $e \equiv_\nu f$ and $\pi[e], \pi[f] \in X$, then $\pi[e] \equiv_\mu \pi[f]$; (iii) If $e \not\equiv_\nu f$, then there is $\pi[x]$ s.t. either exactly one of $\pi[e], \pi[f]$ is in X, or both are and $\pi[e] \not\equiv_\mu \pi[f]$. It is clear that, if ν and ν' have properties (i)–(iii), then $\equiv_\nu = \equiv_{\nu'}$, and also that the fregean semantics has these properties.

[9] Hodges introduces this term in view of what Husserl writes about 'Bedeutungskategorien' in *Logische Untersuchungen*.

[10] μ could be given by the usual game-theoretic semantics for sentences; see the next example. Everything we say below generalizes to the case when $\mathcal{M}$ is instead a parameter of the relevant functions; see also Sect. 21.4.2.

[11] We tend to believe since Tarski's 1933 truth definition that this is the 'right' semantics. But at that time, introducing *assignments* was a non-trivial achievement, not something one could simply read off an intuitive notion of truth for sentences. Instead of considering assignments in $M^{FV(\varphi)}$ we could use (as Tarski did) assignments in M^{Var} to *all* variables. The reason for incorporating $FV(\varphi)$ in the meaning of φ is to ensure the Husserl property: formulas with distinct free variables should differ in meaning (cf. φ and $\varphi \wedge x = x$).

Hodges takes ν to be the 'usual' semantics in the 2011 paper, but his 2001 paper uses (with $\mathcal{M}$ as a parameter): $\nu'(\varphi) = \langle FV(\varphi), [\varphi]_{\mathcal{M}}\rangle$, where $[\varphi]_{\mathcal{M}} = \{\psi : \varphi \sim_\mu \psi \ \&\ \mathcal{M} \models \forall\overline{x}(\varphi \leftrightarrow \psi)\}$ ($\overline{x}$ lists the variables in $FV(\varphi)$). Of course $\equiv_\nu = \equiv_{\nu'}$, and ν' is definable from $\equiv_\nu$ (since $\nu'(\varphi) = \langle FV(\varphi), \{\psi : \psi \equiv_\nu \varphi\}\rangle$); so maybe ν is more deserving of the label 'usual'.

$$\exists x/\{y_1, \ldots, y_n\}$$

read intuitively as 'there is an x which is independent of $y_1, \ldots, y_n$ such that …'. The initial semantics μ for *sentences* was given in terms of games with imperfect information, but there is no obvious extension to formulas with free variables (since these games do not have an obvious notion of subgame). Nevertheless, μ is cofinal, husserlian, and can be shown to be compositional for sentences. So again the total fregean extension *exists*. But this time there is no well-known way to find 'natural' semantic values.

What Hodges does in Hodges (1997) is recursively define a new satisfaction relation $\mathcal{M}, S \models \varphi$, where S is a *set* of assignments (now often called a *team*), such that

$$\nu(\varphi) = \langle FV(\varphi), \{S \subseteq M^{FV(\varphi)} : \mathcal{M}, S \models \varphi\}\rangle$$

has the desired properties, so its associated synonymy is the fregean synonymy.

Sets of assignments can capture the (in)dependencies between variables characteristic of IF logic. The Dependence Logic of Väänänen (2007) uses essentially the same satisfaction relation as Hodges but for a language with *dependence atoms*, which obviates the need for slashed quantifiers. A different approach to the use of dependence atoms will be discussed in Sect. 21.7.2 below.

21.2.3 Discussion

Hodges' analysis is attractive for its sparseness, and seems to lay bare the essence of compositionality. Even so, questions remain. The initial μ-values represent a free initial choice, crucially affecting the resulting semantics, so the analysis does not produce the compositional fregean values out of nothing. And perhaps most importantly, the fregean values are defined holistically using the behavior of the whole language. But this is also a major virtue of the above style of analysis. Compositionality and Contextuality become two sides of the same coin, what we called earlier the 'internal' and 'external' perspectives on meaning are in harmony. One is reminded of the harmony in logicism between defining natural numbers as equivalence classes of equally large sets under bijections and as structured objects generated by arithmetical operations.[12]

Hodges' analysis of compositionality remains at the abstract level of the fregean values. As we have seen, however, the tarskian semantics for FOL and the new team semantics for IF logic cannot be read off these general considerations, which only tell us under which circumstances fully abstract semantics *exist*. The main issue

[12] More generally, Contextuality sounds more like a category-theoretic slogan: 'know an object by its interactions with similar objects', and as is well-known, such external characterizations often match more set-theoretic descriptions in terms of internal structure.

then becomes: what are interesting or useful denotations, and what are criteria for judging that? To get more of a grip on these issues, in what follows, we will discuss compositionality as a design principle in actual cases, identifying further strategies in designing semantic denotations, and eventually finding additional constraints on what counts as a good semantics.

21.3 Some Classical Challenges

Much of the history of logical semantics can be told as a series of responses to challenges to compositionality. In one version of this script, compositionality has to defend itself against various attacks from the (presumably, evil) realm of contextuality. As will be clear, this is not at all our view, since we stressed how compositionality and contextuality can co-exist without friction. However, there are still non-trivial challenges in how to *manage* this co-existence, and how much contextuality needs to be taken on board. In this section, we briefly survey a few earlier examples, while adding some new ones.

Intensions. Perhaps the first example is Frege's introduction of the distinction between Bedeutung and Sinn in response to the Morning Star—Evening Star Paradox. The Morning Star is the Evening Star, but the discovery by the Baylonians of this identity was not the discovery of the trivial identity that The Morning Star is the Morning Star. Thus, substitution of identicals fails in intensional contexts created by verbs like "discover". In modern terms, Frege's solution introduced *intensions* of expressions as denotations, in addition to the usual extensions, whenever compositional interpretation of a complex modal expression requires this. The same strategy can be seen at work in Montague's famous analysis of natural language, where predicate meanings are intensionalized throughout to deal compositionally with constructions like "John seeks a unicorn", or "The temperature is ninety but it is rising". In contemporary modal logic, intensions are often identified with functions from possible worlds (indices of evaluation interpreted in whatever way is germane to the purpose at hand) to truth values, objects, or other entities in the relevant models.

Higher types. A classical case of 'compositionalization' is Montague's use of generalized quantifiers to interpret noun phrases (a style of analysis from Categorial Grammar). This elegantly replaced earlier analyses like Russell's in Russell (2005), which (in effect) blamed their non-compositionality on the 'misleading' subject-predicate form of natural languages. With complex denotations in a type hierarchy for linguistic expressions, subject-predicate form could be maintained and compositionality restored.

Assignments in logic and language. Yet another well-known set of compositionality issues occurs with the phenomena of anaphora and binding in natural and formal languages. Tarski's semantics introduced the new parameter of *variable assignments*, in addition to models and truth, to account for the semantics of formulas with free variables. This move to an enriched setting returned in later work on natural language, but now in new ways.

DRT and dynamic semantics. In particular, the Discourse Representation Theory of Kamp (1981) analyzes anaphoric relationships in a more linguistically sensitive manner than Montague as driving a process of building successive discourse representation structures for linguistic expressions and texts. Typically, discourse representation structures involve markers for objects representing the anaphoric structure of the text. And when it turned out that the process of building such representations was not compositional, compositional solutions were found after all. For instance, in the dynamic semantics of Groenendijk and Stokhof (1991) expressions no longer denote sets of assignments as with FOL, but sets of pairs of assignments, viewed as state transitions in an evaluation process.[13]

To be sure, there is also an independent non-methodological motivation for adopting a dynamic semantics, namely, providing an account of meaning that records more features of actual language use, as in Austin's dictum:

"Words are what words do", Austin (1962).

Current dynamic semantics also take on board information update by speech acts, or information exchange between participants in communication, Nouwen (2015).

It should be noted that, as we shall see again and again in this paper, compositionality by itself does not force a choice here. A dynamic semantics need not replace existing semantics involving more static denotations. It might also be seen as modelling other aspects of language use such as the fine-structure of evaluation, or in other versions of dynamic semantics, the process of communication. Moreover, mathematically, both static and dynamic structure may be of interest. For instance, while the standard semantics for FOL involves sets (i.e., unary properties) of assignments, its dynamic semantics involves binary relations between assignments. Thus, we now have two sorts of algebras: cylindric algebras of standard denotations and relational algebras for dynamic denotations. The two realms have interesting connections, and developing them in tandem throws more light upon language than choosing just one.

At this point, we reach a bridge to another field. The dynamic semantics solution was inspired by the Hoare-style semantics of imperative programming languages, where programs denote the pairs of input and output states of their successful terminating executions. These and other parallels with computer science, known since the 1970s, bring us to a general issue that we think should inform modern discussions of compositionality more generally: semantic ideas about programming languages and computation can enrich more traditional discussions of compositionality in philosophy and linguistics.

Semantics of programs. The issues and techniques used in analyzing compositionality are not confined to logic and linguistics. Striking parallels were observed in Janssen and van Emde Boas (1981) with the semantics of programming languages,

[13] For the full story of compositionalizing DRT, see Janssen (2011).

where compositional design, in various guises, had been a desideratum in the structured programming methodology of Dijkstra, Hoare, and Scott-Strachey; see Harel (1984). For instance, Montague-style intensionalization was shown to be a good strategy for dealing with the semantics of arrays and other data structures. Janssen (2011) discusses program semantics in terms of mathematical results about initial algebras and polynomials in universal algebra. The earlier-noted coexistence of static and dynamic perspectives returns in computer science as the contrast between *denotational* and *operational* semantics for a programming language, where one often wants both styles plus an analysis connecting the two, Abramsky and Jagadeesan (1994). For Abramsky's powerful category-theoretic approach to compositional analysis and design of programming languages, see Sect. 21.8. In this richer setting, computing generalizes to interactive *game play*, which brings us to our last example.

IF logic revisited. Games are a persistent theme in logic and natural language. A famous compositionality dispute concerns Jaakko Hintikka's game-theoretic perspective on natural languages, first put forward in Hintikka (1973) and developed further in Hintikka and Sandu (2011). Hintikka argued that his IF logic is necessarily non-compositional, a claim that was refuted by Hodges' compositional team semantics for IF logic in the preceding section. Even so, there was also a second claim that game-theoretic semantics fits natural language better than tarskian semantics, since it proceeds 'outside–in', something that also fits with Abramsky's view of computation as unfolding interactive agency. Again this leads us to criteria for a 'good' compositional semantics. Is team semantics also a refutation of Hintikka's second claim? We shall return to these issues in the sections to come.

21.4 Currying

The compositional semantics mentioned so far, as well as many other cases, all work on a schema of (a) *introducing new parameters of evaluation*, and (b) *giving semantic truth conditions* for the given language in terms of these. This is obvious in the, by now, standard semantics of first-order logic or modal logic. But (a) and (b) keep arising. One more recent example is 'public announcement logic' for information update, Plaza (1989), Baltag et al. (1998), where the language contains a new kind of dynamic modalities that refer to what is true in models different from the initial one. A typical PAL truth condition now reads as follows:

(4) $\mathcal{M}, s \models [!\varphi]\psi$ iff $\mathcal{M}, s \models \varphi$ implies $\mathcal{M}|\varphi, s \models \psi$

where $\mathcal{M}|\varphi$ is the model $\mathcal{M}$ *restricted* to the states in which φ holds. In contrast with other familiar logical systems, here the current model is a parameter that cannot be held constant in the truth definition (just as the assignment parameter cannot be held constant in the FOL clauses for quantified formulas).

When more 'active' parameters are added, is the resulting parametric semantics compositional? To even ask this question, we a notion of compositionality that takes the presence of parameters into account.

21.4.1 Contextual Compositionality

The following generalization of compositionality has been around in the literature for a while (see e.g. Pagin and Westerståhl (2010a)). Let μ be a function taking expressions in E (terms in our syntactic algebra $\mathbf{S}$) as arguments and in addition a sequence $\overline{p} = p_1, \ldots, p_m$ of parameters, say, $p_i \in P_i$ for $i = 1, \ldots, m$. Again, μ can be partial in the first argument, so it is a function from $X \times \overline{P}$ to M, where $X \subseteq E$ and $\overline{P} = P_1 \times \ldots \times P_m$.

Definition 7 (*contextual compositionality*)

(a) μ is (contextually) *f-compositional* if, for every operation $\alpha \in \Sigma$ there is a corresponding operation r_α on M such that, whenever $\alpha(e_1, \ldots, e_k) \in X$ and $\overline{p} \in \overline{P}$, then $\mu(\alpha(e_1, \ldots, e_k), \overline{p}) = r_\alpha(\mu(e_1, \overline{p}), \ldots, \mu(e_k, \overline{p}), \overline{p})$.
(b) μ is (contextually) *s-compositional* if, whenever $\mu(e_i, \overline{p}) = \mu(f_i, \overline{p})$ for $i = 1, \ldots, n$, and $\pi[e_1, \ldots, e_n]$ and $\pi[f_1, \ldots, f_n]$ are both in X, we have $\mu(\pi[e_1, \ldots, e_n], \overline{p}) = \mu(\pi[f_1, \ldots, f_n], \overline{p})$.[14]

Again, (a) and (b) are equivalent when X is closed under subterms.

Ordinary compositionality is a special case of contextual compositionality. *Given* some parameters ranging over given contextual features, the semantic value function μ can be compositional, or not. In fact, very often it is *not* compositional in the above sense, even though μ is recursively defined. But in this case, strict (ordinary or contextual) compositionality can be enforced by lambda abstraction over a suitable set of parameters, under very general circumstances. We now proceed to make this observation precise.[15]

21.4.2 A Currying Recipe

Currying arguments of an n-place function F where $n > 1$ is essentially just abstracting over them. For example, if $F\colon P_1 \times \cdots \times P_4 \to M$ and the second and fourth

[14] The literature also considers a stronger version, without $\overline{p}$ as an argument of r_α. However (despite some claims to the contrary), this version has no corresponding natural substitutional version. In particular, it is *not* equivalent to the statement that whenever $\mu(e_i, \overline{p}) = \mu(f_i, \overline{q})$ for $i = 1, \ldots, n$, and $\pi[e_1, \ldots, e_n], \pi[f_1, \ldots, f_n] \in X$, $\mu(\pi[e_1, \ldots, e_n], \overline{p}) = \mu(\pi[f_1, \ldots, f_n], \overline{q})$.

[15] The role of currying for compositionality was first discussed in Lewis (1980), generalized in Pagin (2005) and Westerståhl (2012), and is elaborated still further here.

arguments are curried, the result is the 2-place function $F': P_1 \times P_3 \to M^{P_2 \times P_4}$ defined by

$$F'(p_1, p_3)(p_2, p_4) = F(p_1, \ldots, p_4)$$

So if we let $\mu(\varphi, \mathcal{M}, s) = 1 \Leftrightarrow \mathcal{M}, s \models \varphi$ in the PAL example above, then μ is not (contextually) compositional, but the function obtained by currying both $\mathcal{M}$ and s is compositional; this will follow from Fact 9 below.[16]

Similarly, as we have seen, currying the assignment argument, but not the model argument, in the standard Tarskian definition of satisfaction in first-order logic yields a compositional semantic function whose values, relative to a model, are (characteristic functions of) sets of assignments. Let us look at one more example to see the general pattern.

Example 8 (*Propositional Dynamic Logic*) In the two-component language of PDL, Harel et al. (2000), μ must interpret both formulas and programs, the latter as (characteristic functions of) binary relations between states. That is, $\mu(\varphi, \mathcal{M}, w_1, w_2) = 1 \Leftrightarrow w_1 = w_2$ & $\mathcal{M}, w_1 \models \varphi$, and $\mu(\rho, \mathcal{M}, w_1, w_2) = 1 \Leftrightarrow w_1 R_\rho w_2$. Some relevant defining clauses:

- $\mu([\rho]\varphi, \mathcal{M}, w_1, w_2) = 1$ iff
  $w_1 = w_2 \ \&\ \forall w'(\mu(\rho, \mathcal{M}, w_1, w') = 1 \to \mu(\varphi, \mathcal{M}, w', w') = 1)$
- $\mu(\delta_1; \delta_2, \mathcal{M}, w_1, w_2) = 1$ iff
  $\exists w(\mu(\delta_1, \mathcal{M}, w_1, w) = 1 \ \&\ \mu(\delta_2, \mathcal{M}, w, w_2) = 1)$
- $\mu(\rho^*, \mathcal{M}, w_1, w_2) = 1$ iff
  $\exists n \exists z_0 \ldots \exists z_n(z_0 = w_1 \ \&\ z_n = w_1 \ \&\ \forall i(i < n \to \mu(\rho, \mathcal{M}, z_i, z_{i+1}) = 1))$

Again, μ is not contextually compositional, but currying w_1 and w_2 restores compositionality. The reason is that w_1 and w_2 are 'shifted' in the inductive definition of μ, but $\mathcal{M}$ is not.

The general pattern behind these examples is this. μ is defined by an inductive parametric 'truth definition' with (apart from clauses for atoms) one clause for each syntactic rule $\alpha \in \Sigma$, which we write schematically as follows:

(5) $\quad \mu(\alpha(e_1, \ldots, e_k), \mathcal{M}, \overline{p}) = u$ iff $\Phi_\alpha[\ldots, \mu(e_i, \mathcal{M}, \overline{t_i}[\overline{p'}]), \ldots, \mathcal{M}, \overline{p}, u]$

Here $\overline{p} = p_1, \ldots, p_m$ are parameters and $\mathcal{M}$ is the relevant model parameter; it really should be one of the p_j but is made explicit here to make the format more recognizable. Φ_α is written in some suitable set-theoretic metalanguage.[17] The complex terms of the form $\mu(e_i, \mathcal{M}, \overline{t_i}[\overline{p'}])$ that may occur in Φ_α are to be understood

[16] There is a slight technical complication. Let *Mod* be the class of PAL-models of the form $\mathcal{M} = (W, \{R_a\}_{a \in A}, V)$, and let $|\mathcal{M}| = W$. μ as described is really a (total) function from $\{(\varphi, \mathcal{M}, s):$ $\varphi \in E$ & $\mathcal{M} \in Mod$ & $s \in |\mathcal{M}|\}$, which is not strictly speaking a cartesian product. It is trivial but slightly cumbersome to redefine the domain so that it becomes such a product. In the following we ignore this subtlety.

[17] It contains names for the various terms on the right-hand side of ((5)), but for ease of reading we haven't distinguished objects from their names here.

as follows. $\overline{t_i}[\overline{p'}]$ is a sequence of m (possibly) complex terms $t_{i1}[\overline{p'}], \ldots, t_{im}[\overline{p'}]$, in which parameters among $\overline{p'} = p'_1, \ldots, p'_{k'}$ (for some k') may occur. Here each p'_l is either one of $p_1, \ldots, p_m$ or a new parameter, which is bound inside Φ_α (by a quantifier or other variable-binding operator).

To help in formulating a general observation, let us say that a parameter p_j is *fixed* in Φ_α, if whenever p_j occurs in a term $\mu(\alpha_i, \mathcal{M}, \overline{t_i}[\overline{p}])$, we have $t_{ij}[\overline{p'}] = p_j$. If that is not the case, we say that p_j is *shifted*.

It should be fairly clear that this format covers most common truth definitions, in particular those exemplified above with the logical systems FOL, PAL, and PDL.[18] Now we have the following fact.

Fact 9 Let the semantics μ be defined with a recursive clause of the form (5) for each syntactic operator.

(a) If some parameter in some clause is shifted, then (under very general circumstances) μ is not contextually compositional.
(b) If μ' is obtained by currying all shifted parameters, and possibly other parameters as well, then μ' is contextually compositional. In particular, if all parameters are curried, or if no parameter is shifted, μ' is compositional.

Summing up, in most recursive truth definitions familiar from logic and formal semantics, the parametric meaning function is not strictly speaking compositional, but currying shifted parameters ensures compositionality, although at the cost of using higher-order semantic values.

21.4.3 Connections: Currying and Fregean Semantics

The compositional tarskian semantics for FOL is obtained by currying the assignment parameter. But it also resulted—up to synonymy—with the Lifting Lemma in Sect. 21.2.2 (see Example 5). There is in fact a natural connection between the two methods that may be worth spelling out.

In this subsection, let μ be a total reursively defined parametric semantics, and let μ' result from currying all the parameters: $\mu'(e)(\overline{p}) = \mu(e, \overline{p})$. As we know, μ' is compositional. For simplicity, we also assume that μ is 'husserlian', in the sense that if $\mu(e, \overline{p}) = \mu(f, \overline{p})$, then $e \sim f$.[19]

Next, for each tuple $\overline{p}$, define the unary function $\mu_{\overline{p}}$ by setting

[18] For example, consider the second PDL-clause above. E can be taken to be the union of the disjoint sets of formulas and programs, inductively defined from atomic proposition letters and program symbols, so μ is total. There is a syntactic operation $\alpha \in \Sigma$ such that $\alpha(\delta_1, \delta_2) = \delta_1; \delta_2$. The semantic clause then has the form $\mu(\alpha(\delta_1, \delta_2), \mathcal{M}, w_1, w_2) = 1$ iff $\Phi_\alpha[\mu(\delta_1, \mathcal{M}, \overline{t_1}[\overline{w'}]), \mu(\delta_2, \mathcal{M}, \overline{t_2}[\overline{w'}])]$ where $\overline{w'} = w_1, w_2, w$, w is quantified, and $\overline{t_1}[\overline{w'}] = t_{11}[\overline{w'}], t_{12}[\overline{w'}]$, where $t_{11}[\overline{w'}] = w_1$ and $t_{12}[\overline{w'}] = w$; similarly for $\overline{t_2}[\overline{w'}]$. Since $t_{12}[\overline{w'}]$ is not w_2, and $t_{21}[\overline{w'}]$ is not w_1, both w_1 and w_2 are shifted in the clause.

[19] We can write $e \sim f$ rather than $e \sim_\mu f$, since μ is total, so $e \sim f$ just means that e and f have the same category in the syntactic algebra **S**.

$$\mu_{\overline{p}}(e) = \mu(e, \overline{p})$$

The function $\mu_{\overline{p}}$ is in general *not* compositional. In fact, we have:

Fact 10 μ is (contextually) compositional iff each $\mu_{\overline{p}}$ is compositional.

Hodges' applications of the Lifting Lemma concern extending a partial compositional semantics to a total one. But the Lemma holds also when the given semantics is total, except in this case it has effect only when that semantics is *not* compositional, as with the $\mu_{\overline{p}}$. Recall the fregean synonymies $\equiv^F_{\mu_{\overline{p}}}$ from Definition 3. The following holds in general.

Fact 11 $\equiv_{\mu'} \; = \; \bigcap_{\overline{p} \in \overline{P}} \equiv^F_{\mu_{\overline{p}}}$

This connects the curried compositional semantics with the fregean one presented in Sect. 21.2, be it in a rather weak way. Here is a concrete example showing how, with some extra assumptions, the connection becomes much stronger.

Say that our language *contains propositional logic* if it has a category of formulas (using $\varphi, \psi, \ldots$ for these) whose values under μ, relative to a model $\mathcal{M}$ and other parameters, are 0 or 1, and μ respects tautological consequence.[20] Also, a *universal operator* U is a unary formula operator such that $\mu(U\varphi, \mathcal{M}, \overline{p}) = 1$ iff for all $\overline{q}$ in $\mathcal{M}$, $\mu(\varphi, \mathcal{M}, \overline{q}) = 1$. Now fix $\mathcal{M}$ and parameters $\overline{p_0}$, and let $\mu_0 = \mu_{\mathcal{M}, \overline{p_0}}$. Again, μ_0 is usually not compositional.

Fact 12 If μ contains propositional logic and has a universal operator, then

$$\equiv_{\mu'} \; = \; \equiv^F_{\mu_0}$$

So in this case the curried semantics *is* the fregean semantics (up to synonymy). FOL is an example satisfying the assumptions; another is intensional logic with a universal modality. Relative to a model $\mathcal{M}$, the compositional curried semantics whose values are sets of assignments is—up to synonymy—what you get by applying Hodges' construction to the non-compositional semantics μ_0, where s_0 is a fixed assignment and $\mu_0(\varphi) = 1$ iff $\mathcal{M}, s_0 \models \varphi$.

21.5 General Issues in Compositional Semantics

We now move to state some rules of thumb that seem to apply when searching for a good compositional semantics. Our discussion results in a check-list of points to keep in mind that occur in many concrete cases.

---

[20] That is, writing $[\![\varphi]\!]^{\mu}_{\mathcal{M}}$ for $\{\overline{p} \colon \mu(\varphi, \mathcal{M}, \overline{p}) = 1\}$, we have:

$$\text{If } \Gamma \vdash_{\mathrm{PL}} \varphi, \text{ then for all } \mathcal{M}, \; \bigcap_{\psi \in \Gamma} [\![\psi]\!]^{\mu}_{\mathcal{M}} \subseteq [\![\varphi]\!]^{\mu}_{\mathcal{M}}.$$

21.5.1 The Danger of Over-Interpretation: Do Not Invert

A striking phenomenon found with many proposed compositional semantics is a tendency to invert the moral of the solution. For instance, in propositional modal logic, propositions denote intensions, i.e., functions from indices to truth values. It then seems reasonable to identify propositions with sets of worlds. And once this is done, it is tempting to identify the set of propositions with the complete set of all sets of worlds. But this is unwarranted. The compositional solution merely says that propositions correspond to some sets of worlds: letting in all of them is a major step to a second-order perspective on the semantics.

Here, and in many similar well-known semantics, a good recommendation is the following counsel of caution:

(I) *Do Not Invert.*

Indeed, for modal logic, there is a philosophical rationale to merely correlating the intuitive notion of a proposition with a set of worlds without making a total converse identification that would produce huge amounts of undefinable and unusable propositions.[21] And there is also a clear mathematical rationale for the same reticence. The representation theory of modal algebras works with 'general frames' that have merely a designated set of sets of worlds as algebraic values, not necessarily the full power set.[22] Similar points can be made about assignments in first-order semantics and functions from variables to objects, where there is no need to assume that the full function space of all maps from variables to objects is available as indices of evaluation. The latter case study will be discussed at length in Sect. 21.6.3 below, after first pointing to some further reasons for heeding the Do Not Invert injunction.

21.5.2 Avoiding Triviality

Every proposal for compositionalization, even when of the non-inverted kind, needs to face a threat of triviality. A compositional semantics for any given language can almost always be found with enough industry in manipulating semantics or syntax. Thus, a very general precaution is this:

(II) *Don't make too many, or too few, meaning distinctions.*

There might be pre-theoretic intuitions about *how many* meanings should be available, or put in terms of expressive power, (at least) how many distinctions the given language should be able to make. At the extreme end of making too few distinctions:

[21] Likewise, in current 'set liftings' of possible worlds to 'state-based' hyper-intensional semantics, there is no need to assume that all sets are available as states, or structured situations.

[22] Stated in other terms, when quantifying over propositions in modal logic, Henkin models may be preferable to full second-order models, cf. Andréka et al. (2014).

if every expression means the same, compositionality holds trivially. And at the other end: if all expressions mean different things, compositionality is again trivial. Semantics of the latter kind have in fact been proposed, for example, by making the expression itself a part of its meaning. This holds for one suggestion in Zadrozny (1994); see Westerståhl (1998) for further discussion.

We note in passing that the last observation also cautions us that compositionality in itself is no guarantee for learnability, communicability, or any other of the good properties linguistic meanings are supposed to have. A language with no synonymy relation except identity may well be completely unlearnable. Compositionality is at best a *necessary* condition.

21.5.3 The Role of Impossibility Results

Avoiding triviality is good, but how? The following positive constraints are sometimes at work in the search for a new semantics to replace an old one, e.g. because it was partial, or not compositional.

(III) *For some subset X of expressions, the proposed semantics should agree with the benchmark given by some prior semantics for X.*
(IV) *The values of the proposed semantics should be of a specified kind.*

(III) applies to Hodges' set-up in Sect. 21.2. It can be taken in a strong and a weak sense: (III.i) the new semantic *values* of expressions in the set X should be the same as the old ones, or (III.ii) the new semantics should preserve given *synonymies*, or non-synonymies, in X. Together with (IV), mathematical impossibility results can sometimes guide the search for a good semantics. We look at three familiar examples.

Example 13 (*IF logic in a bit more detail*) Let E be the set of IF-formulas, X the subset of IF-sentences, and $\mu_{\mathcal{M}}(\varphi)$ the truth value of a sentence φ in a model $\mathcal{M}$ according to Hintikka's game-theoretic semantics. As we saw in Example 6, the existence of a compositional (fregean) semantics for all of E which agrees with $\mu_{\mathcal{M}}$ on sentences (in the sense of (III.i)) is then guaranteed. Cameron and Hodges (2001) showed that here constraint (IV) makes a difference: there is no compositional semantics for E which agrees with $\mu_{\mathcal{M}}$ on sentences and whose values on formulas are sets of assignments in $\mathcal{M}$. In particular, it showed that if ν is total, compositional, and agrees with μ on sentences, then for any $n \geq 2$ there is a model $\mathcal{M}$ with n elements such that the number of distinct values $\nu_{\mathcal{M}}(\varphi(x))$ of IF-formulas with one free variable is (much) greater than 2^n. Thus, these values cannot all be subsets of M.[23]

This impossibility result in a sense vindicates Hintikka's claim that IF logic has no compositional semantics. But note how strong constraint (III) is here. In effect,

[23] Galliani (2011) generalizes this result to infinite models; this needs an extra requirement on ν, to do with how $\nu_{\mathcal{M}}$ relates to $\nu_{\mathcal{M}'}$ when $\mathcal{M}$ and $\mathcal{M}'$ are isomorphic.

it builds the expressive power of sentences of IF logic (which is that of Existential Second-Order logic), and hence the IF consequence relation, into any candidate semantics. If the main goal is to extend FOL in order to deal explicitly with (in)dependence between variables, that may already seem too much. We sketch an alternative, more sensitive approach in Sect. 21.7.2.

Finally, observe that if we change the force of (III) to just require agreement with μ on the set Y of FOL-formulas—which holds for the team semantics of IF logic—there is no similar impossibility result. In fact, various uninteresting total compositional extensions of μ, satisfying this version of (III) and also (IV), then exist. Note that Y is not cofinal, rather, E is an *end-extension* of Y. As Hodges (2001) observes, it follows, since μ is also husserlian, that the *one-point extension* μ^1 of μ, which gives formulas in Y the old values, but all formulas in $E - Y$ the same distinct value $*$, is in fact the fregean extension (up to synonymy).[24]

Of course, no one would consider μ^1 a useful semantics. Moral: by themselves, conditions (III) and (IV) do not guarantee non-triviality.

Example 14 (*Modal logic*) Next, let E be the set of basic modal formulas, generated by

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \Box\varphi$$

Here we might take (III) to require that $\neg$ and $\wedge$ preserve their standard meaning. A problem with this, which did not arise in the IF example, is that we haven't made clear what a *model* is supposed to be. Until we do, neither is the 'standard' meaning of $\neg$ and $\wedge$ clear. For example, if we generalize the two-valued PL semantics to n-valued semantics, which is a way to implement (IV), there are many candidates for a standard meaning of these connectives. But this is actually not a problem, since there are well-known impossibility results: most modal logics are not determined by any finite-valued truth tables, Ballarin (2017). Historically, a many-valued approach to modal logic was indeed attempted (by Łukasiewicz), interpreting $\Diamond\varphi$ using a separate truth value for 'possible', but the impossibility results put an end to that endeavor.

Note, however, that we have now moved from semantic constraints to logical ones. If a logic is seen as a set of (valid) formulas closed under certain inference rules, or more generally a consequence relation, an impossibility result of the kind mentioned says that no compositional semantic function of a particular kind is *sound and complete* for that logic. This is a strong interpretation of the following kind of constraint that we formulate more generally and loosely:

(V) *A new compositional semantics must respect some given (non-)inferences.*

We will return to this constraint in much greater detail in the next section.

Example 15 (*Intuitionistic logic*) The case of intuitionistic propositional logic IPL is similar to the preceding example, except there is no addition to the syntax, we

[24] Without the Husserl property, a total compositional extension can also be shown to exist, but this is less trivial; see Westerståhl (2004).

still have: $\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \rightarrow \varphi)$. Classical models (bivalent truth value assignments) are unacceptable because they yield validities deemed false. And already Gödel (1932) proved that no finite-valued truth tables determine IPL.[25] Again, this is a strong use of (V). A weaker use might be to require e.g. that $\varphi \wedge \neg\varphi$ is not valid, or that IPL validity agrees with PL validity for formulas without $\vee$ and $\rightarrow$. All of these constraints concern inference. Semantic requirements like the above (III) seem harder to make sense of in this case, absent a common notion of model.

The negative results mentioned in the last two examples ruled out a particular kind of compositional semantics, showing that finite 'matrices' do not determine these logics. On the positive side, study of the 'matrix method' using infinite matrices led to (compositional) *algebraic* semantics; see Goldblatt (2006), Ballarin (2017).

Coda: other perspectives.[26] Our discussion highlighted inference, but other technical perspectives exist. Van Benthem (1986) discussed a compositional extension problem similar to the one in Sect. 21.2.2: when can a new syntactic operation be added to a homomorphically interpreted syntactic algebra satisfying certain equational constraints, in a way that does not 'disturb' the given interpretation? The setting of the results there was classical model theory; it would be interesting to rethink them in the more abstract Hodges framework.

21.5.4 Discussion

In order to evaluate a particular compositional solution, more specific versions of (III)–(V) must come into play. If we view Compositionality as an empirical claim, one consideration is evident: does the proposed semantics mirror how the given empirical phenomenon works? An instance of this is Hintikka's earlier-mentioned criticism of proposed compositional semantics for his IF logic. On his view, it is an evident empirical feature of natural language interpretation that it proceeds outside–in, so any compositional semantics suggesting that the process works inside-out is off the mark. We return to the interesting contrast between outside–in and inside–out accounts in game semantics in Sect. 21.8, but in principle we find this sort of disagreement entirely legitimate.

[25] Interestingly, both IPL and basic modal logic still stay close to finite truth tables in that they have the *finite model property*: each satisfiable formula is true at some point in some finite model. Locally, a set of finite truth tables still works.

[26] Other constraints than (III)–(V) have been considered in the linguistic literature, from strict requirements on syntax—e.g. that the only operation is concatenation, possibly also 'wrapping', or that grammar rules should be context-free—to specific ideas about the corresponding semantics—e.g. that concatenation should correspond to function application. (Zadrozny (1994) achieves this, at the cost of using non-wellfounded sets, but also of making the meaning function 1–1, thereby violating (II).) These constraints are closely tied to specific theories of the syntax-semantics interface in natural language, and will not be discussed here.

So, did Hodges' team semantics refute Hintikka's claim that IF logic is non-compositional? As we have seen, this is no simple yes/no question, and pondering answers highlights issues about what a good semantics is meant to do. A strict application of (III) and (IV) rules out team semantics, but we have noted that one may keep (IV) (semantic values as sets of assignments) while relenting a bit on (III), and still get a compositional semantics. This however changes both syntax and logic, not much, but enough to make the logic decidable; Sect. 21.7.2 below has details. And the outside–in versus inside–out issue is yet a further aspect. So the jury seems to be out on the merits of Hintikka versus Hodges. The best answer for now is: It depends.

In general, the plausibility of type (IV) constraints must come from independent prior intuitions about what meanings are supposed to be (or do). For Tarski's assignments, a credible case can be made that these correspond to our intuitive picture of the contexts in which linguistic utterances takes place; cf. Discourse Representation Theory, or the situation-semantic view of utterance situations as containing realistically interpreted 'anchors', Barwise and Perry (1999). But, say, whether modal intensions match Frege's intuitive pre-theoretical notion of Sinn is already a much more debatable issue.

Finally, more general methodological considerations about utility, fruitfulness, or explanatory power are clearly relevant too when evaluating a proposed compositional semantics. Making these precise raises difficult perennial questions in the general philosophy of science, which we shall not go into here. Instead, in the next section we continue with major logical dimensions in judging proposed compositional semantics, in terms of language design, inference, and general views of what a semantics should deliver.

21.6 Compositional Semantics and Inference

21.6.1 Empirical Perspectives

Requirements (V) and the closely related (III) above have a solid tradition. Around 1980, Barbara Partee emphasized that the purpose of a semantics of natural language is not just to explain how expressions acquire their meanings, but also to account for our prior intuitions about valid or invalid inferences. Language users have intuitions about inference (one reason why we can teach logic courses without mass revolts), though they will usually be partial, rather than total. Any proposed semantics should then at least agree with these intuitions, while providing a plausible extrapolation to inferences not covered by the original intuitions. In Partee's terms:

> Inferences are part of the data.[27]

[27] We leave aside the trickier philosophical issue of what it means for a formal semantics to 'explain' the prior inferential intuitions.

We have already discussed different ways of taking this. One is that particular concrete sentences are to imply each other, say, "John walks slowly" should imply "John walks". Similarly for preservation of (non-)synonymies. Of course, which sentences are actually synonymous is always up for debate. Some scholars tend to the view that, with the right sense of "synonymous", there are almost no non-trivial synonymies in natural languages. As we have seen, this trivializes the compositionality problem. We need notions of synonymy that, ideally, support the good work compositionality is supposed to do. But there are still many choices. One might be to identify it more or less with logical equivalence, but more fine-grained ('hyperintensional') variants also make sense. For example, one might well consider the formulas p and $p \wedge p$ to be synonymous, but deny that $\neg(p \wedge q)$ is synonymous with $\neg p \vee \neg q$.[28]

However, preserving particular (non-)synonymies may seem too modest, and it also does not suggest general design features for a semantics. Intuitions in the semantic literature are often stated in more general schematic terms such as the following principle of upward Monotonicity for "all":

> From any sentence of the form "All A are B", one may infer that
> "All A are C", if C is a 'weaker' predicate implied by B.

This principle is intelligible to language users, it covers infinitely many cases, and it puts clear constraints on the semantic interpretation of the quantifier expression "all", Peters and Westerståhl (2006). Likewise, intuitively, some schema may be judged invalid, in which case the proposed semantics should produce at least one counterexample (and probably even more: a general explanation of the non-validity).

Van Benthem (1986) outlines a formal approach to the preservation of (non)-inference in terms of the existence, or lack thereof, of compositional translations of the source language, equipped with a given partial inference relation, and a partial non-inference relation, into a target formal language with standard notions of consequence and non-consequence. The author shows that the risk of triviality lurks here as well: without significant constraints, such translations may always exist.

It would be interesting to analyze the entanglement of compositionality and intuitions about inference in natural language at the abstraction level of Hodges' minimal requirements for compositionality presented in Sect. 21.2 above. However, we leave this matter for further investigation.

Contextuality. As a final comment, the present topic of preserving intuitions about valid and invalid inference patterns in compositional semantics again demonstrates our theme of the entanglement of compositionality with *contextuality*. Inference patterns have infinitely many instances going far beyond single expressions of the language. And more importantly, inferences from a single given sentence will usually go beyond subformulas, or even more general syntactic 'parts' of that formula. In other words, looking for a compositional semantics under inferential constraints on the valid reasoning to come out of the semantics takes on board contextuality, in both senses discussed in Sect. 21.1.

[28] Peters and Westerståhl (2006 Sect. 11.3) has a detailed discussion of various notions of synonymy.

21.6.2 A Logical Systems Perspective

Any semantics supports a notion of universal validity and valid consequence. Often, it supports even more than one natural candidate, as different notions of consequence can be defined over the same class of models with respect to the same language, Blamey (2001). While in cases such as the semantics of classical first-order logic, people often just accept the received 'standard' logical system, another viewpoint is possible. One might think that the valid consequences to come out of the semantics are themselves a criterion for judging whether the proposed compositional solution achieves what we want it to do. This is of course just the intent of Point (V) in the general discussion of Sect. 21.5, which highlighted the important, though usually only partial, nature of prior inferential constraints. We now take this theme a bit further in terms of logical systems.

A compositional semantics must always be analyzed critically for the validities it produces: as long as there is some latitude in setting up the semantics, these are not 'forced' uniquely. Especially with the earlier-mentioned schematic view on inferential constraints, Point (V) leads us to thinking in terms of full-fledged *logical systems*. And in such a perspective, further system considerations may come in beyond accounting for individual valid or non-valid schemata, especially when prior intuitions are incomplete, and admit of precisification and extrapolation. Such considerations may be general system properties of intelligibility blocking opaque or ad-hoc ways of coding up given inferential intuitions. But also further criteria such as desirable theoretical meta-properties of the system, or more practical computational complexity of the proposed proof system can play a legitimate role. We now proceed to a case study for this system-oriented style of thinking, which also brings further issues of its own.

21.6.3 Case Study: Tarski Semantics Revisited

The Do Not Invert maxim (I) also applies to the semantics of first-order logic. While predicate-logical formulas need variable assignments as indices of evaluation, the requirement of compositionality per se does not force us to assume that the full space of all functions from variables to objects is available as assignments. Letting go of this inversion leads to the broader class of 'generalized assignment models'

$$\mathcal{M} = (D, I, A)$$

consisting of a first-order $\mathcal{M} = (D, I)$ plus a set A of 'available' assignments. Motivations for this move include algebraic simplicity (Németi, 1985), but also the natural phenomenon of dependence, which is beyond the range of standard first-order semantics: with 'gaps' in the function space, changes in the value for one variable may be correlated with changes in value for another variable. Standard Tarski models are not lost, being the special case where $A = D^{VAR}$. In these generalized models, we set

$$\mathcal{M}, s \models \exists x \varphi \text{ iff there exists } t \in A \text{ with } s =^x t \text{ and } \mathcal{M}, t \models \varphi$$

where $s =^x t$ says that $s(y) = t(y)$ for all variables y distinct from x.

The result is a perfectly compositional semantics for the first-order language. Even so, it has some unusual features. For instance, given the preceding stipulation, the truth value of a formula φ may depend on values that the current assignment gives to variables that do not occur in φ; see below.[29]

First-order logic deconstructed. The motivation for the preceding semantics had to do with the theme of inference. It might be expected that the basic logic of quantification is simple, like that of the Boolean operations, and the undecidability of first-order logic then comes as a surprise. So, what are its causes? An analysis requires decoupling the goal of a compositional semantics from the high cost in complexity of the notion of validity. In the above analysis, the latter complexity arises from the negotiable mathematical 'wrappings' of Tarski's definition, namely, the assumption of a regular second-order mathematical object, viz. the presence of the full function space D^{VAR}. After all, we know from Gödel's theorems that the first-order theory of regular mathematical structures may be complex. Here is how this deconstruction works.

Fact 16 The logic CRS consisting of the first-order validities on generalized assignment models has a simple complete axiomatization and is decidable.

On this basis, the semantic content of additional first-order validities can be determined. A non-valid principle in generalized assignment models such as $\exists x \exists y \varphi \to \exists y \exists x \varphi$ expresses *independence* of the variables x, y. More precisely, it imposes the following condition on the set A of admissible assignments: whenever $s =^x t =^y u$, there exists an assignment v with $s =^y v =^x u$. This imposes an existential confluence property on A that facilitates embedding of undecidable tiling problems, Blackburn et al. (2001). A body of theory exists for CRS, Andréka et al. (1998), including connections with the Guarded Fragment of first-order logic, arguably making the above compositional semantics 'useful' and 'interesting'—if only, as a way of throwing new light on what makes standard Tarski semantics tick.

Discussion. The preceding is not a defense of CRS, which for us just serves as a case study of 'loosening up' the uniqueness of a compositional semantics and the tight fit with one logic to be validated by it. There are many more examples of this phenomenon. But there is still more to this particular case study.

New aspects of compositionality? As a final point of interest, some features of CRS suggest further aspects to compositionality not brought to light so far. Consider the following objection to CRS semantics. *Locality* fails, in that truth values of formulas φ may depend on values of variables that do not occur in φ. Locality is a requirement

[29] More abstract modal models generalize these first-order models still further to 'states' and appropriate relations between them for making compositional interpretation work.

close to compositionality, though one not implied by our analysis so far: the semantics of CRS is perfectly algebraic. However, Locality is reinstated in the system LFD to be discussed in Sect. 21.7.2: there, values of the current assignment for the variables in a formula determine its truth value. Failures of Locality occur in other logics, too, such as IF logic, or extensions of PAL such as 'arbitrary announcement logic'. Repairs are sometimes made, but not always, since interesting new phenomena may come to light. For instance, the failure of Locality in IF logic models the natural phenomenon of *signaling* in games. This was initially treated as a problem, but is now seen as a desirable characteristic feature of the IF language, see Mann et al. (2011).

21.7 The Other Side of the Coin: Language Design

A compositional semantics is usually chosen so as to match a given language. But this language need not be sacrosanct: something may give on both sides.

21.7.1 Patterns and Issues in Language Change

Changing the original language. Compositional solutions have also changed initial ideas for the syntax of natural language, Janssen (2001), and the same is true for programming languages. However, this phenomenon seems largely a matter of avoiding getting trapped in ill-formulated problems in the first place.

Improved fit to the chosen models. A more interesting role for language redesign arises even when a compositional semantics is successful by the criteria discussed so far. Once an appealing semantics is there, with well-motivated structures and truth conditions, its very success generates a question. Is the original language the best medium for describing these structures, bringing out their crucial structure? Or can it be extended with new notions that are interpreted in the already established compositional style?

CRS is a case in point. With restricted sets of assignments A, there is now room for significant *polyadic quantifiers* $\exists \mathbf{x}\varphi$, which say that there is an available assignment different from the current one at most in its values for the $\mathbf{x}$ such that φ holds. Given the failure of $\exists x \exists y \varphi \rightarrow \exists y \exists x \varphi$, this no longer reduces to iterated single quantifiers as was the case on standard models.

In practice, this now means that CRS has a larger repertoire for formalizing concrete arguments couched in natural or scientific language—and it would be of interest to see how it fares on traditional formalization projects carried out at a time when FOL was still the only game in town.

There are many further examples of the continuing interplay of language design and semantic analysis. For instance, in the semantics of temporal expressions in natural language, flows of time were introduced which then themselves suggested new temporal and modal operators, whether or not realized already in natural language, Goranko and Rumberg (2020). Similar language redesign occurs in temporal logics for computational processes whose expressive power may go beyond the original needs of some specific programming practice, Stirling (1993). For another example, logics interpreted over information states tend to 'split' classical vocabulary, witness the Boolean and compositional conjunctions of relevant or linear logic, which reflect the new options available in the semantics. A final example is epistemic logic with its new notions of common and distributed knowledge that go beyond the original epistemic language originating in philosophy, Rendsvig (2016).

Language redesign is an ongoing process bringing to light legitimate options. In particular, when a compositional semantics has been found introducing a family of new parameters, one may or may not make these parameters explicit in the syntax of the logic. And thus, for instance, in the 1970s, a long debate raged as to whether all first-order properties of temporal flows should be made explicit in tense logics for describing discourse in natural language, (van Benthem, 1977).

Implicit versus explicit. This language redesign process is a force for new system building, but it may take some time to kick in before the conservativeness inherent in the program of 'giving a semantics' for a given initial language is overcome. For instance, when one only thinks of finding a 'topological semantics' for the basic language of intuitionistic or modal logic, it may take a long time before one sees that formalizing elementary reasoning in mathematical topology quickly needs notions beyond those simple languages, and the same is even more true when topology gets mixed with geometry. And indeed, it took a very long time before such extensions started being studied: cf. Aiello et al. (2007).

Language design interacts with valid inference. Likewise, one may view the contrast between intuitionistic logic and epistemic logic as being one of how much structure of information models is made explicit in the language. This again illustrates how issues of language design are entangled with the validities one endorses. Intuitionistic logic is weaker than classical logic, but the very absence of classical laws like Excluded Middle encodes significant features of an information- or knowledge-based view of truth: silence is informative. By contrast, epistemic logic has explicit operators for knowledge of agents, but with these in place, the base logic can remain classical, based on the traditional notion of truth. A general study of the 'implicit' versus the 'explicit' methodology, and the many new questions to which this gives rise, is made in van Benthem (2019).

Natural versus formal language. Of course, it is easier to redesign formal languages than natural languages that have evolved through history in a community of users.[30]

[30] Programming languages are an intermediate case: they have been designed at some point in history, and can be redesigned in principle, but once they have an established community of users, redesign becomes harder, and ad-hoc patches may be the norm.

Even so, one might speculate to which extent natural language is malleable, too, perhaps even by theoretical semantic considerations.

21.7.2 *Case Study: From CRS to Modal Dependence Logic*

Several of the above themes concerning language redesign are clarified by the following case study, which also shows how ideas in existing established semantics can be reassembled in surprising ways, and offers yet another illustration of the entanglement of compositional interpretation with considerations of inference.

Explicit dependence atoms. Language redesign can have many motivations. A concrete case continues with the earlier topic of generalized first-order semantics in Sect. 21.6.3. As an analysis of dependence, the logic CRS is 'implicit', van Benthem (2019). It merely reinterprets the existing language of first-order logic, and locates information about (in-)dependence of variables in failures of classical laws. An alternative would be to introduce explicit *vocabulary for dependence* on the same models. For this purpose, one can use atoms $D_x y$ for global dependence, interpreted in generalized assignment models $\mathcal{M}$ as saying that, if two assignments $s, t \in A$ agree on the value of x, they also agree on that of y. To find a compositional semantics for this language, Väänänen (2007) adopts Hodges' 'team semantics' for Hintikka's IF logic (cf. Sect. 21.2.2). The result is a second-order dependence logic quantifying over sets of assignments and lifting propositional connectives to this setting, which is undecidable, indeed non-axiomatizable, though some fragments are better-behaved. It is often suggested in the recent literature that this second-order system is the favored outcome of compositional analysis.

A modal approach. However, our earlier point returns. Compositionality can often be achieved in various ways, and the complexity of the resulting logic may guide our choice. The following logic of functional dependence LFD takes *local*, not global, dependence as its basic notion, where $D^s_x y$ holds if any assignment t agreeing with s on the value of x also agrees on that of y. LFD uses the following syntax: $\varphi ::= Px_1 \ldots x_k \mid D_X y \mid \neg\varphi \mid \varphi \wedge \varphi \mid D_X \varphi$, where X is a finite set of variables. Models are still generalized assignment models $\mathcal{M} = (D, I, A)$, where I maps predicate letters P into predicates of the same arity over D. The two highlights of the truth definition are as follows:

$$\mathcal{M}, s \models D_X y \ \text{ iff } \ D^s_X y$$
$$\mathcal{M}, s \models D_X \varphi \ \text{ iff for all } t \in A \text{ with } s =_X t, \mathcal{M}, t \models \varphi$$

Here $s =_X t$ means that s and t agree on the variables in X. So, evaluation takes place, modal-style, at single assignment inside sets of assignments, where the basic quantifier or modality $D_X \varphi$ says that the current values of the variables in the set X fix the truth of the formula φ. Global dependence and global modalities are definable from local ones using the fact that the special case $D_\emptyset$ expresses the universal modality quantifying over all assignments in A.

As mentioned, *locality* holds in LFD, in the following sense: Define the *free variables* of a formula by the following recursion: (a) $free(Px_1 \dots x_k) = \{x_1, \dots, x_k\}$, (b) $free(\neg\varphi) = free(\varphi)$, (c) $free(\varphi \wedge \psi) = free(\varphi) \cup free(\psi)$, (d) $free(D_X y) = free(D_X \varphi) = X$.[31] Then:

$$\text{If } free(\varphi) \subseteq X \text{ and } s =_X t, \text{ then } \mathcal{M}, s \models \varphi \Leftrightarrow \mathcal{M}, t \models \varphi.$$

Here is a key feature of this analysis of dependence language, Baltag and van Benthem (2021).

Fact 17 Validity in LFD is decidable.

LFD also has Hilbert- and Gentzen-style axiomatizations supporting Interpolation and Beth definability theorems. As for the above theme of language redesign, the language supports extensions with identity atoms, modalities for independence, and dynamic modalities for model change. In this manner, LFD can analyze notions of dependence in topology, dynamical systems, and games.[32] But whatever the merits (and drawbacks) of LFD, for us here, it just illustrates the interest in rethinking of what look like definitive compositional semantics.

Thus once more, the needs of providing a compositional semantics do not fix one unique logic for an important area of reasoning, in this case, for dependence. Further considerations matter, of the sort identified in the above analysis.

21.8 Perspectives from Games

This paper has provided a broad canvas for understanding compositionality as it functions in logical semantics at the interface of linguistics and philosophy. In particular, we identified some explicit design patterns and criteria for judging success that occur across the board, and that deserve attention.

In this section, we connect up with a major feature of Abramsky's work: its crucial use of *games*. What follows is a light discussion of some key features of the long-standing game perspective in logic, language and computation, and how it might affect received views of compositionality in philosophy and linguistics. In this setting, we will sketch some distinctive features of Abramsky's game semantics for programs, which is in the tradition of proof theory and categorical logic rather than model theory. However, a full discussion doing justice to this program with its many ramifications would require a separate paper.

[31] Thus, not even the notion of a free variable is written in stone.

[32] Here concrete notions of dependence add additional valid principles. For instance, dependence in vector spaces satisfies the Steinitz Principle $D_{X \cup \{y\}} z \to (D_X z \vee D_{X \cup \{z\}} y)$.

21.8.1 Games in Logic, Language, and Computation

We start with some basic background about interfaces of logic and games.[33]

Games in logic. Games have long been used in logic for a variety of tasks. Perhaps the most basic one of these is *semantic evaluation* of formulas in given models. Originally developed for FOL in the 1960s, semantic games now exist for modal logic, fixed point logics (such as the μ-calculus and the logic μ(FOL) extending FOL with monotonic fixed points), Gurevich and Shelah (1986), Kolaitis and Vardi (1992), IF logic and dependence logics, resource sensitive formalisms such as linear logic, but also logics with transfinite conjunctions, or infinitely deep alternations of quantifiers.

Evaluation games usually have two players: Proponent (Verifier, Myself) and Opponent (Falsifier, Nature) with opposite goals (establishing the truth or falsehood of the formula). Also, most games are sequential: only one player moves at any stage, as dictated by the syntax of the formula: the leading propositional connective, modality or quantifier determines the turns. Play is typically finite, ending when atomic formulas are reached (with the winner determined by the truth value of the atom), but games with infinite play also occur, e.g., in languages that contain fixed-points or infinitely deep formulas, in which case a winning rule has to be specified for the infinite plays. Evaluation games are usually games of perfect information, though famously, imperfect information is needed in the games that match the Hintikka-Sandu IF logic or Väänänen's DL, two extensions of classical first-order logic discussed in earlier sections.

Typically, logical evaluation games have the following basic property:

A formula φ *is true in a model* M *iff the Verifier has*
a winning strategy in the φ*-game played w.r.t.* M.

Likewise, falsity matches the existence of a winning strategy for the Falsifier. Thus, *strategies* are a key notion here, which may be viewed as more fine-structured 'reasons' for the truth or falsity of the formula.

In addition to games for evaluating formulas, there are other important logical games. In particular, *Ehrenfeucht-Fraïssé games* are widely used in model theory for analyzing the expressive power of logical languages, and unlike evaluation games, they analyze two given models in parallel without being guided by the syntax of one particular formula. Winning strategies for the 'Duplicator' in these games correspond with structural analogies between models, such as potential isomorphism up to some depth, while winning strategies for 'Spoiler' match up with formulas true in one model and false in the other.

More relevant to us here, in addition to semantic scenarios, there are two-player games for analyzing argumentation. Formal *proof games* were initiated by Paul

[33] For a much more comprehensive survey, including contacts with game theory, see van Benthem (2014).

Lorenzen, Lorenzen (1955), in a study of the foundations of logic, and the resulting 'dialogical paradigm' has had an influence reaching into philosophy and argumentation theory. These games, played between a Proponent of the conclusion, and an Opponent granting the premises, are not played w.r.t. given models: they probe internal structure and links between premises and conclusions. Again, the key notion is that of a strategy. A winning strategy for the Proponent in argumentation corresponds to a proof for the claim from the premises, Felscher (2001). Interestingly, winning strategies for the Opponent in this game match up with constructions of *counter-models* for the claim w.r.t. the premises.

While evaluation games and proof games clearly share some formal features, they analyze different logical notions: truth versus consistency or derivability. This coexistence reflects the ever-intriguing interplay of Model Theory and Proof Theory as two fundamental perspectives or working styles in logic.

Games in natural language. Games can also be understood as taking the dynamic semantics of language use mentioned in Sect. 21.3 to a multi-agent setting. After all, natural language is typically a medium for communication between different agents who produce and analyze speech or text in turn.

A well-known paradigm here is the 'game-theoretic semantics' of Hintikka (1973), Hintikka and Sandu (2011), introduced in Sect. 21.3, whose main features are as in evaluation games for FOL. Such a game may be viewed as a use of the relevant sentence in a concrete situation (context, model), and while 'winning' or 'losing' may record truth values, the winning strategies are what determines the total meaning.[34]

Yet, in the study of natural language, too, other types of game exist. For instance, in pragmatics, there is a long tradition of *signaling games* starting with Lewis' work on conventions and the emergence of meaning, Lewis (1969), and continuing with further themes in Skyrms (1996, 2010), van Rooij (2004), Zollman (2005), and others.[35] But the earlier dialogical argumentation games, too, make sense for natural language, since much of discourse is about maintaining consistency rather than direct checking of truth against some independently available reality. For more on comparing and connecting various game-theoretic approaches to the syntax, semantics and pragmatics of natural language, cf. van Benthem (2008).

As said, the use of games for natural language is related to current dynamic semantics. But in a way, games are 'more dynamic': unpacking meaning now involves a *process* of evaluation by which speakers and listeners interact with the given sentence and with other agents in a finite or infinite dialogue.

[34] Hintikka makes a suggestive distinction here between 'literal' and 'strategic' meaning.

[35] There are major differences between competitive logical games and signaling games for natural language. Signaling games are partly cooperative, with coordination as a shared goal (though players may have further conflicting goals: maximum informativity for listeners, minimum effort for speakers). Thus, signaling games typically settle on Nash equilibria, indicating satisfactory mutual understanding rather than brute winning strategies.

Games in computation. Games also play an important role in the model-theoretic semantics of computation. For instance, evaluation games for fixed-point logics model nested recursions in sequential computational processes. These games go beyond simple logical evaluation games in allowing infinite play. The widely used *parity games* for the modal μ-calculus produce infinite histories with a winning condition based on which fixed-points in the given formula are unfolded infinitely often.[36] There is an extensive theory of these games, with striking results such as the Positional Determinacy Theorem stating that memory-free winning strategies suffice for parity games. The use of games and strategies in this area is deeply connected with Automata Theory, Thomas et al. (2002).

The possible infinity of play in some model-theoretic evaluation games (or in the earlier-mentioned Ehrenfeucht-Fraïssé games) fits with contemporary views of computing as an open-ended process, which may or may not terminate, may or may not compute any function, but whose main output is *behavior* across time. This same behavioral perspective, now also with an emphasis on concurrency and interleaved action, is central to the proof-theoretic tradition in the semantics of computation, culminating in Abramsky's work. To get there, compositionality challenges had to be met on the way.

Abramsky's game semantics. Lorenzen's original system had a mixture of 'logical rules' determining which moves of defense and attack players could make and 'procedural rules' determining how to schedule attacks over time, whether repetitions of defenses were allowed, and so on. The resulting dialogues could be infinite, at least for first-order logic, in case the initial claim does not follows from the premises. By manipulating procedural rules, the analysis could then be made to fit intuitionistic logic, classical logic, or yet other proof systems. While this mixture of logical and procedural rules may be a realistic model of actual argumentation, it makes it hard to see a compositional pattern.

This challenge is addressed in Abramsky's work, Abramsky and Jagadeesan (1994), Abramsky (1997), with several innovations inspired by a computational view of the content of logical systems. Formulas now denote games between a System and its Environment. Here, proposition letters now stand for subgames which can themselves have any complexity. Moreover, to avoid imposing classical Excluded Middle by design, non-determined infinite games are the paradigm. This is not just a technical fix for blocking some law, it also means that play can now keep returning to subgames, allowing for a much richer view of processes modeled by logical formulas.

Next, logical operators are interpreted as *game constructions* for new games out of given ones. For instance, as in most logical evaluation games, negation corresponds to dualizing the roles of players. Importantly, here, a significant 'split' occurs leading to a richer repertoire of logical notions. One natural disjunction is choice between subgames right at the start, as in most evaluation games, but an equally natural 'delayed disjunction' occurs with *parallel play* of two games where one designated

[36] Formally, when play hits a propositional fixed-point variable, no evaluation takes place against the model, but a return to the body of its matching subformula.

player has the right to *switch* from one game to the other, with a suitable winning condition for the resulting infinite plays. This richer game setting absorbes some procedural rules (as far as needed) into the semantics of the logical operators, and transforms the Lorenzen setting. In particular, there is now a notion of an implication game $A \to B$ with the subgames A, B played in parallel, where winning strategies for the System can be viewed as exemplifying the implication. The resulting logical inferences validated by these games turned out to match up eventually with *linear logic*, Abramsky and Jagadeesan (1994).

The mathematical setting for all this are *categories* whose objects are games, and whose morphisms are strategies in implication games. In the exact design of these categories, various further compositionality challenges are encountered, as explained in Abramsky (1997). One approach uses partial strategies and a simple notion of composition of morphisms, but it does not extend to total strategies in the usual game-theoretic sense. This problem is solved by Abramsky's restriction to *winning strategies* defined in just the right way on parallel disjunction and implication games, whose details need not concern us here.[37]

This is the start of a much broader program. Infinite parallel games, with a pattern of moves or calls to subgames are an excellent model of computational processes where input can be consulted, information can be passed internally, and output produced. Surprisingly, powerful memory-free 'logical' strategies in such games like 'Copy-Cat' turn out to capture essentials of interaction.[38] Enrichments and variations of this game semantics handle a wide variety of features of programming in a compositional manner. And beyond program semantics, game semantics is a theory of communicating processes, whose relation to earlier frameworks such as Process Algebra is discussed in Abramsky (2010).

Abramsky feels that this approach, with the above formulated in category-theoretic terms with games as objects and strategies as morphisms, offers an abstraction level that is more finely grained than many model-theoretic semantics, while avoiding the excessive syntactic detail of some proof systems:

> Game Semantics mediates between [...] operational and denotational semantics, combining the good structural properties of one with the ability to model computational fine structure of the other.

Aside: Model theoretic and proof theoretic semantics. Much more could be said about the proof-theoretic flavor of Abramsky's semantics in terms of categories of games, whose roots go back to the Curry-Howard Isomorphism in the lambda calculus and

---

[37] Finding a good notion of composition of morphisms in a category is not exactly the same as solving a compositionality problem in our general sense, but the two tasks are related. Abramsky explains the issues in terms of validating the Cut Rule in the given proof system, and what requirements this puts on composition of total winning strategies in games for $A \to B$ and $B \to C$ to winning strategies for the game $A \to C$, making sure that episodes in the subgame B remain hidden, and that play does not get stuck forever in these subgames.

[38] Accordingly, in Abramsky's view, the matching tensor product in categories is more fundamental than the sequential composition in other logics of computing. For *truly concurrent* game semantics with simultaneous moves by all players, cf. Abramsky (2003, 2006).

the Brouwer-Heyting-Kolmogorov semantics of intuitionistic logic. Here, we just note that the general approach in this paper covers both proof-theoretic and model-theoretic views, as it includes *algebraic semantics*, a pattern common to both. The coexistence of model theory and proof theory in the study of logical systems and natural language raises many intriguing and often unresolved questions (cf. van Benthem (1998) for discussion) that we cannot address here.

21.8.2 General Themes

Our earlier points about compositionality return with games in logic and computation, though with new twists.

For instance, Abramsky's compositional game semantics for linear logic, IF logic, and their generalizations, Abramsky and Jagadeesan (1994), Abramsky (2006), Abramsky and Väänänen (2009), is based on parametrizing the context using parallel composition of strategies. At an abstract level, this demonstrates the technique of Currying of Sect. 21.4, with contextuality now internalized by a new parameter: the *environment* to which a player's strategy is responding (or the information of the player about her environment). Strategy composition works by 'plugging in' one player's environment into the other players' strategies, thus achieving compositionality in a simple and natural way.[39]

Another key point that returns in game semantics is the legitimate variety of compositional solutions, discussed in Sect. 21.5.1. In the model-theoretic realm, this occurs, for instance, with the modal μ-calculus. Infinite parity games are one natural evaluation game, but so are finite second-order games that involve picking objects, but also sets. This variety of games may seem less obvious in a proof-theoretic setting, when the target of the analysis is one specific logical system, but it can arise even there. For instance, van Benthem (2014) discusses different proof games for classical first-order logic, one based on Lorenzen dialogues, another on semantic tableaux. And in principle, one could even have more model-theoretic and more proof-theoretic games for the same language and logic. Indeed, this variety is only to be expected, since games are a mathematical realm with much more structure than the usual sparser semantic denotations.

Finally, the richer ontology of games also illustrates the theme of language redesign of Sect. 21.7. A game semantics often suggests redesign of the language one started with. The splitting of disjunction in Abramsky's game semantics and the resulting focus on parallel game constructions was a good example. An illustration inside classical logic is the alternative syntactic view of quantifiers as denoting atomic games rather than unary logical operators in Abramsky (2008), van Benthem (2014).

---

[39] Obtaining a yet more abstract compositionality result for game semantics in the style of Hodges' result in Sect. 21.2 seems an open (if not yet very well-defined) problem.

Beyond these methodological concerns or matters of detailed game design, here is what we ourselves see as the major new ideas in current game semantics.

First, *multi-agent* interaction seems crucial to how language functions, and the Agent/Environment methodology fits very well with the realities of how we use language to learn about and cope with the world. Also, *concurrency* has not yet been prominent in the semantics of natural language, even though much of how language functions and evolves on a large scale is not in face-to-face sequential interactions, but in concurrent events of language use in large communities. But also, the shift from finite to *infinite* games, and from final output to ongoing *behavior* seems to fit natural language use very well. Linguistic activity as a whole can be viewed as a non-terminating 'operating system' that provides the arena where terminating tasks can be performed. However, to us, most food for thought is provided by the following more radical perspective.

Inverting the direction: from construction to analysis. In unfolding a game, game semantics takes the view of an agent who 'observes' a sentence (in its context), and gradually analyzes its meaning, by taking it apart, rather than constructing it recursively from the bottom. This inverts the standard semantic perspective that is often associated with compositionality: interpreting expressions outside–in, as Hintikka recommended, rather than inside–out.

All this makes eminent sense for natural languages, especially if one also takes on board pragmatics and understanding up to a point that suffices for action, while acknowledging the role of questions or elucidations to increase depth of understanding as needed. And this is not just a vague metaphor. The outside–in perspective relates to the *co-algebraic view* on semantics, Moss (1999), Pattinson (2003), Cirstea et al. (2011). Coalgebra provides a general mathematical model of dynamical systems that proceeds by co-recursion, rather than recursion. Syntactic operators decompose or unfold an initial state into its components or successors. By gradually parsing a sentence in this top-down manner following its syntactic tree, we gradually 'observe' or analyze a semantic model, or even the structure of reality itself—and this process could well be infinite.

Stated a bit provocatively, on this view, sentences do not provide complete meanings or thoughts: they are a tip of a mental iceberg. And in line with this, reality is not something that we construct out of smallest units, but a complex something that we can only observe to a certain extent by interacting with it. And thus the quest for compositionality achieves a certain grandeur: it might be seen as a way of keeping this learning process comprehensible.

21.9 Further Aspects of Compositionality

This paper may have made some unusual and occasionally controversial comments, but its agenda and modus operandi fit squarely in the established philosophical and mathematical tradition of studying compositionality by logicians. As a counterpoint, it may be useful to observe that our community is surrounded by a broader world today where compositionality is either conceived very differently, or even as something to be abandoned altogether without giving up on mathematical precision, learnability, or computational efficiency.

Cognitive perspectives. Recent studies of compositionality in cognitive science concern the historical emergence of recursive syntax with matching compositional meanings that support a bootstrapping analysis of how meaningful language arose over time. A good example is found in Steinert-Threlkeld (2020). This work shows how compositionality can emerge in principle in evolutionary scenarios for language development, and it is backed up by computational simulations of linguistic scenarios for communication.

Distributional semantics. A major current challenge to compositional semantics is the emergence of distributional semantics, where word meanings are computed from co-occurrence frequencies in large corpora, represented in high-dimensional vector spaces, Evert (2016). A guiding motto for distributional semantics is Zellig Harris's oft-quoted dictum, cited in Sect. 21.1, that "If you want to know the meaning of a word, look at the company it keeps". However, as we have seen in Hodges' analysis in Sect. 21.2, Contextuality can co-exist with Compositionality, so it may be too early to say whether compositional semantics is truly at odds with its distributional rival.[40]

In this connection it may be worth noting that a computational linguistic framework like Data-Oriented Parsing, Bod (2008), combines more compositional views of syntax and semantics with a probabilistic memory of connections in a large corpus of earlier experiences.

Even so, it may be a good idea to confront current discussions of compositionality in language and meaning with some of these new realities.

21.10 Conclusion

Compositionality is a perennial methodological issue that has spawned a huge literature, ranging from particular empirical phenomena to the design of semantics for whole languages, with perhaps the most general abstract perspective to be found in mathematics. One of the reasons why it continues to generate debate is its intriguing relationship with what looks like its antipode: contextuality.

In this paper, we have tried to add some new, or at least under-appreciated considerations to received wisdom. At the most abstract level, we identified a ubiquitous 'currying pattern' in setting up a semantics that brings together contextuality and compositionality. Next, looking at more specific case studies, we identified a list of basic concerns that help in understanding what a proposed compositional semantics does, and does not, achieve. In particular, we emphasized the entanglement of com-

[40] Bonnay (2019) makes a similar point about machine learning systems for natural language processing.

positional semantics with the target set of desired valid inferences, and the dynamic role of language redesign once a semantics has been proposed. We believe that all this results in a more relaxed, but also, a richer attitude toward compositional semantics as an ongoing enterprise, generating new designs, as well as new logical questions and results.

Finally, we have looked at the role of games and game semantics for logic and computation as an area where further intuitions come into play, including radical departures from standard constructive views of meaning to outside-in analyses of complex potentially never-ending behavior. Here computer scientists may well have something to teach to linguists and philosophers.

Acknowledgements Over the years, we have been inspired by our colleague Theo Janssen whose pioneering work on compositionality is still well-worth reading after all these years. The continuing influence of Wilfrid Hodges will also be evident throughout. More concretely for now, we thank the editors of this volume, and the two referees for this chapter. The third author's work on the paper was supported by a grant from the Swedish Research Council, no. 2016-02458. In addition we profited from feedback by Peter Pagin and Jouko Väänänen. But most of all, we thank Samson Abramsky for providing the inspiration and impetus to write this paper.

References

Abramsky, S. (1997). Semantics of interaction. In *Proceedings of the 1996 CLiCS Summer School, Isaac Newton Institute, Cambridge* (pp. 1–31). Cambridge, UK: Cambridge University Press.

Abramsky, S. (2003). Sequentiality versus concurrency in games and logic. *Mathematical Structures in Computer Science, 13*, 531–565.

Abramsky, S. (2006). Socially responsive, environmentally friendly logic. Acta Philosophica Fennica. In T. Aho & A. V. Pietarinen (Eds.), *Truth and Games: Essays in Honour of Gabriel Sandu* (Vol. 78, pp. 17–45). Helsinki: Societas Philosophica Fennica.

Abramsky, S. (2008). Information, processes, and games. In P. Adriaans & J. van Benthem (Eds.), *Handbook of the Philosophy of Information* (pp. 483–549). Amsterdam: Elsevier Science.

Abramsky, S. (2010). From csp to game semantics. In A. W. Roscoe, C. Jones & K. Wood (Eds.), *Reflections on the Work of C.A.R. Hoare* (pp. 33–45). Springer Science Publishers.

Abramsky, S., & Jagadeesan, J. (1994). Games and full completeness for multiplicative linear logic. *Journal of Symbolic Logic, 59*(2), 543–574.

Abramsky, S., & McCusker, G. (1999). Full abstraction for idealized Algol with passive expressions. *Theoretical Computer Science, 227,* 3–42.

Abramsky, S., & Väänänen, J. (2009). From IF to BI: A tale of dependence and separation. *Synthese, 167,* 207–230.

Aiello, M., Pratt, I., & van Benthem, J. (Eds.). (2007). *Handbook of Spatial Logic*. Springer.

Andréka, H., Németi, I., & Sain, I. (2001). Algebraic logic. In D. Gabbay & F. Guenthner (Eds.), *Handbook of Philosophical Logic* (2nd ed., Vol. 2). Dordrecht: Springer Science Publishers.

Andréka, H., Németi, I., & van Benthem, J. (1998). Modal languages and bounded fragments of predicate logic. *Journal of Philosophical Logic, 27,* 217–274.

Andréka, H., van Benthem, J., Bezhanishvili, N., & Németi, I. (2014). Changing a semantics: opportunism or courage? In M. Manzano, I. Sain & E. Alonso (Eds.), *The Life and Work of Leon Henkin: Essays on his Contributions* (pp. 307–338). Birkhäuser: Springer.

Austin, J. L. (1962). *How to do things with words*. Harvard University Press.

Ballarin, R. (2017). Modern origins of modal logic. In E. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy*. Stanford: Metaphysics Research Lab, Stanford University.

Baltag, A., Moss, L., & Solecki, S. (1998). The logic of public announcements, common knowledge, and private suspicions. In *Proceedings of the Seventh TARK conference (1998)* (pp. 43–56). San Francisco: Morgan Kaufmann Publishers.

Baltag, A., & van Benthem, J. (2021). A simple logic of functional dependence. *Journal of Philosophical Logic, 50,* 939–1005.

Barwise, J., & Perry, J. (1999). *Situations and Attitudes* (p. 1983). Stanford: MIT Press. Reprinted by CSLI Publications.

Blackburn, P., de Rijke, M., & Venema, Y. (2001). *Modal Logic*. Cambridge UK: Cambrdige University Press.

Blamey, S. (2001). Partial logic. In D. Gabbay & F. Guenthner (Eds.), *Handbook of Philosophical Logic* (2nd ed., Vol. 5, pp. 261–353). Dordrecht: Springer Science Publishers.

Blok, W., & Pigozzi, D. (1989). *Algebraizable Logics*, volume 396. American Mathematical Society. Also as a Classic Reprint, Advanced Reasoning Forum, Providence.

Bod, R. (2008). The data-oriented parsing approach: Theory and application. In *Studies in computational intelligence* (pp. 307–348). Heidelberg: Springer.

Bonnay, D. (2019). *Compositionality in the light of deep learning*. Lecture Presentation: Department of Philosophy, Université Paris Nanterre.

Cameron, P. J., & Hodges, W. (2001). Some combinatorics of imperfect information. *Journal of Symbolic Logic, 66,* 673–684.

Carnielli, W., & d'Ottaviano, I. (1997). Translations between logical systems: A manifesto. *Logique et Analyse, 15,* 67–82.

Cirstea, C., Kurz, A., Pattinson, D., Schröder, L., & Venema, Y. (2011). Modal logics are coalgebraic. *The Computer Journal, 54,* 31–41.

Dummett, M. (1973). *Frege*. London: Philosophy of Language. Duckworth.

Evert, S. (2016). *Distributional semantics: A practical introduction*. ESSLI: Bolzano. In ESSLI Lecture Notes.

Felscher, W. (2001). Dialogues as a foundation for intuitionistic logic. *Handbook of Philosophical Logic* (Vol. III, pp. 341–372). Dordrecht: Kluwer Academic Publishers.

Frege, G. (1884). *Grundlagen der Arithmetik*, translated by J. Austin as *Foundations of Arithmetic*. Oxford: Blackwell.

Galliani, P. (2011). Sensible semantics of imperfect information. LNCS. In M. Banerjee & A. Seth (Eds.), *Logic and its Applications* (Vol. 6521, pp. 79–89). Berlin/Heidelberg: Springer.

Glüer, K., & Pagin, P. (2021). *Switcher Semantics*. From Propositions to Ingredient Sense: Under contract with Oxford University Press.

Gödel, K. (1932). Zum intuitionistischen Aussagenkalkül. In *Anzeiger der Akademie von Wissenschaften in Wien* (Vol. 69, pp. 65–66). Akademie von Wissenschaften in Wien. Repr. and transl. in Feferman et al. (Eds.) (1932) *Kurt Gödel. Collected Works*, OUP, 1986 (Vol. I, pp. 222–225).

Goldblatt, R. (2006). Mathematical modal logic. a view of its evolution. In D. Gabbay & J. Woods (Eds.), *Logic and the Modalities in the Twentieth Century*, volume 7 of *Handbook of Modal Logic* (pp. 1–98). Elsevier.

Goranko, V., & Rumberg, A., (2020). *Temporal Logic, In Stanford Electronic Encyclopedia of Philosophy.* CSLI: Stanford University.

Groenendijk, J., & Stokhof, M. (1991). Dynamic predicate logic. *Linguistics & Philosophy, 14,* 39–100.

Gurevich, Y., & Shelah, S. (1986). Fixed-point extensions of first-order logic. *Annals of Pure and Applied Logic, 32,* 265–280.

Harel, D. (1984). Dynamic logic. In D. Gabbay & F. Guenthner (Eds.), *Handbook of Philosophical Logic* (Vol. II). Dordrecht: D. Reidel.

Harel, D., Kozen, D., & Tiuryn, J. (2000). *Dynamic Logic*. Cambridge, MA: The MIT Press.

Hintikka, J. (1973). *Logic*. Oxford: Language Games and Information. Clarendon Press.

Hintikka, J., & Sandu, G. (2011). Game-theoretical semantics. In J. van Benthem & A. ter Meulen (Eds.), *Handbook of logic and language* (2nd ed., pp. 415–466). Amsterdam: Elsevier.
Hinzen, W., Machery, E., & Werning, M. (Eds.). (2012). *The Oxford Handbook of Compositionality*. Oxford: Oxford University Press.
Hodges, W. (1997). Compositional semantics for a language of imperfect information. *Logic Journal of the IGPL, 5,* 539–563.
Hodges, W. (2001). Formal features of compositionality. *Journal of Logic, Language and Information, 10,* 7–28.
Hodges, W. (2011). From sentence meanings to full semantics. In J. van Benthem et al. (Ed.), *Proof, Computation and Agency* (pp. 261–276). Springer.
Janssen, T. (2001). Frege, contextuality and compositionality. *Journal of Logic, Language and Information, 10,* 115–136.
Janssen, T. (2011). Compositionality. In J. van Benthem & A. Ter Meulen (Eds.), *Handbook of Logic and Language* (2nd ed., pp. 495–554). Amsterdam: Elsevier.
Janssen, T., & van Emde Boas. P. (1981). Some remarks on compositional semantics. In D. Kozen (Ed.), *Logic of Programs*, number 131 in Lecture notes in computer science (pp. 137–149). Berlin: Springer.
Kamp, H. (1981). A theory of truth and semantic representation. In J. Groenendijk, T. Janssen, & M. Stokhof (Eds.), *Formal Methods in the Study of Language* (pp. 277–322). Amsterdam: Mathematisch Centrum.
Kolaitis, P.G., & Vardi, M.Y. (1992). Fixpoint logic versus infinitary logic in finite-model theory. In A. Scedrov (Ed.), *Proceedings of the Seventh Annual IEEE Symposium on Logic in Computer Science, LICS* (pp. 46–57). IEEE Computer Society Press, June 1992.
Lewis, D. (1969). *Convention. A Philosophical Study*. Cambridge: Harvard University Press.
Lewis, D. (1980). Index, context, and content. In S. Kanger & S. Öhman (Eds.). *Philosophy and Grammar*. Dordrecht: D. Reidel.
Lorenzen, P. (1955). *Einführung in die Operative Logik und Mathematik*. Berlin: Springer.
Mann, A., Sandu, G., & Sevenster, M. (2011). *Independence-Friendly Logic*. Cambridge: A Game-Theoretic Approach. Cambridge University Press.
Montague, R. (1970). Universal grammar. *Theoria, 36*, 373–398. Reprinted in [?], 222–246, 1970.
Moss, L. (1999). Coalgebraic logic. *Annals of Pure and Applied Logic, 96*, 277–317 (Erratum published *Annals of Pure and Applied Logic, 99*, 241–259, 1999).
Németi, I. (1985). The equational theory of cylindric relativized set algebras is decidable. Preprint 63/85, Mathematical Institute, Hungarian Academy of Sciences, Budapest.
Nouwen, R., Brasoveanu, A., van Eijck, J., & Visser, A. (2015). Dynamic semantics. Stanford Encyclopedia of Philosophy.
Pagin, P. (2012). Communication and the complexity of semantics. In Hinzen et al. (2012) (pp. 510–529). Oxford University Press.
Pagin, P., & Westerståhl, D. (2010a). Compositionality I. Definitions and variants. *Philosophy Compass, 5*(3), 250–264.
Pagin, P., & Westerståhl, D. (2010b). Compositionality II. Arguments and problems. *Philosophy Compass, 5*(3), 265–282.
Pagin, P., & Westerståhl, D. (2010c). Pure quotation and general compositionality. *Linguistics & Philosophy, 33*(5), 381–415.
Pagin, P. (2005). Compositionality and context. In G. Preyer & G. Peter (Eds.), *Contextualism in Philosophy* (pp. 303–48). Oxford: Oxford University Press.
Pattinson, D. (2003). Coalgebraic modal logic: Soundness, completeness and decidability of local consequence. *Theoretical Computer Science, 309,* 177–193.
Peters, S., & Westerståhl, D. (2006). *Quantifiers in Language and Logic*. Oxford: Oxford University Press.
Plaza, J. (1989). Logics of public communications. In *Proceedings 4th International Symposium on Methodologies for Intelligent Systems* (pp. 201–216).
Rendsvig, R. (2016). Epistemic logic. In *Stanford electronic encyclopedia of philosophy*.

Russell, B. (2005). On denoting. *Mind, 14,* 479–493.
Skyrms, B. (1996). *Evolution of the Social Contract*. Cambridge: Cambridge University Press.
Skyrms, B. (2010). *Signals Evolution, Learning & Information*. New York: Oxford University Press.
Steinert-Threlkeld, S. (2020). Towards the emergence of non-trivial compositionality. *Philosophy of Science* (forthcoming).
Stirling, C. (1993). Modal and temporal logics. In *Handbook of Theoretical Computer Science, Semantics* (Vol. 2, pp. 477–563). Amsterdam: North-Holland.
Thomas, W., Graedel, E., & Wilke, T. (Eds.). (2002). *Automata. Logics, and Infinite Games. Lecture Notes in Computer Science* (Vol. 2500). Berlin: Springer.
Väänänen, J. (2007). *Dependence Logic: A New Approach to Independence Friendly Logic*. Cambridge: Cambridge University Press.
van Benthem, J. (1977). Tense logic and standard logic. *Logique et Analyse, 80,* 395–437.
Van Benthem, J. (1986). The logic of semantics. Number 29 in Studies in Linguistics and Philosophy*Essays in Logical Semantics* (pp. 198–214). Dordrecht: Springer.
van Benthem, J. (1998). Proofs, labels and dynamics in natural language. Festschrift for Dov Gabbay. In U. Reyle & H.-J. Ohlbach (Eds.), *We Will Show Them* (pp. 31–41). Dordrecht: Kluwer Academic Publishers.
van Benthem, J. (2008). Games that make sense: Logic, language and multi-agent inter- action. In K. Apt & R. van Rooij (Eds.), *New Perspectives on Games and Interaction* (pp. 197–209). Amsterdam: Amsterdam University Press.
van Benthem, J. (2014). *Logic in Games*. Cambridge, MA: The MIT Press.
van Benthem, J. (2019). Implicit and explicit stances in logic. *Journal of Philosophical Logic, 48,* 541–601.
van Rooij, R. (2004). Evolution of conventional meaning and conversational principles. *Synthese, 139,* 331–366.
Westerståhl, D. (1998). On mathematical proofs of the vacuity of compositionality. *Linguistics & Philosophy, 21,* 635–643.
Westerståhl, D. (2004). On the compositional extension problem. *Journal of Philosophical Logic, 33,* 549–582.
Westerståhl, D. (2012). Compositionality in Kaplan style semantics. In Hinzen et al. (2012) (pp. 192–219). Oxford University Press.
Zadrozny, W. (1994). From compositional to systematic semantics. *Linguistics & Philosophy, 17,* 329–342.
Zollman, K. (2005). Talking to neighbors: The evolution of regional meaning. *Philosophy of Science, 72*(1), 69–85.

Chapter 22
Compact Inverse Categories

Robin Cockett and Chris Heunen

Abstract We prove a structure theorem for compact inverse categories. The Ehresmann-Schein-Nambooripad theorem gives a structure theorem for inverse monoids: they are inductive groupoids. A particularly nice case due to Clifford is that commutative inverse monoids become semilattices of abelian groups. It has also been categorified by Hoehnke and DeWolf-Pronk to a structure theorem for inverse categories as locally complete inductive groupoids. We show that in the case of compact inverse categories, this takes the particularly nice form of a semilattice of compact groupoids. Moreover, one-object compact inverse categories are exactly commutative inverse monoids. Compact groupoids, in turn, are determined in particularly simple terms of 3-cocycles by Baez-Lauda.

Keywords Compact category · Inverse category · Inverse monoid · Semilattice · Groupoid

Samson Abramsky's influence[1] on the categorical semantics of quantum computation may be traced to the article (Abramsky & Coecke, 2004) which promoted the use of compact dagger categories for modelling quantum protocols. Subsequently his cate-

[1]Robin, in particular, would like to acknowledge Samson's influence which stretches back to well before the mutual interests discussed in this article. He recalls fondly a "hike" with Samson in Lake Louise, during a Higher Order Banff workshop meeting in the early 1990s. Samson, oblivious to the spectacular scenery, spent the hike explaining his ideas on interaction categories (Abramsky, 1993).

Date: January 13, 2023. Supported by EPSRC Fellowship EP/R044759/1. We thank Peter Hines for pointing out that the proof of Proposition 9 could be simplified, Martti Karvonen for the idea of the proof of Lemma 23, and Phil Scott for pointing out Theorem 5.

R. Cockett
University of Calgary, 2500 University Drive NW, Calgary, AB T2N 1N4, Canada
e-mail: robin@ucalgary.ca

C. Heunen (✉)
University of Edinburgh, Informatics Forum, Edinburgh EH8 9AB, UK
e-mail: chris.heunen@ed.ac.uk

A. Palmigiano and M. Sadrzadeh (eds.), *Samson Abramsky on Logic and Structure in Computer Science and Beyond*, Outstanding Contributions to Logic 25,
https://doi.org/10.1007/978-3-031-24117-8_22

gorical exposition of the “no cloning theorem” (Abramsky, 2008) helped to delineate the divide between the quantum and the classical world. Quantum computation subsequently became one of Samson’s major interests and he continued to influence the subject (Abramsky & Heunen, 2016; Abramsky and Heunen, 2012; Abramsky, 2013, 2004; Abramsky & Coecke, 2005; Abramsky, 2005, 2008; Abramsky & Duncan, 2006).

However, Samson was well-aware of another important source of dagger categories, namely, inverse categories which provide the semantics for another important brand of computation, namely reversible computation. Reversible computation shares many features with quantum computation and, indeed, was another interest of Samson’s. In Abramsky (2005) Abramsky made a surprising link between reversible computing and linear combinatory algebras.

In this chapter we pick up again on the link between reversible and quantum computation by examining the relationship between compact dagger categories and inverse categories. In particular, we investigate the structure of inverse categories that are also compact dagger categories.

22.1 Introduction

Inverse monoids model partial symmetry (Lawson, 1998), and arise naturally in many combinatorial constructions (Duncan & Paterson, 1985). The easiest example of an inverse monoid is perhaps a group. There is a structure theorem for inverse monoids, due to Ehresmann-Schein- Nambooripad (1958; 1960); Schein (1979); Nambooripad (1979), that exhibits them as inductive groupoids. The latter are groupoids internal to the category of partially ordered sets with certain extra requirements. By a result of Clifford (1941); Jarek (1964), the inductive groupoids corresponding to commutative inverse monoids can equivalently be described as semilattices of abelian groups.

A natural typed version of an inverse monoid is an inverse category (Kastl, 1979; Cockett & Lack, 2002). This notion can for example model partial reversible functional programs (Giles, 2014). The easiest example of an inverse category is perhaps a groupoid. Hoehnke and DeWolf-Pronk have generalised the ESN theorem to inverse categories, exhibiting them as locally complete inductive groupoids. This paper investigates ‘the commutative case’, thus fitting in the bottom right cell of Fig. 22.1.

objects	general case	commutative case
one	inductive groupoid [38]	semilattice of abelian groups [16, 31]
many	locally inductive groupoid [30, 18]	semilattice of compact groupoids

Fig. 22.1 Overview of structure theorems for inverse categories

However, let us emphasise two ways in which Fig. 22.1 is overly simplified. First, the term ‘commutative case’ is misleading: we mean considering compact inverse categories. More precisely, we prove that compact inverse categories correspond to semilattices of compact groupoids. Compact inverse categories are only commutative in that their endohomset of scalars is always commutative. In particular, the categorical composition of the compact inverse category can be as noncommutative as you like. We expect that the tensor product also need not be symmetric. But compact categories are interesting in their own right: they model quantum entanglement (Heunen & Vicary, 2019); they model linear logic (Seely, 1989); and they naturally extend traced monoidal categories modelling feedback (Joyal et al., 1996).

Second, our result is not a straightforward special case of Hoehnke-DeWolf-Pronk (1989; 2018), nor of Clifford (1941; 1964), but instead rather a common categorification. We prove that one-object compact inverse categories are exactly commutative inverse monoids. Semilattices of groupoids are a purely categorical notion, whereas ordered groupoids have more ad hoc aspects. Compact groupoids are also known as 2-groups or crossed modules, and have fairly rigid structure themselves, due to work by Baez and Lauda (2004). We take advantage of this fact to ultimately show that there is a (weak) 2-equivalence of (weak) 2-categories of compact inverse categories, and semilattices of 3-cocycles.

Section 22.2 starts by recalling the ESN structure theorem for inverse monoids, and its special commutative case due to Clifford in a language that the rest of the paper will follow. Section 22.3 discusses the generalisation of the ESN theorem to inverse categories due to Hoehnke, DeWolf, and Pronk, and its relation to semilattices of groupoids. Section 22.4 is the heart of the paper, and considers additional structure on inverse categories that was hidden for inverse monoids. It shows that the construction works for compact inverse categories, and argues that this is the right generalisation of inverse monoids in this sense. After all this theory, Sect. 22.5 lists examples. We have chosen to treat examples after theory; that way they can illustrate not just compact inverse categories, but also the construction of the structure theorem itself. Section 22.6 then moves to a 2-categorical perspective, to connect to the structure theorem for compact groupoids due to Baez and Lauda. Finally, Sect. 22.7 discusses the many questions left open and raised in the paper.

22.2 Inverse Monoids

An *inverse monoid* is a monoid where every element x has a unique element $x^\dagger$ satisfying $x = xx^\dagger x$ and $x^\dagger = x^\dagger xx^\dagger$ (Lawson, 1998). Equivalently, the monoid carries an involution $\dagger$ such that $x = xx^\dagger x$ and $xx^\dagger yy^\dagger = yy^\dagger xx^\dagger$ for all elements x and y. Inverse monoids and homomorphisms (that automatically preserve the involution) form a category **InvMon**, and commutative inverse monoids form a full subcategory **cInvMon**. This section recalls structure theorems for inverse monoids. In general they correspond to inductive groupoids by the Ehresmann-Schein-Nambooripad theorem Ehresmann (1958, 1960); Nambooripad (1979); Schein (1979), that we now recall.

Definition 1 A *(bounded meet-)semilattice* is a partially ordered set with a greatest element $\top$, in which any two elements s and t have a greatest lower bound $s \wedge t$. A *morphism of semilattices* is a function f satisfying $f(\top) = \top$ and $f(s \wedge t) = f(s) \wedge f(t)$.

We regard a semilattice as a category by letting elements be objects and having a unique morphism $s \to t$ when $s \leq t$, that is, when $s \wedge t = s$. We will disregard size issues altogether; either by restricting to small categories throughout the article, or by allowing semilattices (and monoids) that are large—the only place it seems to matter is Lemma 23 below. Recall that a *groupoid* is a category whose every morphism is invertible.

Definition 2 An *ordered groupoid* is a groupoid internal to the category of partially ordered sets and monotone functions, together with a choice of *restriction* $(f|A) \colon A \to B$ for each $f \colon A' \to B$ and $A \leq A'$ satisfying $(f|A) \leq f$. Explicitly, the sets G_0 and G_1 of objects and arrows are partially ordered, and the functions

$$\begin{tikzcd} G_0 \arrow[r, "\mathrm{id}"] & G_1 \arrow[l, bend right, "\mathrm{dom}"'] \arrow[l, bend left, "\mathrm{cod}"] \arrow[loop above, "\mathrm{inv}"] & G_2 \arrow[l, "\mathrm{comp}"'] \end{tikzcd}$$

are all monotone, where $G_2 = \{(g, f) \in G_1^2 \mid \mathrm{dom}(g) = \mathrm{cod}(f)\}$ is ordered by $(g, f) \leq (g', f')$ when $g \leq g'$ and $f \leq f'$. An *inductive groupoid* is an ordered groupoid whose partially ordered set of objects forms a semilattice.

A morphism of ordered groupoids is a functor F that is monotone in morphisms, that is, $F(f) \leq F(g)$ when $f \leq g$. Inductive groupoids and their morphisms form a category **IndGpd**.

Theorem 3 *There is an equivalence* **InvMon** $\simeq$ **IndGpd**.

Proof (*Proof sketch*) See Lawson (1998, Sect. 4.2) or DeWolf and Pronk (2018) for details. An inverse monoid M turns into an inductive groupoid as follows. Objects are idempotents $ss^\dagger = s \in M$. Every element of M is a morphism $x \colon x^\dagger x \to xx^\dagger$. The identity on s is s itself, and composition is given by multiplication in M. Inverses are given by $x^{-1} = x^\dagger$. The order $x \leq y$ holds when $x = yx^\dagger x$. The restriction of $x \colon x^\dagger x \to xx^\dagger$ to $s^\dagger s = s \leq x^\dagger x$ is xs. $\square$

Observe from the proof of the previous theorem that commutative inverse monoids correspond to inductive groupoids where every morphism is an endomorphism. Moreover, the endohomsets are abelian groups. Hence commutative inverse monoids correspond to a semilattice of abelian groups.

Definition 4 A *semilattice over* a subcategory $\mathbf{V}$ of $\mathbf{Cat}$ is a functor $F\colon \mathbf{S}^{\mathrm{op}} \to \mathbf{V}$ where $\mathbf{S}$ is a semilattice and all categories $F(s)$ have the same objects. A *morphism of semilattices* $F \to F'$ *over* $\mathbf{V}$ is a morphism of semilattices $\varphi\colon \mathbf{S} \to \mathbf{S}'$ together with a natural transformation $\theta\colon F \Rightarrow F' \circ \varphi$. Write $\mathbf{SLat}[\mathbf{V}]$ for the category of semilattices over $\mathbf{V}$ and their morphisms.

The ordinary category of semilattices can be recovered by choosing $\mathbf{V}$ to be the category containing as its single object the terminal category $\mathbf{1}$. See also Romanowska and Smith (1997). In the commutative case, the ESN theorem simplifies, as worked out by Clifford and Jarek (1941; 1964). The following formulation chooses $\mathbf{V} = \mathbf{Ab}$, regarding an abelian group as a one-object category.

Theorem 5 *If M is a commutative inverse monoid, then*

$$\mathbf{S} = \{s \in M \mid ss^\dagger = s\}, \qquad s \wedge t = st, \qquad \top = 1,$$

is a semilattice, and for each $s \in \mathbf{S}$,

$$F(s) = \{x \in M \mid xx^\dagger = s\}$$

is an abelian group with multiplication inherited from M and unit s, giving a semilattice of abelian groups $F\colon \mathbf{S}^{\mathrm{op}} \to \mathbf{Ab}$ by $F(s \leq t)(x) = sx$.

If $F\colon \mathbf{S}^{\mathrm{op}} \to \mathbf{Ab}$ is a semilattice of abelian groups, then $M = \coprod_{s\in\mathbf{S}} F(s)$ is a commutative inverse monoid under

$$\begin{aligned}
xy &= F(s \wedge t \leq s)(x) \cdot F(s \wedge t \leq t)(y) && \text{if } x \in F(s),\ y \in F(t),\\
x^\dagger &= x^{-1} \in F(s) && \text{if } x \in F(s),\\
1 &= 1 \in F(\top).
\end{aligned}$$

This gives an equivalence $\mathbf{cInvMon} \simeq \mathbf{SLat}[\mathbf{Ab}]$.

Proof First, let M be a commutative inverse monoid. To see that $\mathbf{S}$ is a semilattice, it suffices to show that it is a commutative idempotent monoid. Commutativity is inherited from M, and idempotence follows from the fact that M is an inverse monoid: $(xx^\dagger)^2 = xx^\dagger xx^\dagger = xx^\dagger$. Next we verify that each $F(s)$ is an abelian group. It is closed under multiplication: if $x, y \in F(s)$, then $(xy)(xy)^\dagger = xx^\dagger y^\dagger y = ss^\dagger = s$ so also $xy \in F(s)$. It has s as a unit: if $x \in F(s)$, then $sx = xx^\dagger x = x$. The inverse of $x \in F(s)$ is given by $x^\dagger$, because $xx^\dagger = s$ by definition. Furthermore, the diagram F is functorial: clearly $F(s \leq t) \circ F(r \leq s)(x) = rx = F(r \leq t)(x)$, and $F(s \leq s)(x) = sx = xx^\dagger x = x$. It is also well-defined: if $s \leq t$ and $x \in F(t)$, then $sx(sx)^\dagger = sxx^\dagger s^\dagger = sts^\dagger = ss^\dagger = s$ so $sx \in F(t)$.

Now let $F \in \mathbf{SLat}[\mathbf{Ab}]$. Then $1 \in F(\top)$ acts as a unit in M: if $x \in F(s)$ then $x1 = F(s \leq s)(x) \cdot F(s \leq \top)(1) = x \cdot 1 = x \in F(s)$. The multiplication is clearly associative and commutative, so M is an abelian monoid. It is an inverse monoid because $xx^\dagger x = xx^{-1}x = x$ is computed within $F(s)$.

Next we move to morphisms. Given a morphism $f: M \to M'$ of commutative inverse monoids, define a morphism $F \to F'$ of their associated semilattices of abelian groups as follows: $\varphi: \mathbf{S} \to \mathbf{S}'$ is just $\varphi(s) = f(s)$, and $\theta_s: F(s) \to F'(f(s))$ is just $\theta_s(x) = f(x)$. This is clearly functorial **cInvMon** $\to$ **SLat**[**Ab**].

Conversely, given a morphism $(\varphi, \theta): F \to F'$ of semilattices of abelian groups, define a homomorphism $M \to M'$ of their associated commutative inverse monoids by $F(s) \ni x \mapsto \theta_s(x) \in F(\varphi(s))$. This is clearly functorial **SLat**[**Ab**] $\to$ **cInvMon**.

Finally, turning a commutative inverse monoid M into a semilattice of abelian groups and that in turn into a commutative inverse monoid ends up with the exact same monoid M. A semilattice of abelian groups $F: \mathbf{S} \to \mathbf{Ab}$ gets mapped to the inverse monoid $\coprod_s F(s)$, which in turn gets mapped to the following semilattice of abelian groups $G: \mathbf{T} \to \mathbf{Ab}$. The semilattice $\mathbf{T}$ is given by $\{t \in F(s) \mid s \in \mathbf{S}, t = tt^\dagger\} = \{t \in F(s) \mid s \in \mathbf{S}, t = tt^{-1} = 1\} = \{1 \in F(s) \mid s \in \mathbf{S}\}$; clearly $s \mapsto 1 \in F(s)$ is an isomorphism $\varphi: \mathbf{S} \to \mathbf{T}$. The abelian group $G(\varphi(s))$ is given by $\{x \mid xx^\dagger = s\} = \{x \mid 1 = xx^{-1} = s\} = \{x \in F(s)\}$; clearly $x \mapsto x$ is a natural isomorphism $\theta_s: F(s) \to G(\varphi(s))$. Thus $G \simeq F$, and the two functors implement an equivalence. □

22.3 Inverse Categories

This section extends the previous one to a typed setting. A *dagger category* is a category with a contravariant involution † that acts as the identity on objects. A *dagger functor* is a functor between dagger categories satisfying $F(f^\dagger) = F(f)^\dagger$. An *inverse category* is a dagger category where $f = ff^\dagger f$ and $ff^\dagger gg^\dagger = gg^\dagger ff^\dagger$ for any pair of morphisms f and g with the same domain (Cockett & Lack, 2002). Equivalently, it is a category where every morphism $f: A \to B$ allows a unique morphism $f^\dagger: B \to A$ satisfying $f = ff^\dagger f$ and $f^\dagger = f^\dagger ff^\dagger$; thus every functor between inverse categories is in fact a dagger functor. Inverse categories and (dagger) functors form a category **InvCat**, and groupoids and functors form a full subcategory **Gpd**. The ESN theorem extends to inverse categories, as worked out by Hoehnke and DeWolf-Pronk (1989; 2018).

Definition 6 A *locally complete inductive groupoid* is an ordered groupoid with a partition of the semilattice G_0 of objects into semilattices $\{M_i\}$ such that two objects are comparable if and only if they are in the same semilattice M_i. Locally complete inductive groupoids form a subcategory **lcIndGpd** of **IndGpd** of those functors that preserve greatest lower bounds of objects.

Theorem 7 *There is an equivalence* **InvCat** $\simeq$ **lcIndGpd**.

Proof (*Proof sketch*) See Hoehnke (1989); DeWolf and Pronk (2018) for details. An inverse category $\mathbf{C}$ turns into a locally complete inductive groupoid as follows. Objects are idempotents $ff^\dagger$ for some endomorphism $f: A \to A$ in $\mathbf{C}$. These partition into the semilattices of idempotents on a fixed object A. Every morphism $f: A \to B$ of $\mathbf{C}$ becomes a morphism $f^\dagger f \to ff^\dagger$. The identity on $ff^\dagger$ is $ff^\dagger$

itself, and composition is inherited from $\mathbf{C}$. Inverses are given by $f^{-1} = f^{\dagger}$. The order $f \leq g$ holds when $f = gf^{\dagger}f$; clearly two identity morphisms are comparable exactly when they endomorphisms on the same object. The restriction of $f : f^{\dagger}f \to ff^{\dagger}$ to $s^{\dagger}s = s \leq f^{\dagger}f$ is fs. $\square$

Lemma 8 *If $F : \mathbf{S}^{\mathrm{op}} \to \mathbf{Gpd}$ is a semilattice of groupoids, there is a well-defined inverse category $\mathbf{C}$ with the same objects as $F(\top)$ and morphisms*

$$\mathbf{C}(A, B) = \coprod_{s \in \mathbf{S}} F(s)\big(A, B\big).$$

If (φ, θ) is a morphism $F \to F'$ of semilattices of groupoids, then there is a dagger functor $\mathbf{C} \to \mathbf{C}'$ between their associated categories, given by $A \mapsto \theta_{\top}(A)$ on objects and $F(s) \ni f \mapsto \theta_s(f) \in F'(\varphi(s))$ on morphisms. This gives a functor $\mathbf{SLat}[\mathbf{Gpd}] \to \mathbf{InvCat}$.

Proof The composition of $f \in F(s)(A, B)$ and $g \in F(t)(A, B)$ is given by $F(s \wedge t \leq t)(g) \circ F(s \wedge t \leq s)(f) \in F(s \wedge t)(A, C)$; this is clearly associative. The identity on A is given by $\mathrm{id}_A \in F(\top)(A, A)$: if $f \in F(s)(A, B)$, then $f \circ \mathrm{id}_A = F(s \wedge \top \leq \top)(\mathrm{id}_A) \circ F(s \wedge \top \leq s)(f) = \mathrm{id} \circ F(s \leq s)(f) = f$. The dagger of $f \in F(s)(A, B)$ is given by $f^{-1} \in F(s)(B, A)$; this clearly is an inverse category. $\square$

Combining Theorem 7 and Lemma 8, we see that a semilattice of groupoids $F : \mathbf{S}^{\mathrm{op}} \to \mathbf{Gpd}$ gives rise to a locally complete inductive groupoid $\mathbf{G}$ where:

- objects are $\coprod_{A \in F(\top)} \coprod_{s \in \mathbf{S}} \{f^{\dagger}f \mid f \in F(s)(A, A)\}$;
- there is an arrow $(f^{\dagger}f)_{A,s} \to (ff^{\dagger})_{B,s}$ for each $f \in F(s)(A, B)$;
- the composition of $f \in F(s)(A, B)$ and $g \in F(t)(B, C)$ is computed as $F(s \wedge t \leq t)(g) \circ F(s \wedge t \leq s)(f)$.

Not every locally complete inductive groupoid comes from a semilattice of groupoids in this way. Instead, locally complete inductive groupoids correspond to certain functors $\mathbf{S}^{\mathrm{op}} \to \mathbf{Gpd}$ where $\mathbf{S}$ may be a disjoint union of several semilattices; a 'multi-semilattice' of groupoids.

Notice that the objects of $\mathbf{G}$ are doubly-indexed: once by an object of the category $F(\top)$, and once by an element of the semilattice $\mathbf{S}$. Locally complete inductive groupoids and semilattices of groupoids have different ways of bookkeeping the same data, each emphasising one of these two indices. In the remainder of the paper, we will prefer to work with semilattices of groupoids rather than the more general locally complete inductive groupoids for two reasons. First, the extra structure we will consider does not require 'multi-semilattices', but instead is uniform enough so semilattices suffice. Second, semilattices of groupoids form a purely categorical concept, whereas ordered groupoids require extra conditions on groupoids internal to the category of partially ordered sets that are somewhat ad hoc. For example, this perspective will later enable us to remove the restriction that all groupoids in a semilattice of groupoids must have the same objects; see Lemma 23 below.

22.4 Compact Inverse Categories

There is another way to categorify inverse monoids, that takes advantage of a degree of commutativity. Instead of moving from inverse monoids to inverse categories, in this section we move to *compact inverse categories*. The presence of the tensor product means that the latter specialise to commutative inverse monoids in the one-object case. By a *compact inverse category* we mean an inverse category that is also a compact dagger category under the same dagger (Heunen & Vicary, 2019). Here, a dagger category is compact when it is symmetric monoidal, $(f \otimes g)^\dagger = f^\dagger \otimes g^\dagger$ for all morphisms f and g, all coherence isomorphisms are inverted by their own daggers, and every object A allows an object A^* and a morphism $\eta_A \colon I \to A^* \otimes A$ satisfying

$$\mathrm{id}_A = \lambda_A \circ (\varepsilon \otimes \mathrm{id}_A) \circ \alpha \circ (\mathrm{id}_A \otimes \eta) \circ \rho_A^{-1} \tag{22.1}$$

for $\varepsilon = \sigma \circ \eta^\dagger$ where σ is the swap map. Let us first show that compact inverse categories indeed generalise commutative inverse monoids, because the property of compactness is hidden in the one-object case. Recall that any commutative monoid may be regarded as a one-object monoidal category, and vice versa.

Proposition 9 *One-object compact (dagger/inverse) categories are exactly commutative (involutive/inverse) monoids.*

Proof Let M be a commutative monoid. Regard it as a one-object monoidal category. The one object is the tensor unit, and in any monoidal category, the tensor unit I is its own dual $I^* = I$, since $\eta = \lambda_I^{-1}$ and $\varepsilon = \rho_I$ satisfy (22.1) by coherence (Heunen & Vicary, 2019, Lemma 3.6). If the monoid is involutive/inverse, then the category is clearly dagger/inverse.

Conversely, a one-object (dagger) category is clearly an (involutive) monoid. If the category is monoidal, then the monoid is necessarily that of scalars $I \to I$, where tensor and composition coincide and are commutative (Abramsky, 2005). □

We now set out to generalise Theorem 5 to compact inverse categories $\mathbf{C}$. They have the right modicum of commutativity to take advantage of Lemma 8: the monoid $\mathbf{C}(I, I)$ of *scalars* is always commutative, any morphism $f \colon A \to B$ can be multiplied with a scalar $s \colon I \to I$ to give $s \bullet f = \lambda \circ (s \otimes f) \circ \lambda^{-1}$. Any compact category has a partial *trace* that turns morphisms $A \otimes U \to B \otimes U$ into maps $A \to B$; the special case $A = B = I$ of the total trace turns any endomorphism $f \colon A \to A$ into the scalar $\mathrm{Tr}(f) = \varepsilon \circ (f \otimes \mathrm{id}_{A^*}) \circ \sigma \circ \eta \colon I \to I$. Furthermore, any morphism $f \colon A \to B$ has a *dual* $f^* = (\mathrm{id}_{A^*} \otimes \varepsilon_B) \circ (\mathrm{id}_{A^*} \otimes f \otimes \mathrm{id}_{B^*}) \circ (\eta_A \otimes \mathrm{id}_{B^*}) \colon B^* \to A^*$, satisfying $\mathrm{Tr}(f^*) = \mathrm{Tr}(f)^*$ when $A = B$. We will write $\mathrm{tr}(f)$ instead of $\mathrm{Tr}(f)^*$. The form of the following lemma resembles the categorical no-cloning theorem (Abramsky, 2008), and is the heart of the matter.

Lemma 10 *In a compact inverse category, any endomorphism f equals* $\mathrm{tr}(f) \bullet \mathrm{id}$.

Proof Let $f : A \to A$ be an endomorphism. Compactness provides $\eta : I \to A^* \otimes A$ and $\varepsilon : A \otimes A^* \to I$ satisfying the snake equations. In terms of $g = \varepsilon \otimes \mathrm{id}_A$ and $h = \mathrm{id}_A \otimes \eta^\dagger = \mathrm{id}_A \otimes (\varepsilon \circ \sigma)$, and suppressing coherence isomorphisms, these equations read $gh^\dagger = \mathrm{id}_A = hg^\dagger$. It follows that

$$hh^\dagger = gh^\dagger hh^\dagger = gh^\dagger = \mathrm{id}_A,$$
$$g^\dagger h = g^\dagger gh^\dagger h = h^\dagger hg^\dagger g = h^\dagger g.$$

Therefore $g = hh^\dagger g = hg^\dagger h = h$, and so

$$f = g \circ (\mathrm{id}_A \otimes f^* \otimes \mathrm{id}_A) \circ h^\dagger = h \circ (\mathrm{id}_A \otimes f^* \otimes \mathrm{id}_A) \circ h^\dagger = \mathrm{Tr}(f^*) \bullet \mathrm{id}_A.$$

□

Proposition 11 *A compact dagger category is a compact inverse category if and only if every morphism f satisfies $f = \mathrm{tr}(ff^\dagger) \bullet f$.*

Proof Suppose we're given a compact inverse category. By Lemma 10, the endomorphism $ff^\dagger$ equals $\mathrm{tr}(ff^\dagger ff^\dagger) \bullet \mathrm{id} = \mathrm{tr}(ff^\dagger) \bullet \mathrm{id}$. Hence $f = \mathrm{tr}(ff^\dagger) \bullet f$.

Conversely, suppose given a compact dagger category in which every morphism satisfies $f = \mathrm{tr}(ff^\dagger) \bullet f$. We will prove that this is a *restriction category* with $\bar{f} = \mathrm{tr}(ff^\dagger) \bullet \mathrm{id}$, by verifying the four axioms (Cockett & Lack, 2002).

First, $f\bar{f} = \mathrm{tr}(ff^\dagger) \bullet f = f$. Second, $\bar{f}\bar{g} = \mathrm{tr}(ff^\dagger) \bullet \mathrm{tr}(gg^\dagger) \bullet \mathrm{id} = \bar{g}\bar{f}$ if $\mathrm{dom}(f) = \mathrm{dom}(g)$. Third,

$$\begin{aligned}
&\mathrm{tr}(ff^\dagger)^\dagger \circ \mathrm{tr}(ff^\dagger) \\
&= (\varepsilon \otimes \varepsilon) \circ (\sigma \otimes \mathrm{id}) \circ (ff^\dagger \otimes \mathrm{id} \otimes ff^\dagger \otimes \mathrm{id}) \circ (\mathrm{id} \otimes \sigma) \circ (\eta \otimes \eta) \\
&= \varepsilon \circ (ff^\dagger ff^\dagger \otimes \mathrm{id}) \circ \sigma \circ \eta \\
&= \varepsilon \circ (ff^\dagger \otimes \mathrm{id}) \circ \sigma \circ \eta \\
&= \mathrm{tr}(ff^\dagger) \qquad (*)
\end{aligned}$$

by Lemma 10. Therefore, for $\mathrm{dom}(f) = \mathrm{dom}(g)$:

$$\begin{aligned}
\overline{g\bar{f}} &= \overline{\mathrm{tr}(ff^\dagger) \bullet g} \\
&= \mathrm{tr}\left[\mathrm{tr}(ff^\dagger)^\dagger \bullet \mathrm{tr}(ff^\dagger) \bullet gg^\dagger\right] \bullet \mathrm{id} \\
&= \mathrm{tr}(ff^\dagger)^\dagger \bullet \mathrm{tr}(ff^\dagger) \bullet \mathrm{tr}(gg^\dagger) \bullet \mathrm{id} \\
&= \mathrm{tr}(ff^\dagger) \bullet \mathrm{tr}(gg^\dagger) \bullet \mathrm{id} \\
&= \bar{g}\bar{f}.
\end{aligned}$$

Fourth, $\bar{g}f = \mathrm{tr}(gg^\dagger) \bullet f = \mathrm{tr}(gg^\dagger) \bullet \mathrm{tr}(ff^\dagger) \bullet f$, and $f\overline{gf} = \mathrm{tr}(gff^\dagger g^\dagger) \bullet f$. The two are equal by a similar computation as $(*)$.

Finally, taking $g = f^\dagger$ shows that $\bar{f} = \mathrm{tr}(ff^\dagger) \bullet \mathrm{id} = gf$ by Lemma 10 and similarly $\bar{g} = fg$. Therefore the category is compact inverse (Cockett & Lack, 2002, Theorem 2.20). □

Next we build up to generalise Theorem 5, starting with the replacement for abelian groups. A *compact groupoid* is a compact dagger category where any morphism f is inverted by $f^\dagger$.

Lemma 12 *Compact groupoids are precisely compact inverse categories with invertible scalars.*

Proof Let $\mathbf{C}$ be a compact inverse category with invertible scalars. By Lemma 10, all endomorphisms are invertible. Let $f : A \to B$ be any morphism. Then $ff^\dagger$ is an isomorphism, and so f is (split) monic. Because $f = ff^\dagger f$, it follows that $ff^\dagger = \mathrm{id}_B$. Similarly $f^\dagger f$ is an isomorphism, so f is (split) epic, whence $f^\dagger f = \mathrm{id}_A$. Thus f is invertible. □

We can now show that any compact inverse category is a semilattice of compact groupoids. Write **CptInvCat** for the category of compact inverse categories and (strong) monoidal dagger functors, and **CptGpd** for the full subcategory of compact groupoids and (strong) monoidal functors.

Proposition 13 *If* $\mathbf{C}$ *is a compact inverse category, then*

$$\mathbf{S} = \{s \in \mathbf{C}(I, I) \mid ss^\dagger = s\}, \qquad s \wedge t = st, \qquad \top = \mathrm{id}_I,$$

is a semilattice, and for each $s \in \mathbf{S}$*, there is a compact groupoid* $F(s)$ *with the same objects as* $\mathbf{C}$ *and morphisms*

$$F(s)(A, B) = \{f \in \mathbf{C}(A, B) \mid \mathrm{tr}(ff^\dagger) = s\},$$

giving a semilattice $F : \mathbf{S}^{\mathrm{op}} \to$ **CptGpd** *of compact groupoids* $F(s \leq t)(f) = s \bullet f$.

The assignment $\mathbf{C} \mapsto F$ *extends to a functor* **CptInvCat** $\to$ **SLat**[**CptGpd**] *by sending a morphism* $G : \mathbf{C} \to \mathbf{C}'$ *to*

$$\varphi(s) = \psi_0^{-1} \circ G(s) \circ \psi_0, \qquad \theta_s(A) = G(A), \qquad \theta_s(f) = G(f),$$

where $\psi_0 : I' \to G(I)$ *is the structure isomorphism.*

Proof First, $\mathbf{S}$ is a commutative idempotent monoid by definition.

Next, we verify that $F(s)$ is a compact groupoid. Composition is well-defined: if $f : A \to B$ and $g : B \to C$ satisfy $\mathrm{tr}(ff^\dagger) = s = \mathrm{tr}(gg^\dagger)$, then by Lemma 10 and linearity and cyclicity of trace:

$$
\begin{aligned}
\operatorname{tr}\left((gf)(gf)^\dagger\right) &= \operatorname{tr}(g^\dagger g f f^\dagger)\\
&= \operatorname{tr}\left[(\operatorname{tr}(g^\dagger g) \bullet \mathrm{id}_B) \circ (\operatorname{tr}(ff^\dagger) \bullet \mathrm{id}_B)\right]\\
&= \operatorname{tr}(g^\dagger g) \bullet \operatorname{tr}(ff^\dagger) \bullet \operatorname{tr}(\mathrm{id}_B)\\
&= \operatorname{tr}(ff^\dagger) \bullet \operatorname{tr}(\mathrm{id}_B)\\
&= \operatorname{tr}\left[\mathrm{id}_B \circ (\operatorname{tr}(ff^\dagger) \bullet \mathrm{id}_B)\right]\\
&= \operatorname{tr}(\mathrm{id}_B \circ ff^\dagger)\\
&= \operatorname{tr}(ff^\dagger)\\
&= s.
\end{aligned}
$$

It is clear that $s \bullet \mathrm{id}_A$ play the role of identities in $F(s)$. The category $F(s)$ is monoidal, because if $\operatorname{tr}(ff^\dagger) = s = \operatorname{tr}(gg^\dagger)$, then $\operatorname{tr}((f \otimes g)(f \otimes g)^\dagger) = \operatorname{tr}(ff^\dagger \otimes gg^\dagger) = \operatorname{tr}(ff^\dagger)\operatorname{tr}(gg^\dagger) = s$. It also inherits the dagger from $\mathbf{C}$: if $\operatorname{tr}(ff^\dagger) = s$, then also $\operatorname{tr}(f^\dagger f) = \operatorname{tr}(ff^\dagger) = s$. Consequently, $F(s)$ inherits the property of being an inverse category from $\mathbf{C}$. Moreover, $F(s)$ is a compact dagger category: the units and counits are given by $s \bullet \eta_A$ and $s \bullet \varepsilon_A$. Finally, scalars $x \in F(s)(I, I)$ are those scalars $x \in \mathbf{C}(I, I)$ satisfying $x^\dagger x = s$, and form an abelian group with inverse $x^\dagger$ and unit s: for $xs = xx^\dagger x = x$; if $x^\dagger x = s = y^\dagger y$ then $(xy)^\dagger(xy) = x^\dagger x y^\dagger y = s^\dagger s = s$; and $xx^\dagger = s$. Lemma 12 therefore makes $F(s)$ a compact groupoid. Notice that F is a well-defined functor: if $s \le t$ and $\operatorname{tr}(ff^\dagger) = t$, then $st = t$, so $\operatorname{tr}((sf)(sf)^\dagger) = ss^\dagger \operatorname{tr}(ff^\dagger) = st = s$.

Now consider morphisms. If $\mathbf{G}\colon \mathbf{C} \to \mathbf{C}'$ is a monoidal dagger functor, say with structure isomorphisms $\psi_0\colon I' \to G(I)$ and $\psi_{A,B}\colon G(A) \otimes' G(B) \to G(A \otimes B)$, then it is easy to see that φ is a semilattice homomorphism, and that θ_s is a well-defined monoidal dagger functor that is moreover natural in s, because monoidal functors preserve dual objects and hence traces. Finally, it is clear that the assignment $G \mapsto (\varphi, f)$ is functorial. □

Notice that $\mathbf{S}$ contains all *dimension* scalars $\dim(A) = \operatorname{tr}(\mathrm{id}_A)$.

Lemma 14 *If* $F\colon \mathbf{S}^{\mathrm{op}} \to$ **CptGpd** *is a semilattice of compact groupoids, then the category* $\mathbf{C}$ *of Lemma 8 is a compact inverse category, and this gives a functor* **SLat[CptGpd]** $\to$ **CptInvCat**.

Proof Define the tensor product on objects on $\mathbf{C}$ as in $F(\top)$, and set the tensor unit I in $\mathbf{C}$ to be that of $F(\top)$. The fact that $F(s \le \top)$ are monoidal functors gives structure isomorphisms $\psi_s\colon A \otimes_s B \to A \otimes B$, where we write $\otimes_s$ for the tensor product in $F(s)$, and $\psi\colon I_s \to I$, where we write I_s for the tensor unit in $F(s)$. Define the tensor product of $f \in F(s)(A, B)$ and $g \in F(t)(C, D)$ to be

$$\psi_{s\wedge t} \circ \left(F(s \wedge t \le s)(f) \otimes_{s\wedge t} F(s \wedge t \le t)(g)\right) \circ \psi_{s\wedge t}^{-1}$$

in $F(s \wedge t)\big(A \otimes C, B \otimes D\big)$. Taking coherence isomorphisms and dual objects as in $F(\top)$, a tedious but straightforward calculation proves that the triangle and pentagon axioms are satisfied, that the snake equations are satisfied, and that $\mathbf{C}$ is a compact inverse category.

An even more tedious but still straightforward calculation shows that the functor induced by a morphism of semilattices of compact groupoids is monoidal. $\square$

Theorem 15 *The functors of Proposition 13 and Lemma 14 implement an equivalence* $\mathbf{CptInvCat} \simeq \mathbf{SLat}[\mathbf{CptGpd}]$.

Proof Starting with a compact inverse category $\mathbf{C}$, turning it into a semilattice of compact groupoids F, and turning that into compact inverse category again, results in the exact same compact inverse category $\mathbf{C}$. For example, the old homset $\mathbf{C}(A, B)$ equals the new homset $\coprod_{s \in \mathbf{C}(I,I) \mid ss^\dagger = s} \{f \in \mathbf{C}(A, B) \mid \mathrm{tr}(ff^\dagger) = s\}$ because any morphism f in $\mathbf{C}$ is of the form $s \bullet f$ for some scalar $ss^\dagger = s = \mathrm{tr}(ff^\dagger)$ by Proposition 11. Similarly, the new tensor product of $f \in F(s)(A, B)$ and $g \in F(t)(C, D)$ is

$$\begin{aligned}
&\psi_{s\wedge t} \circ \big(F(s \wedge t \leq s)(f) \otimes F(s \wedge t \leq s)(g)\big) \circ \psi_{s\wedge t}^{-1} \\
&= \psi_{s\wedge t} \circ (stf \otimes stg) \circ \psi_{s\wedge t}^{-1} \\
&= \psi_{s\wedge t} \circ (st \bullet (f \otimes g)) \circ \psi_{s\wedge t}^{-1} \\
&= (st \bullet (f \otimes g)) \circ \psi_{s\wedge t} \circ \psi_{s\wedge t}^{-1} \\
&= (s \bullet f) \otimes (t \bullet g) \\
&= f \otimes g,
\end{aligned}$$

again by Proposition 11, and because the natural isomorphism ψ cooperates with unitors and hence scalar multiplication, and so equals the old tensor product.

Now start with a semilattice of compact groupoids $F : \mathbf{S}^{\mathrm{op}} \to \mathbf{CptGpd}$. Lemma 14 turns it into a compact inverse category $\mathbf{C}$, which in turn becomes the following semilattice of compact groupoids $G : \mathbf{T}^{\mathrm{op}} \to \mathbf{CptGpd}$. The semilattice $\mathbf{T}$ is

$$\coprod_{s \in \mathbf{S}} \{t \in F(s)(I, I) \mid tt^\dagger = t\} = \coprod_{s \in \mathbf{S}} \{\mathrm{id}_I \in F(s)(I, I)\}$$

because each $F(s)$ is a groupoid, so $s \mapsto \mathrm{id}_I \in F(s)(I, I)$ is a semilattice isomorphism $\varphi : \mathbf{S} \to \mathbf{T}$. The construction of Proposition 13 gives $G(\varphi(s))$ the same objects as $F(\top)$. Morphisms $A \to B$ in $G(\varphi(s))$ are $f : A \to B$ in $F(t)(A, B)$ for some $t \in \mathbf{S}$ satisfying $\varphi(s) = \mathrm{tr}(ff^\dagger)$. Because $F(t)$ is a groupoid, s must be t, so $G(\varphi(s))$ and $F(t)$ have the exact same homsets and identities, and we may take θ to be the identity functor. Going through the construction of G shows that θ is in fact a monoidal dagger functor. $\square$

22.5 Examples

This section lists examples of compact inverse categories $\mathbf{C}$. For each example we will indicate how Proposition 13 works by writing $\mathbf{C}_0$ for the semilattice $\mathbf{S}$ and $\mathbf{C}_s$ for the compact groupoid $F(s)$.

Example 16 (*The fundamental compact groupoid*) Any topological space X with a fixed chosen point $x \in X$ gives rise to a compact groupoid $\mathbf{C}$:

- The objects of $\mathbf{C}$ are paths from x to x, more precisely, continuous functions $f\colon [0,1] \to X$ with $f(0) = f(1) = x$.
- The arrows $f \to g$ are homotopy classes of paths, more precisely, continuous functions $h\colon [0,1]^2 \to X$ such that $h(s,0) = f(s)$, $h(s,1) = g(s)$, and $h(0,t) = h(1,t) = x$, where h and h' are identified when there is a continuous function $H\colon [0,1]^3 \to X$ with $H(s,t,0) = h(s,t)$, $H(s,t,1) = h'(s,t)$, $H(s,0,u) = f(s)$, $H(s,1,u) = g(s)$, and $H(0,t,u) = H(1,t,u) = x$.
- The tensor product of objects is composition of paths according to some fixed reparametrisation, the tensor unit is the constant path. Reparametrisation leads to associators and unitors.
- Dual objects are given by reversal of paths.
- The dagger is given by reversal of homotopies.
- The unit η_f is the "birth of a double loop", a homotopy that "grows" from the constant path to the path $f^\dagger \circ f$ by travelling progressively further along f before travelling back along $f^\dagger$.
- The counit ε_f is the "contraction of a double loop", a homotopy that "shrinks" from the path $f^\dagger \circ f$ to the constant path.

In this case $\mathbf{C}_0$ is a one-element semilattice, and $\mathbf{C}_s = \mathbf{C}$ is already a groupoid.

Example 17 Any abelian group $\mathbf{C}$, considered as a discrete monoidal category, is a compact groupoid. In this case $\mathbf{C}_0$ is a one-element semilattice, and $\mathbf{C}_s = \mathbf{C}$ is already a groupoid.

Lemma 18 *If $\mathbf{C}$ is a compact (dagger/inverse) category, and S a family of (dagger) idempotents, then $\mathrm{Split}_S(\mathbf{C})$ is again (dagger/inverse) compact.*

In terms of Theorem 15, $\mathrm{Split}_S(\mathbf{C})_0 \simeq \mathbf{C}_0$, and $\mathrm{Split}_S(\mathbf{C})_s = \mathrm{Split}_{S_s}(\mathbf{C}_s)$, where $S_s = \{p \in S \mid \mathrm{tr}(p) = s\}$.

Proof Let $p\colon A \to A$ be in S. Define $\eta_p = (p^* \otimes p) \circ \eta_A\colon \mathrm{id}_I \to p \otimes p^*$ and $\varepsilon_p = \varepsilon_A \circ (p \otimes p^*)\colon p^* \otimes p \to \mathrm{id}_I$; these are well-defined morphisms in $\mathrm{Split}_S(\mathbf{C})$. Then indeed the snake equations hold: $p = (\varepsilon_A \otimes p) \circ (p \otimes p^* \otimes p) \circ (p \otimes \eta_A) = (\varepsilon_p \otimes p) \circ (p \otimes \eta_p)$. If $\mathbf{C}$ has a dagger, then so does $\mathrm{Split}_S(\mathbf{C})$, and $\eta_p = (\varepsilon_p \circ \sigma)^\dagger$. $\square$

Example 19 If $\mathbf{C}$ and $\mathbf{D}$ are compact inverse categories, then so is $\mathbf{C} \times \mathbf{D}$. In this case $(\mathbf{C} \times \mathbf{D})_0 \simeq \mathbf{C}_0 \times \mathbf{D}_0$, and $(\mathbf{C} \times \mathbf{D})_{(s,t)} = \mathbf{C}_s \times \mathbf{D}_t$. If $\mathbf{C}$ and $\mathbf{D}$ are compact groupoids, then so is $\mathbf{C} \times \mathbf{D}$.

Example 20 If $\mathbf{C}$ is a compact inverse category, and $\mathbf{G}$ is a groupoid, then $[\mathbf{G}, \mathbf{C}]_\dagger$, the category of functors $F : \mathbf{G} \to \mathbf{C}$ satisfying $F(f^{-1}) = F(f)^\dagger$ and natural transformations, is again a compact inverse category.

In this case $([\mathbf{G}, \mathbf{C}]_\dagger)_0 \simeq \mathbf{C}_0$, and $([\mathbf{G}, \mathbf{C}]_\dagger)_s$ has as morphisms natural transformations whose every component is in $\mathbf{C}_s$.

Proof If $\alpha : F \Rightarrow G$ is a natural transformation, its dagger is given by $(\alpha^\dagger)_A = (\alpha_A)^\dagger : G(A) \to F(A)$; naturality of $\alpha^\dagger$ follows from naturality of α together with the conditions $F(f)^\dagger = F(f^{-1})$ and $G(f)^\dagger = G(f^{-1})$. This makes $[\mathbf{G}, \mathbf{C}]_\dagger$ into a dagger category. It inherits the property $\alpha = \alpha\alpha^\dagger\alpha$ componentwise from $\mathbf{C}$, and is therefore an inverse category.

The tensor product of objects is given by $(F \otimes G)(A) = F(A) \otimes G(A)$, and on morphisms by $(F \otimes G)(f) = F(f) \otimes G(f)$. The tensor unit is the functor that is constantly I. Because the coherence isomorphisms in $\mathbf{C}$ are unitary, this makes $[\mathbf{G}, \mathbf{C}]_\dagger$ into a well-defined dagger symmetric monoidal category.

Finally, the dual object of $F : \mathbf{G} \to \mathbf{C}$ is given by $F^*(A) = F(A)^*$ and $F^*(f) = F(f)_*$. The unit $\eta_F : I \Rightarrow F^* \otimes F$ is given by $(\eta_F)_A = \eta_{F(A)}$, and the counit by $(\varepsilon_F)_A = \varepsilon_{F(A)}$. These are natural because any morphism $f : A \to B$ in $\mathbf{G}$ satisfies $ff^\dagger = \mathrm{id}_A$, whence $(F(f)_* \otimes F(f)) \circ \eta_{F(A)} = (\mathrm{id}_{B^*} \otimes f) \circ (\mathrm{id}_{B^*} \otimes f^\dagger) \circ \eta_{F(B)} = \eta_{F(B)}$. This makes $[\mathbf{G}, \mathbf{C}]_\dagger$ a compact inverse category. $\square$

22.6 Compact Groupoids

This section moves to a 2-categorical perspective, to connect to a characterisation of compact groupoids. A compact groupoid is the same thing as a *coherent 2-group* (Baez & Lauda, 2004). It is also known as a *crossed module*. Compact groupoids are classified by two abelian groups G and H and an element of the third cohomology group of G with coefficients in H, as worked out by Baez and Lauda (2004). The following proposition makes this more precise. In the nonsymmetric case, G need not be abelian, and there is an additional action of G on H.

Proposition 21 *A compact groupoid* $\mathbf{C}$ *is, up to equivalence, defined by the following data:*

- *the (abelian) group G of isomorphism classes of objects of* $\mathbf{C}$*, under $\otimes$, with unit I, and inverse given by dual objects;*
- *the abelian group H of scalars* $\mathbf{C}(I, I)$ *under composition with unit* id_I *and inverse $\dagger$;*
- *the conjugation action $G \times H \to H$ that takes (A, s) to* $\mathrm{tr}(A \otimes s) = s$*;*
- *the 3-cocycle $G \times G \times G \to H$ that takes (A, B, C) to* $\mathrm{Tr}(\alpha_{A,B,C})$.

The above data form the objects of a (weak) 2-category **Cocycle**, with 1- and 2-cells as in Baez and Lauda (2004, Theorem 43).

Proof (*Proof sketch*) See Baez and Lauda (2004, Sect. 8). The trick is the following. First, we may assume that $\mathbf{C}$ is skeletal. Then, we may adjust the tensor product such that all unitors and units and counits (but not the associators!) are identities. The pentagon equation ensures that the trace of the associator is in fact a 3-cocycle. $\square$

The proof of Theorem 15 is the only place where we have used that in a semilattice F of categories all $F(s)$ must have the same objects. It was needed because if the functor θ_s is to be an isomorphism, it must give a bijection between the objects of $F(s)$ and $F(\top)$. We now move to a (weak) 2-categorical perspective to remove this restriction.

Definition 22 Redefine the category $\mathbf{SLat}[\mathbf{V}]$ of Definition 4 to become a (weak) 2-category as follows:

- 0-cells are functors $F : \mathbf{S}^{\mathrm{op}} \to \mathbf{V}$ for some semilattice $\mathbf{S}$;
- 1-cells $F \to F'$ consist of a morphism $\varphi : \mathbf{S} \to \mathbf{S}'$ of semilattices and a natural transformation $\theta : F \Rightarrow F' \circ \varphi$;
- 2-cells $(\varphi, \theta) \to (\varphi', \theta')$ exist when $\varphi \leq \varphi'$ and then are natural transformations $\gamma : \theta \Rrightarrow \theta' \circ (\mathrm{id} * (\varphi \leq \varphi'))$.

Composition is by pasting.

$$\begin{tikzcd}
\mathbf{S}^{\mathrm{op}} \arrow[dd, bend right, "\varphi"'] \arrow[dd, bend left, "\varphi'"] \arrow[dd, phantom, "\leq"] \arrow[drr, bend left, "F"] & & \\
{} \arrow[rr, phantom, "\theta \Downarrow \overset{\gamma}{\Rrightarrow} \Downarrow \theta'"] & & \mathbf{V} \\
\mathbf{S}'^{\mathrm{op}} \arrow[urr, bend right, "F'"'] & &
\end{tikzcd}$$

Write $\mathbf{SLat}_{=}[\mathbf{CptGpd}]$ for the full sub-2-category where all categories $F(s)$ have the same objects.

To be precise, in $\mathbf{SLat}[\mathbf{CptGpd}]$, 2-cells γ are modifications: for each $s \in \mathbf{S}$ and $A \in F'(\varphi(s))$, there is a morphism $\gamma_{s,A} : \theta_s(A) \to \theta'_s\big(F'\big(\varphi(s) \leq \varphi'(s)\big)(A)\big)$ that is natural in s as well as A.

Lemma 23 *There is a (weak) 2-equivalence* $\mathbf{SLat}[\mathbf{CptGpd}] \simeq \mathbf{SLat}_{=}[\mathbf{CptGpd}]$.

Proof First, observe that two 0-cells $F, G : \mathbf{S}^{\mathrm{op}} \to \mathbf{CptGpd}$ are equivalent in $\mathbf{SLat}[\mathbf{CptGpd}]$ exactly when there is a natural monoidal equivalence $F(s) \simeq G(s)$. Therefore, it suffices to construct, for each F, such a G such that each $G(s)$ has the same objects. Let κ_s be the cardinality of the objects of $F(s)$, and let κ be the maximum of all κ_s. Define $G(s)$ to be equal to $F(s)$, except that we add κ isomorphic copies of the tensor unit I. There is an obvious monoidal structure on $G(s)$, and by construction there is a monoidal equivalence $F(s) \simeq G(s)$, so that $G(s)$ is automatically a compact groupoid. We may furthermore relabel the objects of $G(s)$ to be ordinal numbers, so that all $G(j)$ have the same objects. $\square$

Theorem 24 *There is a (weak) 2-equivalence* $\mathbf{CptInvCat} \simeq \mathbf{SLat}[\mathbf{Cocycle}]$*, where* $\mathbf{CptInvCat}$ *has natural transformations as 2-cells.*

Proof The (weak) 2-equivalence $\mathbf{CptGpd} \simeq \mathbf{Cocycle}$ of Baez and Lauda (2004, Theorem 43) induces a (weak) 2-equivalence $\mathbf{SLat}[\mathbf{CptGpd}] \simeq \mathbf{SLat}[\mathbf{Cocycle}]$ by postcomposition. Combine this with the equivalence $\mathbf{SLat}_=[\mathbf{CptGpd}] \simeq \mathbf{SLat}[\mathbf{CptGpd}]$ of Lemma 23 and the equivalence $\mathbf{CptInvCat} \simeq \mathbf{SLat}_=[\mathbf{CptGpd}]$ of Theorem 15; the latter still holds after the change of Definition 22. □

22.7 Concluding Remarks

We conclude by discussing the many questions left open and raised in this paper. First, one could investigate generalising the results in this paper from categories to semicategories. Second, one could investigate generalising the results in this paper from compact categories to monoidal categories where every object has a dual.

22.7.1 Traced Inverse Categories

Inverse categories provide semantics for reversible programs, but higher-order aspects of reversible programming remain unclear. Compact categories are closed and hence provide semantics for higher-order programming. Theorem 15 shows that compact inverse categories are, in a sense, degenerate. But one of the most interesting aspects of higher-order programming, tail recursion, doesn't need compact categories for semantics, and can already be modeled in traced monoidal categories. (But see also Kaarsgaard et al. (2017).) Now every traced monoidal category can be monoidally embedded in a compact category (Joyal et al., 1996). One can prove that there exists a left dagger biadjoint to the forgetful functor from dagger compact categories to dagger traced categories. There is also a left adjoint to the forgetful functor from compact inverse categories to compact dagger categories, but the latter is not faithful. Hence there is a left dagger biadjoint to the forgetful functor from compact inverse categories to traced inverse categories, but its unit does not embed any traced inverse category into a compact inverse category. Therefore Theorem 15 does not show that all traced inverse categories degenerate. Indeed, the category **PInj** of sets and injections is the universal inverse category (Kastl, 1979), and is also traced (Hines, 1997; Haghverdi & Scott, 2006), but it fails Lemma 10, irrespective of which tensor product it carries, as the swap map on the two element set is not a scalar multiple of the identity. That leaves a valid question: what do traced inverse categories look like?

22.7.2 Idempotents

A *subunit* in a monoidal category $\mathbf{C}$ is a subobject $r\colon R \rightarrowtail I$ for which $r \otimes \mathrm{id}_R$ is invertible (Moliner et al., 2019); they form a semilattice $\mathrm{ISub}(\mathbf{C})$. The following lemma shows that in compact inverse categories, up to splitting idempotents, the semilattice $\mathbf{C}_0$ is precisely that of subunits. See also Schwab and Schwab (2015) for structure theorems of inverse categories in which all idempotents split.

Lemma 25 *Let* $\mathbf{C}$ *be a compact inverse category.*

(a) A map $r\colon R \to I$ *is a subunit if and only if* $r^\dagger r = \mathrm{id}$.
(b) Any subunit r *induces an element* $rr^\dagger$ *of* $\mathbf{C}_0$.
(c) If idempotents split in $\mathbf{C}$*, any element of* $\mathbf{C}_0$ *is* $rr^\dagger$ *for a unique subunit* r*; this gives an isomorphism between the semilattices* $\mathbf{C}_0$ *and* $\mathrm{ISub}(\mathbf{C})$.

Proof For (a), first notice that if $r\colon R \to I$ is monic, then because $r = rr^\dagger r$ in fact r is an isometry, that is, $r^\dagger r = \mathrm{id}$. We will show that for isometries r, the condition that $r \otimes \mathrm{id}_R$ is invertible holds automatically, with the inverse being $r^\dagger \otimes \mathrm{id}_R$. It suffices to show that $(r \otimes \mathrm{id}_R)(r^\dagger \otimes \mathrm{id}_R) = \mathrm{id}_{I\otimes R}$. But

$$\mathrm{id}_{I\otimes R} = \mathrm{id}_I \otimes (r^\dagger r) = \mathrm{id}_I \otimes (r^\dagger(rr^\dagger)r) = (rr^\dagger) \otimes (r^\dagger r) = (rr^\dagger) \otimes \mathrm{id}_R.$$

Thus the subunits are precisely the (subobjects represented by) isometries.

Part (b) is obvious: if r is an isometry, then $s = rr^\dagger\colon I \to I$ satisfies $s = ss^\dagger$.

Part (c) follows from Cockett and Lack (2002, Lemma 2.25), as does the fact that the maps of (b) and (c) are each other's inverses. It is easy to see that both maps preserve the order structure using (Moliner et al., 2019, Proposition 2.8). □

Now there are two ways to 'localise' $\mathbf{C}$ to $r \in \mathrm{ISub}(\mathbf{C})$. The localisation $\mathbf{C}\big|_r$ according to Moliner et al. (2019) has objects A such that $r \otimes \mathrm{id}_A$ is invertible, and all morphisms between those objects. The localisation $\mathbf{C}_{rr^\dagger}$ above has all objects, but only those morphisms f satisfying $\mathrm{tr}(ff^\dagger) = rr^\dagger$. These two localisations are different. The former localises with respect to the tensor product, whereas the latter localises with respect to composition.

Generally, taking semilattices of categories is a completion procedure. Does it generalise to (weak) 2-categories? If so, the above may be the special cases of a single object and of unique 2-cells, and could form a higher-categorical analogue of the Eckmann-Hilton argument in the Baez-Dolan stabilisation hypothesis (Baez & Dolan, 1995). Is there a relationship with Hayashi (1985)?

22.7.3 Internal Descriptions

Groupoids are precisely special dagger Frobenius algebras in the category **Rel** of sets and relations (Heunen et al, 2013). Compact groupoids are precisely special dagger

Frobenius algebras in the category **Rel**(**Gp**) of relations over the regular category of groups, see Gran et al. (2019). Can inverse categories similarly be described as certain monoids in a category of relations?

22.7.4 Bratteli Diagrams and C-Algebras*

Describing compact inverse categories through a diagram of groupoids resembles describing an AF C*-algebra as a diagram of finite-dimensional C*-algebras (Amini et al, 2015). It is very fruitful to work with this so-called Bratteli diagram directly rather than with the C*-algebra itself. More generally, inverse semigroups are a popular way to generate C*-algebras (Duncan & Paterson, 1985), as it is easier to work with the inverse semigroup directly, and moreover this captures many important classes of C*-algebras (see e.g. Starling (2016)): AF C*-algebras, graph C*-algebras, tiling C*-algebras, self-similar group C*-algebras, subshift C*-algebras, C*-algebras of ample étale groupoids, and C*-algebras of Boolean dynamical systems. There is also a multiply-typed version building a C*-algebra from a so-called higher rank graph (Kumjian & Pask, 2000). Can one similarly generate a C*-algebra from a compact inverse category, and is there a relationship to these other constructions? A first step might be to extend (Linckelmann, 2013) to possibly infinite categories by adding a norm.

References

Abramsky, S. (1993). Interaction categories. In G. Burn, S. Gay, & M. Ryan, (Eds.). Theory and formal methods. Berlin: Springer (1993).

Abramsky, S. (2004). High-level methods for quantum computation and information. In *Logic in computer science* (pp. 410–414). IEEE Computer Society.

Abramsky, S. (2005). Abstract scalars, loops, and free traced and strongly compact closed categories. In *Conference on Algebra and Coalgebra* (Vol. 3629, pp. 1–31). Lecture notes in computer science. Berlin: Springer.

Abramsky, S. (2008). *Mathematics of Quantum Computing and Technology*. Temperley-Lieb algebra: From knot theory to logic and computation via quantum mechanics (pp. 415–458). Taylor and Francis (2008).

Abramsky, S. (2008). No-cloning in categorical quantum mechanics. In *Semantic techniques for quantum computation* (pp. 1–28). Cambridge University Press, Cambridge.

Abramsky, S., & Coecke, B. (2004). A categorical semantics of quantum protocols. In *Logic in computer science* (Vol. 19, pp. 415–425). IEEE Computer Society.

Abramsky, S., & Heunen, C. (2012). *Clifford lectures* (Vol. 71, pp. 1–24), H*-algebras and nonunital Frobenius algebras: First steps in infinite-dimensional categorical quantum mechanics. American Mathematical Society.

Abramsky, S., & Heunen, C. (2016). *Logic and algebraic structures in quantum computing and information*. Operational theories and categorical quantum mechanics (pp. 88–122). Number 45 in lecture notes in logic. Cambridge University Press, Cambridge.

Abramsky, S. (2005). A structural approach to reversible computation. *Theoretical Computer Science, 347*(3), 441–464.
Abramsky, S. (2013). Coalgebras, chu spaces, and representations of physical systems. *Journal of Philosophical Logic, 42*(3), 551–574.
Abramsky, S., & Coecke, B. (2005). Abstract physical traces. *Theory and Applications of Categories, 14,* 111–124.
Abramsky, S., & Duncan, R. (2006). A categorical quantum logic. *Mathematical Structures in Computer Science, 16*(3), 469–489.
Amini, M., Elliott, G. A., & Golestani, N. (2015). The category of Bratteli diagrams. *Canadian Journal of Mathematics, 67,* 990–1023.
Baez, J. C., & Dolan, J. (1995). Higher-dimensional algebra and topological quantum field theory. *Journal of Mathematical Physics, 36,* 6073–6105.
Baez, J. C., & Lauda, A. (2004). Higher-dimensional algebra V: 2-groups. *Theory and Applications of Categories, 12,* 423–491.
Clifford, A. H. (1941). Semigroups admitting relative inverses. *Annals of Mathematics, 42*(4), 1037–1049.
Cockett, J. R. B., & Lack, S. (2002). Restriction categories I: Categories of partial maps. *Theoretical Computer Science, 270*(1–2), 223–259.
DeWolf, D., & Pronk, D. (2018). The Ehresmann-Schein-Nambooripad theorem for inverse categories. *Theory and Applications of Categories, 33*(27), 813–831.
Duncan, J., & Paterson, A. L. T. (1985). C*-algebras of inverse semigroups. *Proceedings of the Edinburgh Mathematical Society, 28,* 41–58.
Ehresmann, C. (1958). Gattungen von lokalen Strukturen. *Jahresbericht der Deutschen Mathematiker-Vereinigung, 60,* 49–77.
Ehresmann, C. (1960). Catégories inductives et pseudogroupes. *Annales de l'Institut Fourier, 10,* 307–332.
Giles, B. G. (2014). An investigation of some theoretical aspects of reversible computing. Ph.D. thesis, University of Calgary.
Gran, M., Heunen, C., & Tull, S. (2019). Monoidal characterisation of groupoids and connectors. *Topology and Applications*.
Haghverdi, E., & Scott, P. (2006). A categorical model for the geometry of interaction. *Theoretical Computer Science, 350,* 252–274.
Hayashi, S. (1985). Adjunction of semifunctors: Categorical structures in nonextensional lambda calculus. *Theoretical Computer Science, 41,* 95–104.
Heunen, C., & Vicary, J. (2019). *Categories for Quantum Theory*. Oxford: Oxford University Press.
Heunen, C., Contreras, I., & Cattaneo, A. S. (2013). Relative Frobenius algebras are groupoids. *Journal of Pure and Applied Algebra, 217,* 114–124.
Hines, P. (1997). The algebra of self-similarity and its applications. Ph.D. thesis, University of Wales.
Hoehnke, H. -J. (1989). On certain classes of categories and monoids constructed from abstract Mal'cev clones II. In *Conference on Universal Algebra and Related Topics* (Vol. B62, pp. 30–64). Potsdamer Forschungen.
Jarek, P. (1964). Commutative regular semigroups. *Colloquium Mathematicum, XI, 1*(2), 195–208.
Joyal, A., Street, R., & Verity, D. (1996). Traced monoidal categories. *Mathematical Proceedings of the Cambridge Philosophical Society, 119*(3), 447–468.
Kaarsgaard, R., Axelsen, H. B., & Glück, R. (2017). Join inverse categories and reversible recursion. *Journal of Logical and Algebraic Methods in Programming, 87,* 33–50.
Kastl, J. (1979). *Algebraische Modelle, Kategorien und Gruppoide*, chapter Inverse categories (Vol. 7, pp. 51–60). Studien zur Algebra und ihre Anwendungen. Akademie-Verlag.
Kumjian, A., & Pask, D. (2000). Higher rank graph C*-algebras. *New York Journal of Mathematics, 6,* 1–20.
Lawson, M. V. (1998). *Inverse semigroups*. World Scientific.

Linckelmann, M. (2013). On inverse categories and transfer in cohomology. *Proceedings of the Edinburgh Mathematical Society, 56,* 187–210.

Moliner, P. E., Heunen, C., & Tull, S. (2019). Tensor topology. *Journal of Pure and Applied Algebra, 224*(10), 106378.

Nambooripad, K. S. S. (1979). *Structure of regular semigroups*. American Mathematical Society.

Romanowska, A. B., & Smith, J. D. H. (1997). Duality for semilattice representations. *Journal of Pure and Applied Algebra, 115*(3), 289–308.

Schein, B. M. (1979). On the theory of inverse semigroups and generalised groups. *American Mathematical Society Translations, 113*(2), 89–122.

Schwab, E., & Schwab, E. D. (2015). On inverse categories with split idempotents. *Archivum Mathematicum, 51,* 13–25.

Seely, R. A. G. (1989). Linear logic, *-autonomous categories and cofree coalgebras. *Contemporary Mathematics, 92,* 371–382.

Starling, C. (2016). C*-algebras of Boolean inverse monoids—Traces and invariant means. *Documenta Mathematica, 21,* 809–840.

Chapter 23
Reductive Logic, Proof-Search, and Coalgebra: A Perspective from Resource Semantics

Alexander V. Gheorghiu, Simon Docherty, and David J. Pym

Abstract The reductive, as opposed to deductive, view of logic is the form of logic that is, perhaps, most widely employed in practical reasoning. In particular, it is the basis of logic programming. Here, building on the idea of uniform proof in reductive logic, we give a treatment of logic programming for BI, the logic of bunched implications, giving both operational and denotational semantics, together with soundness and completeness theorems, all couched in terms of the resource interpretation of BI's semantics. We use this set-up as a basis for exploring how coalgebraic semantics can, in contrast to the basic denotational semantics, be used to describe the concrete operational choices that are an essential part of proof-search. The overall aim, toward which this paper can be seen as an initial step, is to develop a uniform, generic, mathematical framework for understanding the relationship between the deductive structure of logics and the control structures of the corresponding reductive paradigm.

23.1 Introduction

While the traditional deductive approach to logic begins with premisses and in step-by-step fashion applies proof rules to derive conclusions, the complementary reductive approach instead begins with a putative conclusion and searches for premisses sufficient for a legitimate derivation to exist by systematically reducing the space of possible proofs. A first step in developing a mathematical theory of reductive logic, in the setting of intuitionistic and classical logic, has been given in Pym & Ritter (2004).

A. V. Gheorghiu (✉) · S. Docherty · D. J. Pym
University College London, London, England
e-mail: alexander.gheorghiu.19@ucl.ac.uk

S. Docherty
e-mail: s.docherty@ucl.ac.uk

D. J. Pym
e-mail: d.pym@ucl.ac.uk

A. Palmigiano and M. Sadrzadeh (eds.), *Samson Abramsky on Logic and Structure in Computer Science and Beyond*, Outstanding Contributions to Logic 25,
https://doi.org/10.1007/978-3-031-24117-8_23

Not only does this picture more closely resemble the way in which mathematicians actually prove theorems and, more generally, the way in which people solve problems using formal representations, it also encapsulates diverse applications of logic in computer science such as the programming paradigm known as *logic programming*, the proof-search problem at the heart symbolic of AI, automated theorem proving, precondition generation in program verification, and more—see, for example, Kowalski (1986). It is also reflected at the level of truth-functional semantics—the perspective on logic utilized for the purpose of model checking and thus verifying the correctness of industrial systems—wherein the truth value of a formula is calculated according to the truth values of its constituent parts.

The definition of a system of logic may be given *proof-theoretically* as a collection of rules of inference that, when composed, determine proofs; that is, formal constructions of arguments that establish that a conclusion is a consequence of some assumptions:

$$\frac{\text{Premiss}_1 \ldots \text{Premiss}_k}{\text{Conclusion}} \Downarrow$$

This systematic use of symbolic and mathematical techniques to determine the forms of valid deductive argument defines *deductive logic*: conclusions are inferred from assumptions.

This is all very well as a way of defining what proofs are, but it relatively rarely reflects either the way in which logic is used in practical reasoning problems or the method by which proofs are actually found. Rather, proofs are more often constructed by starting with a desired, or putative, conclusion and applying the rules of inference 'backwards'. In this usage, the rules are sometimes called *reduction operators*, read from conclusion to premisses, and denoted

$$\frac{\text{Sufficient Premiss}_1 \ldots \text{Sufficient Premiss}_k}{\text{Putative Conclusion}} \Uparrow$$

Constructions in a system of reduction operators are called *reductions* and this view of logic is termed 'reductive' (Pym & Ritter, 2004). The space of reductions of a (provable) putative conclusion is larger than its space of proofs, including also failed searches.

The background to these issues, in the context of capturing human reasoning, is discussed extensively in, for example, the work of Kowalski (1979) and Bundy (1985). Reductive reasoning lies at the heart of widely deployed proof assistants such as LCF (Gordon et al., 1979), HOL (Gordon & Melham, 1993), Isabelle (Paulson, 1994), Coq (The Coq Proof Assistant, 2023; Bertot & Castéran, 2004), Twelf (The Twelf Project, 2023; Pfenning & Schürmann, 1999), and more; most of which can be seen as building on Milner's theory of tactical proof Milner (1984)—that is, 'tactics' and 'tacticals' for proof-search—which gave rise to the programming language ML (Paulson, 1996).

In deductive logic, if one has proved a set of premisses for a rule, then one immediately has a proof of the conclusion by an application of the rule; and, usually, the order in which the premisses are established is of little significance. In reductive logic, however, when moving from a putative conclusion to some choice of sufficient premisses, a number of other factors come into play. First, the order in which the premisses are attempted may be significant not only for the logical success of the reduction but also for the efficiency of the search. Second, the search may fail, simply because the putative conclusion is not provable. These factors are handled by a *control process*, which is therefore a first-class citizen in reductive logic.

Logic programming can be seen as a particular form of implementation of reductive logic. While this perspective is, perhaps, somewhat obscured by the usual presentation of Horn-clause logic programming with SLD-resolution (HcLP)—see, for example, Lloyd (1984) among many excellent sources—it is quite clearly seen in the presentation of logic programming for intuitionistic logic (hHIL) in Miller (1989), which generalizes to the higher-order case (Miller & Nadathur, 2012).

Miller's presentation uses an operational semantics that is based on a restriction of Gentzen's sequent calculus LJ to hereditary Harrop formulas, given by the following mutually inductive definition:

$$\begin{aligned} D &::= A \in \mathbb{A} \mid \forall x\, D \mid G \to A \mid D \wedge D \\ G &::= A \in \mathbb{A} \mid \exists x\, G \mid D \to G \mid G \wedge G \mid G \vee G \end{aligned}$$

—here, $\mathbb{A}$ is a denumerable set of propositional letters.

The basic idea is that a program is a list of 'definite' clauses D, and a query is given by a 'goal' formula G, so that a *configuration* in HhIL is given by a sequent

$$D_1, \ldots D_k \vdash G$$

We are assuming that all formulas in the sequent are closed, and therefore suppress the use of quantifiers.

The execution of a configuration $P \vdash G$ is defined by the search for a 'uniform' proof,

- Apply right rules of LJ reductively until G is reduced to an atom (here we are assuming some control strategy for selecting branches);
- At a sequent $P \vdash A$, choose (assuming some selection strategy) a definite clause of the form $G \to A$ in the program, and apply the $\to L$ rule of LJ reductively to yield the goal $P \vdash G'$. More precisely, due to the (suppressed) quantifiers, we use *unifiers* to match clauses and goals; that is, in general, we actually compute a substitution θ such that, for a clause $G' \to B$ in the program, we check $A\theta = B\theta$ and proceed to the goal $P \vdash G'\theta$. This step amounts to a 'resolution'.

If all branches terminate successfully, we have computed an overall, composite substitution θ such that $P \vdash G\theta$ is valid in intuitionistic logic. This θ is the output of the computation.

The denotational semantics of this logic programming language amounts to a refinement of the term model construction for the underlying logic that, to some extent at least, follows the operational semantics of the programming language; that is, the control process for proof-search that delivers computation.

In Miller's setting, the denotational semantics is given in terms of the least fixed point of an operator T on the complete lattice of Herbrand interpretations. The operator is generated by the implicational clauses in a program, and its least fixed point characterizes the space of goals that is computable by the program in terms of the resolution step in the operational semantics. Similar constructions are provided for a corresponding first-order dependent type theory in Pym (1990).

A general perspective may be offered by proof-theoretic semantics, where model-theoretic notions of validity are replaced by proof-theoretic ones (Schroeder-Heister, 2008, 2017). In the setting of term models, especially for reductive logic (Pym, 2019), it is reasonable to expect a close correspondence between these notions.

In this paper, we study logic programming for *the logic of Bunched Implications* (BI) (O'Hearn & Pym, 1999)—which can be seen as the free combination of intuitionistic propositional logic and multiplicative intuitionistic linear logic—in the spirit of Miller's analysis for intuitionistic logic, building on early ideas of Armelín (2001; 2002). We emphasise the fundamental rôle played by the reductive logic perspective; and, we begin the development of a denotational semantics using coalgebra, that incorporates representations of the control processes that guide the construction of proof-search procedures. This is a natural choice if one considers that coalgebra can be seen as the appropriate algebraic treatment of systems with state (Jacobs, 2016; Komendantskaya et al., 2016; Pym, 2019). The choice of BI serves to highlight some pertinent features of the interaction between reduction operators and control; and, results in an interesting, rational alternative to the existing widely studied substructural logic programming languages based on Linear Logic (Hodas & Miller, 1994; Hodas, 1994; Cervesato et al., 2000; Hodas et al., 2002; López & Polakow, 2005). That is, we discuss how the coalgebraic model offers insight into control, generally speaking, and the context-management problem for multiplicative connectives, in particular, and how a general approach to handling the problem can be understood abstractly.

The motivations and application of BI as a logic are explained at some length in Sect. 23.2, which also sets up proof-search in the context of our view of reductive logic, in general, and the corresponding notions of hereditary Harrop formulas and uniform proof, in particular. The bunched structure of BI's sequent calculus renders these concepts quite delicate. We introduce BI from both a semantic, truth-functional perspective and a proof-theoretic one, through its sequent calculus. Before that, though, we begin with a motivation for BI as a logic of resources. While this interpretation is formally inessential, it provides a useful way to think the use of the multiplicative connectives and the computational challenges to which they give rise. Furthermore, the resource interpretation of BI's semantics stands in contrast with the proof-theoretic resource-interpretation of Linear Logic, and so helps to illustrate informally how BI and Linear Logic differ. In the sequel, we also give a precise, purely logical distinction between these two logics.

In Sect. 23.3, we establish the least fixed point semantics of logic programming with BI, again in the spirit of Miller's presentation.

While the least fixed point semantics is logically natural and appealing in its conceptual simplicity, its major weakness is that despite being constructed from an abstraction of behaviour, it lacks any explicit representation of the control processes that are central to the operational semantics. These include the selection operations alluded to above in our sketch of uniform proofs, and choices such as depth-first versus breadth-first navigation of the space of searches. In Sect. 23.4, we introduce the coalgebraic framework as a model of behaviour. The value of the choice of BI to illustrate these ideas—in addition to BI's value as logic—can now been seen: it has enough structure to challenge the approach (cf. work of Komendantskaya, Power, and Schmidt (Komendantskaya et al., 2016)) without becoming conceptually too complex to understand clearly.

In Sect. 23.5, we illustrate the ideas introduced in this paper with a simple example based on databases. Finally, in Sect. 23.6, we summarize our contribution and consider some directions for further research.

23.2 Logic Programming with Resource Semantics

We study proof-search in the logic of Bunched Implications (BI) (O'Hearn & Pym, 1999) not because it is easy, but because it is not. As a logic BI has a well-behaved metatheory admitting familiar categorical, algebraic, and truth-functional semantics which have the expected dualities (Pym et al., 2004; Galmiche et al., 2005; Pym, 2002; Docherty, 2019; Pym, April 2019), as well as a clean proof theory including Hilbert, natural deduction, sequent calculi, tableaux systems, and display calculi (Pym, 2002; Galmiche et al., 2005; Brotherston, 2012; Docherty, 2019). Moreover, BI has found significant application in program verification as the logical basis for Reynolds' Separation Logic, which is a specific theory of (Boolean) BI, and its many derivatives (Ishtiaq & O'Hearn, 2001).

What makes proof-search in the logic both difficult and interesting is the interaction between its additive and multiplicative fragments.

23.2.1 The Logic of Bunched Implications

We begin with a brief summary of the semantics and proof theory of BI. One way to understand it is by contrast with Girard's Linear Logic (LL) (Girard, 1987), which is well-known for its computational interpretation developed by Abramsky (1993) and others.

In LL, the structural rules of Weakening and Contraction are regulated by modalities, ! (related to $\Box$ in modal logic) and ? (related to $\Diamond$ in modal logic). To see how this works, consider the left and right (single-conclusioned) sequent calculus rules for the ! modality,

$$\frac{\Gamma, \phi \vdash \psi}{\Gamma, !\phi \vdash \psi} \quad \text{and} \quad \frac{!\Gamma \vdash \phi}{!\Gamma \vdash !\phi}$$

and note that Weakening and Contraction arise as

$$\frac{\Gamma \vdash \psi}{\Gamma, !\phi \vdash \psi}\ W \quad \text{and} \quad \frac{\Gamma, !\phi, !\phi \vdash \psi}{\Gamma, !\phi \vdash \psi}\ C,$$

respectively.

From our perspective, there are two key consequences of this set-up. First, proof-theoretically, the relationship between intuitionistic (additive) implication, $\rightarrow$, and linear (multiplicative) implication, $\multimap$, is given by Girard's translation; that is,

$$\phi \rightarrow \psi \equiv (!\phi) \multimap \psi$$

Second, more semantically, LL has a rudimentary interpretation as a logic of resource via the so-called *number-of-uses* reading, in which, in a sequent $\Gamma \vdash \psi$, the number of occurrences of a formula ϕ in Γ determines the number of times ϕ may be 'used' in ψ. The significance of the modality ! can now be seen: if $!\phi$ is in Γ, then ϕ may be used any number of times in ψ, including zero, and this reading is wholly consistent with the forms of Weakening and Contraction.

The relationship between logic and structure offered in the above reading has been characterized by Abramsky (2008) as the *intrinsic* views. By comparison, BI is, perhaps, the prime example of the *descriptive* view of resource; that is, in contrast to the case for LL, in the resource interpretation of BI a proposition is not a resource itself, but a declaration about the state of some resources. Nonetheless, the constructive reading of BI's implications mean that one can read a sequent, in particular formulas in a context, as things available for the constructions of the formula. This tension is explored further in Sect. 23.3. Moreover, although our discussion of logic programming with BI will be guided by the resource semantics, it is in no way essential to the logic's metatheory.

The original presentation of BI emphasised it as an interesting system of formal logic. Technically, it is the combination of intuitionistic logic (IL) and multiplicative intuitionistic linear logic (MILL), and is conservative over both.

Definition 23.2.1 (*Formulas, Additive and Multiplicative Connectives*) Let $\mathbb{A}$ be a denumerable set of propositional letters. The formulas of BI are defined by the grammar:

$$\phi, \psi ::= \top \mid \bot \mid \top^* \mid A \in \mathbb{A} \mid \phi \wedge \psi \mid \phi \vee \psi \mid \phi \rightarrow \psi \mid \phi * \psi \mid \phi \mathrel{-\!\!*} \psi$$

Symbols in $\{\wedge, \vee, \rightarrow, \top, \bot\}$ are additive connectives, and symbols in $\{*, \mathrel{-\!\!*}, \top^*\}$ are multiplicative connectives.

Following the resource discussion above we introduce BI from a semantic (truth-functional) perspective, before moving on to its proof theory.

Informally, a judgement $m \models \phi \wedge \psi$ is a declaration that the resource m satisfies ϕ and satisfies ψ, meanwhile the judgement $m \models \phi * \psi$ says that it can be *split* into two parts n and n' such that $n \models \phi$ and $n' \models \psi$.

Example 23.2.2 Consider the case where resources are gold coins and composition is summation. An atomic proposition is a brand of chocolate, and a judgement is a valuation of the cost.

Suppose chocolate bar A costs two gold coins, and chocolate bar B costs three, then we may write $3 \models A \wedge B$ to say that three gold coins suffice for both chocolates. Moreover, $7 \models A * B$ since seven gold coins may be split into two pile of three and four coins, and $3 \models A$ and $4 \models B$. Notice the persistence of the judgement; that is, since $2 < 3$ and two gold coins suffice for A, so do three. □

While any set of objects can, in some sense, be considered an abstract notion of resource, some simple assumptions are both natural and useful. For example, from our experience of resources it makes sense to assume that elements can be *combined* and *compared*. This indicates an appropriate structure for the abstract treatment of them: ordered monoids,

$$\mathcal{M} = \langle \mathbb{M}, \sqsubseteq, \circ, e \rangle$$

in which $\sqsubseteq$ is a preorder on $\mathbb{M}$, $\circ$ is a monoid on $\mathbb{M}$, and e is a unit for $\circ$. That is, combination is given by $\circ$ and comparison given $\sqsubseteq$. Examples of such structures are given by the natural numbers, $\langle \mathbb{N}, \geq, +, 0 \rangle$ and, in Separation Logic, by concatenation and containment of blocks of computer memory.

Such monoids form a semantics of BI with satisfaction given as in Fig. 23.1, but with Beth's treatment of disjunction, and with $\perp$ never satisfied—see Lambek & Scott (1986); Galmiche et al. (2005, 2019). However, for our semantics of logic programming with BI, we are able to work in a much simpler setting, without $\perp$, and in which occurrences of disjunction are restricted to the right-hand sides of sequents, meaning that the simpler Kripke clause for disjunction is adequate.

Definition 23.2.3 *(Resource Frame, Resource Model)* A resource frame is a structure $\langle \mathbb{M}, \sqsubseteq, \circ, e \rangle$ where $\sqsubseteq$ is a preorder; $\circ : \mathbb{M} \times \mathbb{M} \to \mathbb{M}$ is commutative, associative, and bifunctorial—that is, $m \sqsubseteq m' \;\&\; n \sqsubseteq n' \implies m \circ n \sqsubseteq m' \circ n'$—and e is a unit for $\circ$.

A resource frame together with a monotonic valuation $[\![-]\!] : \mathbb{A} \to \mathcal{P}(\mathbb{M})$ becomes a resource model under the satisfaction relation in Fig. 23.1.

Many different types of proof system—Hilbert-type, natural deduction, λ-calculus, tableaux, and display—are available for BI, but here we restrict attention to sequent calculi since they are more amenable to a concept of *computation as proof-search*. Here the presence of two quite different implications ($\to$ and $-\!*$) means that contexts in the sequent calculus require two different context-formers:

$$
\begin{array}{lll}
m \models A & \text{iff} & m \in [\![A]\!] \\
m \models \phi \wedge \psi & \text{iff} & m \models \phi \text{ and } m \models \psi \\
m \models \phi * \psi & \text{iff} & \exists n, n' \in \mathbb{M} : m \sqsubseteq n \circ n' \text{ and } n \models \phi \text{ and } n' \models \psi \\
m \models \phi \vee \psi & \text{iff} & m \models \phi \text{ or } m \models \psi \\
m \models \phi \to \psi & \text{iff} & \forall n \sqsubseteq m : n \models \phi \implies n \models \psi \\
m \models \phi \mathrel{-\!\!*} \psi & \text{iff} & \forall n \in \mathbb{M} : n \models \phi \implies n \circ m \models \psi \\
m \models \top & \text{iff} & m \in \mathbb{M} \\
m \models \top^* & \text{iff} & m \sqsubseteq e
\end{array}
$$

Fig. 23.1 Frame satisfaction

one that admits structural rules of Weakening and Contraction, and another that does not,

$$
\frac{\Gamma; \phi \implies \psi}{\Gamma \implies \phi \to \psi} \qquad\qquad \frac{\Gamma, \phi \implies \psi}{\Gamma \implies \phi \mathrel{-\!\!*} \psi}
$$

As a consequence, contexts in BI sequents are not a flat data structure, such as a sequence or a multiset, but instead are tree structures called *bunches*, a term that derives from the relevance logic literature (see, for example, Read (1988)).

Definition 23.2.4 (*Bunch*) Bunches are constructed with the following grammar:

$$
\Gamma ::= \phi \mid \varnothing_+ \mid \varnothing_\times \mid \Gamma; \Gamma \mid \Gamma, \Gamma
$$

where ϕ is a formula; $\varnothing_+$ and $\varnothing_\times$ are the additive and multiplicative units respectively, and, the symbols ; and , are the additive and multiplicative context-formers, respectively. The set of all bunches is denoted $\mathbb{B}$.

The two context-formers individually behave as the comma for IL and MILL, respectively, resulting in the following generalization of equivalence under permutation:

Definition 23.2.5 *(Coherent Equivalence)* Two bunches are coherent when $\Gamma \equiv \Gamma'$, where $\equiv$ is the least equivalence relation on bunches satisfying

– Commutative monoid equations for $\varnothing_\times$,
– Commutative monoid equations for $\varnothing_+$,
– Congruence: $\Delta \equiv \Delta'$ implies $\Gamma(\Delta) \equiv \Gamma(\Delta')$.

Let $\Gamma(\Delta)$ denote that Δ is a sub-bunch of Γ, then let $\Gamma(\Delta)[\Delta \mapsto \Delta']$—abbreviated to $\Gamma(\Delta')$, where no confusion arises—be the result of replacing the occurrence of Δ by Δ'.

Definition 23.2.6 (*Sequent*) A sequent is a pair of a bunch Γ, called the context, and a formula ϕ, and is denoted $\Gamma \implies \phi$.

Note that a bunch Γ can be read as formula $\lfloor \Gamma \rfloor$ by replacing all additive and multiplicative context-formers with additive and multiplicative conjunctions respectively. Satisfaction extends to bunches by $\models \Gamma$ iff $\models \lfloor \Gamma \rfloor$.

$$\frac{\Delta(\Delta') \Longrightarrow \chi}{\Delta(\Delta';\Delta'') \Longrightarrow \chi}\,W \qquad \frac{\Delta \Longrightarrow \phi}{\Delta' \Longrightarrow \phi}\,E_{(\Delta\equiv\Delta')} \qquad \frac{\Delta(\Delta';\Delta') \Longrightarrow \phi}{\Delta(\Delta') \Longrightarrow \phi}\,C$$

$$\frac{\Delta' \Longrightarrow \phi \qquad \Delta(\Delta'',\psi) \Longrightarrow \chi}{\Delta(\Delta',\Delta'',\phi \mathrel{-\!\!*} \psi) \Longrightarrow \chi}\mathrel{-\!\!*} L \qquad \frac{\Delta,\phi \Longrightarrow \psi}{\Delta \Longrightarrow \phi \mathrel{-\!\!*} \psi}\mathrel{-\!\!*} R \qquad \frac{\Delta(\varnothing_{\times}) \Longrightarrow \chi}{\Delta(\top^{*}) \Longrightarrow \chi}\,\top^{*}L$$

$$\frac{\Delta' \Longrightarrow \phi \qquad \Delta(\Delta'';\psi) \Longrightarrow \chi}{\Delta(\Delta';\Delta'';\phi \to \psi) \Longrightarrow \chi} \to L \qquad \frac{\Delta;\phi \Longrightarrow \psi}{\Delta \Longrightarrow \phi \to \psi} \to R \qquad \frac{\Delta(\varnothing_{+}) \Longrightarrow \chi}{\Delta(\top) \Longrightarrow \chi}\,\top L$$

$$\frac{\Delta(\phi,\psi) \Longrightarrow \chi}{\Delta(\phi * \psi) \Longrightarrow \chi}\,{*L} \qquad \frac{\Delta \Longrightarrow \phi \qquad \Delta' \Longrightarrow \psi}{\Delta,\Delta' \Longrightarrow \phi * \psi}\,{*R} \qquad \frac{\Delta(\phi) \Longrightarrow \chi \qquad \Delta(\psi) \Longrightarrow \chi}{\Delta(\phi \vee \psi) \Longrightarrow \chi}\,\vee L$$

$$\frac{\Delta(\phi;\psi) \Longrightarrow \chi}{\Delta(\phi \wedge \psi) \Longrightarrow \chi}\,\wedge L \qquad \frac{\Delta \Longrightarrow \phi \qquad \Delta' \Longrightarrow \psi}{\Delta;\Delta' \Longrightarrow \phi \wedge \psi}\,\wedge R \qquad \frac{\Delta \Longrightarrow \phi_i}{\Delta \Longrightarrow \phi_1 \vee \phi_2}\,\vee R_i$$

$$\frac{}{\Delta(\bot) \Longrightarrow \phi}\,\bot \qquad \frac{}{A \Longrightarrow A}\,\mathrm{Ax.} \qquad \frac{}{\varnothing_{\times} \Longrightarrow \top^{*}}\,\top^{*}R \qquad \frac{}{\varnothing_{+} \Longrightarrow \top}\,\top R$$

Fig. 23.2 Sequent calculus **LBI**

Figure 23.2 presents the sequent calculus **LBI** which is sound and complete for BI with respect to the model theory presented above Pym (2002). Note that the system has the subformula property, meaning that analytic proofs are complete, and it is therefore directly for the study of proof-search. Moreover, the system admits Cut (Brotherston, 2012; Gheorghiu & Marin, 2021) (see Boolos (1984) for the computational value of suitably formulated Cut-based approaches to proof-search),

$$\frac{\Delta \Longrightarrow \phi \qquad \Gamma(\phi) \Longrightarrow \psi}{\Gamma(\Delta) \Longrightarrow \psi}\,\mathrm{Cut}$$

The completeness of **LBI** has been established both relative to the Beth semantics discussed above and to a Grothendieck topological semantics (Pym et al., 2004). Relaxing composition of resources to be a partial function does yield a Kripke semantics, the completeness of which follows from a duality with the algebraic semantics (Docherty & Pym, 2018; Docherty, 2019). In fact, soundness and completeness theorems for BI have been considered in a number settings; see O'Hearn & Pym (1999); Pym et al. (2004); Pym (2002); Pym (2023); Galmiche et al. (2005); Brotherston (2012); Brotherston & Villard (2015); Larchey-Wendling (2016); Docherty (2019) for a range of discussions and explanations.

Finally, while, in LL, Girard's translation gives intuitionistic implication as derived from linear implication and the modality !, no such translation is available in BI. This can be seen quite readily in the setting of BI's categorical semantics, which is given in terms of *bicartesian doubly closed categories* (O'Hearn & Pym, 1999; Pym, 2002). That is, it can be shown that there cannot in general exist an endofunctor ! on such a category that satisfies Girard's translation.

23.2.2 Goal-Directed Proof-Search

Reduction operators in reductive logic are the set of operations available for computation, but this computation only becomes a procedure, specifically a *proof-search procedure*, in the presence of a *control* flow. It is this step which begins the definition of an operational semantics of a *logic programming language* (LP). Instantiating Kowalski's maxim (Kowalski, 1979):

Proof-search = Reductive Logic + Control

Here reductive logic provides the structure of the computation and control is the mechanism determining the individual steps. There are essentially two parts which must be decided: which rule to use, and which set of sufficient premisses to choose upon application. A mathematically appealing conceptional framework via the use of coalgebra for the former, and choice functions for the latter, is offered in Sect. 23.4, for now we offer a more traditional presentation.

The original logic programming language (HcLP Kowalski (1974)) is still widely used in the many Prolog variants that are available. It is based on the Horn clause fragment of first-order logic and uses Cut as the *only* rule, thereby solving the first control problem. The second problem is solved by showing that the use of fixed *selection functions* (Kowalski & Kuehner, 1971) result in a terminating procedure. This landmark result may perhaps be regarded as the second theorem of reductive logic, the first being Gentzen's completeness of analytic proofs (Gentzen, 1969).

A more general approach to proof-search is *goal-directed* proof-search which can be defined for a sequent $P \implies G$ as follows:

- If G is complex, apply right rules indefinitely until one has sub-goals $P_i \implies A_i$, where the A_i are atomic goals;
- If G is atomic, *resolve* it relative to the program by invoking a left rule.

Proofs constructed subject to this strategy are called *uniform* proofs (Miller et al, 1991; Pym & Harland, 1994), and may be generalized even further, resulting in the *focusing principle* (Andreoli, 1992), which is now a common phenomenon in proof theory. Though BI's sequent calculus admits the focusing principle (Gheorghiu & Marin, 2021), we choose to work with the simpler case as it is sufficient for illustrating how proof-search arises from the perspective of reductive logic.

Definition 23.2.7 (*Uniform Proof*) A proof is said to be uniform when, in its upward reading, it is composed of phases where all available right rules are applied before left rules (including structural rules).

Not all valid sequents in BI admit uniform proofs, but the existence can be guaranteed for the clausal *hereditary Harrop* fragment.

Definition 23.2.8 (*Hereditary Harrop BI*) Let D be a metavariable for a *definite clause*, G for a goal formula:

$$D ::= \top^* \mid \top \mid A \in \mathbb{A} \mid D \wedge D \mid D * D \mid G \to A \mid G \mathrel{-\!\!*} A$$
$$G ::= \top^* \mid \top \mid A \in \mathbb{A} \mid G \wedge G \mid G \vee G \mid G * G \mid D \to G \mid D \mathrel{-\!\!*} G$$

Unfortunately, in BI, uniform proof-search is not sufficient for goal-directedness. The problem can be remedied by a judicious goal-directed instance of Cut which creates two branches that each are also independently goal-directed. As a simulation of reasoning, this transformation represents the appeal to a lemma specifically to prove a desired goal form a collection hypotheses; a perspective which has a mechanical implementation in HcLP by a device called *tabling* (Miller & Nigam, 2007).

Definition 23.2.9 (*Resolution Proof*) A uniform proof is a resolution proof if the right sub-proof of an implication rule consists of a series of weakening followed by a Cut on the atom defined by the implication.

Example 23.2.10 The following proof is uniform, but not goal-directed as the goal A is not principle in the first $\mathrel{-\!\!*} L$ inference:

$$\begin{prooftree}
\AxiomC{$B \implies B$}
\AxiomC{$A \implies A$}
\RightLabel{E}
\UnaryInfC{$\varnothing_\times, A \implies A$}
\RightLabel{$\top^* L$}
\UnaryInfC{$\top^*, A \implies A$}
\RightLabel{$\mathrel{-\!\!*} L$}
\BinaryInfC{$B, B \mathrel{-\!\!*} \top^*, A \implies A$}
\end{prooftree}$$

The problem can be remedied by the introduction of a cut,

$$\begin{prooftree}
\AxiomC{$B \implies B$}
\AxiomC{$\varnothing_\times \implies \top^*$}
\RightLabel{$\top^* L$}
\UnaryInfC{$\top^* \implies \top^*$}
\RightLabel{$\mathrel{-\!\!*} L$}
\BinaryInfC{$B, B \mathrel{-\!\!*} \top^* \implies \top^*$}
\AxiomC{$A \implies A$}
\RightLabel{E}
\UnaryInfC{$\varnothing_\times, A \implies A$}
\RightLabel{$\top^* L$}
\UnaryInfC{$\top^*, A \implies A$}
\RightLabel{Cut}
\BinaryInfC{$B, B \mathrel{-\!\!*} \top^*, A \implies A$}
\end{prooftree}$$

More complicated cases include the possibility that A is the atom defined by the implication, in which case one can also make a judicious use of Cut to keep the proof goal-directed.

The control régime of resolution proofs can be enforced by augmenting $\to L$, $\mathrel{-\!\!*} L$ and Ax to rules to encode the necessary uses of Cut such that resolution proofs are identified with uniform proofs in the system which has the replacement rules. For multiplicative implication, the use of Cut then allows porting the additional multiplicative content to the left sub-proof, resulting in a single thread of control. To express this control régime as rules, we require presenting bunches in *canonical form*—see Armelín (2001; 2002). A bunch Γ is in canonical form iff the left-hand branch of Γ is either a proposition, a unit or a canonical bunch of the opposite (additive or multiplicative) type, and the right-hand branch of Γ is a canonical bunch. It is easy to see that every bunch is equivalent to a bunch in canonical form. The purpose of canonical forms is to let us represent bunches in a convenient way: instead of writing a bunch like $\Gamma_1, (\Gamma_2, (\ldots, (\Gamma_{n-1}, \Gamma_n)))$ we can write it as $\Gamma_1, \Gamma_2, \ldots., \Gamma_{n-1}, \Gamma_n$ with no loss of information.

Lemma 23.2.11 (Based on Armelín, 2001; 2002) *The following rules, where* $\alpha \in \{\top, \top^*\} \cup \mathbb{A}$ *and the bunches are all written in canonical form, are admissible, and replacing* $\rightarrow L$, $-\!\!* \, L$ *and* Ax *with them in* ***LBI*** *does not affect the completeness of the system, with respect to the hereditary Harrop fragment:*

$$\frac{P; G \to \alpha \implies G \qquad Q \implies \top^*}{Q, (P; G \to \alpha) \implies \alpha}\,Res_1 \qquad \frac{P \implies G}{P, G -\!\!* \, \alpha \implies \alpha}\,Res_2 \qquad \frac{\Gamma \implies \top^*}{\Gamma, \alpha \implies \alpha}\,Res_3$$

Moreover, uniform proofs in the resulting system are complete for the hereditary Harrop fragment of BI.

23.2.3 Operational Semantics

To distinguish the metatheory of BI from the logic programming language, we use $P \vdash G$ to denote that a goal G is being queried relative to a program P. To enforce the structure of uniform proof-search, we restrict the system modified according to Lemma 23.2.11 further resulting in the *resolution* system.

Definition 23.2.12 (*Program, Goal, Configuration, State*) A program P is any bunch in which formulas are definite clauses; and, a goal is a (goal) formula G. A configuration is either empty (denoted $\Box$), or is pair of a program P and goal G denoted $P \vdash G$; and a state is a list of configurations. The set of all programs is denoted by $\mathbb{P}$.

Milner's *tactics and tacticals* (Milner, 1984) provide a conceptual theory to support reductive logic proof-search and its mechanization. A tactic (τ) is a mapping taking a configuration to a list of next configurations, together with a deduction operator verifying the correctness of the reduction. A tactical is a composition of tactics.

In the following labelled transition system, the C_i are configurations, the S_i are states, $\sqcup$ is concatenation of lists, and δ is a deduction operator (i.e., a rule):

$$\frac{C_0 \longrightarrow S_0 \qquad \dots \qquad C_n \longrightarrow S_n}{[C_0, \dots, C_n] \longrightarrow S_0 \sqcup \dots \sqcup S_n} \quad \frac{\delta(P_0 \vdash G_0, \dots, P_n \vdash G_n) = P \vdash G}{P \vdash G \longrightarrow [P_0 \vdash G_0, \dots, P_n \vdash G_n]}\,\tau$$

In practice, we are given $P \vdash G$ and compute the list of sufficient premisses, and not the other way around, thus the condition of τ ought to be a reduction operator,

$$\frac{[P_0 \vdash G_0, \dots, P_n \vdash G_n] \in \rho(P \vdash G)}{P \vdash G \longrightarrow [P_0 \vdash G_0, \dots, P_n \vdash G_n]}\,\tau$$

Of course, deduction and reduction operators are dual meaning that it is only the use of τ that is unchanged. Note the non-determinism, which introduces the use of a *choice* (or, with some particular agenda, *selection*) function.

We restrict the application of the inference rules syntactically so that they may only be applied correctly with respect to resolution proofs. For example, a left rule can only be applied when the goal is an atom, in which case the the positive left rules ($\top L$, $\top^* L$, $\wedge L$, $* L$) can be applied eagerly, handled by clausal decomposition.

Definition 23.2.13 (*Clausal Decomposition*) Clausal decomposition of programs is as follows:

$$\begin{array}{lll}
[\top^*] := [\varnothing_\times] := \varnothing_\times & [\top] := [\varnothing_+] := \varnothing_+ & [A] := A \\
[P; Q] := [P]; [Q] & [D_1 \wedge D_2] := [D_1]; [D_2] & [G \to A] := G \to A \\
[P, Q] := [P], [Q] & [D_1 * D_2] := [D_1], [D_2] & [G \mathrel{-\!\!*} A] := G \mathrel{-\!\!*} A
\end{array}$$

It remains to apply either an implication left rule or an axiom, which is made goal directed by Lemma 23.2.11. Their completeness requires the use of weakening, so we introduce a weak coherence ordering ($\leqq$) as follows:

$$P \geqq Q \iff \cfrac{\cfrac{P \implies \bot}{\{ \text{ Weakening and Exchange } \}}}{Q \implies \bot}$$

Though the use of this ordering seems to reintroduce a lot of non-determinism, the choice it offers is still captured by the use of a selection function. That is, once a clause has been *selected* in the program, the weakenings may be performed in a *goal-directed* way to bring the clause to the top of the bunch. For example, for the $\mathrel{-\!\!*}$ resolution, one may alternate phases of removing additive data in the neighbourhood of the clause and permuting data to makes the desired context-former principal.

Example 23.2.14 The following may illustrate the process:

$$\cfrac{\cfrac{\cfrac{(Q_0, (Q_1; Q_2)), G \mathrel{-\!\!*} \alpha \implies \bot}{Q_0, ((Q_1; Q_2), G \mathrel{-\!\!*} \alpha) \implies \bot}\,\mathrm{E}}{Q_0, ((Q_1; Q_2), (Q_3; (G \mathrel{-\!\!*} \alpha)) \implies \bot}\,\mathrm{W}}{Q_0, ((Q_1; Q_2), (Q_3; (Q_4; G \mathrel{-\!\!*} \alpha)) \implies \bot}\,\mathrm{W}$$

Anything additively combined with the clause is removed (using weakening), and when something is multiplicatively combined, the bunch is re-ordered so that the context-former of the clause is principal. In the case in which Q_0 has been additively combined, it would also have to be removed.

Definition 23.2.15 (*Resolution System*) The resolution system is composed of the rules in Fig. 23.3. The first two rules are called the initial rules, the following three the resolution rules (Cut-resolution, $\mathrel{-\!\!*}$-resolution, and $\to$-resolution respectively), and the final five are the decomposition rules.

Theorem 23.2.16 (Soundness and Completeness) *Let P be a program and G a goal.*

- *Soundness. Any execution of $P \vdash G$ is a resolution proof of $P \implies G$.*
- *Completeness. Any resolution proof of $P \implies G$ is an execution of $P \vdash G$.*

Proof This follows from Lemma 23.2.11.

$P \vdash \top$	$\Leftarrow$	Always
$P \vdash \top^*$	$\Leftarrow$	$[P] \equiv \varnothing_\times; R$
$P \vdash A$	$\Leftarrow$	$[P] \leqq S, A$ and $S \vdash \top^*$
$P \vdash A$	$\Leftarrow$	$[P] \leqq Q, G \multimap\!\!* A$ and $Q \vdash G$
$P \vdash A$	$\Leftarrow$	$[P] \leqq S, (Q; G \to A)$ and $S \vdash \top^*$ and $Q;G \to A \vdash G$
$P \vdash G_1 \vee G_2$	$\Leftarrow$	$P \vdash G_1$ or $P \vdash G_2$
$P \vdash G_1 \wedge G_2$	$\Leftarrow$	$P \vdash G_1$ and $P \vdash G_2$
$P \vdash G_1 * G_2$	$\Leftarrow$	$P \equiv R; (Q, R)$ and $Q \vdash G_1$ and $R \vdash G_2$
$P \vdash D \to G$	$\Leftarrow$	$P; [D] \vdash G$
$P \vdash D \multimap\!\!* G$	$\Leftarrow$	$P, [D] \vdash G$

Fig. 23.3 Resolution system

The traditional approach for applying a selection function follows from the application of a backtracking schedule. For example, taking contexts to be lists one may simply choose to always attempt the leftmost clause first when using a resolution rule; and, having made a choice, one then progresses until success or failure of the search, returning to an earlier stage in the computation in case of the latter. This forms one possibility which has been called *depth-first search with leftmost selection* (Lloyd, 1984; Kowalski, 1986). Another example of a schedule is *breadth-first search with leftmost selection*, where after one step of reduction one immediately backtracks so that every possibility is tried as soon as possible. These two choices are the extremes of a range of possibilities which have different advantages and disadvantages including complexity (Plaisted and Zhu, 1997). The later is always safe, that is will always terminate when there is a proof, but the same is not true for the former as seen in Sect. 23.4.3. Whatever the choice, assuming a safe one is chosen, the result is the programming language hHBI.

The use of a selection function is sufficient to make proof-search effective, but is limited in that such functions are fundamentally nothing more than a prescribed pattern of *guessing*. To improve the situation one may study *control* as a phenomenon in itself; for example, one may use proof theory to restrict the choices as was done with the restriction to uniform proofs. In Sect. 23.4, we offer a framework in which multiple choices can be handled simultaneously, resulting in a partial determinization of the proof-search procedure. This method works in cases in which other control mechanisms do not.

23.3 Denotational Semantics

Traditionally, formal systems are given semantics of two different flavours: operational semantics, which specifies the local behaviour, and denotational semantics, which maps expressions into abstract mathematical entities. A basic requirement of the denotational semantics is that it is *faithful*—meaning that the mathematical structures respect the behaviour prescribed by the operational semantics—and that it is *adequate*—meaning that the dynamics of the denotations determine the operational semantics. The most exacting requirement is that equality in the denotational semantics characterizes equivalence in syntax, in which case the denotational semantics is said to be a *full abstraction*.

A standard construction of a denotational semantics for a programming language is the greatest fixed point of behaviour (see Turi & Plotkin (1997) for more details). Often, including for hHBI, all the constructs of the formal system are finite, so the semantic domain admits an inductive algebraic formulation: for hHBI, *the least Herbrand model*.

Below, we implement a strategy outlined by Apt (1982), employing the method presented by Miller (1989) for this construction. The correctness of the semantics is offered in Theorem 23.3.4, where we use the nomenclature of mathematical logic for the notions of faithfulness and adequacy: soundness and completeness.

23.3.1 Denotations and Interpretations

We develop a denotational semantics of hHBI within the setting of resource reading of BI by instantiating the resource model given in Definition 23.2.3. What is required is an *interpretation* of programs as some collection of resources. A simple choice is to interpret a given program as the set of atomic formulas which it satisfies, this being a subset of the Herbrand base ($\mathbb{H}$). To be precise, the models are given by a special interpretation $J : \mathbb{P} \to \mathcal{P}(\mathbb{H})$ such that a particular configuration executes if and only if the programs satisfies the goal,

$$P \vdash G \iff J(P) \models G$$

In the propositional case studied above, the Herbrand base $\mathbb{H}$ is the set of atomic propositions $\mathbb{A}$, though in Sect. 23.5 the language is generalized to a first-order setting and it's semantics can be analogous handled by choosing the set of ground formulas instead.

To give a resource frame, one must be able to combine and compare resource elements (here, programs). The first comes for free using the multiplicative context-former (or equivalently, multiplicative conjunction), but the second is more subtle. From a programming perspective, the ordering follows from the fact that a program may evolve during execution, and since execution is determined by provability, one may simply choose the *Cut* ordering:

$$P \preccurlyeq Q \iff P \vdash \lfloor Q \rfloor$$

The ordering makes sense because it is *internally monotone*; that is, if a program gets larger then the set of associated atoms does not decrease. The resulting frame is adequate for a resource model when interpretation satisfies a basic precondition,

$$A \in I(P) \implies P \vdash A$$

$I, P \Vdash \top$	iff	Always
$I, P \Vdash \top^*$	iff	$P \equiv \varnothing_\times ; R$
$I, P \Vdash A$	iff	$A \in I(P)$
$I, P \Vdash G_1 \vee G_2$	iff	$I, P \Vdash G_1$ or $I, P \Vdash G_2$
$I, P \Vdash G_1 \wedge G_2$	iff	$I, P \Vdash G_1$ and $I, P \Vdash G_2$
$I, P \Vdash G_1 * G_2$	iff	$\exists Q, R \in \mathbb{P} : P \preccurlyeq Q \circ R$ and $I, Q \Vdash G_1$ and $I, R \Vdash G_2$
$I, P \Vdash D \to G$	iff	$I, ([D];P) \Vdash G$
$I, P \Vdash D \mathrel{-\!\!*} G$	iff	$I, ([D];P) \Vdash G$

Fig. 23.4 Program satisfaction

A primitive example of such an interpretation is $I_\perp : P \mapsto \emptyset$.

Definition 23.3.1 (*Monoid of Programs, Interpretation*) The monoid of programs is the structure $\langle \mathbb{P}, \preccurlyeq, \circ, \varnothing_\times \rangle$, where $\circ$ is multiplicative composition. An interpretation of the monoid of programs is an order reversing mapping $I : \mathbb{P} \to \mathcal{P}(\mathbb{A})$.

Unfortunately, the use of the Cut ordering reintroduces non-analyticity into the semantics. Some of this may be handled using the adjunction between the implications and their respective conjunctions, resulting in the simpler *program* satisfaction relation of Fig. 23.4 in which bunches are written using the notation for canonical forms. One can verify by induction on goals that it is still internally monotone; that is, for any interpretation I, if $P \preccurlyeq Q$ then $I, Q \Vdash G$ implies $I, P \Vdash G$.

However, since the accessibility condition of the implication clauses in program satisfaction is a restriction of the corresponding clauses in the frame satisfaction relation—that is, program satisfaction replaces the accessibility condition by a particular program extension—it actually forms a new semantics altogether. The advantage is that this new semantics is more amenable to computation.

Lemma 23.3.2 (Adequacy) *If $A \in I(P)$ implies $P \vdash A$, then $I, P \Vdash G$ implies $P \vdash G$*

Proof By induction on the structure of the goal formula.

Base case. There are the sub-cases to consider for $G \in \{\top, \top^*\} \cup \mathbb{A}$, but in either case it follows by equivalence of the respective clauses between program satisfaction and the resolution system.

Inductive case.

- $G = G_1 \vee G_2$. By definition, $I, P \Vdash G_1$ or $I, P \Vdash G_2$, so by the induction hypothesis, $P \vdash G_1$ or $P \vdash G_2$, whence $P \vdash G_1 \vee G_2$.
- $G = G_1 \wedge G_2$. By definition, $I, P \Vdash G_1$ and $I, P \Vdash G_2$, so by the induction hypothesis, $P \vdash G_1$ and $P \vdash G_2$, whence $P \vdash G_1 \wedge G_2$.
- $G = G_1 * G_2$. By definition, $P \preccurlyeq Q \circ R$ with $I, Q \Vdash G_1$ and $I, R \Vdash G_1$. It follows from the induction hypothesis that $Q \vdash G_1$ and $R \vdash G_2$, so that $Q, R \vdash G_1 * G_2$, whence $P \vdash G_1 * G_2$.
- $G = D \to G'$. By definition, $P; [D] \Vdash G'$, so by the induction hypothesis, $P; [D] \vdash G'$, so $P \vdash D \to G'$.
- $G = D \mathrel{-\!\!*} G'$. By definition, $P, [D] \Vdash G'$, so by the induction hypothesis, $P, [D] \vdash G'$, so $P \vdash D \to G'$.

□

The restriction to program satisfaction now increases the set of possible frames, since, for example, one no longer requires bifunctoriality. Therefore, another candidate for the ordering on frames is simply to make the clauses of program satisfaction match the clauses for resolution proofs, so one has the following extension ordering:

$$P \leq Q \iff \exists R \in \mathbb{P} : \ P \equiv Q; R$$

However, such a choice does *not* yield a resource frame (precisely because the relation is not bifunctorial), but does yield an adequate semantics with respect to program satisfaction. The extension ordering is strictly *coarser* than the provability ordering; that is, $P \leq Q \implies P \preccurlyeq Q$, but not the other way around as witnessed by the following:

$$\cfrac{\cfrac{\varnothing_\times \vdash \top^*}{\varnothing_\times; \top^* \to A; \varnothing_+ \vdash A}}{\varnothing_\times; \top^* \to A \vdash \top \to A}$$

One has $\varnothing_\times; \top^* \to A \preccurlyeq \top \to A$, but the former is not an extension of the latter. In fact, the two ordering both give full abstractions relative to different notions of equality which predictably are equality of programs and equality of declarations:

$$\begin{array}{lcl} P = Q \quad \text{mod} \ \leq & \iff & P \equiv Q \\ P = Q \quad \text{mod} \ \preccurlyeq & \iff & \varnothing_\times \vdash \lfloor P \rfloor \mathrel{*\!\!-\!\!*} \lfloor Q \rfloor \end{array}$$

23.3.2 Least Fixed Point

Reading the resources semantics as a possible-worlds interpretation of the logic, we use the symbols $w, u, v...$ for programs. The adequacy condition requires that a world is inhabited by nothing other than the propositional formulas that it satisfies, given by an interpretation $I(w)$.

Observe that since $\mathcal{P}(\mathbb{A})$ forms a complete lattice, so do interpretations (with $I_\perp$ as the bottom),

$$\begin{array}{lcl} I_1 \sqsubseteq I_2 & \iff & \forall w (I_1(w) \subseteq I_2(w)) \\ (I_1 \sqcup I_2)(w) & := & I_1(w_1) \cup I_2(w_2) \\ (I_1 \sqcap I_2)(w) & := & I_1(w_1) \cap I_2(w_2) \end{array}$$

The derived ordering essentially requiring that larger interpretations see more of the worlds than the smaller interpretations.

Combining the adequacy condition in Lemma 23.3.2 with the well-behaved structure of the resolution system suggests that one can inductively improve the vacuous interpretation $I_\perp$ by unfolding the resolution system. To this end, consider the T-operator on interpretations.

Definition 23.3.3 (*T -Operator*) The T operator is defined as follows:

$$T(I)(w) := \{A \mid [w] \leqq u, A \text{ and } I, u \Vdash \top^*\} \cup \tag{23.1}$$

$$\{A \mid [w] \leqq u, G \multimap A \text{ and } I, u \Vdash G\} \cup \tag{23.2}$$

$$\{A \mid [w] \leqq u, (v; G \to A) \text{ and } I, u \Vdash \top^* \text{ and } I, (v; G \to A) \Vdash G\} \tag{23.3}$$

Observe that the three parts of the definition correspond exactly to the resolution clauses of execution, so T precisely incorporates *one* resolution step. Applying it indefinitely, therefore, corresponds to performing an arbitrary number of resolutions. This is handled mathematically by the use of a least fixed-point.

It follows from the Knaster-Tarski Theorem (Knaster & Tarksi (1928); Tarski (1955); Apt & Emden (1982)) that if T is monotone and continuous, then the following limit operator is well-defined:

$$T^\omega(I_\perp) := I_\perp \sqcup T(I_\perp) \sqcup T^2(I_\perp) \sqcup T^3(I_\perp) \sqcup \dots$$

These conditions are shown to hold in Lemma 23.3.5(3) and Lemma 23.3.5(4), respectively. The interpretation may thenceforth be suppressed, so that $P \Vdash G$ denotes $T^\omega(I_\perp), P \Vdash G$.

The completeness of the semantics follows immediately from the definition of T, since it simply observes the equivalence between resolution and application of the T-operator; that is, that T extends correctly. Soundness, on the other hand, requires showing that every path during execution is eventually considered during the unfolding.

Theorem 23.3.4 (Soundness and Completeness) $P \vdash G \iff P \Vdash G$

Proof of Completeness ($\Longleftarrow$). From Lemma 23.3.2, it suffices to show that the adequacy condition holds:

$$A \in T^\omega I_\perp(P) \implies P \vdash A$$

It follows from Lemma 23.3.5(2) that the antecedent holds only if there exists $k \in \mathbb{N}$ such that $T^k I_\perp, P \Vdash A$. We proceed by induction on k, the base case $k = 0$ being vacuous as $I_\perp(P) = \emptyset$. For the inductive step, observe that $A \in T^k I_\perp(P)$ only if at least one of (1), (2), or (3) holds in Definition 23.3.3 for $I = T^{k-1} I_\perp$. In either case, the result follows from the induction hypothesis and the correspondence between (1), (2), and (3), with the resolution clauses of Fig. 23.3.

The completeness proof is a repackaging of Miller's familiar ordinal induction proof (Miller, 1989). The difference is simply to pull out the sub-induction on the structure of the formula as an instance of Lemma 23.3.2.

Proof of Soundness ($\implies$). By induction on the height N of executions.

Base case. $N = 1$. It must be that the proof of $P \vdash G$ follows from the application of an initial rule. Hence, either $G = \top$, or $G = \top^*$ and $P \equiv \varnothing_\times; R$. In either case, $P \Vdash G$ follows immediately.

Inductive case. We consider each of the cases for the last inference.

- Cut-resolution. By definition, $[P] \leqq Q, A$, where $Q \vdash \top^*$ with height $N' < N$. So, by the induction hypothesis, $Q \Vdash \top^*$, so that $A \in T(T^\omega(I_\bot))(P) = T^\omega(I_\bot)(P)$, whence $P \Vdash A$.
- $-\!\!*$-resolution. By definition, $[P] \leqq Q, G' -\!\!* A$, where $Q \vdash G'$ with height $N' < N$. So, by the induction hypothesis, $Q \Vdash G'$, so that $A \in T(T^\omega(I_\bot))(P) = T^\omega(I_\bot)(P)$, whence $P \Vdash A$.
- $\to$-resolution. By definition, $[P] \equiv Q, (R; G' \to A)$, where $Q \vdash \top^*$ and $R \vdash G'$, with heights $N', N'' < N$. So, by the induction hypothesis, $Q \Vdash \top^*$ and $R \Vdash G'$, hence $A \in T(T^\omega I_\bot)(P) = T^\omega I_\bot(P)$, whence $P \Vdash A$.
- $G = G_1 \vee G_2$. By definition, $P \vdash G_i$ for some $i \in \{1, 2\}$ with height $N' < N$. So, by the induction hypothesis, $P \Vdash G_i$ for some $i \in \{1, 2\}$, so $P \Vdash G_1 \vee G_2$.
- $G = G_1 \wedge G_2$. By definition, $P \vdash G_i$ for all $i \in \{1, 2\}$ with heights $N', N'' < N$. So, by the induction hypothesis, $P \Vdash G_i$ for all $i \in \{1, 2\}$, so $P \Vdash G_1 \wedge G_2$.
- $G = G_1 * G_2$. By definition, $P \equiv Q \circ R$ where $Q \vdash G_1$ and $R \vdash G_2$ with heights $N', N'' < N$. So, by the induction hypothesis, $Q \Vdash G_1$ and $R \Vdash G_2$, so from reflexivity of provability $P \Vdash G_1 * G_2$.
- $G = D \to G'$. By definition, $P; [D] \vdash G'$ with height $N' < N$. So, by the induction hypothesis, $P; [D] \Vdash G'$, so $P \Vdash D \to G'$.
- $G = D -\!\!* G'$. By definition, $P, [D] \vdash G'$ with height $N' < N$. So, by the induction hypothesis, $P, [D] \Vdash G'$, so $P \Vdash D -\!\!* G'$. □

The remainder of this subsection shows that $T^\omega(I_\bot)$ is well defined.

Lemma 23.3.5 *Let $I_0 \sqsubseteq I_1 \sqsubseteq \ldots$ be a collection of interpretations, let $w \in \mathbb{P}$, and let G be a goal, then,*

1. $I_1, w \Vdash G \implies I_2, w \Vdash G$. *(Persistence of Satisfaction)*
2. $\sqcup_{i=1}^{\infty} I_i, w \Vdash G \implies \exists k \in \mathbb{N} : I_k, w \Vdash G$. *(Compactness of Satisfaction)*
3. $T(I_0) \sqsubseteq T(I_1)$. *(Monotonicity of T)*
4. $T(\sqcup_{i=0}^{\infty} I_i) = \sqcup_{i=0}^{\infty} T(I_i)$. *(Continuity of T)*

Proof of Lemma 23.3.5*(*1*)* By induction on the structure of the goal formula. Throughout, assume $I_1, w \Vdash G$, otherwise the statement holds vacuously.

Base case. The cases $G \in \{\top, \top^*\}$ are immediate since satisfaction is independent of interpretation, meanwhile the case in which $G \in \mathbb{A}$ follows from the definition of the ordering on interpretations.

Inductive case. Assume, inductively, that the claim holds for any goal smaller than G independent of program.

- $D \to G$. By definition, $I_1, w; [D] \Vdash G$, so, by the induction hypothesis, it follows that $I_2, w; [D] \Vdash G$, thus $I_2, w \Vdash D \to G$.

- $D \multimap G$. By definition, $I_1, (w, [D]) \Vdash G$, so, by the induction hypothesis, it follows that $I_2, (w, [D]) \Vdash G$, thus $I_2, w \Vdash D \multimap G$.
- $G_1 * G_2$. By definition, $w \preccurlyeq u \circ v = u, v$ such that $I_1, u \Vdash G_1$ and $I_1, v \Vdash G_2$ and so, by the induction hypothesis, it follows that $I_2, u \Vdash G_1$ and $I_2, v \Vdash G_2$, so that $I_2, w \Vdash G_1 * G_2$.
- $G_1 \wedge G_2$. By definition, $I_1, w \Vdash G_1$ and $I_1, w \Vdash G_2$, so, by the induction hypothesis, $I_2, w \Vdash G_1$ and $I_2, w \Vdash G_2$, and so $I_2, w \Vdash G_1 \wedge G_2$.
- $G_1 \vee G_2$. By definition, $I_1, w \Vdash G_1 \vee G_2$, then $I_1, w \Vdash G_1$ or $I_1, w \Vdash G_2$, so, by the induction hypothesis, $I_2, w \Vdash G_1$ or $I_2, w \Vdash G_2$, and so $I_2, w \Vdash G_1 \vee G_2$. $\square$

***Proof of Lemma* 23.3.5*(*2*)*.** By induction on the structure of the goal formula. Throughout, assume $\sqcup_{i=1}^{\infty} I_i, w \Vdash G$ otherwise the statement holds vacuously.

Base case. The cases $G \in \{\top, \top^*\}$ are immediate since satisfaction is independent of interpretation. For the case in which $G \in \mathbb{A}$, note that $\left(\sqcup_{i=0}^{\infty}\right)(w) = \bigcup_{i=0}^{\infty} I_i(w)$, so by the definition of satisfaction $G \in I_k(w)$, for some k, and so $I_k, w \Vdash G$.

Inductive case. Assume, inductively, that the claim holds for any goal smaller than G independent of program.

- $G_1 \vee G_2$. By definition, $\sqcup_{i=1}^{\infty} I_i, w \Vdash G_1$ or $\sqcup_{i=1}^{\infty} I_i, w \Vdash G_2$, so, by the induction hypothesis, $\exists k \in \mathbb{N}$ such that $I_k, w \Vdash G_1$ or $I_k, w \Vdash G_2$, whence $I_k, w \Vdash G_1 \vee G_2$.
- $G_1 \wedge G_2$. By definition, $\sqcup_{i=1}^{\infty} I_i, w \Vdash G_1$ and $\sqcup_{i=1}^{\infty} I_i, w \Vdash G_2$, so, by the induction hypothesis, $\exists m, n \in \mathbb{N}$ such that $I_m, w \Vdash G_1$ and $I_n, w \Vdash G_2$. Let $k = \max(m, n)$, from Lemma 23.3.5(1), it follows that $I_k, w \Vdash G_1$ and $I_k \Vdash G_2$. Therefore, $I_k, w \Vdash G_1 \wedge G_2$.
- $G_1 * G_2$. By definition, $w \preccurlyeq u, v$ such that $\sqcup_{i=1}^{\infty} I_i, u \Vdash G_1$ and $\sqcup_{i=1}^{\infty} I_i, v \Vdash G_2$. It follows from the induction hypothesis, that $\exists m, n \in \mathbb{N}$ such that $I_m, u \Vdash G_1$ and $I_n, u \Vdash G_1$. Let $k = \max(m, n)$, it follows from Lemma 23.3.5(1) that $I_k, u \vdash G_1$ and $I_k, v \Vdash G_2$. Therefore, $I_k, (u, v) \Vdash G_1 * G_2$, thus $I_k, w \Vdash G_1 * G_2$.
- $D \to G'$. By definition, $\sqcup_{i=1}^{\infty} I_i, (w; [D]) \Vdash G_1$, so, by the induction hypothesis, $\exists k \in \mathbb{N}$ such that $I_k, (w; [D]) \Vdash G'$. Thus, by definition, $I_k, w \Vdash D \to G'$.
- $D \multimap G'$. By definition, $\sqcup_{i=1}^{\infty} I_i, (w, [D]) \Vdash G'$, so, by the induction hypothesis, $\exists k \in \mathbb{M}$ with $I_k, (w, [D]) \Vdash G'$. Thus, by definition, $I_k, w \Vdash D \multimap G'$. $\square$

Proof of Lemma* 23.3.5*(*3*) Let $w \in \mathbb{P}$ be arbitrary, and suppose $A \in TI_0(w)$, then we must show $A \in TI_1(w)$. There are three cases, corresponding to the definition of T.

(1) Suppose $[w] \leqq u, A$ such that $I_0, u \Vdash \top^*$. Then, from Lemma 23.3.5(1), it follows that $I_1, u \Vdash I$, so that $A \in TI_1(w)$.
(2) If $[w] \leqq u, G \multimap A$ such that $I_0, u \Vdash G$, then, from Lemma 23.3.5(1), it follows that $I_1, u \Vdash G$, so that $A \in TI_1(w)$.
(3) If $[w] \leqq u, (v; G \to A)$ such that $I_0, u \Vdash \top^*$ and $I_0, v \vdash G$, then, from Lemma 23.3.5(1), it follows that $I_1, u \Vdash \top^*$ and $I_1, v \vdash G$, so that $A \in TI_1(w)$. $\square$

Proof of Lemma* 23.3.5*(*4*) We consider each direction of the inclusion independently.

First, $\sqcup_{i=0}^{\infty} T(I_i) \sqsubseteq T(\sqcup_{i=0}^{\infty} I_i)$. Let $j \geq 0$, then $I_j \sqsubseteq \sqcup_{i=1}^{\infty} I_i$, so from Lemma 23.3.5(3) it follows that $T(I_j) \sqsubseteq T(\sqcup_{i=1}^{\infty} I_i)$. Since j was arbitrary, $\sqcup_{i=1}^{\infty} T I_i \sqsubseteq T(\sqcup_{i=1}^{\infty} I_i)$

Second, $T(\sqcup_{i=0}^{\infty} I_i) \sqsubseteq \sqcup_{i=0}^{\infty} T(I_i)$. Let $w \in \mathbb{P}$ be arbitrary, and suppose $A \in T(\sqcup_{i=1}^{\infty} I_i)(w)$, then it suffices to $\exists k \in \mathbb{N}$ such that $A \in T(I_k)$. Here are three cases, corresponding to the definition of T.

(1) If $[w] \leqq u$, A and $\sqcup_{i=1}^{\infty} I_i, u \Vdash \top^*$, then it follows from Lemma 23.3.5(2) that $\exists k \in \mathbb{N}$ such that $I_k, u \Vdash \top^*$. Therefore, by definition, $A \in T I_k(w)$.
(2) If $[w] \leqq u$, $G \mathrel{-\!\!*} A$ and $\sqcup_{i=1}^{\infty} I_i, u \Vdash G$, then it follows from Lemma 23.3.5(2) that $\exists k \in \mathbb{N}$ such that $I_k, u \Vdash \top^*$. Therefore, by definition, $A \in T I_k(w)$.
(3) If $[w] \leqq u$, $(v; G \to A)$ and $\sqcup_{i=1}^{\infty} I_i, u \Vdash \top^*$ and $\sqcup_{i=1}^{\infty} I_i, v \Vdash G$, then it follows from Lemma 23.3.5(2) that $\exists m, n \in \mathbb{N}$ such that $I_m, u \Vdash \top^*$ and $I_m, v \Vdash G$. Let $k = \max(m, n)$, then from 23.3.5(1), it follows that $I_k, u \Vdash \top^*$ and $I_k, v \Vdash G$, so $A \in T I_k(w)$. $\square$

23.4 Coalgebraic Semantics

The fixed point semantics for hHBI is appealing in its conceptional simplicity and relationship to the resource interpretation of BI, but leaves something to be desired as a semantics of proof-search as it lacks explicit witnessing of the control structures. Therefore we look for a denotational semantics of the behaviour of proof-search itself; that is, an abstraction of the operational semantics. In the past, models for classical, intuitionistic, and linear logics have included dialogue games (Pym & Ritter, 2004; Miller & Saurin, 2006) in the spirit of Abramsky's full abstraction for PCF (Abramsky et al., 2000). However, in the case of logic programming, the uniform treatment of Turi & Plotkin (1997) via coalgebraic methods, an essential feature of modern approaches to operational semantics in general, has been successfully implemented for HcLP (Komendantskaya et al., 2011; Komendantskaya & Power, 2011; Komendantskaya et al., 2016; Bonchi & Zanasi, 2015), and is the method demonstrated here. Though we give a coalgebraic model of proof-search specifically for the resolution system of the hereditary Harrop fragment of BI, which defines the behaviour of hHBI, we make every attempt to keep the approach general. The choice of coalgebra to model proof-search is both natural and sensible when one considers it an abstract mathematical framework for the study states and transitions.

The deductive view of logic is inherently algebraic; that is, rules in a proof system can be understood as functions, called deduction operators, which canonically determine algebras for a functor. The space of valid sequents is defined by the recursive application of these operators, which instantiates the inductive definition of proofs. In contrast, the reductive paradigm of logic is coalgebraic; that is, the reduction operators can be mathematically understood as coalgebras. The much larger space of sequents (or configurations) explored during proof-search is corecrusively generated, and reductions are coinductively defined. This framework immediately allows the modelling of an interpretation of hHBI with parallel execution, analogous to the treatment of HcLP discussed above (Komendantskaya et al., 2011; Komendantskaya & Power, 2011; Komendantskaya et al., 2016; Bonchi & Zanasi, 2015).

The parallel semantics and the familiar sequential one using a backtracking-schedule share a common problem: the size of the set of premisses upon application of a reduction operator. We propose a general approach to ameliorate the situation using algebraic constraints. The mechanism can be understood in the coalgebraic setting as a lifting of a single step of proof-search and works on the metalevel of reductive logic, so is independent of the particular execution semantics (sequential or parallel). In Sect. 23.5, we show that this method closer simulates intelligent reasoning than the immediate use of selection functions.

23.4.1 Background

Endofunctors on the category of sets and functions are a suitable mathematical framework for an abstract notion of structure, and throughout we will use the word *functor* exclusively for such mappings. We may suppress the composition symbol and simply write $\mathcal{GF}$ for the mapping which first applies $\mathcal{F}$ and then applies $\mathcal{G}$; similarly, we may write $\mathcal{F}X$ for the application of functor $\mathcal{F}$ on a set X.

There are numerous functors used throughout mathematics and computer science, for example elements of the *flat polynomial* functors,

$$\mathcal{F} ::= \mathcal{I} \mid \mathcal{K}_A \mid \mathcal{F} \times \mathcal{F} \mid \mathcal{F} + \mathcal{F}$$

Here $\mathcal{I}$ is the identity functor (i.e., the mapping fixing both objects and functions); $\mathcal{K}_A$ is the constant functor for a given set A, which is defined by mapping any set to the set A and any arrow to the identity function on A; $\mathcal{F} \times \mathcal{G}$ is the cartesian product of $\mathcal{F}$ and $\mathcal{G}$; and, $\mathcal{F} + \mathcal{G}$ is the disjoint union of $\mathcal{F}$ and $\mathcal{G}$.

Occasionally one can transform one functor into another uniformly. That is, one can make the transformation componentwise, so that the actions on sets and function cohere.

Definition 23.4.1 *(Natural Transformation)* A collection of functions indexed by sets $\mathfrak{n} := (\mathfrak{n}_X)$ is a natural transformation between functors $\mathcal{F}$ and $\mathcal{G}$ if and only if $\mathfrak{n}_X : \mathcal{F}X \to \mathcal{G}X$ and if $f : X \to Y$ then then the following diagram commutes:

$$\begin{tikzcd}
\mathcal{F}X \arrow[r, "\mathfrak{n}_X"] \arrow[d, "\mathcal{F}(f)"'] & \mathcal{G}X \arrow[d, "\mathcal{G}(f)"] \\
\mathcal{F}(Y) \arrow[r, "\mathfrak{n}_Y"'] & \mathcal{G}(Y)
\end{tikzcd}$$

Every functor $\mathcal{F}$ admits at least one natural transformation called the identity: $\mathfrak{i}_X := I_{\mathcal{F}X}$, where $I_{\mathcal{F}X}$ is the identify function on $\mathcal{F}X$. As an abuse of notation, we use the notation of function types when speaking about natural transformation; that is, we may write $\mathfrak{n} : \mathcal{F} \to \mathcal{G}$ to denote that $\mathfrak{n}$ is a natural transformation between $\mathcal{F}$ and $\mathcal{G}$.

There are two particularly well-behaved classes of functors, called *monads* and *comonads*, that are useful abstractions of data-type and behaviour-type when modelling computation.

Definition 23.4.2 (*Monad and Comonad*) Let $\mathcal{T}$ be a functor. It is a (co)monad if there are natural transformations $\mathfrak{u} : \mathcal{I} \to \mathcal{T}$ and $\mathfrak{m} : \mathcal{T}^2 \to \mathcal{T}$ (resp. $\mathfrak{u} : \mathcal{T} \to \mathcal{I}$ and $\mathfrak{m} : \mathcal{T} \to \mathcal{T}^2$) satisfying the following commutative diagrams:

$$
\begin{tikzcd}
(\mathcal{T}\mathcal{T}\mathcal{T})X \arrow[r, "\mathcal{T}\mathfrak{m}_X"] \arrow[d, "\mathfrak{m}_{\mathcal{T}X}"'] & (\mathcal{T}\mathcal{T})X \arrow[d, "\mathfrak{m}_X"] \\
(\mathcal{T}\mathcal{T})X \arrow[r, "\mathfrak{m}_X"'] & \mathcal{T}X
\end{tikzcd}
\qquad
\left(\text{resp.}\quad
\begin{tikzcd}
(\mathcal{T}\mathcal{T}\mathcal{T})X & (\mathcal{T}\mathcal{T})X \arrow[l, "\mathcal{T}\mathfrak{m}_X"'] \\
(\mathcal{T}\mathcal{T})X \arrow[u, "\mathfrak{m}_{\mathcal{T}X}"] & \mathcal{T}X \arrow[l, "\mathfrak{m}_X"] \arrow[u, "\mathfrak{m}_X"']
\end{tikzcd}
\right)
$$

$$
\begin{tikzcd}
\mathcal{T}X \arrow[r, "\mathfrak{u}_{\mathcal{T}X}"] \arrow[d, "\mathcal{T}\mathfrak{u}_X"'] \arrow[dr, "\mathcal{I}"] & (\mathcal{T}\mathcal{T})X \arrow[d, "\mathfrak{m}_X"] \\
(\mathcal{T}\mathcal{T})X \arrow[r, "\mathfrak{m}_X"'] & \mathcal{T}X
\end{tikzcd}
\qquad
\left(\text{resp.}\quad
\begin{tikzcd}
\mathcal{T}X & (\mathcal{T}\mathcal{T})X \arrow[l, "\mathfrak{u}_{\mathcal{T}X}"'] \\
(\mathcal{T}\mathcal{T})X \arrow[u, "\mathcal{T}\mathfrak{u}_X"] & \mathcal{T}X \arrow[l, "\mathfrak{m}_X"] \arrow[u, "\mathfrak{m}_X"'] \arrow[ul, "\mathcal{I}"']
\end{tikzcd}
\right)
$$

The natural transformations $\mathfrak{u}$ and $\mathfrak{m}$ are often called the (co)unit and (co)multiplication of the (co)monad. There is an abundance of examples of monads; for example, the powerset functor $\mathcal{P}$, which takes sets to their powersets and functions to their direct image functions, is a monad whose unit is the singleton function and whose multiplication is the union operator. Simple examples of comonads are less common. However, since they are used to define behaviour-type, modelling operational semantics will involve defining one: the proof-search comonad.

Under relatively mild conditions, there is a canonical way to construct a (co)monad from a functor: the (co)free construction. Heuristically, this is the indefinite application of the functor structure until a fixed-point is reached. The cofree construction is analogous.

Example 23.4.3 Consider the functor $\mathcal{F}_A : X \mapsto \mathsf{nil} + A \times X$, where nil is the emptyset. One can generate the least fixed point $\mathcal{L}_A$ for $\mathcal{F}_A$ by the ω-chain in the following diagram:

$$\mathsf{nil} \longrightarrow \mathsf{nil} + A \times \mathsf{nil} \longrightarrow \mathsf{nil} + A \times (\mathsf{nil} + A \times \mathsf{nil}) \longrightarrow ...$$

The arrows are inductively defined by extending with the unique function out of the emptyset. The mapping $A \mapsto L_A$ defines the free functor $\mathcal{L}_A$, which can be understood as structuring elements of A as lists (identified with products). It is a monad whose unit is the single-element list constructor $a \mapsto a$::(nil::nil...) and whose multiplication concatenation.

Given a space structured by a functor $\mathcal{F}$, one can define actions which respect the structure. There are two directions: either one wants the domain to be structured, in which case one has an $\mathcal{F}$-algebra, or the codomain, in which case one has a $\mathcal{F}$-coalgebra. When structure represents a data-type (resp. a behaviour-type) given by a (co)monad, one may add extra conditions on the (co)algebra that make it a co(module). These functions are used to give abstract models of data and behaviour.

Definition 23.4.4 (*Algebra and Coalgebra*) Let $\mathcal{T}$ be a functor, then a function $\alpha : \mathcal{T}X \to X$ is called an $\mathcal{T}$-algebra and any function $\beta : X \to \mathcal{T}X$ is a $\mathcal{T}$-coalgebra. If $\langle \mathcal{T}, \mathfrak{u}, \mathfrak{m} \rangle$ is a monad (resp. comonad), then α (resp. β) is a module (resp. comodule) when the following diagrams commute:

$$\begin{tikzcd} X \arrow[r, "\mathfrak{u}_X"] \arrow[dr, "\mathfrak{i}_X"'] & \mathcal{T}X \arrow[d, "\alpha"] \\ & X \end{tikzcd} \qquad \left(\text{resp.} \quad \begin{tikzcd} X & \mathcal{T}X \arrow[l, "\mathfrak{u}_X"'] \\ & X \arrow[ul, "\mathfrak{i}_X"] \arrow[u, "\alpha"'] \end{tikzcd} \right)$$

$$\begin{tikzcd} \mathcal{T}\mathcal{T}X \arrow[r, "\mathcal{T}\alpha"] \arrow[d, "\mathfrak{m}_A"'] & \mathcal{T}X \arrow[d, "\alpha"] \\ \mathcal{T}X \arrow[r, "\alpha"'] & X \end{tikzcd} \qquad \left(\text{resp.} \quad \begin{tikzcd} \mathcal{T}\mathcal{T}X & \mathcal{T}X \arrow[l, "\mathcal{T}\alpha"'] \\ \mathcal{T}X \arrow[u, "\mathfrak{m}_A"] & X \arrow[l, "\alpha"] \arrow[u, "\alpha"'] \end{tikzcd} \right)$$

The abstract modelling of operational semantics witnesses both the use of algebra and coalgebra: the former for specifying the constructs, and the latter for specifying the transitions. In the best case the two structures cohere, captured mathematically by the mediation of a natural transformation called a distributive law, and form a bialgebra.

Definition 23.4.5 *(Distributive Law)* A distributive law for a functor $\mathcal{G}$ over a functor $\mathcal{F}$ is a natural transformation $\partial : \mathcal{G}\mathcal{F} \to \mathcal{F}\mathcal{G}$.

Definition 23.4.6 (*Bialgebra*) Let $\partial : \mathcal{G}\mathcal{F} \to \mathcal{F}\mathcal{G}$ be a distributive law, and let $\alpha : \mathcal{G}X \to X$ be an algebra and $\beta : X \to \mathcal{F}X$ be a coalgebra. The triple (X, α, β) is a ∂-bialgebra when the following diagram commutes:

$$\begin{tikzcd} \mathcal{G}X \arrow[r, "\alpha"] \arrow[d, "\mathcal{G}\beta"'] & X \arrow[r, "\beta"] & \mathcal{F}X \\ \mathcal{G}\mathcal{F}X \arrow[rr, "\partial_X"'] & & \mathcal{F}\mathcal{G}X \arrow[u, "\mathcal{F}\alpha"'] \end{tikzcd}$$

There are additional coherence condition which may be applied for when one has a monad or a comonad structure.

In Turi's and Plotkin's bialgebraic models of operational semantics (Turi & Plotkin, 1997), the algebra supplies the structure of the syntax and the coalgebra supplies the behaviour of execution, and under relatively mild conditions (i.e. the coalgebra structure preserves weak pullbacks) forms are even a *full* abstractions (with respect to bisimulation).

23.4.2 Reduction Operators

To model proof-search coalgebraically, we first model the key components: reduction operators. These are the essential elements in reductive logic and proof-search, and they readily admit coalgebraic treatment by dualizing the algebraic perspective of deductive logic. We begin, however, by modelling the syntax of hHBI.

The construction of the list monad $\mathcal{L}$ in Example 23.4.3 is a model of the grammar of infinite lists,

$$\ell ::= x :: \ell$$

The variable x represents a data element from a set A, so is interpreted by the functor $\mathcal{K}_A$; the ℓ represents a list, the thing being modelled, so is interpreted by identity $\mathcal{I}$; and the :: constructor represents the pairing of the two components, so is interpreted as the product. In the BNF format, the | symbol represents a choice of productions, so it is modelled by the sum of functors, therefore the base for the *free* construction in Example 23.4.3 is $X \mapsto \mathsf{nil} + A \times X$.

Modelling configurations $P \vdash G$ as abstract data structures works analogously; that is, we use the productions in the context-free grammars for definite clauses and goals in Definition 23.2.8 to generate functors to which we apply the free construction. There are several pairs in these grammar(s) that ought to be distinguished, such as $\phi \wedge \psi$ and $\phi * \psi$, so rather than simply using a binary product, one signs the pairs, resulting in functors of the following shape:

$$\wedge : X \mapsto X \times \{\wedge\} \times X \qquad \text{and} \qquad * : X \mapsto X \times \{*\} \times X$$

The functors will be written using infix notation to simplify presentation. The X is a set of variables upon which the construction takes place, at the end of the construction the variables will be replaced by the constants $\mathbb{A}$.

Modelling the syntax of the hereditary Harrop fragment of BI is slightly more elaborate because of the mutual induction taking place over definite clauses and goals. Consequently, we first consider mappings of two variables delineating the structural difference of the two types of formulas:

$$\begin{aligned}
\mathcal{F}_G(X, Y) &:= X \wedge X \;+\; X * X \;+\; X \vee X \;+\; Y \to X \;+\; Y -\!\!* X \;+\; \mathcal{K}_{\{\top^*\}} \;+\; \mathcal{K}_{\{\top\}} \\
\mathcal{F}_D(X, Y) &:= Y \wedge Y \;+\; Y * Y \;+\; X \to Y \;+\; X -\!\!* Y \;+\; \mathcal{K}_{\{\top^*\}} \;+\; \mathcal{K}_{\{\top\}}
\end{aligned}$$

To recover the free construction, we unfold the inductive definition simultaneously and use the first and second projection function π_1 and π_2 to put the right formulas in the right place. That is, we consider the following functor:

$$\mathcal{F}(Z) := \mathcal{F}_G(\pi_1 Z, \pi_2 Z) \times \mathcal{F}_D(\pi_1 Z, \pi_2 Z)$$

Here sets Z contain products. The functors defining $\mathcal{F}$ are sufficiently simple and the category sufficiently well-behaved that the free construction yields a limit,

$$\begin{aligned} Z \longrightarrow\ & Z + \mathcal{F}Z = \\ & Z + \underbrace{\mathcal{F}_G\big(\pi_1(Z + \mathcal{F}Z),\ \pi_2(Z + \mathcal{F}Z)\big)}_{\mathcal{F}_G(\pi_1 Z + \mathcal{F}_G Z,\, \pi_2 Z + \mathcal{F}_D Z)} \times \underbrace{\mathcal{F}_D\big(\pi_1(Z + \mathcal{F}Z),\ \pi_2(Z + \mathcal{F}Z)\big)}_{\mathcal{F}_D(\pi_1 Z + \mathcal{F}_G(Z),\, \pi_2 Z + \mathcal{F}_D(Z))} \longrightarrow \ldots \end{aligned}$$

Each transition in the construction is the embedding of the previously constructed set within the next which contains it as a component of its disjoint union; for example, the first arrow embeds Z in $Z + \mathcal{F}(Z)$. This embedding is in fact a natural transformation $\mathcal{I} \to \mathcal{I} + \mathcal{F}$. Hence, as the construction continues more and more stages of the inductive definition of goals and definite clauses are captured, and the limit, therefore, contains all the possible goals and definite clauses at once.

Let $\hat{\mathcal{F}}$ denote the free functor for $\mathcal{F}$, then the goal formulas and definite clauses are recovered via the first and second projections,

$$\hat{\mathcal{F}}_G(X) := \pi_1 \hat{\mathcal{F}}(X, X) \qquad \hat{\mathcal{F}}_D(X) := \pi_2 \hat{\mathcal{F}}(X, X)$$

By fixing the set of atomic proposition $\mathbb{A}$ (i.e., the base of the inductive construction), the disjoint union present in the free construction means that all goal formulas and definite clauses are present in $\hat{\mathcal{F}}_G(\mathbb{A})$ and $\hat{\mathcal{F}}_D(\mathbb{A})$, respectively.

The model of bunches is comparatively simple since there is no mutual induction, so it simply requires the free monad $\hat{\mathcal{B}}$ for the functor

$$\mathcal{B} := \hat{\mathcal{D}} + \fatsemi + \ ; \ + \mathcal{K}_{\{\varnothing_\times\}} + \mathcal{K}_{\{\varnothing_+\}}$$

Configurations are pairs of programs and goals, so their abstract data-structure is given by the functor $\mathcal{G}(X) := \hat{\mathcal{B}}(X) \times \hat{\mathcal{F}}(X)$. In particular, configurations are modelled by elements of $\mathcal{G}(\mathbb{A})$. Below, we use the abstract data structure and the formal grammar interchangeably.

Example 23.4.7 The configuration $(A \wedge \top) \mathrel{-\!\!*} A \vdash A$, where $A \in \mathbb{A}$, is modelled by the tuple $\big(((A, \wedge, \top), \mathrel{-\!\!*}, A), A\big)$, which is nothing more than the typical encoding of the syntax-tree as an ordered tree.

A rule R in a sequent calculus is typically given by a rule figure together with correctness conditions, and can be understood mathematically as relation $\mathbf{R}$ on sequents that holds only when the first sequent (the conclusion) is inferred from the remaining (the premisses) by the rule.

Example 23.4.8 The exchange rule E in **LBI** is comprised of the following figure, together with correctness condition $\Delta' \equiv \Delta$:

$$\frac{\Delta \implies \phi}{\Delta' \implies \phi}\,\mathrm{E}$$

Consider the following:

$$\begin{array}{l}(\varnothing_\times \implies A)\,\mathbf{E}\,((\varnothing_\times, \varnothing_\times) \implies A)\\(\varnothing_\times \implies A)\,\mathbf{E}\,((\varnothing_\times; \varnothing_\times) \implies A)\end{array}$$

Both have the right shape, but only the first respects the correctness condition and so is the only one of the two that holds.

This understanding immediately lends itself to the modelling of a rule as a non-deterministic partial function δ, an example of the functions we called deduction operators in Sect. 23.2.3,

$$\delta : \mathcal{LG}(\mathbb{A}) \rightharpoonup \mathcal{PG}(\mathbb{A}) : \ell \mapsto \{(P, G) \in \mathcal{G}(\mathbb{A}) \mid (P, G)\,\mathbf{R}\,\ell\}$$

In the study of proof-search, we consider the inverse action, since we have a putative conclusion and look for sufficient premisses,

$$\delta^{-1} : \mathcal{G}(\mathbb{A}) \to \mathcal{PLG}(\mathbb{A}) : (P, G) \mapsto \{\ell \in \mathcal{LG}(\mathbb{A}) \mid \delta(\ell) = (P, G)\}$$

Example 23.4.9 Let R be the $\to$-resolution rule in the modified sequent calculus (Lemma 23.2.11), then the deduction operator and its inverse are the following:

$$\begin{array}{lll}\delta & : [(P; G \to A \vdash G), (Q \vdash A)] & \mapsto (Q, (P; G \to A)) \vdash A\\ \delta^{-1} & : (Q, (P; G \to A)) \vdash A & \mapsto [(P; G \to A \vdash G), (Q \vdash A)]\end{array}$$

It is worth comparing this presentation to the one given in Lemma 23.2.11; where δ represents the downward reading, δ^{-1} represents the upward one. □

We make now an important assumption for proof-search, as presented herein, to be effective: the proof-search space must be finitely branching. Rules whose upward reading renders an infinite set of sufficient premisses (e.g., E) cannot be handled with step-by-step reduction since the breadth-first approach to search may fail. At first this assumption seems to be hugely restrictive, since almost all common proof-systems seemingly fail this criterion; however, it is not to be understood as injunction against such rules, merely a requirement to *control* them. For example, the E rule is confined in the resolution system (Fig. 23.3) to the weak coherence ordering in Sect. 23.2.3, and an algorithmic approach to its use is offered. Hence, we drop the $\mathcal{P}$ structure in deduction operators and their inverses, and replace it with the finite powerset monad $\mathcal{P}_f$.

Moreover, since there is no logical constraint on the order in which sufficient premisses are verified, we may simplify the structure of the states by replacing the list

$$
\begin{array}{rlcl}
\rho_\top : & P \vdash \top & \mapsto & \{\Box\} \\
\rho_{\top^*} & P \vdash \top^* & \mapsto & \{\Box \mid [P] \equiv \varnothing_\times ; R\} \\
\rho_{res_1} : & P \vdash A & \mapsto & \{\{Q \vdash A\} \mid [P] \leqq Q, A\} \\
\rho_{res_2} : & P \vdash A & \mapsto & \{\{Q \vdash G\} \mid [P] \leqq Q, G \mathrel{-\!\!*} A\} \\
\rho_{res_3} : & P \vdash A & \mapsto & \{\{Q \vdash \top^* , R; G \to A \vdash G\} \mid [P] \leqq Q, (R; G \to A)\} \\
\rho_\vee : & P \vdash G_1 \vee G_2 & \mapsto & \{\{P \vdash G_1\}, \{P \vdash G_2\}\} \\
\rho_\wedge : & P \vdash G_1 \vee G_2 & \mapsto & \{\{P \vdash G_1 , P \vdash G_2\}\} \\
\rho_* : & P \vdash G_1 * G_2 & \mapsto & \{\{Q \vdash G_1 , R \vdash G_2\} \mid P \equiv (Q, R); S\} \\
\rho_\to : & P \vdash D \to G_2 & \mapsto & \{\{P; [D] \vdash G\}\} \\
\rho_{-*} : & P \vdash D \mathrel{-\!\!*} G_2 & \mapsto & \{\{P, [D] \vdash G\}\}
\end{array}
$$

Fig. 23.5 Reduction operators for the resolution system

structure $\mathcal{L}$ with the finite powerset monad $\mathcal{P}_f$ too; indeed this simplification is tacit in the labelled transition system as all the possible configurations in a possible state are explored simultaneously. When the construction in Example 23.4.9 is performed for the resolution system in Fig. 23.3, with the simplification on the structure of states and collections of premisses, one forms the coalgebras in Fig. 23.5, which are formally the *reduction operators* for the resolution system.

Definition 23.4.10 (*Reduction Operator*) A reduction operator for a logic L with sequent structure $\mathcal{G}(\mathbb{A})$ is a coalgebra $\rho : \mathcal{G}(\mathbb{A}) \to \mathcal{P}_f\mathcal{P}_f\mathcal{G}(\mathbb{A})$ such that $\rho(C) = \{P_0, ..., P_n\}$ if and only if the following inference is L-sound:

$$\frac{P_0 \qquad \ldots \qquad P_n}{C}$$

The would-be atomic steps of proof-search, a reductive inference, can be decomposed into two steps in the coalgebraic model of rules above: the application of a reduction operator, and the selection of a set of sufficient premisses,

$$\mathcal{G}(\mathbb{A}) \xrightarrow{\rho} \mathcal{P}_f\mathcal{P}_f\mathcal{G}(\mathbb{A}) \xrightarrow{\sigma} \mathcal{P}_f\mathcal{G}(\mathbb{A})$$

The choices presented by these steps represent the control problems of proof-search, as discussed in Sect. 23.2.3; that is, the choice of rule, and the choice of instance.

Simply by this presentation one has insight into the proof-search behaviour. For example, the disjointedness of the defined portions of the reduction operators for the operational and initial rules means the choice of application is deterministic for a non-atomic goal. Therefore, one may coalesce the operators into a single goal-destructor:

$$\rho_{\mathrm{op}} := \rho_\vee + \rho_\wedge + \rho_* + \rho_\to + \rho_{-*} + \rho_\top + \rho_{\top^*}$$

Moreover, since there is no *a priori* way to know which resolution rule to use, one in principle tests all of them, so uses a reduction operator of the shape:

$$\rho_{\text{res}} : P \vdash A \mapsto \rho_{res_1}(P \vdash A) \cup \rho_{res_2}(P \vdash A) \cup \rho_{res_3}(P \vdash A)$$

The presentation of ρ_{op} and ρ_{res} as operators is simple when working with coalgebras, but it is not at all clear how to present them as a single rule figures. Attempts toward such presentations are the *synthetic* rules derived from focused proof systems (Chaudhuri et al., 2016). It is thus that the first control problem vanishes for hereditary Harrop fragment of BI (and in logic programming in general), since one may then say that *the* reduction operator for resolution proofs in the hereditary Harrop fragment of BI is $\rho := \rho_{\text{op}} + \rho_{\text{res}}$.

23.4.3 The Proof-Search Space

The construction of reduction operators as above is an interpretation of a reduction step. We now turn to modelling reduction proper; that is, we construct a coalgebraic interpretation of the *proof-search space*—the structure explored during proof-search—itself. In practice, we simply formalize the heuristic exploratory approach of reductive inference as the corecursive application of reduction operators.

Computations, such as proof-search, can be understood as sequences of actions, and the possible traces can be collected into tree structures where different paths represent particular threads of execution. In logic programming, such trees appear in the literature as *coinductive derivation trees* (CD-trees) (Komendantskaya et al., 2011; Komendantskaya & Power, 2011; Komendantskaya et al., 2016; Bonchi & Zanasi, 2015), and an *action* is one step of reductive inference. Typically one distinguishes the two components, the reduction operator and the choice function, by using an intermediary node labelled with a $\bullet$, sometimes called an *or*-node, as it represents the disjunction of sets of sufficient premisses.

Example 23.4.11 Let $P = ((\varnothing_\times; B), C \mathrel{-\!\!*} A, (B; B \to A)); A$ and $G = \top \to A$, then the (P)CD-tree for $P \vdash G$ in the resolution system is the following:

$$\begin{tikzcd}
 & & & \bullet \arrow[r] & \Box & & \\
P \vdash G \arrow[r] & \bullet \arrow[r] & P; \varnothing_+ \vdash A \arrow[r] \arrow[ur] \arrow[dr] & \bullet \arrow[r] & (\varnothing_\times; B), (B; B \to A) \vdash C & & \\
 & & & \bullet \arrow[r] \arrow[dr] & B; B \to A \vdash B \arrow[r] & \bullet \arrow[r] & \Box \\
 & & & & \varnothing_\times; B \vdash \top^* \arrow[r] & \bullet \arrow[r] & \Box
\end{tikzcd}$$

At the first bifurcation point, the three • nodes represent from top to bottom the choice of the unique member of $\rho_{res_i}(P; \varnothing_\times \vdash A)$ for $i = 1, 2, 3$. In the case of ρ_{res_1} and ρ_{res_3}, the procedure continues and terminates successfully; meanwhile, for ρ_{res_2}, it fails.

The bullets • serve only as *punctuation* separating the possible choice functions, so the actual coinductive derivation tree is the tree without them. That is, the punctuation helps to define the proof-search space, but is not part of it.

Definition 23.4.12 (*(Punctuated) Coinductive Derivation Tree*) A punctuated coinductive derivation tree (PCD-tree) for a sequent S is a tree satisfying the following:

- The root of the tree is S
- The root has $|\rho(S)|$ children labelled •
- For each • there is a unique set $\{S_0, ..., S_n\} \in \rho(S)$ so that the the children are PCD-trees for the S_i.

A saturated coinductive derivation tree for a sequent (CD-tree) is the tree constructed from the PCD-tree for the sequent constructed by connecting the parents of •-nodes directly to the children, removing the node itself.

The CD-trees model reduction only (as opposed to proof-search) since the representation of a control régime is lacking; nonetheless, we shall see in Sect. 23.4.4 that a model of proof-search is immediately available. A possible approach to at least represent the controls used in a particular search is to replace the bullets with integers that enumerate the order of preference in the possible choices.

The CD-structure on a set X of sequents is formally the cofree comonad $C(X)$ on the $\mathcal{P}_f\mathcal{P}_f$ functor, the behaviour-type of reduction, constructed inductively as follows:

$$\begin{cases} Y_0 & := X \\ Y_{\alpha+1} & := X \times \mathcal{P}_f\mathcal{P}_f Y_\alpha \end{cases}$$

Each stage of the construction yields a coalgebra $\rho_\alpha : X \to Y_\alpha$ defined inductively as follows, where I is the identify function:

$$\begin{cases} \rho_0 & := I \\ \rho_{\alpha+1} & := I \times (\mathcal{P}_f\mathcal{P}_f\rho_\alpha \circ \rho) \end{cases}$$

For some limit ordinal λ, the coalgebra $\rho_\lambda : \mathcal{G}(\mathbb{A}) \to C(\mathcal{G}(\mathbb{A}))$ precisely maps a configuration to its CD-tree. To show that this model of the proof-search space is faithful we must show that every step, represents a valid reduction; meanwhile, to show that it is adequate we must prove that every proof is present.

Definition 23.4.13 (*Controlled Subtree of CD-tree*) A subtree $\mathcal{R}$ of $\rho_\lambda(P, G)$ is controlled if and only if is a tree extracted from the PCD-tree for $P \vdash G$ by taking the root node and connecting it to the heads of the reduction trees of all the children of one •-node. It is successful if and only if the leaves are all $\Box$.

Observe that the application of a choice function, determined by a control régime, is precisely the choosing of a particular $\bullet$-node at each stage of the extraction.

Example 23.4.14 The following is an example of a controlled sub-tree from the example (P)CD-tree above:

$$P \vdash G \longrightarrow P; \varnothing_{\times} \vdash A \longrightarrow \square$$

The first choice of $\bullet$ is trivial (as there is only one) and the second choice is the upper path.

We do not claim that every controlled sub-tree in a CD-tree is finite; in fact, this is demonstrably not the case.

Example 23.4.15 Consider the PCD-tree for $A; A \to A \vdash A$:

$$\begin{tikzcd}
A \to A; A \vdash A \arrow[r] & \bullet \arrow[r] \arrow[dr] & A \to A; A \vdash A \arrow[r] & \bullet \arrow[r] \arrow[dr] & A \to A; A \vdash A \arrow[r] & \ldots \\
 & & \square & & \square &
\end{tikzcd}$$

Every finite execution of the configuration is successful; however, there is an infinite path which represents an attempt at proof-search that never terminates, but also never reaches a invalid configuration. This further demonstrates the care that is required when implementing controls because the *depth-first search with leftmost selection* régime here fails.

Theorem 23.4.16 (Soundness and Completeness) *A tree labelled with sequents is a proof of a configuration $P \vdash G$ in the resolution system if and only if it is a successful controlled sub-tree of $\rho_{\lambda}(P, G)$.*

Proof Immediate by induction on the height of proofs and the definition of reduction tree. $\square$

23.4.4 Choice and Control

The difference between modelling the proof-search space and modelling proof-search is subtle, and comes down to the handling of the space of sufficient premisses. Throughout we have restricted attention to a traditional sequential paradigm for computation, where one chooses one particular collection of sufficient premisses, we now show that an alternative parallel model is immediately available from the constructions above. A notion of control is, nonetheless, ever present, and so we also show how the coalgebra offers a unifying perspective for a particular approach to it that is independent of the particular execution semantics.

It is the τ rule in the operational semantics of Sect. 23.2.3 that invokes the reduction operators. It expresses precisely the two-step understanding of reductive inference: apply a reduction operator, choose a collection of sufficient premisses. The usual way to interpret it is to perform these action in sequence as read, yielding the backtracking-schedule approach to computation studied so far; however, it may be interpreted simply *extracting* a correct reduction from the proof-search space. In this latter reading the traces of the proof-search space can be understood as being in a superposition, forming a parallel semantics of computation. This reading has been thoroughly studied for HcLP (Gupta & Santos Costa, 1994; Gupta et al., 2001, 2007) and this idea of parallelism can be captured proof-theoretically by hypersequent calculi (Harland, 2001; Kurokawa, 2009).

The coalgebraic model of the proof-search space immediately offers a coalgebraic model of parallel proof-search; that is, the controlled sub-tree extraction from $\rho_\lambda : \mathcal{G}(\mathbb{A}) \to \mathcal{C}(\mathcal{G}(\mathbb{A}))$. Indeed, this is precisely analogous to the parallel model of HcLP (Komendantskaya et al., 2011; Komendantskaya & Power, 2011; Komendantskaya et al., 2016). The coalgebraic approach has the advantage over more traditional algebraic models in that it allows for infinite searches, thereby extending the power of logic programming to include features such as corecursion (Gupta et al., 2007; Simon et al., 2007; Komendantskaya et al., 2016).

Indeed, the parallel semantics is amenable to a more accurate model by unpacking the algebraic structure of the state-space, yielding a bialgebraic semantics. Observe then that in the structure $\mathcal{P}_f\mathcal{P}_f$ for reduction operators, the external functor structures the set of choices, and the internal one structures the states themselves (the collections of sufficient premisses). The outer one is disjunctive, captured diagrammatically by the *or*-nodes represented by $\bullet$ in the PCD-trees; meanwhile, the inner one is conjunctive since every premiss needs to be verified. There is no distributive law of $\mathcal{P}_f$ over $\mathcal{P}_f$, but there is a distributive law of $\mathcal{L}$ over $\mathcal{P}_f$ which coheres with this analysis:

$$[X_1, ..., X_N] \mapsto \{[x_1, ..., x_n] \mid x_i \in X_i\}$$

This suggests a bialgebraic model for the parallel reading of the operational semantics given in Sect. 23.2.3, obtained by performing the same cofree comonad construction for the behaviour, but with $\mathcal{P}_f\mathcal{L}$ instead.

These models are studied partly to let one reason about computation, and perhaps use knowledge to improve behaviour. For example, in the sequential semantics a programmer may purposefully tailor the program to the selection function to have better behaviour during execution, meanwhile in the parallel approach the burden is shifted to the machine (or, rather, theorem prover) which may give more time to branches that are more promising.

Example 23.4.17 While generating the (P)CD-tree in Example 23.4.11 a theorem prover can ignore the branch choosing the $C \multimap A$ in the program since it is clear that it will never be able to justify C as the atom appears nowhere else in the context.

The problem being handled in either case is how best to explore the space of possible reductions. The two approaches, parallel and sequential, both suffer from the *amount* of non-determinism in the reduction operator ρ. In fact, this problem is exponentially increasing with each inference made as each collection of premisses represents another branch in the CD-tree. Moreover, in practice, with any additional features in the logic the problem compounds so that such reasoning becomes increasingly intractable.

As a case study, consider the plethora of research on *resource management* in linear logic programming (Hodas & Miller, 1994; Hodas, 1994; Cervesato et al., 2000; Hodas et al., 2002; López & Polakow, 2005) (which has sequential execution), to deal with the problem of context decomposition in operators such as ρ_*. These solution work for linear logic, but the *input-output* method, which underpins many of these mechanisms, forces a commitment to depth-first search, which is not always ideal. Moreover, the solution simply do not work hHBI because of the presence of another context-former.

One approach to improve the situation, given a particular reduction system, is to attempt to make the operators deterministic. Occasionally, it may be possible to quotient different paths or choices in the CD-tree by enriching the calculus with algebraic labels that can be used to differentiate the threads at a later point. That is, suppose we wish to perform a search in a system $\mathbf{L}'$, then we may simplify the problem by appealing to a more well-behaved $\mathbf{L}$,

$$\text{Proof-search in } \mathbf{L}' = \text{ Reduction in } \mathbf{L} + \text{ Controls } \mathcal{A}$$

For the resource management problem this is captured by the *resource distribution via boolean constraints* (Harland & Pym, 2003, 1997) mechanism, which is uniformly applicable to LL and BI.

Modifying the rules of the sequent calculus so that sequents carry labels from a set $\mathcal{X}$ of boolean variables, one has inferences of the form:

$$\frac{\phi_0[e_0 f_0], ..., \phi_n[e_n f_n] \Rightarrow \psi_0[e] \qquad \phi_0[e_0 \bar{f}_0], ..., \phi_{G+1}[e_n \bar{f}_n] \Rightarrow \psi_1[e]}{\phi_0[e_0], ..., \phi_n[e_n] \implies \psi_0 * \psi_1[e]}\ e = 1$$

The possible choices for decomposing the context can be understood as assignments $A : \mathcal{X} \to \{0, 1\}$ subject to the constraints collected during proof-search, such as $A(e) = 1$. Let $\mathcal{B}$ denote boolean algebra, then the boolean-constraints approach to resource distribution instantiates the equation,

$$\mathbf{BI} = \mathbf{LJ} + \mathcal{B}$$

Example 23.4.18 Consider the BI sequent $(A, (B; C), B \mathrel{-\!\!*} D); E \implies D * A$, and consider the following reduction tree with boolean labels:

$$\frac{\dfrac{\dfrac{x_0 = 0 \qquad x_1 = 1}{A[x_0], B[x_1] \Rightarrow B[x_3]}}{A[x_0], (B[x_1]; C[x_1]) \Rightarrow B[x_3]} \qquad \mathcal{D}}{(A[x_0], (B[x_1]; C[x_2]), (B \mathrel{-\!\!*} D)[x_3]); E[x_4] \Rightarrow D * A[x_5]}\ x_3, x_5 = 1, x_1 = x_2$$

where $\mathcal{D}$ is

$$\dfrac{\dfrac{\bar{x}_0 y_0 = 0 \qquad \bar{x}_1 y_1 \bar{x}_2 y_1 = 0 \qquad x_3 y_3 = 1}{A[\bar{x}_0 y_0], (B[\bar{x}_1 y_1]; C[\bar{x}_2 y_1]), D[x_3 y_3] \Rightarrow D[x_5]} \qquad \dfrac{\bar{x}_0 \bar{y}_0 = 1 \qquad \bar{x}_1 \bar{y}_1 \bar{x}_1 \bar{y}_1 = 0 \qquad x_3 \bar{y}_3 = 0}{A[\bar{x}_0 \bar{y}_0], (B[\bar{x}_1 \bar{y}_1]; C[\bar{x}_1 \bar{y}_1]), D[x_3 \bar{y}_3 = 0] \Rightarrow A[x_5]}}{\dfrac{(A[\bar{x}_0], (B[\bar{x}_1]; C[\bar{x}_2]), D[x_3]) \Rightarrow D * A[x_5]}{(A[\bar{x}_0], (B[\bar{x}_1]; C[\bar{x}_2]), D[x_3]); E[x_4] \Rightarrow D * A[x_5]}\, x_5 = \bar{x}_0 + \bar{x}_1 + \bar{x}_2 + x_3}$$

Each reductive inference replaces the distribution condition of the corresponding rule in **LBI** with a side-condition encoding it, thereby making it deterministic, without loss of expressivity. An assignment satisfying all the constraints is,

$$x_0 = 0, x_1 = 1, x_2 = 1, x_3 = 1, x_4 = 1, x_5 = 1, y_0 = 0, y_1 = 1, y_3 = 1$$

which instantiates to the following **LBI**-proof,

$$\dfrac{\dfrac{B \Rightarrow B}{(B; C) \Rightarrow B}\,W \qquad \dfrac{\dfrac{D \Rightarrow D \qquad A \Rightarrow A}{A, D \Rightarrow D * A}\,{*R}}{(A, D); E \Rightarrow D * A}\,W}{(A, (B; C), (B \mathrel{-\!\!*} D)); E \Rightarrow D * A}\,{-\!\!*\, L}$$

Indeed, not only does the boolean constraints system have the same expressive power as **LBI**, but generally each reduction offers more information. For example, another assignment available for the above is the same but with $x_4 = 0$, meaning that we have at the same time found a proof of $A, (B; C), (B \mathrel{-\!\!*} D) \Rightarrow D * A$. □

This approach to control can be understood abstractly in the coalgebraic setting as a lifting of a proof-search step. Let $(\mathcal{G}.\mathcal{A})(\mathbb{A})$ be sequents labelled with variables from an $\mathcal{A}$-algebra together with $\mathcal{A}$-algebra equations, let ρ be a reduction operator for $\mathbf{L}'$, and let π be a deterministic reduction operator for $\mathbf{L}$, then the mechanism is captured by the following diagram:

$$\begin{tikzcd}
(\mathcal{G}.\mathcal{A})(\mathbb{A}) \arrow[rr, "\pi"] & & \mathcal{L}(\mathcal{G}.\mathcal{A})(\mathbb{A}) \arrow[d, "\omega"] \\
\mathcal{G}(\mathbb{A}) \arrow[u, "\alpha"] \arrow[r, "\rho"'] & \mathcal{P}_f \mathcal{L}\mathcal{G}(\mathbb{A}) \arrow[r, "\sigma"'] & \mathcal{L}\mathcal{G}(\mathbb{A})
\end{tikzcd}$$

The function α is the application of labels, and the function ω is a *valuation* determined by the choice function σ. Using the constructing of reduction operators from Sect. 23.2.3, with a formal meta-logic for the relations representing rules, one can give suitable conditions under which such a lifting is possible.

Moreover, though the best case is a fully deterministic π, this is not required since the top-path may still contain two facets and be an improvement as long as the reduction operator is, in general, more deterministic.

23.5 Computation

23.5.1 Predicate BI

As a programming language hHBI should compute some *thing* which is determined by the goal, with constraints given by the program. Therefore, following the discussion in the introduction, we increase the expressivity of the language by extending the logic with predicates and quantifiers.

The standard approach for turning a propositional logic into predicate logic begins with the introduction of a set of terms $\mathbb{T}$, typically given by a context-free grammar which has three disjoint types of symbols: variables, constants, and functions. The propositional letters are then partitioned $\mathbb{A} := \bigcup_{i<\omega} \mathbb{A}_i$ into classes of predicates/relations of different arities, such that the set of atomic formulas is given by elements $A(t_0, ..., t_i)$, where $t_0, ..., t_i$ are terms and $A \in \mathbb{A}_i$. In the model theory, the Herbrand universe is the set of all ground terms $\overline{\mathbb{T}}$ (terms not containing free variables), and the Hebrand base is the set of all atomic formulas (instead of atomic propositions).

The extra expressivity of predicate logic comes from the presence of two quantifiers: the universal quantifier $\forall$ and the existential quantifier $\exists$, which for the hereditary Harrop fragment gives the following grammar for formulas:

$$\begin{aligned} D &::= ... \mid \forall x\, A \mid \forall x(G \to A) \mid \forall x(G \mathrel{-\!\!*} A) \\ G &::= ... \mid \exists x\, G \end{aligned}$$

Formally, programs and goals are not constructed out of arbitrary formulas, but only out of *sentences*: formulas containing no free-variables. However, since in this fragment the quantifiers are restricted to different types of formulas (and the sets of variables and constants are disjoint) they may be suppressed without ambiguity. For example, the formulas $A(x)$ regarded as a goal is unambiguously existentially quantified, whereas when regarded as a definite clause it is unambiguously universally quantified.

Rules for the quantifiers require the use of a mappings from $\theta : \mathbb{T} \to \overline{\mathbb{T}}$ that are fixed on $\overline{\mathbb{T}} \subseteq \mathbb{T}$, which are uniquely determined by their assignment of variables. Such a function becomes a *substitution* under the following action:

$$\phi\theta := \begin{cases} A(\theta(t_0), ..., \theta(t_n)) & \text{if } \phi = A(t_0, ..., t_n) \\ \psi_0\theta \circ \psi_1\theta & \text{if } \phi := \psi_0 \circ \psi_1 \text{ for any } \circ \in \{\land, \lor, \to, *, \mathrel{-\!\!*}\} \\ \phi & \text{if } \phi \in \{\top, \top^*\} \end{cases}$$

The resolution system (Fig. 23.3) is thus extended with the operators in Fig. 23.6 which incorporate the quantifier rules. Observe that substitution is used to match a definite clause with the goal, and for this reason is traditionally called a *unifier*. Since execution is about finding some term (some element of the Herbrand universe) which

$P \vdash \exists x G$	$\Leftarrow$	$P \vdash G\theta$
$P \vdash A$	$\Leftarrow$	$P \equiv R; (Q, \forall x B)$ and $Q \vdash \top^*$ and $\exists \theta :\ B\theta = A\theta$
$P \vdash A$	$\Leftarrow$	$P \equiv R; (Q, \forall x(G \mathrel{-\!\!*} B))$ and $\exists \theta :\ Q \vdash G\theta$ and $A\theta = B\theta$
$P \vdash A$	$\Leftarrow$	$\begin{cases} P \equiv R; (S, (Q; \forall x(G \to B)))\text{ and } S \vdash \top^* \text{ and} \\ \exists \theta :\ Q; \forall x(G \to B) \vdash G\theta \text{ and } A\theta = B\theta \end{cases}$

Fig. 23.6 Unification rules

satisfies the goal, one may regard the thing being computed as the combined effect of the substitutions witnessed along the way, often called the *most general unifier*.

The presentation of the quantifier rules seems to have a redundancy since in each unification rule the goal A is supposed to already be ground, and thus by definition $A\theta = A$. However, in practice one may postpone commitment for existential substitution until one of the unification rules is used, yielding much better behaviour. This is already implicitly an example of the *control via algebraic constraints* phenomenon from Sect. 23.4.2.

The introduction of quantifiers into the logic programming language offered here is minimal, and much more development is possible (Armelín, 2002). An attempt at a full predicate BI is found in Pym (1999, 2023), but its metatheory is not currently adequate—see (Pym, 2023). The intuition follows from the intimate relationship between implication and quantification in intuitionistic logics such as IL and MILL. The intended reading of an implication $A \to B$ in BI is a constructive claim of the existence of a procedure which turns a proof A into a proof of B. Therefore, a proof of an existential claim $\exists x A(x)$ involves generating (or showing how to generate) an object t for which one can prove $A(t)$; similarly, a proof of a universal claim $\forall x A(x)$ is an procedure which takes any object t and yields a proof of $A(t)$ (Dummett, 1977). The presence of both additive ($\to$) and multiplicative ($\mathrel{-\!\!*}$) implications in BI results in the possibility of both additive (resp. $\{\exists, \forall\}$) and multiplicative (resp. $\{\mathfrak{Z}, \not{\mathcal{L}}\}$) quantifiers. Intuitively, the difference between $\forall$ and $\not{\mathcal{L}}$ is that for $\not{\mathcal{L}} x \phi$ makes a claim about objects x *separate* from any other term appearing in ϕ, thus it may be read *for all new*; similarly for the relationship between the $\exists$ and $\mathfrak{Z}$ quantifiers. This behaviour is similar to the freshness quantifier (И) from nominal logic (Pitts, 2003), which is the familiar universal quantifier ($\forall$) together with an exclusivity condition, and has a well understood metatheory.

23.5.2 Example: Databases

Abramsky and Väänänen (2009) have shown that BI is the natural propositional logic carried by the Hodge's compositional semantics (Hodges, 1997a, b) for the logics of informational dependence and independence by Hintikka et al. (1989; 2002; 2007), and in doing so introduced several connectives not previously studied in this setting. We develop here an example illustrating how BI's connectives indeed offer a useful, and natural, way to represent the combination of pieces of data.

Fig. 23.7 Columns of electives

Column 1	Column 2	Column 3
Al(gebra)	Lo(gic)	Da(tabases)
Pr(obability)	Ca(tegories)	Co(mpilers)
Gr(aphs)	Au(tomata)	AI

Logic programming has had a profound effect on databases both theoretically, providing a logical foundation, and practically, by extending the power to incorporate reasoning capabilities (Grant & Minker, 1992). Standard relational database systems are the fundamental information storage solution in data management, but have no reasoning abilities meaning information is either stored explicitly or is not stored at all. One may combine a database with a LP language resulting in a *deductive* database, which extends the capabilities of such relational databases to included features such as multiple file handling, concurrency, security, and inference.

A deductive database combines two components. The *extensional* part contains the atomic facts (ground atoms), and is the type of data that can exist in a relational database; meanwhile the *intensional* part contains inference rules and represents primitive reasoning abilities relative to a knowledgebase. In the case of hHLP, if there are no recursive rules in the intensional database then it corresponds to views in a relational database. However, even without recursion the two connectives of hHBI offers extra abilities as demonstrated in the following.

Suppose Unseen University offers a computer science course. To have a well-rounded education, students must select one module from each of three columns in Fig. 23.7, with the additional constraint that to complete the course students must belong to a particular stream, A or B. Stream A contains Al, Gr, Lo, Ca, Au, Co, AI, and students must pick one from each column, stream B contains the complement. This compatibility information for modules may be stored as an extensional database provided by the bunch ED = (Col1,Col2,Col3) with each Col$i$ bunch defined as follows:

$$\begin{array}{lcl} \text{Col1} & := & \text{A(Al) ; A(Gr) ; B(Pr) ; B(Gr)} \\ \text{Col2} & := & \text{A(Lo) ; A(Ca) ; A(Au) ; B(Ca) ; B(Au)} \\ \text{Col3} & := & \text{A(Co) ; A(AI) ; B(Da) ; B(Co) ; B(AI)} \end{array}$$

Let x be a list of subjects, then the logic determining CS courses for the $Astr$(eams) and $Bstr$(eams) respectively is captured by an intensional database ID given by the following bunch, where π_i is the ith projection function:

$$\begin{array}{c} Astr\big(\pi_0(x), \pi_1(x), \pi_2(x)\big) \to str(x) \quad ; \quad Bstr\big(\pi_0(x), \pi_1(x), \pi_2(x)\big) \to str(x) \quad ; \\ A(x) * A(y) * A(z) \mathrel{-\!\!*} Astr(x, y, z) \quad ; \quad B(x) * B(y) * B(z) \mathrel{-\!\!*} Bstr(x, y, z) \end{array}$$

The equivalent implementation in hHLP would require a tagging system to show compatibility of the columns; meanwhile the computation can be handled easily, and more importantly logically, in hHBI. Conversely, because hHBI is an extension of hHLP, any program in the latter is automatically executable in the former.

To find the possible combinations of subjects one performs the query $ED\ ;\ ID \vdash str(x)$—note that there are implicit quantifiers throughout. One possible execution is the following:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{Col1 \vdash A(Al) \qquad Col2 \vdash A(Lo)}{Col1, Col2 \vdash A(Al) * A(Lo)} \qquad Col3 \vdash A(AI)}{Col1, Col2, Col3 \vdash A(Al) * A(Lo) * A(AI)}\,(1)}{ED; ID \vdash Astr([Al, Lo, AI])} \qquad \begin{array}{l}\theta := [\pi_0(x) \mapsto Al \\ \quad \pi_1(x) \mapsto Lo \\ \quad \pi_2(x) \mapsto AI]\end{array}}{ED; ID \vdash str([Al, Lo, AI])}\,(2)}{ED; ID \vdash str(x)}$$

The example includes the use of two particular selection functions, on lines (1) and (2), which merit discussion. The first is the choice of decomposition of the context, for which there are six possibilities; the second is the choice of unifier θ, for which there are twenty-seven possible for each choice of definite clause. These two choices require control régimes. The problem at (1) is the context-management problem found in linear logic programming, and can be handled by encoding the *resource distribution via boolean constraints* mechanism (Pym, 1999). Meanwhile, the problem at (2) is common to all logic programming languages: the choice of definite clause. Assuming a preferential selection and backtracking régime, what remains is the choice of unifier, which at this point in the execution is no better than *guessing*. Indeed, this problem is already found in the instantiation of x as the term $[Al, Lo, AI]$!

A student at Unseen University might reason slightly differently about their course, to avoid needless backtracking. For example, they may initially simply determine from the intensional database that to belong to the A stream they need to pick three modules, thus they need to find a list x with three components $y_0 = \pi_0(x)$, $y_1 = \pi_1(x)$, $y_2 = \pi_2(x)$. After this, they realize that the each y_i must belong to a separate column, and look *simultaneously* across the possible choices, keeping track of the constraints as they accumulate. Such reasoning is presented by a reduction with algebraic side-conditions:

$$\dfrac{\dfrac{\dfrac{\vdots \qquad \vdots}{Col1, Col2, Col3 \vdash A(y_0) * A(y_1) * A(y_2)}\ y_0 = \pi_0(x), y_1 = \pi_1(x), y_2 = \pi_2(x)}{ED; ID \vdash Astr(y)}\ y = [\pi_0(x), \pi_1(x), \pi_2(x)]}{ED; ID \vdash str(x)}$$

This form of reasoning is precisely instantiating the control via algebraic constraints mechanism of Sect. 23.4. The hidden part of the computation contains the boolean constraints mechanism for resource distribution, and what is shown is the same kind of lifting used to control quantifier elimination instead. The study of hHBI shows that the meta-theory of control in proof-search is limited and fragmented, but that algebraic constraints offer not only a general approach, but a meaningful one supported by an established mathematical framework for computation.

23.6 Conclusion

We have developed a logic programming language HhBI based on goal-directed proof-search in a fragment of the logic of Bunched Implications. The analysis uses the now traditional approach of uniform proofs and selection functions. However, we begin from the general point of view of reductive logic which guides the entire construction; the object logic BI serves as an extended example where certain complication around control, a defining feature of proof-search, can be explored in more interesting detail. The resulting programming language is nonetheless interesting in itself, being particular suitable for programming systems that have constituent parts which partition the whole. Future work on the language itself includes implementation, and to extend it with further quantifiers as outlined in Sect. 23.5.1.

Though reductive reasoning is pervasive in both theory and practice of formal logic, it has no unified foundation. We have show that coalgebra is a robust framework for providing a uniform study via formally defined reduction operators; and, that such an approach coheres with contemporary mathematical approaches, such as bialgebra, to operational semantics. Developing the mathematical theory is a substantial ongoing area of research, for which the above may be regarded as a first step toward generalizing the perspective. Future work includes providing a systematic approach the meta-theory based on some general specification of an object logic, as opposed to developing the theory on a case by case basis.

Proof-search follows from reductive logic by the application of a control regime. However, the abstract study of control is currently limited and fragmented lacking even a uniform language of discourse. Once more the coalgebraic framework appears to be effective since it resolves the problem into two facets (i.e., choice of rule, and choice of instance) each of which may be understood simply as the application of a coalgebra. This allows for an abstract perspective of traditional approaches; for example, for the choice-of-rule problem it show that the right rules can be condensed into a single reduction operator without additional non-determinism allowing uniform proofs to form a foundation for logic programming.

The second control problem, choice-of-instance, is more subtle. The incremental addition of features in the base logic for an logic programming language results in drastically more complicated handling of selection functions; consider, for example, the study of context management in linear logic programming as discussed in Sect. 23.4.4. An alternate approach from the use of a backtracking schedule is to make the reduction operators deterministic by lifting to a better behaved reduction system in which the possible choices are encoded as simple algebraic equations. The coalgebraic framework gives a clear account of this works. Moreover, in Sect. 23.5.2 we show that the algebraic constraints approach mimics the action of reasoning about a problem before executing it.

Future work includes developing the meta-theory of control, with an initial step being a precise mathematical development of this lifting. More generally, we would seek to develop a bialgebraic framework able to model generically—in the sense of a 'logical framework'—not only logical structure, but also control structure, in an integrated way.

Acknowledgements We thank Pablo Armelín for the initiating the study on logic programming with the logic of Bunched Implications, and Fabio Zanasi and Sonia Marin for many valuable discussions concerning the algebraic and coalgebraic approaches we have described. This work has been partially supported by the UK's EPSRC through research grant EP/S013008/1.

References

Abramsky, S. (1993). Computational Interpretations of Linear Logic. *Theoretical Computer Science, 111*(1–2), 3–57.

Abramsky, S. (2008). Information, processes and games. *Philosophy of information, 8,* 483–549.

Abramsky, S., & Väänänen, J. (2009). From IF to BI. *Synthese, 167*(2), 207–230.

Abramsky, S., Jagadeesan, R., & Malacaria, P. (2000). Full abstraction for PCF. *Information and Computation, 163*(2), 409–470.

Andreoli, J.-M. (1992). Logic programming with focusing proofs in linear logic. *Journal of logic and Computation, 2*(3), 297–347.

Apt, K., & Emden, M. (1982). Contributions to the theory of logic programming. *Journal of the ACM,29*, 841–862.

Armelín, P. (2002). *Programming with bunched implications*. Ph.D. thesis, University College London.

Armelín, P., & Pym, D. J. (2001). Bunched logic programming. In *International joint conference on automated reasoning,* (pp. 289–304).

Bertot, Y., & Castéran, P. (2004). *Interactive theorem proving and program development*. Berlin, Heidelberg: Springer.

Bonchi, F., & Zanasi, F. (2015). Bialgebraic semantics for logic programming. *Logical Methods in Computer Science,11*(1).

Boolos, G. (1984). Don't eliminate cut! *Journal of Philosophical Logic, 13*(4), 373–378.

Brotherston, J., & Villard, J. (2015). Sub-classical boolean bunched logics and the meaning of par. In *24th EACSL annual conference on computer science logic* (pp. 325–342).

Brotherston, J. (2012). Bunched logics displayed. *Studia Logica, 100*(6), 1223–1254.

Bundy, A. (1985). *The computer modelling of mathematical reasoning*. Academic Press Professional, Inc.

Cervesato, I., Hodas, J. S., & Pfenning, F. (2000). Efficient resource management for linear logic proof search. *Theoretical Computer Science, 232*(1–2), 133–163.

Chaudhuri, K., Marin, S., & Straßburger, L. (2016). Focused and synthetic nested sequents. In *Proceedings of 19th international conference on foundations of software science and computation structures* (pp. 390–407).

Docherty, P. (2019). *Bunched logics: A uniform approach*. Ph.D. thesis, University College London.

Docherty, S., & Pym, D. (2018). A stone-type duality theorem for separation logic via its underlying bunched logics. *Electronic Notes in Theoretical Computer Science, 336,* 101–118.

Dummett, M. (1977). *Elements of intuitionism*. Logic Guides: Clarendon Press.

Galmiche, D., Marti, M., & Méry, D. (2019). Relating labelled and label-free bunched calculi in BI logic. In *Proceedings of 28th international conference on automated reasoning with anamytic tableaux and related methods* (pp. 130–146).

Galmiche, D., Méry, D., & Pym, D. (2005). The semantics of BI and resource tableaux. *Mathematical Structures in Computer Science,15*(6), 1033.

Gentzen, G. (1969). *The collected papers of Gerhard Gentzen*. North-Holland Publishing Company.

Gheorghiu, A., & Marin, S. (2021). Focused proof-search in the logic of bunched implications. In *24th international conference on foundations of software science and computation structures*.
Girard, J.-Y. (1987). Linear logic. *Theoretical Computer Science, 50*(1), 1–101.
Gordon, M. J., & Melham, T. F. (1993). *Introduction to HOL: A theorem proving environment for higher order logic*. Cambridge University Press.
Gordon, M. J., Wadsworth, C. P., & Milner, R. (1979). Edinburgh LCF. *A mechanised logic of computation*. Berlin, Heidelberg: Springer.
Grant, J., & Minker, J. (1992). The impact of logic programming on databases. *Communications of the ACM, 35*(3), 66–81.
Gupta, G., & Santos Costa, V. (1994). Optimal implementation of and-or parallel prolog. *Future Generation Computer Systems,10*(1), 71 – 92.
Gupta, G., Bansal, A., Min, R., Simon, L., & Mallya, A. (2007). Coinductive logic programming and its applications. In *Logic Programming* (Vol. 27, pp. 27–44).
Gupta, G., Pontelli, E., Ali, K. A. M., Carlsson, M., & Hermenegildo, M. V. (2001). Parallel execution of prolog programs: A survey. *ACM Transactions on Programming Languages and Systems,23*(4), 472–602.
Harland, J. (2001). An algebraic approach to proof-search in sequent calculi. In *Presented at the international joint conference on automated reasoning*.
Harland, J., & Pym, D. J. (1997). Resource-distribution via Boolean Constraints. In *Automated Deduction—CADE-14* (pp. 222–236).
Harland, J., & Pym, D. J. (2003). Resource-distribution via Boolean Constraints. *ACM Transactions on Computational Logic, 4*(1), 56–90.
Hintikka, J. (2002). Hyperclassical logic (a.k.a. IF Logic) and its implications for logical theory. *Bulletin of Symbolic Logic,8*(3), 404–423.
Hintikka, J., & Sandu, G. (1989). Informational Independence as a Semantical Phenomenon. In Fenstad, J. E., Frolov, I. T., & Hilpinen, R. (Eds.), *Logic, methodology and philosophy of science VIII*, *Studies in logic and the foundations of mathematics* (Vol. 126, pp. 571–589). Elsevier.
Hodas, J. S. (1994). *Logic programming in intutionistic linear logic: Theory, design and implementation*. Ph.D. thesis, University of Pennsylvania.
Hodas, J. S., López, P., Polakow, J., Stoilova, L., & Pimentel, E. (2002). A tag-frame system of resource management for proof search in linear-logic programming. In *International workshop on computer science logic* (pp. 167–182).
Hodas, J. S., & Miller, D. (1994). Logic programming in a fragment of intuitionistic linear logic. *Information and Computation, 110*(2), 327–365.
Hodges, W. (1997). Some Strange Quantifiers. Ehrenfeucht*Structures in logic and computer science: A selection of essays in honor of A* (pp. 51–65). Berlin, Heidelberg: Springer.
Hodges, W. (1997). Compositional semantics for a language of imperfect information. *Logic Journal of the IGPL, 5*(4), 539–563.
Ishtiaq, S. S., & O'Hearn, P. W. (2001). BI as an assertion language for mutable data structures. *ACM SIGPLAN Notices, 36*(3), 14–26.
Jacobs, B. (2016). *Introduction to coalgebra: Towards mathematics of states and observation*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press.
Knaster, B., & Tarksi, A. (1928). Un Théorème sur les Fonctions d'Ensembles. *Annales de la Societe Polonaise de Mathematique,6,* 133–134.
Komendantskaya, E., & Power, J. (2011). Coalgebraic semantics for derivations in logic programming. In *International Conference on Algebra and Coalgebra in Computer Science*, (pp. 268–282). Springer.
Komendantskaya, E., McCusker, G., & Power, J. (2011). Coalgebraic semantics for parallel derivation strategies in logic programming. In *International Conference on Algebraic Methodology and Software Technology,* (Vol. 13, pp. 111–127).
Komendantskaya, E., Power, J., & Schmidt, M. (2016). Coalgebraic logic programming: From semantics to implementation. *Journal of Logic and Computation, 26*(2), 745–783.

Kowalski, R. (1974). Predicate logic as programming language. In *IFIP congress,* (Vol. 74, pp. 569–544).

Kowalski, R. (1986). *Logic for problem-solving*. North-Holland Publishing Co.

Kowalski, R., & Kuehner, D. (1971). Linear resolution with selection function. *Artificial Intelligence,2*, 227–260.

Kowalski, R. (1979). Algorithm = Logic + Control. *Communications of the ACM, 22*(7), 424–436.

Kurokawa, H. (2009). Hypersequent calculi for intuitionistic logic with classical atoms. *Annals of Pure and Applied Logic, 161*(3), 427–446.

Lambek, J., & Scott, P. (1986). *Introduction to higher order categorical logic*. Cambridge University Press.

Larchey-Wendling, D. (2016). The formal strong completeness of partial monoidal boolean BI. *Journal of Logic and Computation, 26*(2), 605–640.

Lloyd, J. W. (1984). *Foundations of logic programming, Symbolic computation*. Springer.

López, P., & Polakow, J. (2005). Implementing efficient resource management for linear logic programming. In *International conference on logic for programming artificial intelligence and reasoning,* (Vol. 11, pp. 528–543).

Miller, D., & Nadathur, G. (2012). *Programming with higher-order logic*. Cambridge University Press.

Miller, D., & Nigam, V. (2007). Incorporating tables into proofs. In *International workshop on computer science logic* (pp. 466–480).

Miller, D. (1989). A logical analysis of modules in logic programming. *Journal of Logic Programming, 6*(1–2), 79–108.

Miller, D., & Saurin, A. (2006). A game semantics for proof search: Preliminary results. *Electronic Notes in Theoretical Computer Science, 155,* 543–563.

Miller, D., Nadathur, G., Pfenning, F., & Scedrov, A. (1991). Uniform proofs as a foundation for logic programming. *Annals of Pure and Applied Logic, 51*(1), 125–157.

Milner, R. (1984). The use of machines to assist in rigorous proof. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences,312*(1522), 411–422.

O'Hearn, P. W., & Pym, D. J. (1999). The logic of bunched implications. *The Bulletin of Symbolic Logic, 5*(2), 215–244.

Paulson, L. C. (1994). *Isabelle: A generic theorem prover*. Berlin, Heidelberg: Springer.

Paulson, L. C. (1996). *ML for the working programmer*. Cambridge University Press.

Pfenning, F., & Schürmann, C. (1999). System description: Twelf — a meta-logical framework for deductive systems. In *Automated deduction — CADE-16* (pp. 202–206).

Pitts, A. M. (2003). Nominal logic, a first order theory of names and binding. *Information and Computation, 186*(2), 165–193.

Plaisted, D. A., & Zhu, Y. (1997). *The efficiency of theorem proving strategies*. Springer.

Pym, D. J. (1990). *Proofs, Search and computation in general logic*. Ph.D. thesis, University of Edinburgh.

Pym, D. J. (1999). On Bunched Predicate Logic. In *Symposium on logic in computer science* (Vol. 14, pp. 183–192).

Pym, D. J. (2002). *The Semantics and Proof Theory of the Logic of Bunched Implications*. Netherlands, Dordrecht: Springer.

Pym, D. J. (2019). Reductive logic and proof-theoretic semantics. In *Proceedings of the third Tübingen conference on proof-theoretic semantics: assessment and future perspectives*. https://publikationen.uni-tuebingen.de/xmlui/handle/10900/93935.

Pym, D. J. (2023). Errata and Remarks for 'The Semantics and Proof Theory of the Logic of Bunched Implications'. http://www.cantab.net/users/david.pym/BI-monograph-errata.pdf.

Pym, D. J., & Harland, J. A. (1994). A uniform proof-theoretic investigation of linear logic programming. *Journal of logic and Computation,4*(2), 175–207.

Pym, D. J., & Ritter, E. (2004). *Reductive logic and proof-search: Proof theory, semantics, and control*. Oxford Logic Guides. Clarendon Press.

Pym, D. J. (2019). Resource semantics: Logic as a modelling technology. *ACM SIGLOG News, 6*(2), 5–41.

Pym, D. J., O'Hearn, P. W., & Yang, H. (2004). Possible worlds and resources: The semantics of BI. *Theoretical Computer Science, 315*(1), 257–305.

Read, S. (1988). *Relevant logic*. Basil Blackwell.

Schroeder-Heister, P. (2008). Proof-theoretic versus model-theoretic consequence. In *The logica yearbook 2007* (pp. 187–200). College Publications.

Schroeder-Heister, P. (2017). The definitional view of atomic systems in proof-theoretic semantics. In *The logica yearbook 2016* (pp. 185–200). College Publications.

Simon, L., Bansal, A., Mallya, A., & Gupta, G. (2007). Co-logic programming: Extending logic programming with coinduction. In Arge, L., Cachin, C., Jurdziński, T., & Tarlecki, A. (Eds.), *Automata, languages and programming* (pp. 472–483).

Tarski, A. (1955). A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics, 5*(2), 285–309.

The Coq Proof Assistant. (2023). Internet. https://coq.inria.fr/.

The Twelf Project. (2023). Internet. http://twelf.org/wiki/Main_Page.

Turi, D., & Plotkin, G. (1997). Towards a mathematical operational semantics. In *Proceedings of twelfth annual IEEE symposium on logic in computer science* (pp. 280–291).

Väänänen, J. (2007). *Dependence logic: A new approach to independence friendly logic*. Cambridge University Press.

Chapter 24
Lambek–Grishin Calculus: Focusing, Display and Full Polarization

Giuseppe Greco, Michael Moortgat, Valentin D. Richard, and Apostolos Tzimoulis

Abstract *Focused sequent calculi* are a refinement of sequent calculi, where additional side-conditions on the applicability of inference rules force the implementation of a proof search strategy. Focused cut-free proofs exhibit a special normal form that is used for defining identity of sequent calculi proofs. We introduce a novel focused display calculus **fD.LG** and a fully polarized algebraic semantics $\mathbb{FP}.\mathbb{LG}$ for Lambek–Grishin logic by generalizing the theory of *multi-type calculi* and their algebraic semantics with *heterogenous consequence relations*. The calculus **fD.LG** has *strong focalization* and it is *sound and complete* w.r.t. $\mathbb{FP}.\mathbb{LG}$. This completeness result is in a sense stronger than completeness with respect to standard polarized algebraic semantics, insofar as we do not need to quotient over proofs with consecutive applications of shifts over the same formula. We also show a number of additional results. **fD.LG** is sound and complete w.r.t. LG-algebras: this amounts to a semantic proof of the so-called *completeness of focusing*, given that the standard (display) sequent calculus for Lambek–Grishin logic is complete w.r.t. LG-algebras. **fD.LG** and the focused calculus **fLG** of Moortgat and Moot are equivalent with respect to proofs, indeed there is an effective translation from **fLG**-derivations to **fD.LG**-derivations and vice versa: this provides the link with operational semantics, given that every **fLG**-derivation is in a Curry–Howard correspondence with a directional $\overline{\lambda}\mu\widetilde{\mu}$-term.

G. Greco (✉) · A. Tzimoulis
Vrije Universiteit, Amsterdam, The Netherlands
e-mail: g.greco@vu.nl

A. Tzimoulis
e-mail: a.tzimoulis@vu.nl

M. Moortgat
Utrecht University, Utrecht, The Netherlands
e-mail: m.j.moortgat@uu.nl

V. D. Richard
École Normale Supérieure Paris-Saclay, Cachan, France
e-mail: valentin.richard@loria.fr

A. Palmigiano and M. Sadrzadeh (eds.), *Samson Abramsky on Logic and Structure in Computer Science and Beyond*, Outstanding Contributions to Logic 25,
https://doi.org/10.1007/978-3-031-24117-8_24

Keywords Lambek–Grishin calculus · Multi-type display calculi · Focused sequent calculi · Polarized logics · Heterogeneous algebras · Weakening relations · Semantics of proofs

2012 ACM Subject Classification Theory of computation proof theory · Algebraic semantics

24.1 Introduction

The problem of the identity of proofs is a fundamental one. It has been actively investigated in philosophy and mathematics (when do two proofs correspond to the same argument?), and in computer science (when do two algorithms correspond to the same program?). A logic can be presented by different formalisms. Sequent calculi often exhibit syntactically different proofs of the very same end-sequent. Some of these proofs differ from each other by trivial permutations of inference rules. Other formalisms, like natural deduction calculi or proof nets, are less sensitive to inference rule permutations and are usually taken as benchmarks for defining identity of proofs. *Focused sequent calculi* (Andreoli, 1992, 2001; Miller, 2004) make use of syntactic restrictions on the applicability of inference rules achieving three main goals: (i) the proof search space is considerably reduced without losing completeness, (ii) every cut-free proof comes in a special normal form, (iii) a criterion for defining identity of sequent calculi proofs. Being able to identify or tell apart two proofs has far-reaching consequences. In particular, in the tradition of parsing-as-deduction (Lambek, 1958, 1961), various logical systems—and notably various extensions of the Lambek calculus—have been proposed to recognise not only whether sentences are syntactically well-formed, but also to capture different semantic readings by 'genuinely different' proofs in the type logic (Bernardi & Moortgat, 2007; Moortgat & Moot, 2012).

In this paper, we focus on the minimal Lambek–Grishin logic and we provide a novel algebraic and proof-theoretic analysis of the focused Lambek–Grishin calculus (Moortgat & Moot, 2012). More generally, this analysis leads to the identification of a new class of display calculi and their natural algebraic semantics. The gist of the analysis is to generalise (and refine) *multi-type display calculi* (Frittella et al., 2014) and *heterogeneous algebras* (Birkhoff & Lipson, 1970) admitting not only heterogeneous operators, but also *heterogeneous consequence relations* (see Jung et al., 1999), now interpreted as *weakening relations* (Kurz et al., 2019) (i.e. a natural generalisation of partial orders). Here, we introduce a specific instance of this class tailored for the signature of the Lambek–Grishin logic and we plan to provide the full picture as future work. In particular, we plan to show that if a calculus belongs to this class, then it enjoys cut-elimination, aiming at generalizing the cut-elimination meta-theorem in the tradition of display calculi (see Wansing, 2002). Moreover, we conjecture that any *displayable logic* (Greco et al., 2016) can be equivalently presented as an instance of this class. Given that the cut-elimination provided in Sect. 24.5 is fully modular (i.e. it is preserved by adding additional structural rules 'closed under mutations': cfr. Definition 35 and conditions C_6'' and C_6'' in Definition 37) we do not expect specific problems in providing the full picture. The next paragraph summarises the main features of this analysis in general terms, without special reference to Lambek–Grishin logic.

In the case of focused sequent calculi, the distinction between *positive* versus *negative* formulas is the key ingredient for organising proofs in so-called *phases*. The distinction is proof-theoretically relevant in that it reflects a fundamental distinction between logical introduction rules: the left introduction rules for positive connectives are *invertible* while the right introduction rules are *non-invertible* in general, and vice versa for negative connectives. We observe that this distinction is also semantically grounded, indeed the main connective of a positive formula (in the original language of the logic) is a left adjoint/residual and the main connective of a negative formula (in the original language of the logic) is a right adjoint/residual. Proofs in *focalized normal form* (see Moortgat & Moot, 2012) are cut-free proofs organised in three phases: two focused phases (either positive or negative) and one non-focused phase (also called neutral phase). A focused positive (resp. negative) phase in a derivation is a proof-section (see Definition 19) where a formula is decomposed as much as possible only by means of non-invertible logical rules for positive (resp. negative) connectives. This formula and all its immediate subformulas in this proof-section are said 'in focus'. All the other rules are applied only in non-focused phases. So, each derivable sequent has at most one formula in focus. Moreover, the interaction between two focused phases is always mediated by a non-focused phase.

So-called *shift operators*—usually denoted as $\uparrow$ and $\downarrow$ (Hamano & Scott, 2007; Hamano & Takemura, 2010; Bastenhof, 2012)—are often considered to polarize a focused sequent calculus, i.e. as a tool to control the interplay between positive and negative formulas and the interaction between phases. Shifts are adjoint unary operators that change the polarity of a formula, where $\uparrow$ goes from positive to negative, $\downarrow$ goes from negative to positive, and $\uparrow \dashv \downarrow$. In this paper, we consider positive and negative formulas as formulas of different sorts.[1] We also distinguish between positive (resp. negative) *pure* formulas and positive (resp. negative) *shifted* formulas, i.e. formulas under the scope of a shift operator. So, we end up considering four different sorts, each of which is interpreted in a different sub-algebra. Therefore, in this setting shifts are heterogeneous operators, where $\uparrow$ gets split into $\uparrow$ (from positive pure formulas into negative shifted formulas) and $\upharpoonleft$ (from positive shifted formulas into negative pure formulas), $\downarrow$ gets split into $\downarrow$ (from negative pure formulas into positive shifted formulas) and $\downharpoonleft$ (from negative shifted formulas into positive pure formulas), $\uparrow \dashv \downharpoonleft$ and $\upharpoonleft \dashv \downarrow$. Moreover, the composition of two shifts is still either a closure or an interior operator (by adjunction), but we do not assume that it is an identity. We call a presentation of a logic with the features described above *full* polarization.

[1] Note in the literature on multi-type display calculi 'types' is used instead of 'sorts'.

The present investigation was largely inspired by the work of Samson Abramsky, specifically by his contributions on the Geometry of Interactions program (see Abramsky & Jagadeesan, 1994b) and his categorical and game-theoretic perspective on the semantics of linear logic. In particular, Abramsky and Jagadeesan (1994a) introduce a game-theoretic semantics for the multiplicative fragment of linear logic expanded with the so-called MIX rule, and show a strong form of completeness they call *full completeness*. This notion is inherently stronger than standard completeness (that is with respect to provability), indeed it requires "a semantic characterization of the space of proofs of a given logic". More precisely, given a logic **L** and the appropriate categorical model $\mathcal{M}$ where formulas are interpreted as objects and proofs as morphisms, $\mathcal{M}$ is fully complete for $\mathcal{L}$ if every morphism $[\![\pi]\!] : [\![A]\!] \longrightarrow [\![B]\!]$ in $\mathcal{M}$ is the denotation of some proof π of $A \vdash B$ in **L**. Abramsky and Melliès (1999) extend the full completeness result to the multiplicative-additive fragment of linear logic with respect to a new concurrent form of games semantics.

First of all, let us emphasize some differences between the present approach and the two papers cited above. Leaving aside that in this paper we focus on the Lambek–Grishin logic, notice that in Abramsky and Jagadeesan (1994a) and Abramsky and Melliès (1999) the identity of proofs is addressed via proof nets, where here we work with focused sequent calculi. We do not consider this an essential difference from a conceptual point of view. Most importantly, the notion of full completeness makes sense, strictly speaking, only for a categorical semantics. What we do here is to take seriously the idea that different semantics, even algebraic semantics, could be more tightly or less tightly connected to a logic, in the sense that they reflect more closely or less closely the structure of proofs. As already mentioned above, we observe that in polarized algebraic semantics the composition of two shifts is an identity: this means that sequences of shifts could occur in well-formed sequents and their interpretation makes perfect sense. On the other hand, shifts are witnesses of phase transitions, so sequents admitting sequences of two or more shifts do not necessarily have a focused proof. The usual solution adopted in the literature is to consider only sequents where $\uparrow$ (resp. $\downarrow$) does not immediately occur under the scope of $\downarrow$ (resp. $\uparrow$). Only those sequents certainly have a focused proof, if derivable. This is of course without loss of generality, given that $\uparrow\downarrow\uparrow=\uparrow$, $\downarrow\uparrow\downarrow=\downarrow$, $\uparrow\downarrow \varphi = \varphi$ and $\downarrow\uparrow \varphi = \varphi$ (and provided a proof of the completeness of focusing). Here we consider sequents as syntactical objects (not as equivalence classes) and we internalize the previous restriction thanks to a more expressive language (pure and shifted formulas are of different types) coupled with an opportune design of the calculus (arbitrary derivable well-formed sequents have a focused proof: no need to confine to a subclass of well-formed sequents). This refinement is also reflected in fully polarized algebras, given that different types are interpreted in different sub-algebras. The proof-theoretically relevant distinction between non-invertible logical rules versus invertible logical rules (plus structural rules) is kept both at the syntactical level (language plus calculus) and at the semantical level. In particular, the fact that only logical non-invertible rules can be applied in focused phases is no longer just a syntactical constraint but it is reflected in the algebraic interpretation, namely no other rule is validated in the relevant sub-algebras. Analogously, the fact that structural rules can be applied only in neutral phases is reflected in the algebraic interpretation. In this sense, we may argue that polarized algebras are more tightly connected to focused calculi than standard algebraic semantics, and even more so fully polarized algebras. Nonetheless, the distinction we can directly capture is still coarser than in the case of operational or categorical semantics: we can tell apart focused versus non-focused proofs, but, in general, we cannot always tell apart two different focused proofs of the very same end-sequent (to do so explicitly, we still need to couple the calculus with directional $\overline{\lambda}\mu\widetilde{\mu}$-terms).

We observe that the notion of a weakening relation (see Sect. 24.2.1), the key ingredient for defining fully polarized algebras, has a natural categorical presentation (see Kurz et al., 2019) and we plan to provide a categorical presentation of the present approach in future work, where the notion of full completeness can be rigorously defined.

The paper is structured as follows. In Sect. 24.2 we introduce fully polarized LG-algebras $\mathbb{FP}.\mathbb{LG}$. In Sect. 24.3 we introduce the focused display calculus **fD.LG** for the minimal Lambek–Grishin logic and we prove that it has the strong focalization property. In Sect. 24.4, we show that the calculus **fD.LG** is sound and complete w.r.t. $\mathbb{FP}.\mathbb{LG}$ and LG-algebras, and in Sect. 24.5 that **fD.LG** has canonical cut-elimination. Section 24.6 provides the effective translation between derivations of the calculi **fD.LG** and **fLG**.

24.2 Algebraic Semantics

In this section we first recall the definition of Lambek–Grishin algebras, weakening relations and their properties. Then, we define fully polarized LG-algebras.

24.2.1 Preliminaries

Lambek–Grishin Algebras

The basic Lambek–Grishin logic **LG** (Moortgat, 2009) is the pure logic of residuation in the signature that expands the (non-unital, non-associative) Lambek calculus (Lambek, 1961) with the so-called Grishin connectives (i.e. a co-tensor $\oplus$ and its residuals $\otimes$, $\oslash$). **LG** is complete w.r.t. Lambek–Grishin algebras defined below.

Definition 1 A basic Lambek–Grishin algebra $\mathbb{G} = (G, \leq, \otimes, \oplus, \backslash, \otimes, /, \oslash)$ is a partially ordered algebra endowed with six binary operations compatible with the order $\leq$. Moreover, the following residuation laws hold:

$$B \leq A \backslash C \ \text{ iff } \ A \otimes B \leq C \ \text{ iff } \ A \leq C \,/\, B \qquad C \oslash B \leq A \ \text{ iff } \ C \leq A \oplus B \ \text{ iff } \ A \obslash C \leq B \tag{24.1}$$

Weakening Relations and Collages

In this paper we use weakening relations (Jung et al., 1999; Kurz et al., 2019; Galatos & Jipsen, 2016, 2019; Bilkova et al., 2013) to interpret the heterogeneous consequence relations of the calculus introduced in Sect. 24.3.1. Weakening relations can be viewed as the order-theoretic equivalents of profunctors (Benabou, 1973) (aka distributors or bimodules), which have already been considered in models of polarized logic (Hamano & Scott, 2007). In particular, partial orders are weakening relations where $\mathcal{A} = \mathcal{B}$ and $\leq_{\mathcal{A}} = \leq_{\mathcal{B}}$. We use $\preccurlyeq \subseteq \mathcal{A} \times \mathcal{B}$, $\preccurlyeq^{\mathcal{B}}_{\mathcal{A}}$ and $\mathcal{A} \nrightarrow \mathcal{B}$ interchangeably to denote a weakening relation with source $\mathcal{A}$ and target $\mathcal{B}$, and $\preccurlyeq_{\mathcal{A}}$ as an abbreviation for $\preccurlyeq^{\mathcal{A}'}_{\mathcal{A}}$. Given two relations R and S, we use RS to denote composition of relations.

Definition 2 A **weakening relation** is a relation $\preccurlyeq \subseteq \mathcal{A} \times \mathcal{B}$ on two partially ordered set $(\mathcal{A}, \leq_{\mathcal{A}})$ and $(\mathcal{B}, \leq_{\mathcal{B}})$ that is *compatible with the orders* $\leq_{\mathcal{A}}$ and $\leq_{\mathcal{B}}$ in the following sense

$$\frac{A' \leq_{\mathcal{A}} A \qquad A \preccurlyeq B \qquad B \leq_{\mathcal{B}} B'}{A' \preccurlyeq B'}$$

Definition 3 Given two weakening relations $\preccurlyeq_{\mathcal{A}} \subseteq \mathcal{A} \times \mathcal{A}'$ and $\preccurlyeq_{\mathcal{B}} \subseteq \mathcal{B} \times \mathcal{B}'$, we say that the order-preserving functions $L : \mathcal{A} \to \mathcal{B}$ and $R : \mathcal{B}' \to \mathcal{A}'$ form a **heterogeneous adjoint pair** $L \dashv^{\preccurlyeq_{\mathcal{B}}}_{\preccurlyeq_{\mathcal{A}}} R$ if for every $A \in \mathcal{A}$ and $B' \in \mathcal{B}'$,

$$L(A) \preccurlyeq_{\mathcal{B}} B' \text{ iff } A \preccurlyeq_{\mathcal{A}} R(B') \qquad \begin{tikzcd} \mathcal{A}' \arrow[r, phantom, "\top"] & \mathcal{B}' \arrow[l, bend right, "R"'] \\ \mathcal{A} \arrow[u, "\preccurlyeq_{\mathcal{A}}"] \arrow[r, bend right, "L"'] & \mathcal{B} \arrow[u, "\preccurlyeq_{\mathcal{B}}"'] \end{tikzcd} \tag{24.2}$$

If $\mathcal{A}' = \mathcal{A}$, $\preccurlyeq_{\mathcal{A}} = \leq_{\mathcal{A}}$, $\mathcal{B}' = \mathcal{B}$ and $\preccurlyeq_{\mathcal{B}} = \leq_{\mathcal{B}}$, we recover the usual definition of adjunction.

Remark 4 Heterogeneous adjunctions also appear in the theory of Chu spaces. The case we present here corresponds to Chu_2 morphisms (Pratt, 1999).

Proposition 5 *If $L \dashv^{\preccurlyeq_{\mathcal{B}}}_{\preccurlyeq_{\mathcal{A}}} R$ is a heterogeneous adjunction, then it defines a weakening relation $\preccurlyeq \subseteq \mathcal{A} \times \mathcal{B}'$ by $A \preccurlyeq B'$ iff $L(A) \preccurlyeq_{\mathcal{B}} B'$, which is also equivalent to $A \preccurlyeq_{\mathcal{A}} R(B')$. We say that $\preccurlyeq$ is **the weakening relation represented by** $L \dashv^{\preccurlyeq_{\mathcal{B}}}_{\preccurlyeq_{\mathcal{A}}} R$.*

The proof only requires unfolding of the definitions (see Appendix 1).

Definition 6 If $\preccurlyeq \subseteq \mathcal{A} \times \mathcal{B}$ is a weakening relation, then the relation $\leq_{\mathcal{A} \sqcup \mathcal{B}} := \leq_{\mathcal{A}} \sqcup \preccurlyeq \sqcup \leq_{\mathcal{B}}$ defined on the disjoint union $\mathcal{A} \sqcup \mathcal{B}$ is an order. We call it the **collage order** on $\mathcal{A} \sqcup \mathcal{B}$.

The collage order $(\mathcal{A} \sqcup \mathcal{B}, \leq_{\mathcal{A}\sqcup\mathcal{B}})$ corresponds to the collage (Street, 1980) (or cograph) of $\preccurlyeq$ seen as a profunctor. We extend the $\sqcup$ notation to weakening relations.

Definition 7 If we are in the following situation:

$$\begin{tikzcd}
\mathcal{A}' \arrow[loop left, "\leq_{\mathcal{A}'}"] \arrow[r, "\cdot\preceq", "\shortmid" marking] & \mathcal{B}' \arrow[loop right, "\leq_{\mathcal{B}'}"] \\
\mathcal{A} \arrow[loop left, "\leq_{\mathcal{A}}"] \arrow[u, "\preccurlyeq_{\mathcal{A}}", "\shortmid" marking] \arrow[ur, "\preceq", "\shortmid" marking] \arrow[r, "\preceq\cdot"', "\shortmid" marking] & \mathcal{B} \arrow[loop right, "\leq_{\mathcal{B}}"] \arrow[u, "\preccurlyeq_{\mathcal{B}}"', "\shortmid" marking]
\end{tikzcd}$$

and we also have $\preccurlyeq_{\mathcal{A}}\cdot\preceq\ \subseteq \preceq$ and $\preceq\preccurlyeq_{\mathcal{B}}\ \subseteq \preceq$, then the relation $\mathring{\preceq}\ := \cdot\preceq \sqcup \preceq \sqcup \preceq\cdot$ is a weakening relation on the collage orders $\mathcal{A} \sqcup \mathcal{A}'$ and $\mathcal{B} \sqcup \mathcal{B}'$, and we call it the **collage weakening relation**.

24.2.2 *Fully Polarized LG-Algebra*

We write $\mathcal{A}^\partial$ for the order dual of $(\mathcal{A}, \leq_{\mathcal{A}})$, i.e. $A \leq_{\mathcal{A}^\partial} A'$ iff $A' \leq_{\mathcal{A}} A$. We use P, Q (resp. $\dot{P}, \dot{Q}$) for pure (resp. shifted) positive elements, i.e. elements in the poset $\mathbb{P}$ (resp. in $\dot{\mathbb{P}}$); M, N (resp. $\dot{M}, \dot{N}$) for pure (resp. shifted) negative elements, i.e. elements in the poset $\mathbb{N}$ (resp. $\dot{\mathbb{N}}$); $\mathring{P}, \mathring{Q}, \mathring{R}$ (resp. $\mathring{M}, \mathring{N}, \mathring{L}$) for general positive (resp. negative) elements, i.e. elements in the poset $\mathring{\mathbb{P}}$ (resp. $\mathring{\mathbb{N}}$). The letters A, B, C are used whenever we do not need to specify the poset.

An *order-type* over $n \in \mathbb{N}$ is an n-tuple $\varepsilon \in \{1, \partial\}^n$. For any order type ε, we let $\mathbb{A}^\varepsilon := \Pi_{i=1}^n \mathbb{A}^{\varepsilon_i}$. We use $n_h \in \mathbb{N}$ to denote the arity of a connective h. The language $\mathcal{L}_{\mathbb{FP.LG}}(\mathcal{F}, \mathcal{G})$ (from now on abbreviated as $\mathcal{L}_{\mathbb{FP.LG}}$) takes as parameters: two disjoint denumerable sets of proposition letters AtProp^+, elements of which are denoted p, q, and AtProp^-, elements of which are denoted m, n, and two disjoint sets of connectives $\mathcal{F}$ and $\mathcal{G}$:

$$\begin{gathered}
\mathcal{F} = \{\otimes, \oslash, \obslash, \uparrow, \upharpoonleft\} \\
\mathcal{G} = \{\oplus, \backslash, /, \downarrow, \downharpoonleft\} \\
\varepsilon(\otimes) = \varepsilon(\oplus) = (1, 1) \\
\varepsilon(\obslash) = \varepsilon(\backslash) = (\partial, 1) \\
\varepsilon(\oslash) = \varepsilon(/) = (1, \partial) \\
\varepsilon(\downarrow) = \varepsilon(\downharpoonleft) = \varepsilon(\uparrow) = \varepsilon(\upharpoonleft) = (1)
\end{gathered} \tag{24.3}$$

Definition 8 A fully polarized LG-algebra ($\mathbb{FP.LG}$) $\mathbb{A}$ is defined by four posets $(\mathbb{P}, \leq^+)$, $(\dot{\mathbb{P}}, \cdot\leq\cdot^+)$, $(\mathbb{N}, \leq^-)$ and $(\dot{\mathbb{N}}, \cdot\leq\cdot^-)$ together with

- Two adjunctions $\uparrow \dashv \downharpoonleft$ and $\upharpoonleft \dashv \downarrow$

$$\begin{tikzcd} \mathbb{P} \arrow[r, bend left, "\uparrow"] \arrow[r, phantom, "\perp"] & \dot{\mathbb{N}} \arrow[l, bend left, "\downharpoonleft"] \end{tikzcd} \qquad \begin{tikzcd} \dot{\mathbb{P}} \arrow[r, bend left, "\upharpoonleft"] \arrow[r, phantom, "\perp"] & \mathbb{N} \arrow[l, bend left, "\downarrow"] \end{tikzcd} \tag{24.4}$$

We use $\preceq\cdot$ for the weakening relation represented by $\uparrow \dashv \downharpoonleft$ and $\cdot\preceq$ for the weakening relation represented by $\upharpoonleft \dashv \downarrow$.

- Three weakening relations $\preceq\cdot^+ \subseteq \mathbb{P} \times \dot{\mathbb{P}}$, $\preceq \subseteq \mathbb{P} \times \mathbb{N}$ and $\cdot\preceq^- \subseteq \dot{\mathbb{N}} \times \mathbb{N}$ such that for all $P \in \mathbb{P}$ and $N \in \mathbb{N}$ we have

$$\uparrow P \cdot\preceq^- N \quad \text{iff} \quad P \preceq N \quad \text{iff} \quad P \preceq\cdot^+ \downarrow N \tag{24.5}$$

i.e. $\preceq$ is the weakening relation represented by the heterogeneous adjunction $\uparrow \dashv_{\cdot\preceq^-}^{\preceq\cdot^+} \downarrow$.

We define the collage posets $(\mathring{\mathbb{P}}, \mathring{\leq}^+) = (\mathbb{P} \sqcup \dot{\mathbb{P}}, \leq^+ \sqcup \preceq\cdot^+ \sqcup \cdot\leq\cdot^+)$, $(\mathring{\mathbb{N}}, \mathring{\leq}^-) = (\mathbb{N} \sqcup \dot{\mathbb{N}}, \leq^- \sqcup \cdot\preceq^- \sqcup \cdot\leq\cdot^-)$ and the collage weakening relation $\mathring{\preceq} = \preceq\cdot \sqcup \preceq \sqcup \cdot\preceq \subseteq \mathring{\mathbb{P}} \times \mathring{\mathbb{N}}$, summarised in Fig. 24.1.

$$\begin{tikzcd} \dot{\mathbb{P}} \arrow[loop left, "\cdot\leq\cdot^+"] \arrow[rr, "\cdot\preceq"] & & \mathbb{N} \arrow[loop right, "\leq^-"] \\ & & \\ \mathbb{P} \arrow[loop left, "\leq^+"] \arrow[uu, "\preceq\cdot^+"] \arrow[uurr, "\preceq"] \arrow[rr, "\preceq\cdot"'] & & \dot{\mathbb{N}} \arrow[loop right, "\cdot\leq\cdot^-"] \arrow[uu, "\cdot\preceq^-"'] \end{tikzcd}$$

$$\begin{tikzcd} \mathring{\mathbb{P}} \arrow[loop left, "\mathring{\leq}^+"] \arrow[rr, "\mathring{\preceq}"] & & \mathring{\mathbb{N}} \arrow[loop right, "\mathring{\leq}^-"] \end{tikzcd}$$

Fig. 24.1 Weakening relations in $\mathbb{FP.LG}$-algebras

- Six operations (that we call LG-connectives)

$$\begin{array}{lll} \otimes : \mathring{\mathbb{P}} \times \mathring{\mathbb{P}} \to \mathbb{P} & \oslash : \mathring{\mathbb{P}} \times \mathring{\mathbb{N}}^\partial \to \mathbb{P} & \obslash : \mathring{\mathbb{N}}^\partial \times \mathring{\mathbb{P}} \to \mathbb{P} \\ \oplus : \mathring{\mathbb{N}} \times \mathring{\mathbb{N}} \to \mathbb{N} & \backslash : \mathring{\mathbb{P}}^\partial \times \mathring{\mathbb{N}} \to \mathbb{N} & / : \mathring{\mathbb{N}} \times \mathring{\mathbb{P}}^\partial \to \mathbb{N} \end{array}$$

such that the following heterogeneous adjunctions hold

$$\begin{array}{ccccc} \mathring{Q} \mathring{\preceq} \mathring{P} \backslash \mathring{N} & \text{iff} & \mathring{P} \otimes \mathring{Q} \mathring{\preceq} \mathring{N} & \text{iff} & \mathring{P} \mathring{\preceq} \mathring{N} / \mathring{Q} \\ \mathring{P} \oslash \mathring{N} \mathring{\preceq} \mathring{M} & \text{iff} & \mathring{P} \mathring{\preceq} \mathring{M} \oplus \mathring{N} & \text{iff} & \mathring{M} \obslash \mathring{P} \mathring{\preceq} \mathring{N} \end{array} \tag{24.6}$$

Proposition 9 *In any* $\mathbb{FP.LG}$ *we have* $\preceq\cdot^+ \cdot\preceq \;=\; \preceq \;=\; \preceq\cdot\, \cdot\preceq^-$.

Proof We show that $\preceq\cdot^{+}\ \cdot\!\preceq\ = \preceq$. Fix $P \in \mathbb{P}$, $N \in \mathbb{N}$ and assume that $P \preceq\cdot^{+} \dot{Q}$ and $\dot{Q}\ \cdot\!\preceq N$ for some $\dot{Q} \in \dot{\mathbb{P}}$. From $\dot{Q}\ \cdot\!\preceq N$, we conclude that $\dot{Q}\ \cdot\!\preceq\cdot^{+} \downarrow\! N$, for $\cdot\!\preceq$ is the weakening relation represented by $\upharpoonleft\ \dashv\ \downarrow$ (see Proposition 5). From $P \preceq\cdot^{+} \dot{Q}$ and $\dot{Q}\ \cdot\!\preceq\cdot^{+} \downarrow\! N$ we conclude $P \preceq\cdot^{+} \downarrow\! N$, for $\preceq\cdot^{+}$ is a weakening relation compatible with the partial order $\cdot\!\preceq\cdot^{+}$. Therefore, $P \preceq N$ by (24.5).

Now, fix $P \in \mathbb{P}$, $N \in \mathbb{N}$ and assume $P \preceq N$. On the one hand, $P \preceq\cdot^{+} \downarrow\! N$ by (24.5). On the other hand, $\downarrow\! N\ \cdot\!\preceq\cdot^{+} \downarrow\! N$ gives $\downarrow\! N\ \cdot\!\preceq N$ by Proposition 5 on $\cdot\!\preceq$. The equality $\preceq\ = \preceq\cdot\ \cdot\!\preceq^{-}$ is proven in a similar way.

Remark 10 In Kurz et al. (2019), operations that are order-reversing in some coordinate are considered 'problematic', essentially because source and target of weakening relations are, in general, of different types. Remark 2.23 in Kurz et al. (2019) illustrates the concern considering negation and implication in Boolean or Heyting algebras as prototypical examples. We observe that this is an issue only insofar as we confine ourselves to homogeneous operations. In the present setting, the problem is overcome allowing heterogeneous operations. In the case of fully polarized LG-algebras, shifts and the adjoints of shifts are heterogeneous operations (see Definition 8).

24.3 Proof Theory

The focused display LG-calculus **fD.LG** to be presented below has the following distinctive features: (i) homogeneous as well as heterogeneous connectives are considered (*multi-type*), (ii) each rule is closed under uniform substitution within each type (*properness*), (iii) every principal formula is introduced in display (*visibility property* in focused phases) and every structure occurring in a neutral derivable sequent can be isolated either in precedent or, exclusively, in succedent position by means of display postulates (*display property* in neutral phases), and (iv) homogeneous as well as heterogeneous turnstiles are considered (*heterogeneous*).[2]

fD.LG improves on the focused sequent calculus **fLG** for Lambek–Grishin logic of Moortgat and Moot (2012) in a number of respects.[3] The calculus **fLG** as presented there is considered a display calculus given that all the connectives in the language are residuated in each coordinate; nonetheless, the so-called *display postulates* capturing residuation can only be applied in unfocused phases. Therefore, strictly speaking, **fLG** does not meet the definition of 'display property' (even if when restricted to derivable sequents) proposed in the literature in the case of non

[2] Notice that any *multi-type proper display calculus* (see Wansing, 2002; Frittella et al., 2014) has features i–iii (where the display property holds for every derivable sequent) but not iv. See Sect. 24.5 for a discussion of the notions of visibility and display property and the role they play in the canonical cut-elimination theorem.

[3] Moortgat and Moot (2012) also cover the extension of basic Lambek–Grishin logic with linear distributivity postulates, not considered here in the basic calculus. Nonetheless, the relevant structural rules can be added still preserving the cut-elimination result.

focused sequent calculi (see, for instance Wansing, 2002; Goré, 1998). All the connectives of the calculus **fD.LG** introduced in Sect. 24.3.1 are residuated in neutral phases, but they are not in focused phases—a feature that is reflected in the algebraic semantics $\mathbb{FP}.\mathbb{LG}$. In addition, we prove a canonical cut-elimination result for **fD.LG** in Sect. 24.5 whereas **fLG** considers only cut-free derivations.

24.3.1 Focused Display LG-Calculus

The language of **fD.LG** is the Lambek–Grishin display calculus language expanded with shift operators.

Notation 11 *Following the display calculi literature (Greco et al., 2018), we adopt a notation where structural and operational (aka logical) connectives are in a one-to-one correspondence. Moreover, we mark the structural counterpart of a connective* $\star$ *as follows:* $\hat{\star}$ *if* $\star$ *is a left-adjoint or residual,* $\check{\star}$ *if* $\star$ *is a right-adjoint or residual.*

The Lambek–Grishin structural and operational connectives are the following

Structural symbols	$\hat{\otimes}$	$\hat{\oslash}$	$\hat{\obslash}$	$\check{\oplus}$	$\check{\backslash}$	$\check{/}$
Operational symbols	$\otimes$	$\oslash$	$\obslash$	$\oplus$	$\backslash$	$/$

Below we list the structural and operational shift operators. We consider it more convenient not to include the operational adjoints of shifts (in grey cells) in the language of **fD.LG**

Structural symbols	$\check{\downharpoonleft}$	$\check{\downarrow}$	$\hat{\upharpoonleft}$	$\hat{\uparrow}$
Operational symbols	$\downharpoonleft$	$\downarrow$	$\upharpoonleft$	$\uparrow$

For any connective h (either structural or operational), the arity n_h, order-type $\epsilon(h)$, and its classification as $\mathcal{F}$-connective or, exclusively, $\mathcal{G}$-connective are like in (24.3).

Notation 12 *We adopt the following notational convention for formulas and structures:* $\mathring{P} \in \{P, \dot{P}\}$, $\mathring{X} \in \{X, \dot{X}\}$, $\mathring{N} \in \{N, \dot{N}\}$, $\mathring{\Delta} \in \{\Delta, \dot{\Delta}\}$. *For instance, accordingly to this convention, we have that* $\mathring{P} \otimes \mathring{Q} \in \{P \otimes Q, \dot{P} \otimes Q, P \otimes \dot{Q}, \dot{P} \otimes \dot{Q}\}$. *Therefore, general formulas and structures are not a full-fledged sort, but rather an abbreviation.*

The calculus **fD.LG** manipulates formulas and structures defined by the following mutual recursion, where $p \in \mathsf{AtProp}^+$ and $n \in \mathsf{AtProp}^-$:

$$
\begin{array}{rcll}
\mathsf{PurePosFm} \ni P & ::= & p \mid \mathring{P} \otimes \mathring{P} \mid \mathring{P} \oslash \mathring{N} \mid \mathring{N} \obslash \mathring{P} & \text{Pure positive formulas} \\
\mathsf{PureNegFm} \ni N & ::= & n \mid \mathring{N} \oplus \mathring{N} \mid \mathring{P} \backslash \mathring{N} \mid \mathring{N} / \mathring{P} & \text{Pure negative formulas} \\
\mathsf{ShiftedPosFm} \ni \dot{P} & ::= & \downarrow N & \text{Shifted positive formulas} \\
\mathsf{ShiftedNegFm} \ni \dot{N} & ::= & \uparrow P & \text{Shifted negative formulas} \\
\mathsf{GenPosFm} \ni \mathring{P} & ::= & P \mid \dot{P} & \text{General positive formulas} \\
\mathsf{GenNegFm} \ni \mathring{N} & ::= & N \mid \dot{N} & \text{General negative formulas} \\
\mathsf{PurPosStr} \ni X & ::= & P \mid \check{\downharpoonleft} \dot{\Delta} \mid \mathring{X} \hat{\otimes} \mathring{X} \mid \mathring{X} \hat{\oslash} \mathring{\Delta} \mid \mathring{\Delta} \hat{\obslash} \mathring{X} & \text{Pure positive structures} \\
\mathsf{PurNegStr} \ni \Delta & ::= & N \mid \hat{\upharpoonleft} \dot{X} \mid \mathring{\Delta} \check{\oplus} \mathring{\Delta} \mid \mathring{X} \check{\backslash} \mathring{\Delta} \mid \mathring{\Delta} \check{/} \mathring{X} & \text{Pure negative structures} \\
\mathsf{ShiftedPosStr} \ni \dot{X} & ::= & \dot{P} \mid \check{\downarrow} \Delta & \text{Shifted positive structures} \\
\mathsf{ShiftedNegStr} \ni \dot{\Delta} & ::= & \dot{N} \mid \hat{\uparrow} X & \text{Shifted negative structures} \\
\mathsf{GenPosStr} \ni \mathring{X} & ::= & X \mid \dot{X} & \text{General positive structures} \\
\mathsf{GenNegStr} \ni \mathring{\Delta} & ::= & \Delta \mid \dot{\Delta} & \text{General negative structures}
\end{array}
$$

The well-formed sequents are the following (notice that sequents in grey cells are not derivable, cfr. Proposition 14):

Positive sequents	$X \vdash^{+} Y$	$\dot{X} \cdot\!\vdash^{+} Y$	$X \vdash\!\cdot^{+} \dot{Y}$	$\dot{X} \cdot\!\vdash\!\cdot^{+} \dot{Y}$
Negative sequents	$\Delta \vdash^{-} \Gamma$	$\dot{\Delta} \cdot\!\vdash^{-} \Gamma$	$\Delta \vdash\!\cdot^{-} \dot{\Gamma}$	$\dot{\Delta} \cdot\!\vdash\!\cdot^{-} \dot{\Gamma}$
Neutral sequents	$X \vdash \Delta$	$\dot{X} \cdot\!\vdash \Delta$	$X \vdash\!\cdot \dot{\Delta}$	$\dot{X} \cdot\!\vdash\!\cdot \dot{\Delta}$

(24.7)

Notation 13 *We extend the previous conventions to (derivable) sequents as follows:* $\overset{\circ}{\vdash}^{+} \in \{\vdash^{+}, \vdash\!\cdot^{+}, \cdot\!\vdash\!\cdot^{+}\}$, $\overset{\circ}{\vdash}^{-} \in \{\vdash^{-}, \cdot\!\vdash^{-}, \cdot\!\vdash\!\cdot^{-}\}$, $\overset{\circ}{\vdash} \in \{\vdash, \cdot\!\vdash, \vdash\!\cdot\}$. *The reading is supposed to preserve well-formedness. For instance, in a premise of a binary logical rule* $\overset{\circ}{\vdash}^{+} = \vdash^{+}$ *iff* $\mathring{X} = X$ *and* $\mathring{Y} = Y$, *or* $\overset{\circ}{\vdash}^{+} = \vdash\!\cdot^{+}$ *iff* $\mathring{X} = X$ *and* $\mathring{Y} = \dot{Y}$, *and so on. So, for instance, each binary logical rule below denotes nine different rules and each unary logical rule denotes two different rules. Nonetheless, notice that in any actual derivation the instantiation of a logical inference rule is unique and completely deterministic.*

The calculus **fD.LG** consists of the following rules.

Axioms and Cuts

$$
\frac{}{p \vdash^{+} p}\ p\text{-Id} \qquad n\text{-Id}\ \frac{}{n \vdash^{-} n}
$$

$$
\text{P-Cut}\ \frac{\mathring{X} \overset{\circ}{\vdash}^{+} \mathring{P} \qquad \mathring{P} \overset{\circ}{\vdash}^{+} \mathring{Y}}{\mathring{X} \overset{\circ}{\vdash}^{+} \mathring{Y}} \qquad \frac{\mathring{\Gamma} \overset{\circ}{\vdash}^{-} \mathring{N} \qquad \mathring{N} \overset{\circ}{\vdash}^{-} \mathring{\Delta}}{\mathring{\Gamma} \overset{\circ}{\vdash}^{-} \mathring{\Delta}}\ \text{N-Cut} \tag{24.8}
$$

$$
\text{Pn-Cut}\ \frac{\mathring{X} \overset{\circ}{\vdash}^{+} \mathring{P} \qquad \mathring{P} \overset{\circ}{\vdash} \mathring{\Delta}}{\mathring{X} \overset{\circ}{\vdash} \mathring{\Delta}} \qquad \frac{\mathring{X} \overset{\circ}{\vdash} \mathring{N} \qquad \mathring{N} \overset{\circ}{\vdash}^{-} \mathring{\Delta}}{\mathring{X} \overset{\circ}{\vdash} \mathring{\Delta}}\ \text{nN-Cut}
$$

Logical Rules

The logical rules transforming a structural connective in the premise into its logical counterpart in the conclusion are called *translation rules*. All the other logical rules are called *tonicity rules*. In the literature on focused calculi, 'asynchronous' and 'synchronous', respectively, are often used (e.g. in Andreoli, 2001).

$$
\otimes_L\ \frac{\mathring{P}\,\hat{\otimes}\,\mathring{Q} \vdash \mathring{\Delta}}{\mathring{P} \otimes \mathring{Q} \vdash \mathring{\Delta}}
\quad
\frac{\mathring{X} \overset{\circ}{\vdash}{}^{+} \mathring{P} \qquad \mathring{Y} \overset{\circ}{\vdash}{}^{+} \mathring{Q}}{\mathring{X}\,\hat{\otimes}\,\mathring{Y} \vdash^{+} \mathring{P} \otimes \mathring{Q}}\ \otimes_R
\quad
\oplus_L\ \frac{\mathring{N} \overset{\circ}{\vdash}{}^{-} \mathring{\Gamma} \qquad \mathring{M} \overset{\circ}{\vdash}{}^{-} \mathring{\Delta}}{\mathring{N} \oplus \mathring{M} \vdash^{-} \mathring{\Gamma}\,\check{\oplus}\,\mathring{\Delta}}
\quad
\frac{\mathring{X} \vdash \mathring{N}\,\check{\oplus}\,\mathring{M}}{\mathring{X} \vdash \mathring{N} \oplus \mathring{M}}\ \oplus_R
$$

$$
\oslash_L\ \frac{\mathring{P}\,\hat{\oslash}\,\mathring{N} \vdash \mathring{\Delta}}{\mathring{P} \oslash \mathring{N} \vdash \mathring{\Delta}}
\quad
\frac{\mathring{X} \overset{\circ}{\vdash}{}^{+} \mathring{P} \qquad \mathring{N} \overset{\circ}{\vdash}{}^{-} \mathring{\Delta}}{\mathring{X}\,\hat{\oslash}\,\mathring{\Delta} \vdash^{+} \mathring{P} \oslash \mathring{N}}\ \oslash_R
\quad
\backslash_L\ \frac{\mathring{X} \overset{\circ}{\vdash}{}^{+} \mathring{P} \qquad \mathring{N} \overset{\circ}{\vdash}{}^{-} \mathring{\Delta}}{\mathring{P} \backslash \mathring{N} \vdash^{-} X\,\check{\backslash}\,\Delta}
\quad
\frac{\mathring{X} \vdash \mathring{P}\,\check{\backslash}\,\mathring{N}}{\mathring{X} \vdash \mathring{P} \backslash \mathring{N}}\ \backslash_R
$$

$$
\obslash_L\ \frac{\mathring{N}\,\hat{\obslash}\,\mathring{P} \vdash \mathring{\Delta}}{\mathring{N} \obslash \mathring{P} \vdash \mathring{\Delta}}
\quad
\frac{\mathring{N} \overset{\circ}{\vdash}{}^{-} \mathring{\Delta} \qquad \mathring{X} \overset{\circ}{\vdash}{}^{+} \mathring{P}}{\mathring{\Delta}\,\hat{\obslash}\,\mathring{X} \vdash^{+} \mathring{N} \obslash \mathring{P}}\ \obslash_R
\quad
/_L\ \frac{\mathring{N} \overset{\circ}{\vdash}{}^{-} \mathring{\Delta} \qquad \mathring{X} \overset{\circ}{\vdash}{}^{+} \mathring{P}}{\mathring{N} / \mathring{P} \vdash^{-} \mathring{\Delta}\,\check{/}\,\mathring{X}}
\quad
\frac{\mathring{X} \vdash \mathring{N}\,\check{/}\,\mathring{P}}{\mathring{X} \vdash \mathring{N} / \mathring{P}}\ /_R
$$

$$
\downarrow_L\ \frac{N \vdash^{-} \Delta}{\downarrow N \cdot\!\vdash\!\cdot^{+} \check{\downharpoonleft}\,\Delta}
\quad
\frac{\mathring{X} \overset{\circ}{\vdash}{}^{+} \check{\downharpoonleft}\,N}{\mathring{X} \overset{\circ}{\vdash}{}^{+} \downarrow N}\ \downarrow_R
\quad
\uparrow_L\ \frac{\hat{\upharpoonleft}\,P \overset{\circ}{\vdash}{}^{-} \mathring{\Delta}}{\uparrow P \overset{\circ}{\vdash}{}^{-} \mathring{\Delta}}
\quad
\frac{X \vdash^{+} P}{\hat{\upharpoonleft}\,X \cdot\!\vdash\!\cdot^{-} \uparrow P}\ \uparrow_R
\tag{24.9}
$$

Display postulates

Below we use a double inference line to denote two rules: (i) from the premise to the conclusion and (ii) from the conclusion to the premise. We use the same name for both rules.

$$
\begin{array}{rc}
 & \mathring{Y} \overset{\circ}{\vdash} \mathring{X}\,\check{\backslash}\,\mathring{\Delta} \\
\hat{\otimes} \dashv \check{\backslash} & \hline\hline \\
 & \mathring{X}\,\hat{\otimes}\,\mathring{Y} \overset{\circ}{\vdash} \mathring{\Delta} \\
\hat{\otimes} \dashv \check{/} & \hline\hline \\
 & \mathring{X} \overset{\circ}{\vdash} \mathring{\Delta}\,\check{/}\,\mathring{Y}
\end{array}
\qquad\qquad
\begin{array}{cl}
\mathring{X}\,\hat{\oslash}\,\mathring{\Delta} \overset{\circ}{\vdash} \mathring{\Gamma} & \\
\hline\hline & \hat{\oslash} \dashv \check{\oplus} \\
\mathring{X} \overset{\circ}{\vdash} \mathring{\Gamma}\,\check{\oplus}\,\mathring{\Delta} & \\
\hline\hline & \hat{\obslash} \dashv \check{\oplus} \\
\mathring{\Gamma}\,\hat{\obslash}\,\mathring{X} \overset{\circ}{\vdash} \mathring{\Delta} &
\end{array}
\tag{24.10}
$$

$$
\begin{array}{c}
\hat{\upharpoonleft}\,X \cdot\!\vdash\!\cdot^{-} \dot{\Delta} \\
\hline\hline
X \vdash^{+} \check{\downharpoonleft}\,\dot{\Delta}
\end{array}\ \hat{\upharpoonleft} \dashv \check{\downharpoonleft}
\qquad
\begin{array}{c}
\hat{\upharpoonleft}\,X \cdot\!\vdash^{-} \Delta \\
\hline\hline
X \vdash\!\cdot^{+} \check{\downharpoonleft}\,\Delta
\end{array}\ \hat{\upharpoonleft} \dashv \check{\downharpoonleft}
\qquad
\hat{\upharpoonleft} \dashv \check{\downharpoonleft}\
\begin{array}{c}
\dot{X} \cdot\!\vdash\!\cdot^{+} \check{\downharpoonleft}\,\Delta \\
\hline\hline
\hat{\upharpoonleft}\,\dot{X} \vdash^{-} \Delta
\end{array}
$$

Structural Rules

$$
\begin{array}{c}
\mathring{X} \overset{\circ}{\vdash} \Delta \\
\hline\hline
\mathring{X} \overset{\circ}{\vdash}{}^{+} \check{\downharpoonleft}\,\Delta
\end{array}\ \check{\downharpoonleft}\ \hat{\upharpoonleft}\
\begin{array}{c}
X \overset{\circ}{\vdash} \mathring{\Delta} \\
\hline\hline
\hat{\upharpoonleft}\,X \overset{\circ}{\vdash}{}^{-} \mathring{\Delta}
\end{array}
\tag{24.11}
$$

Proposition 14 *Sequents of the form $\dot{X} \cdot\!\vdash^{+} Y$, $\Delta \vdash\!\cdot^{-} \dot{\Gamma}$ and $\dot{X} \cdot\!\vdash\!\cdot\ \dot{\Delta}$ are not derivable.*

Proof By quick induction on the derivation and examination of every rule with the following induction hypothesis: "In a sequent S, if $\dot{X}$ (resp. $\dot{\Delta}$) occurs in S in precedent (resp. succedent) position, then put in display, the succedent (resp. precedent) is either pure negative or shifted positive (resp. pure positive or shifted negative)." The IH works because LG connectives are pure and because it holds for the conclusion of rules involving shifts.

24.3.2 Focalization

In this subsection we first provide a procedural description and a formal definition of *strongly focused proof* of an arbitrary sequent calculus (Definition 20, adapted from Laurent, 2004, Definition 3). Then, we show that **fD.LG** has strong focalization (Theorem 23). In the end we provide some nomenclature and a diagrammatic representation of the 'topology of rules' of **fD.LG**. We use Ψ, Φ to refer to arbitrary structures.

The backward-looking proof search strategy implemented by a focused sequent calculus (see for instance Andreoli, 2001) can be roughly described as follows: (i) pick a formula, (ii) decompose the chosen formula as much as possible via applications of non-invertible logical rules, (iii) once you reach a subformula of the opposite polarity or an atom, then you may apply structural rules or invertible logical rules, (iv) repeat the process. In order to make this informal procedural description precise, we use a couple of preliminary definitions (see for instance Greco, 2018).

Definition 15 (*Signed generation tree*) The positive (resp. negative) generation tree of a structure Ψ, denoted $+\Psi$ (resp. $-\Psi$), is defined by labelling the root node of the generation tree of Ψ with the sign $+$ (resp. $-$), and then propagating the labelling on each remaining node as follows:

> For any node labelled with $h \in \mathcal{F} \cup \mathcal{G}$ of arity $n_h \geq 1$, and for any $1 \leq i \leq n_h$, assign the same (resp. the opposite) sign to its ith child node if the order-type $\epsilon(h, i) = 1$ (resp. if $\epsilon(h, i) = \partial$).

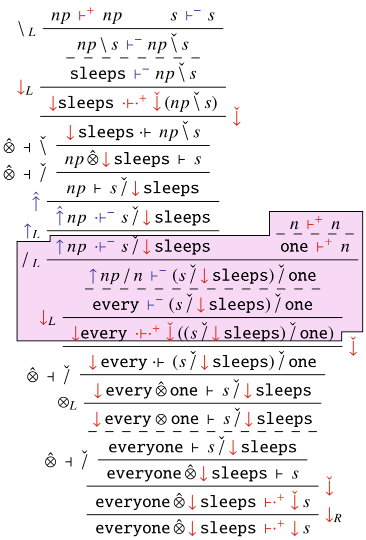
The signed generation tree of a sequent $\Psi\ t\ \Phi$ consists of the signed generation trees $+\Psi$ and $-\Phi$.

Definition 16 (*Skeleton and PIA*) A node in a signed generation tree of a sequent is called skeleton if it is labelled with $+f$ for some $f \in \mathcal{F}$ or with $-g$ for some $g \in \mathcal{G}$. Otherwise, it is called a PIA node.

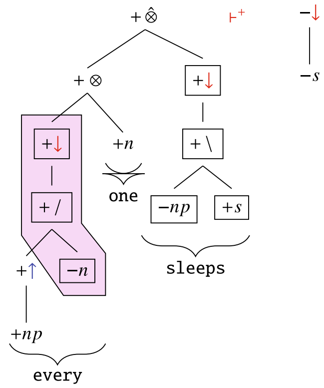
Definition 17 A PIA (resp. skeleton) subtree of a formula A occurring in a sequent is a connected subgraph of the generation tree of A having all nodes being PIA (resp. skeleton) nodes. This subtree is called maximal if it is maximal for inclusion.

In the tradition of parsing-as-deduction (Lambek, 1958, 1961), various extensions of the Lambek calculus have been proposed to recognize whether sentences are syntactically well-formed. See an example derivation in Fig. 24.2a. Here, n stands for *noun* and is a positive atomic formula. Notice that any signed generation tree of a well-formed structure Ψ in the language of **fD.LG** can be partitioned into maximal skeleton subtrees and maximal PIA subtrees.

Definition 18 (*Transition node*) A transition node of a signed generation tree σ is the uppermost node of a maximal skeleton or PIA subtree excluding the root of σ.



(a) Derivation of *Everyone sleeps* in fD.LG. The purple area is the minimal proof-section including all the introduction rules used to build the PIA subtree of everyone showed in figure 2b.



(b) Signed generation tree of the end-sequent in figure 2a. PIA nodes are encapsulated in a rectangle and Skeleton nodes are not. The purple area is the maximal PIA subtree of everyone.

Fig. 24.2 Example of derivation and signed generation tree. Here n is a positive atomic formula

Definition 19 (*Proof-section*) A proof-section π' of a proof-tree π is a connected subgraph of π, such that for every node $S \in \pi'$, if S is not a leaf of π' and it is introduced by a rule application R, then also the premise(s) of R are in π'.

Definition 20 (*Strong focalization*) A sequent proof π is *strongly focalized* if cut-free and, for every formula A occurring in π, every PIA subtree of A is constructed by a proof-section of π containing only tonicity rules.

Proposition 21 *Let h be an operational connective occurring in the generation tree of the end-sequent* Ψ t Φ *in a* ***fD.LG****-proof* π, *and let S be the uppermost sequent in* π *where h occurs. If h is a skeleton node, then it is introduced in S via a translation rule. If h is a PIA node, then it is introduced in S via a tonicity rule.*

Proof Immediate by inspection of the rules of **fD.LG**.

Proposition 22 *Let A be a formula. If a shift labels a node v of the signed generation tree of A, then either v is a transition node or it is the root of A.*

Proof This is due to the presence of shift operators and the polarization of the calculus: For every LG structural connective $\star \in \mathcal{F}$ (resp. $\star \in \mathcal{G}$), (i) the target sort of $\star$ is positive (resp. negative), and (ii) the source sort of the ith argument of $\star$ is positive (resp. negative) iff $\epsilon(\star, i) = 1$.

We can now state the strong focalization property tailored to **fD.LG**.

Theorem 23 *Every cut-free proof in* ***fD.LG*** *is strongly focalized.*

Proof Fix a cut-free **fD.LG**-proof π, a formula A occurring in a sequent of π, and a PIA subtree Σ of A. We prove by induction on Σ that for every subtree Σ' of Σ which is closed by descendent, the subgraph of π formed by the rules introducing the connectives of Σ' is a proof-section of π of end-sequent S, and if $\Sigma' \neq \Sigma$ then S is of the form $(*) : X \vdash^{+} P$ or $N \vdash^{-} \Delta$.

Call h the root of Σ' and R the rule introducing h in π. We decompose $\Sigma' = h(\Sigma_1, \ldots, \Sigma_n)$, with $n \in \{1, 2\}$ (h is a shift or LG connective) and Σ_i a subtree closed by descendent. As h is a PIA node, R is a tonicity rule by Proposition 21.

Case (a): If Σ_i is empty, we let π_i be the tree consisting of the ith premise S_i of R. As S_i is derivable, π_i is a proof-section.

Case (b): If Σ_i is non-empty, we apply the induction hypothesis on Σ_i, yielding a proof-section π_i of π containing only tonicity rules and of end-sequent S_i of the form $(*)$.

Take π' the subgraph made of π_1,..., π_n and the conclusion S of R. In case (a), S is connected to π_i by construction. In case (b), by looking at the rules, the only rules applicable on focused sequents (i.e. sequents of the form $(*)$) are tonicity rules, introducing an operational connective. Therefore, the only possibility is that the rule after π_i is R, so S is connected to π_i. Therefore, π' is a proof-section containing only tonicity rules and introducing all connectives of Σ'.

If $\Sigma \neq \Sigma'$, h is not the root of Σ. Therefore, h is not a transition node and not the root of A, so h is not a shift by Proposition 22. Therefore, S is also of the form $(*)$.

Proposition 24 *Every PIA subtree of a formula occurring in a* ***fD.LG****-sequent contains at least one LG-connective.*

Proof This is due to the fullness of the polarization, i.e. the sort of shifts. The target of $\uparrow$ and $\downarrow$ is shifted but their source sort is pure, i.e. their argument must begin by a LG formula or an atom. In other words, composing $\uparrow$ and $\downarrow$ is impossible.

Proposition 24 forces the focused sections to be uninterrupted from the point of view of LG connectives. In a forward-looking derivation, it is then impossible to defocus a formula A, and then refocus on A. Therefore, translating a **fD.LG** derivation to **fLG** by removing shift rules would preserve strong focalization.

Now we provide the definition of phases and phase transitions tailored to **fD.LG**.

Definition 25 (*Phases and phase transitions*) Let π be a cut-free proof in **fD.LG**. A sequent S occurring in π is *focused* (aka S is in a focused phase of π) if it is positive ($S = \mathring{X} \mathrel{\mathring{\vdash}^{+}} \mathring{Y}$) or negative ($S = \mathring{\Delta} \mathrel{\mathring{\vdash}^{-}} \mathring{\Gamma}$) and no structural shift occurs in S (namely, $\check{\downarrow}$, $\check{\downharpoonleft}$, $\hat{\uparrow}$, $\hat{\upharpoonleft}$). Any other sequent S' occurring in π is *non-focused* (aka S' is in a non-focused phase of π).

A *phase transition* in π is a proof-section π' of π such that the LG-connectives tonicity rules are not applied in π' and its initial-sequent is focused (resp. non-focused) iff its end-sequent is non-focused (resp. focused). A phase transition where the initial-sequent is focused is called *defocusing*, and *focusing* otherwise.

By design of **fD.LG**, the application of a shift logical rule is needed to move from a focused to a non-focused phase (resp. from a non-focused to a focused phase). Therefore, we may say that principal shifted formulas are the gate-keepers of phase transitions. Because **fD.LG** enjoys the subformula property, any formula introduced in a cut-free proof π, and in particular shifted formulas, will also occur in the conclusion of π. Therefore, we may say that shifted formulas are *witnesses* of the relevant proof structure of π. Therefore, we find it useful to introduce the following terminology:

Definition 26 (*Entry-point and exit-point*) The principal formula introduced by $\downarrow_L$ (resp. $\uparrow_R$) is called the positive (resp. negative) *entry-point* of the induced phase transition. The principal formula introduced by $\downarrow_R$ (resp. $\uparrow_L$) is called the positive (resp. negative) *exit-point* of the induced phase transition.

We now provide a diagrammatic representation to visualise the topology of rules and phase transitions tailored to **fD.LG** and, ultimately, make apparent the strong focalization property.

The diagram in Fig. 24.3 depicts the phase transition flow chart of **fD.LG**. The white area contains the generic form of focused sequents, where each of them is either positive or negative (see Definition 25). The grey and yellow areas contain the generic form of non-focused sequents (see Definition 25), where all neutral sequent seat inside the yellow area.[4] We use metavariables S, S', S'' for sequents and arrows to depict rules. Arrows $\xrightarrow{R}$ and $\circ\!\xrightarrow{R}$ points towards the conclusion of the rule R. In particular, $\xrightarrow{R} S$ represents a zeroary rule R (i.e. an axiom) and $S' \xrightarrow{R} S$ represents a unary rule R (here shift logical rules). A double-headed arrow $\leftrightarrow R$ represents an invertible rule (here structural rules introducing or eliminating a shift). $\multimap$ are 'teleporters' where the configuration $S' \multimap$ and $S'' \multimap$ together with $\circ\!\!\rightarrow S$ represents a binary rule with premises S', S'' and conclusion S (i.e. tonicity rules).[5] To exemplify the conventions involving teleporters, let us consider two configurations included in the diagram of Fig. 24.3. (i) Sequents of the form $X \vdash^+ P$ could occur as premises and conclusion of $\otimes_R$, therefore they occurs in the configuration $X \vdash^+ P \multimap$ and $X \vdash^+ P \multimap$ together with $\circ\!\!\rightarrow X \vdash^+ P$. (ii) Sequents of the form $X \vdash^+ P$ could occur as premise of $\backslash_L$ and sequents of the form $N \vdash^- \Delta$ could occur as premise and conclusion of $\backslash_L$, therefore they occur in the configuration $X \vdash^+ P \multimap$ and $N \vdash^- \Delta \multimap$ together with $\circ\!\!\rightarrow N \vdash^- \Delta$. To keep the diagram simple, we do not display tonicity rules and display postulates.

Summing up, the topology of rules is as follows: (i) the white area is closed under axioms and tonicity rules, (ii) the grey area is closed under display postulates for shifts, (iii) the yellow area is closed under any other structural rules (i.e. display postulates for LG-connectives and, whenever we consider analytic extensions of the minimal logic, all the relevant additional structural rules) and translation rules, (iv) the boundary between white and grey areas is crossed only by (non-invertible) shift

[4] Notice that sequents of the form $\dot{X} \cdot\!\vdash^+ Y$, $\Delta \vdash\!\cdot^- \dot{\Gamma}$ and $\dot{X} \cdot\!\vdash\!\cdot \dot{\Delta}$ are not derivable (see Proposition 14) and, therefore, they are not included in the diagram.

[5] Notice that in this case we do not explicitly mention the name of the rule in the diagram.

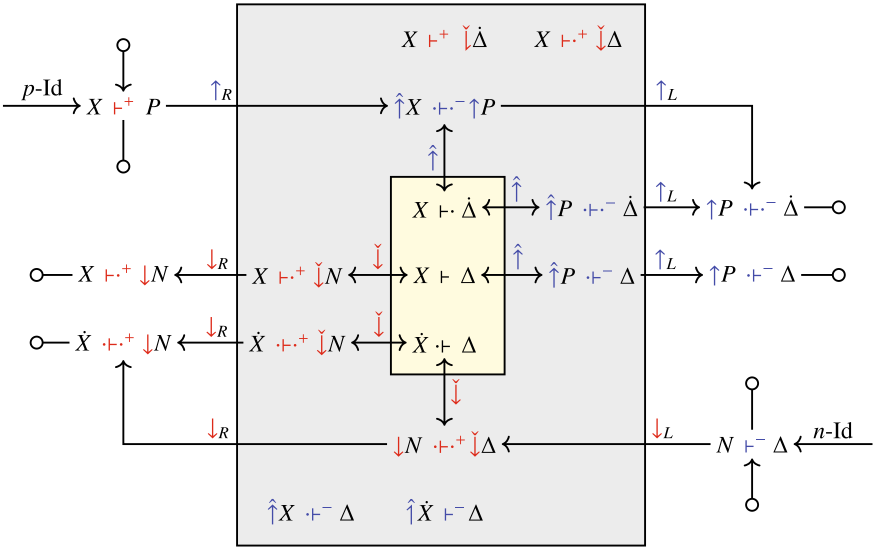


Fig. 24.3 The topology of **fD.LG**-rules and phase transitions

logical rules, and (v) the boundary between grey and yellow area is crossed only by (invertible) shift structural rules.

24.4 Completeness of Focusing

In this section first we prove that the focused calculus **fD.LG** is sound and complete w.r.t. $\mathbb{FP.LG}$-algebras. Then we prove that **fD.LG** is sound and complete w.r.t. LG-algebras: this amounts to a semantic argument showing the so-called completeness of focusing.

24.4.1 Soundness and Completeness w.r.t. $\mathbb{FP.LG}$*-Algebras*

Soundness and completeness are proven as usual in the case of algebraic semantics, where the only departure is that now we consider weakening relations instead of just orders. Soundness is stated as follows:

Theorem 27 *Each rule of the focused display calculus* ***fD.LG*** *is sound under any interpretation in a fully polarized algebra* $\mathbb{FP.LG}$.

Proof Given a $\mathbb{FP.LG}$-algebra $\mathbb{A}$ and an interpretation $(\cdot)^{\mathbb{A}}$, it is straightforward to check by induction on the complexity of proofs that for every sequent S derivable in

fD.LG, the interpretation $(S)^{\mathbb{A}}$ is valid. We leave the proof to the reader. Below we simply recall that interpretations of pure atomic formulas $p^{\mathbb{A}}$ and $n^{\mathbb{A}}$ homomorphically extend to arbitrary formulas, and each consequence relation is interpreted by a weakening relation as follows

t	$\vdash^{+}$	$\vdash\!\cdot^{+}$	$\cdot\!\vdash\!\cdot^{+}$	$\vdash^{-}$	$\cdot\!\vdash^{-}$	$\cdot\!\vdash\!\cdot^{-}$	$\vdash$	$\cdot\!\vdash$	$\vdash\!\cdot$	$\mathring{\vdash}^{+}$	$\mathring{\vdash}^{-}$	$\mathring{\vdash}$
$t^{\mathbb{A}}$	$\leq^{+}$	$\preceq\!\cdot^{+}$	$\cdot\!\leq\!\cdot^{+}$	$\leq^{-}$	$\cdot\!\preceq^{-}$	$\cdot\!\leq\!\cdot^{-}$	$\preceq$	$\cdot\!\preceq$	$\preceq\!\cdot$	$\mathring{\leq}^{+}$	$\mathring{\leq}^{-}$	$\mathring{\preceq}$

(24.12)

In order to prove completeness, we need to introduce the auxiliary notion of standard sequents.

Definition 28 The principal subtree of a structure Ψ is the largest subtree of the signed generation tree of A containing the root and which is either a skeleton subtree or a PIA subtree.

Definition 29 Let Ψ be a structure. We call $\lfloor\!\lfloor \Psi \rfloor\!\rfloor$ (resp. $\lceil\!\lceil \Psi \rceil\!\rceil$) the structure of same sort obtained, when it is defined, by turning every connective of its principal subtree Σ into either

(I) its structural counterpart if Σ is a skeleton subtree of $+\Psi$ (resp. $-\Psi$)
(II) its operational counterpart if Σ is a PIA subtree of $+\Psi$ (resp. $-\Psi$)

and turning all other connectives into their operational counterpart.

Given a well-formed sequent $\Psi\ t\ \Phi$, its **standard sequent** is $\lfloor\!\lfloor \Psi \rfloor\!\rfloor\ t\ \lceil\!\lceil \Phi \rceil\!\rceil$.

To exemplify the instrumental use of standard sequents in proving completeness, consider the following observation. The sequent $p \otimes q \ \vdash^{+}\ p \otimes q$ is not derivable in **fD.LG**, despite the fact that $\leq^{+}$ is a partial order, so in particular $p^{\mathbb{P}} \otimes^{\mathbb{P}} q^{\mathbb{P}} \leq^{+} p^{\mathbb{P}} \otimes^{\mathbb{P}} q^{\mathbb{P}}$ in every $\mathbb{FP}.\mathbb{LG}$-algebra. However, the standard sequent $p \hat{\otimes} q \ \vdash^{+}\ p \otimes q$ is **fD.LG**-derivable and moreover $(p \hat{\otimes} q)^{\mathbb{P}} = (p \otimes q)^{\mathbb{P}} = p^{\mathbb{P}} \otimes^{\mathbb{P}} q^{\mathbb{P}}$. See Lemma 52 in Appendix 1 for a recursive definition of $\lfloor\!\lfloor\ \rfloor\!\rfloor$ and $\lceil\!\lceil\ \rceil\!\rceil$. Completeness is stated as follows:

Theorem 30 *For every $\mathbb{FP}.\mathbb{LG}$-algebra $\mathbb{A}$ and every well-formed sequent $\Psi\ t\ \Phi$ in the language of* ***fD.LG****, if the interpretation $(\Psi\ t\ \Phi)^{\mathbb{A}}$ is valid, then the standard sequent $\lfloor\!\lfloor \Psi \rfloor\!\rfloor\ t\ \lceil\!\lceil \Phi \rceil\!\rceil$ is derivable in* ***fD.LG****.*

Proof We prove completeness by building a syntactic model $\mathbb{A}$. Let $\approx_{\mathbb{P}}$, $\approx_{\dot{\mathbb{P}}}$, $\approx_{\mathbb{N}}$ and $\approx_{\dot{\mathbb{N}}}$ be the equivalence relation generated by (24.13).

$$
\begin{array}{llllll}
\Psi \approx_{\mathbb{P}} \Phi & \text{iff} & \lfloor\!\lfloor \Psi \rfloor\!\rfloor \vdash^{+} \lceil\!\lceil \Phi \rceil\!\rceil & \text{and} & \lfloor\!\lfloor \Phi \rfloor\!\rfloor \vdash^{+} \lceil\!\lceil \Psi \rceil\!\rceil \\
\Psi \approx_{\dot{\mathbb{P}}} \Phi & \text{iff} & \lfloor\!\lfloor \Psi \rfloor\!\rfloor \cdot\!\vdash\!\cdot^{+} \lceil\!\lceil \Phi \rceil\!\rceil & \text{and} & \lfloor\!\lfloor \Phi \rfloor\!\rfloor \cdot\!\vdash\!\cdot^{+} \lceil\!\lceil \Psi \rceil\!\rceil \\
\Psi \approx_{\mathbb{N}} \Phi & \text{iff} & \lfloor\!\lfloor \Psi \rfloor\!\rfloor \vdash^{-} \lceil\!\lceil \Phi \rceil\!\rceil & \text{and} & \lfloor\!\lfloor \Phi \rfloor\!\rfloor \vdash^{-} \lceil\!\lceil \Psi \rceil\!\rceil \\
\Psi \approx_{\dot{\mathbb{N}}} \Phi & \text{iff} & \lfloor\!\lfloor \Psi \rfloor\!\rfloor \cdot\!\vdash\!\cdot^{-} \lceil\!\lceil \Phi \rceil\!\rceil & \text{and} & \lfloor\!\lfloor \Phi \rfloor\!\rfloor \cdot\!\vdash\!\cdot^{-} \lceil\!\lceil \Psi \rceil\!\rceil
\end{array}
\tag{24.13}
$$

In particular, $\approx_{\mathbb{P}}$, $\approx_{\dot{\mathbb{P}}}$, $\approx_{\mathbb{N}}$ and $\approx_{\dot{\mathbb{N}}}$ are congruence relations (by tonicity rules and cut rules, see Appendix 1 for a detailed proof). For any $s \in \{\mathbb{P}, \dot{\mathbb{P}}, \mathbb{N}, \dot{\mathbb{N}}\}$, let $[\Psi]_{\approx_s}$ denote the class of structures Φ such that $\Phi \approx_s \Psi$. We define operations and weakening relations by (24.14).

$$
\begin{array}{c}
[\mathring{X}]_{\approx_{\mathring{\mathbb{P}}}} \otimes^{\mathbb{A}} [\mathring{Y}]_{\approx_{\mathring{\mathbb{P}}}} = [\mathring{X} \,\hat{\otimes}\, \mathring{Y}]_{\approx_{\mathbb{P}}} \quad [\mathring{X}]_{\approx_{\mathring{\mathbb{P}}}} \oslash^{\mathbb{A}} [\mathring{\Delta}]_{\approx_{\mathring{\mathbb{N}}}} = [\mathring{X} \,\hat{\oslash}\, \mathring{\Delta}]_{\approx_{\mathbb{P}}} \quad [\mathring{\Delta}]_{\approx_{\mathring{\mathbb{N}}}} \obslash^{\mathbb{A}} [\mathring{\Delta}]_{\approx_{\mathring{\mathbb{P}}}} = [\mathring{\Delta} \,\hat{\obslash}\, \mathring{Y}]_{\approx_{\mathbb{P}}} \\
[\mathring{\Delta}]_{\approx_{\mathring{\mathbb{N}}}} \oplus^{\mathbb{A}} [\mathring{\Gamma}]_{\approx_{\mathring{\mathbb{N}}}} = [\mathring{\Delta} \,\check{\oplus}\, \mathring{\Gamma}]_{\approx_{\mathbb{N}}} \quad [\mathring{X}]_{\approx_{\mathring{\mathbb{P}}}} \backslash^{\mathbb{A}} [\mathring{\Delta}]_{\approx_{\mathring{\mathbb{N}}}} = [\mathring{X} \,\check{\backslash}\, \mathring{\Delta}]_{\approx_{\mathbb{N}}} \quad [\mathring{\Delta}]_{\approx_{\mathring{\mathbb{N}}}} /^{\mathbb{A}} [\mathring{\Delta}]_{\approx_{\mathring{\mathbb{P}}}} = [\mathring{\Delta} \,\check{/}\, \mathring{Y}]_{\approx_{\mathbb{N}}} \\
\downarrow^{\mathbb{A}} [\Delta]_{\approx_{\mathbb{N}}} = [\check{\downarrow} \Delta]_{\approx_{\dot{\mathbb{P}}}} \quad \downharpoonleft^{\mathbb{A}} [\dot{\Delta}]_{\approx_{\dot{\mathbb{N}}}} = [\check{\downharpoonleft} \dot{\Delta}]_{\approx_{\mathbb{P}}} \quad \uparrow^{\mathbb{A}} [X]_{\approx_{\mathbb{P}}} = [\hat{\uparrow} X]_{\approx_{\dot{\mathbb{N}}}} \quad \upharpoonleft^{\mathbb{A}} [X]_{\approx_{\dot{\mathbb{P}}}} = [\hat{\upharpoonleft} X]_{\approx_{\mathbb{N}}} \\
\text{and for a turnstile } t \text{ from sort } s \text{ to } s',\ [\Psi]_{\approx_s}\ t^{\mathbb{A}}\ [\Phi]_{\approx_{s'}} \text{ iff } \llfloor \Psi \rrfloor\ t\ \llceil \Phi \rrceil \text{ is derivable}
\end{array}
\tag{24.14}
$$

It is not difficult to see that the operations and relations of (24.14) are well-defined, and that the relations are indeed orders or weakening relations. The technical proof is provided in Appendix 1.

We take $\mathbb{P} = \mathsf{PurePosStr}/\approx_{\mathbb{P}}$, $\dot{\mathbb{P}} = \mathsf{ShiftedPosStr}/\approx_{\dot{\mathbb{P}}}$, $\mathbb{N} = \mathsf{PureNegStr}/\approx_{\mathbb{N}}$ and $\dot{\mathbb{N}} = \mathsf{ShiftedNegStr}/\approx_{\dot{\mathbb{N}}}$. It is easy to show that properties (24.4), (24.5), (24.6) of Definition 8 are verified thanks to the corresponding rules of the calculus.

24.4.2 Soundness and Completeness w.r.t. LG-Algebras

Given an LG-algebra $\mathbb{G} = (G, \leq, \otimes^{\mathbb{G}}, /^{\mathbb{G}}, \backslash^{\mathbb{G}}, \oplus^{\mathbb{G}}, \oslash^{\mathbb{G}}, \obslash^{\mathbb{G}})$ we define an $\mathbb{FP}.\mathbb{LG}$ algebra $\mathbb{A}_{\mathbb{G}}$ as follows: We take a copy of G as the domain of any sub-algebra in $\mathbb{A}_{\mathbb{G}}$, by defining shifts as maps sending an element to its copy in the appropriate sub-algebra of $\mathbb{A}_{\mathbb{G}}$, and finally, for each A, B in the appropriate sub-algebra of $\mathbb{A}_{\mathbb{G}}$, for each weakening relation R in $\mathbb{A}_{\mathbb{G}}$, and for each binary operation $\star^{\mathbb{A}_{\mathbb{G}}}$ in $\mathbb{A}_{\mathbb{G}}$, by defining

$$A\ R\ B \text{ iff } A \leq B \quad \text{and} \quad A \star^{\mathbb{A}_{\mathbb{G}}} B \text{ iff } A \star^{\mathbb{G}} B$$

Proposition 31 *For every LG-algebra $\mathbb{G}$, $\mathbb{A}_{\mathbb{G}}$ is an $\mathbb{FP}.\mathbb{LG}$-algebra.*

Proof It is straightforward to check that weakening relations and operations are well-defined, and (24.5) and (24.6) hold, so $\mathbb{A}_{\mathbb{G}}$ is an $\mathbb{FP}.\mathbb{LG}$ algebra accordingly to Definition 8.

Conversely, given an $\mathbb{FP}.\mathbb{LG}$ algebra $\mathbb{A}$ we first define $\pi(\mathbb{A}) = (L, \leq, \otimes^{\mathbb{G}}, /^{\mathbb{G}}, \backslash^{\mathbb{G}}, \oplus^{\mathbb{G}}, \oslash^{\mathbb{G}}, \obslash^{\mathbb{G}})$ by taking $L = \mathbb{P} \sqcup \mathbb{N}$, by defining $A \leq B$ iff $A^+ \mathring{\preceq} B^-$, where

$$\mathring{\mathbb{P}} \ni A^+ = \begin{cases} A & \text{if } A \in \mathbb{P} \\ \downarrow A & \text{if } A \in \mathbb{N} \end{cases} \quad \text{and} \quad \mathring{\mathbb{N}} \ni A^- = \begin{cases} A & \text{if } A \in \mathbb{N} \\ \uparrow A & \text{if } A \in \mathbb{P} \end{cases} \tag{24.15}$$

and by defining the operations as follows

$$
\begin{array}{lll}
A \otimes^{\pi(\mathbb{A})} B := A^+ \otimes^{\mathbb{A}} B^+ & A \oslash^{\pi(\mathbb{A})} B := A^+ \oslash^{\mathbb{A}} B^- & A \obslash^{\pi(\mathbb{A})} B := A^- \obslash^{\mathbb{A}} B^+ \\
A \oplus^{\pi(\mathbb{A})} B := A^- \oplus^{\mathbb{A}} B^- & A \backslash^{\pi(\mathbb{A})} B := A^+ \backslash^{\mathbb{A}} B^- & A /^{\pi(\mathbb{A})} B := A^- /^{\mathbb{A}} B^+.
\end{array}
\tag{24.16}
$$

Proposition 32 *For every* $\mathbb{FP}.\mathbb{LG}$ *algebra* $\mathbb{A}$, $\pi(\mathbb{A})$ *is a pre-order.*

Proof First let us show that $\leq$ is transitive. Assume that $A \leq B$ and $B \leq C$, that is $A^+ \mathrel{\mathring{\precsim}} B^-$ and $B^+ \mathrel{\mathring{\precsim}} C^-$. If $B \in \mathbb{P}$ then $B \mathrel{\mathring{\precsim}} C^-$, which is equivalent to $\uparrow B \mathrel{\mathring{\leq}}^- C^-$ and hence $A^+ \mathrel{\mathring{\precsim}} C^-$, i.e. $A \leq C$. If $B \in \mathbb{N}$ then $A^+ \mathrel{\mathring{\precsim}} B$, which is equivalent to $A^+ \mathrel{\mathring{\leq}}^+ \downarrow B$ and hence again we get $A \leq C$. It is easy to show that $\leq$ is reflexive, so $\leq$ is a pre-order.

Now we define $\mathbb{G}_{\mathbb{A}}$ based on $\pi(\mathbb{A})$ by taking the quotient over $\leq \cap \geq$. Since the operations on $\pi(\mathbb{A})$ are monotone and antitone, $\leq \cap \geq$ is in fact a congruence relation and the operations on $\mathbb{G}_{\mathbb{A}}$ can be defined in the usual way.

Proposition 33 *For every* $\mathbb{FP}.\mathbb{LG}$ *algebra* $\mathbb{A}$, $\mathbb{G}_{\mathbb{A}}$ *is an LG-algebra.*

Proof We need to show that the defined operations are residuated in each coordinate. Assume that $A \in \mathbb{P}$, $B \in \mathbb{N}$ and $C \in \mathbb{P}$:

$$\begin{aligned} A \otimes^{\mathbb{G}_{\mathbb{A}}} B \leq C \ &\text{iff}\ A \otimes^{\mathbb{A}} \downarrow B \mathrel{\mathring{\precsim}} \uparrow C \\ &\text{iff}\ A \mathrel{\mathring{\precsim}} \uparrow C /^{\mathbb{A}} \downarrow B \\ &\text{iff}\ A \leq C /^{\mathbb{G}_{\mathbb{A}}} B \end{aligned}$$

and

$$\begin{aligned} A \otimes^{\mathbb{G}_{\mathbb{A}}} B \leq C \ &\text{iff}\ A \otimes^{\mathbb{A}} \downarrow B \mathrel{\mathring{\precsim}} \uparrow C \\ &\text{iff}\ \downarrow B \mathrel{\mathring{\precsim}} A \backslash^{\mathbb{A}} \uparrow C \\ &\text{iff}\ B \leq A \backslash^{\mathbb{G}_{\mathbb{A}}} C \end{aligned}$$

The rest of the cases are done analogously.

Theorem 34 (Completeness and Soundness) *The system* ***fD.LG*** *is sound and complete with respect to LG-algebras.*

Proof First let us define a translation of *formulas* of **fD.LG** into formulas of the language of LG-algebras. We do so recursively:

- Positive and negative atoms are sent to atoms.
- For each binary connective $\star$ we define $\tau(A \star B)$ to be $\tau(A) \star \tau(B)$.
- Finally $\tau(\downarrow A)$ and $\tau(\uparrow A)$ are defined to be $\tau(A)$.

Let t be an arbitrary turnstile in the language of **fD.LG** and let w be an arbitrary weakening relation of an $\mathbb{FP}.\mathbb{LG}$ algebra. We will show that any sequent $A\ t\ B$ is provable in **fD.LG** if and only if $\tau(A) \vdash \tau(B)$ is provable in the logic of LG-algebras. Since **fD.LG** is sound and complete with respect to $\mathbb{FP}.\mathbb{LG}$ algebras it is enough to show that the sequent is falsified in an $\mathbb{FP}.\mathbb{LG}$ algebra if and only if its translation is falsified in an LG-algebra.

Assume $\mathbb{G} \nvDash \tau(A) \leq \tau(B)$. Then it is immediate that $\mathbb{A}_{\mathbb{G}} \nvDash A\ w\ B$ (since $\downarrow$ and $\uparrow$ are essentially 'identity maps', given that we defined shifts as maps sending an element to its copy in the appropriate sub-algebra of $\mathbb{A}_{\mathbb{G}}$).

For the opposite direction first we make a distinction. We call a formula in normal form, if the outermost connective is binary. It is immediate that it is enough to restrict ourselves to normal form formulas and show that if $\mathbb{A} \nvDash A\ w\ B$ then $\mathbb{G}_{\mathbb{A}} \nvDash \tau(A) \leq \tau(B)$. Notice that if in $\pi(\mathbb{A})$, it is the case that $C \nleq D$ then so is the case in $\mathbb{G}_{\mathbb{A}}$. So assume that $\mathbb{A} \nvDash A\ w\ B$. Then by definition $\pi(\mathbb{A}) \nvDash \tau(A) \leq \tau(B)$. This in turn implies that $\mathbb{G}_{\mathbb{A}} \nvDash \tau(A) \leq \tau(B)$. This concludes the proof.

24.5 Canonical Cut-Elimination

In this section we show that the class of multi-type proper display calculi (Frittella et al., 2014) can be extended to include calculi involving heterogeneous sequents, and that **fD.LG** belongs to that class.

The calculus **fD.LG** and (and any analytic extension with additional structural rules in the neutral phase) departs from the standard (multi-type) display calculi (e.g. Frittella et al., 2014; Greco, 2018) in two key points. Neutral phases are closed under display postulates, so neutral proof sections of **fD.LG**-proofs have the *display property* (Wansing, 2002; Goré, 1998). Focused phases are not closed under any structural rules so, in particular, they are not closed under display postulates. Nonetheless, focused phases have the so-called *visibility property* (see Battilotti et al., 2000, also called *segregation property* in Wansing, 2002), i.e. principal and auxiliary terms occurring in an application of logical rules are in display. Secondly, and most importantly, uniform substitution *within each type* of congruent parametric structures in a rule is generalized to allow substituting instances of structures of a sort s with another sort s' under certain restrictions (see below the notion of *mutation* introduced in Definition 35 and its use in conditions C''_6 and C''_7 of Definition 37).

All the notational conventions and the specific choices we considered in this paper are supposed to easy the reading, keep the notation compact, and allow us to discuss type-mutations. The use of meta-variables in writing rules is standard and it signifies that meta-variables can be instantiated with any (concrete) term in an actual derivation. In the case of multi-type (sequent) calculi we use different alphabets (i.e. disjoints sets of meta-variables) to signify that meta-variables of type s can be instantiated with any (concrete) term of sort s in an actual derivation. Recall that here we further extended the conventions on meta-variables (see Notation 12) and we introduced meta-turnstile symbols (see Notation 13); moreover, also the symbols for binary connectives are somehow meta-symbols given that binary connectives can take input terms of different types: for convenience, in this case we decided not to make this explicit in the syntax (see the definition of the **fD.LG** language in Sect. 24.3.1) and, indeed, the semantic domain of binary connectives is defined on the union of the relevant sub-algebras domains (see how fully polarized LG-algebras are defined in Definition 8).

Eliminating a parametric (possibly heterogeneous) cut amounts to be able to substitute a formula of a sort s by any structure of another (possibly different) sort

s', and keeping derivability. As the arguments of structural connectives have a fixed sort, substitution may lead to a clash of sorts.

To illustrate this idea, let us take the example of (24.17), where we want to move up the cut on the derivable sequent $p \,\hat{\otimes}\, (\downarrow p \backslash n) \vdash\!\cdot^{+} \downarrow n$ to the uppermost occurrence of $\downarrow n$ in π_2. This transformation requires to substitute every parametric occurrence of $\downarrow n$ in π_2 by $p \,\hat{\otimes}\, (\downarrow p \backslash n)$, which is still positive but pure. The problem is that there is an occurrence of $\downarrow n$ under $\hat{\upharpoonleft}$, and that this connective only takes shifted structures as argument. Therefore, we have to 'mutate' $\hat{\upharpoonleft}$ into $\hat{\uparrow}$ so that the sequent stays well-formed. We can check that the instances of the rule $\hat{\upharpoonleft} \dashv \check{\downharpoonright}$ are changed into instances of the rule $\hat{\uparrow} \dashv \check{\downarrow}$. In other words, the mutation $\hat{\upharpoonleft} \mapsto \hat{\uparrow}$ preserves the derivability.

The mutation generated by (24.17) also has an impact on $\hat{\otimes}$, because $\downarrow n$ appears as an argument of $\hat{\otimes}$ in π_2. However, as this connective accepts shifted as well as pure arguments, it does not have to mutate, or equivalently, it mutates into itself $\hat{\otimes} \mapsto \hat{\otimes}$. Last but not least, turnstiles also have to mutate. If we compare the relevant proof sections occurring in the two derivations, we have the following mutations at the level of turnstiles: $\cdot\!\vdash \mapsto \vdash$, $\cdot\!\vdash\!\cdot^{+} \mapsto \vdash\!\cdot^{+}$ and $\vdash^{-} \mapsto \cdot\!\vdash^{-}$.

$$
\dfrac{\begin{array}{c}\pi\\ p \,\hat{\otimes}\, (\downarrow p \backslash n) \vdash\!\cdot^{+} \downarrow n\end{array} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c}\pi_{2.1}\\ \downarrow n \cdot\!\vdash\!\cdot^{+} \check{\downharpoonright} n\\ \vdots\\ \downarrow n \,\hat{\otimes}\, p \vdash\!\cdot \uparrow(\downarrow n \otimes p)\end{array}}{\downarrow n \cdot\!\vdash \uparrow(\downarrow n \otimes p) \mathbin{\check{/}} p}\,\hat{\otimes} \dashv \check{/}}{\downarrow n \cdot\!\vdash\!\cdot^{+} \check{\downharpoonright}(\uparrow(\downarrow n \otimes p) \mathbin{\check{/}} p)}\,\check{\downharpoonright}}{\hat{\upharpoonleft}\downarrow n \vdash^{-} \uparrow(\downarrow n \otimes p) \mathbin{\check{/}} p}\,\hat{\upharpoonleft} \dashv \check{\downharpoonright}}{\downarrow n \cdot\!\vdash\!\cdot^{+} \check{\downharpoonright}(\uparrow(\downarrow n \otimes p) \mathbin{\check{/}} p)}\,\hat{\upharpoonleft} \dashv \check{\downharpoonright}}{\downarrow n \cdot\!\vdash \uparrow(\downarrow n \otimes p) \mathbin{\check{/}} p}\,\check{\downharpoonright}}{p \,\hat{\otimes}\, (\downarrow p \backslash n) \vdash \uparrow(\downarrow n \otimes p) \mathbin{\check{/}} p}\,\text{P-Cut}
\quad \rightsquigarrow \quad
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c}\dfrac{\begin{array}{c}\pi\\ p \,\hat{\otimes}\, (\downarrow p \backslash n) \vdash\!\cdot^{+} \downarrow n\end{array} \qquad \begin{array}{c}\pi_{2.1}\\ \downarrow n \cdot\!\vdash\!\cdot^{+} \check{\downharpoonright} n\end{array}}{p \,\hat{\otimes}\, (\downarrow p \backslash n) \vdash\!\cdot^{+} \check{\downharpoonright} n}\,\text{P-Cut}\\ \vdots\\ (p \,\hat{\otimes}\, (\downarrow p \backslash n)) \,\hat{\otimes}\, p \vdash\!\cdot \uparrow(\downarrow n \otimes p)\end{array}}{p \,\hat{\otimes}\, (\downarrow p \backslash n) \vdash \uparrow(\downarrow n \otimes p) \mathbin{\check{/}} p}\,\hat{\otimes} \dashv \check{/}}{p \,\hat{\otimes}\, (\downarrow p \backslash n) \vdash\!\cdot^{+} \check{\downarrow}(\uparrow(\downarrow n \otimes p) \mathbin{\check{/}} p)}\,\check{\downarrow}}{\hat{\uparrow}(p \,\hat{\otimes}\, (\downarrow p \backslash n)) \cdot\!\vdash^{-} \uparrow(\downarrow n \otimes p) \mathbin{\check{/}} p}\,\hat{\uparrow} \dashv \check{\downarrow}}{p \,\hat{\otimes}\, (\downarrow p \backslash n) \vdash\!\cdot^{+} \check{\downarrow}(\uparrow(\downarrow n \otimes p) \mathbin{\check{/}} p)}\,\hat{\uparrow} \dashv \check{\downarrow}}{p \,\hat{\otimes}\, (\downarrow p \backslash n) \vdash \uparrow(\downarrow n \otimes p) \mathbin{\check{/}} p}\,\check{\downarrow}
\tag{24.17}
$$

We will now discuss the idea of mutation and generalize Belnap's cut-elimination metatheorem. Then we will show that **fD.LG** satisfies the conditions of the metatheorem. Let $\mathcal{C}$ be a multi-type calculus with heterogeneous sequents and let $\mathcal{Q}$ be the set of types of $\mathcal{C}$. We call $\mathcal{S}_{\mathcal{F}}$ (resp. $\mathcal{S}_{\mathcal{G}}$) the set of structural $\mathcal{F}$-connectives (resp. $\mathcal{G}$-connectives), $\mathcal{S} = \mathcal{S}_{\mathcal{F}} \cup \mathcal{S}_{\mathcal{G}}$ and $\mathcal{T}$ the set of turnstiles. For an n-ary connective $H \in \mathcal{S}$ we call *sort of* H, $\mathsf{sort}(H) \in \mathcal{Q}^n$, the n-tuple of types that the connective takes as input. For a turnstile $t \in \mathcal{T}$, we call *sort of* t, $\mathsf{sort}(t) \in \mathcal{Q}^2$, the pair of types that t connects.

Definition 35 (*Mutation*) The *mutation* relation of $\mathcal{C}$, $\mu_{\mathcal{C}} \subseteq \mathcal{S} \times \mathcal{S}$, is an equivalence relation between structural connectives such that:

1. if $H \mu_{\mathcal{C}} H'$ then $H \in \mathcal{S}_{\mathcal{F}}$ if and only if $H' \in \mathcal{S}_{\mathcal{F}}$;
2. if $H \mu_{\mathcal{C}} H'$ then H and H' have the same arity;
3. if $H \mu_{\mathcal{C}} H'$ and $\mathsf{sort}(H) = \mathsf{sort}(H')$ then $H = H'$.

Henceforth we will usually omit the subscript when the context is clear. Informally, the mutation relation describes into which structural connectives the structural connective H can be mutated. It is easy to see that we can extend the relation to (not necessarily well typed) structures recursively on the generation tree of a structure: We say that $\Phi\mu\Psi$ if the generation trees of Φ and Ψ are identical modulo μ. It is not hard to see that by condition 3 of Definition 35, given a structure Φ, there exists at most one well typed structure in $\mu[\Phi] = \{\Psi \mid \Phi\mu\Psi\}$, which we denote with $\mu(\Phi)$. Finally if $\Phi\ t\ \Psi$ is a sequent we denote with $\mu(\Phi\ t\ \Psi)$ the unique (if it exists) well typed sequent of the form $\mu(\Phi)\ t'\ \mu(\Psi)$, where $t' \in \mathcal{T}$. The following is an immediate consequence of condition 3 of Definition 35.

Corollary 36 *For every sequent $\Phi\ t\ \Psi$, every well typed subtree of the generation tree of $\Phi\ t\ \Psi$ is unchanged in $\mu(\Phi\ t\ \Psi)$.*

Definition 37 We adapt the conditions of Frittella et al. (2014, Sect. 3) to define the class of *heterogeneous* multi-type proper display calculi C by modifying C_5', C_5'', C_6', C_7', C_8', C_9 and C_{10} as follows:

- C_1 (Preservation of operational terms) Each operational term occurring in a premise of an inference rule *inf* is a subterm of some operational term in the conclusion of *inf*.
- C_2 (Shape-alikeness of parameters) Congruent parameters (i.e. non-active terms in the application of a rule) are occurrences of the same structure.
- C_2' (Type-alikeness of parameters) Congruent parameters have exactly the same type. This condition bans the possibility that a parameter changes type along its history.
- C_3 (Non-proliferation of parameters) Each parameter in an inference rule *inf* is congruent to at most one constituent in the conclusion of *inf*.
- C_4 (Position-alikeness of parameters) Congruent parameters are either all precedent or all succedent parts of their respective sequents.
- C_5 (Display of principal constituents) If an operational term a is principal in an inference rule *inf*, then a is in display in the conclusion of *inf*.
- C_6'' (Closure under precedent mutations) For every derivable turnstile[6] t' with $\mathsf{sort}(t') = (s', s)$, for every rule

$$\frac{((\Phi_i\ t_i\ \Psi_i)[A]^{pre})_{1\leq i\leq n}}{(\Phi_0\ t_0 \Psi_0)[A]^{pre}}\ \mathrm{R}$$

where A is a parametric operational term in precedent position and of sort s and for every precedent structure Ξ of sort s', $\mu((\Phi_i\ t_i\ \Psi_i)[\Xi/A]^{pre})$ and $\mu((\Phi_0 t_0 \Psi_0)[\Xi/A]^{pre})$ exist for every $1 \leq i \leq n$ and the rule

$$\frac{(\mu((\Phi_i\ t_i\ \Psi_i)[\Xi/A]^{pre}))_{1\leq i\leq n}}{\mu((\Phi_0\ t_0 \Psi_0)[\Xi/A]^{pre})}\ \mathrm{R'}$$

is derivable in the calculus.
- C_7'' (Closure under succedent mutations) For every derivable turnstile t' with $\mathsf{sort}(t') = (s, s')$, for every rule

[6] By derivable turnstile, we mean that there is a derivable sequent on that turnstile.

$$\frac{((\Phi_i\ t_i\ \Psi_i)[A])_{1\leq i\leq n}}{(\Phi_0 t_0 \Psi_0)[A]}\ \mathrm{R}$$

where A is parametric operational term in succedent position and of sort s and for every succedent structure Ξ of sort s', $\mu((\Phi_i\ t_i\ \Psi_i)[\Xi/A])$ and $\mu((\Phi_0 t_0 \Psi_0)[\Xi/A])$ exist for every $1 \leq i \leq n$ and the rule

$$\frac{(\mu((\Phi_i\ t_i\ \Psi_i)[\Xi/A]))_{1\leq i\leq n}}{\mu((\Phi_0 t_0 \Psi_0)[\Xi/A])}\ \mathrm{R'}$$

is derivable in the calculus.

C_8'' (Eliminability of matching principal constituents) This condition requests a standard Gentzen-style checking, which is now limited to the case in which both cut formulas are *principal*, i.e. each of them has been introduced with the last rule application of each corresponding subdeduction. In this case, analogously to the proof Gentzen-style, condition C_8'' requires being able to transform the given deduction into a deduction with the same conclusion in which either the cut is eliminated altogether, or is transformed in one or more applications of the cut rule, involving proper subterms of the original operational cut-term.

C_9' (Uniqueness of turnstiles) There is at most one turnstile per each pair of types.

C_{10}' (Closure under turnstile composition) For every turnstiles $t, t' \in \mathcal{T}$ such that $\mathsf{sort}(t) = (s, s')$ and $\mathsf{sort}(t') = (s', s'')$, there exists a turnstile $t'' \in \mathcal{T}$ such that the following cut is definable and belongs to the system:

$$\frac{\Psi\ t\ A \qquad A\ t'\ \Phi}{\Psi\ t''\ \Phi}\ tt'\text{-Cut}$$

Theorem 38 (Canonical cut-elimination) *Any heterogeneous multi-type proper display calculus satisfying C_2-C_{10}' enjoys cut-elimination. If also C_1 is satisfied, then the calculus enjoys the subformula property.*

Proof The cut-elimination proof provided in Frittella et al. (2014, Sect. 4) essentially follows the proof in Wansing (2002, Sect. 3.3, Appendix A) generalizing it to the case of multi-type sequent systems. Without loss of generality, we can confine ourselves to derivations with exactly one application of cut. As usual in the display calculi literature, we consider two main cases: (i) both cut-formulas are principal (principal moves) and (ii) at least one cut-formula is not principal (parametric moves). The principal moves are guaranteed by condition C_8'', as usual in the display calculi literature (see Wansing, 2002). As to the parametric moves, we essentially follow the proof in Frittella et al. (2014) and we only expand on the parts of which the proof departs from it. We are in the following situation:

$$\begin{array}{ccc}
 & \begin{array}{ccc} \vdots\,\pi_{2.1} & & \vdots\,\pi_{2.n} \\ A_1\ t_{2.1}\ \Phi_1 & \ldots & A_n\ t_{2.n}\ \Phi_n \end{array} \\
\vdots\,\pi & \ddots\vdots\cdot\ \pi_2 \\
\Psi\ t_1\ A & A\ t_2\ \Phi \\
\hline
\multicolumn{2}{c}{\Psi\ t_3\ \Phi}
\end{array}\ t_1t_2\text{-Cut}$$

with the A_is being the uppermost congruent occurrences of A in the proof section π_2. We treat the case of A_i and when it is principal (cfr. case (1) in Frittella et al., 2014). Call s' the sort of Ψ and s the sort of A. By C_2', A_i is also of sort s. By C_{10}', the calculus contains the $t_1 t_{2.i}$-Cut. For every rule R in the section π_2, we track the congruent occurrences $(A_j)_j$ of A in R (thanks to C_2-C_4) and substitute them uniformly by Ψ with the well-typed mutation derivable by C_6''. By a straightforward induction on π_2, its (well-typed) mutation $\mu(\pi_2)$ is a well-formed proof of the end-sequent $\Psi\ t_2'\ \Phi$ by Corollary 36 and C_9'. Moreover, we can cut on $A_i\ t_{2.i}\ \Phi_i$ with $\Psi\ t_1\ A$ (this cut is well-defined and belongs to the calculus thanks to C_5, C_9' and C_{10}') as follows:

$$
\dfrac{\begin{array}{c}\vdots\,\pi\\ \Psi\ t_1\ A\end{array} \qquad \begin{array}{c}\vdots\,\pi_{2.i}\\ A_i\ t_{2.i}\ \Phi_i\\ \vdots\,\pi_2\\ A\ t_2\ \Phi\end{array}}{\Psi\ t_3\ \Phi}\ t_1t_2\text{-Cut}
\quad\rightsquigarrow\quad
\begin{array}{c}
\dfrac{\begin{array}{c}\vdots\,\pi\\ \Psi\ t_1\ A\end{array} \qquad \begin{array}{c}\vdots\,\pi_{2.i}\\ A_i\ t_{2.i}\ \Phi_i\end{array}}{\Psi\ t_1{\circ}t_{2.1}\ \Phi_i}\ t_1t_{2.i}\text{-Cut}\\
\vdots\,\mu(\pi_2)\\
\Psi\ t_3\ \Phi
\end{array}
$$

If A_i has been introduced as a parameter in the conclusion of $\pi_{2.i}$, the proof goes exactly as in case (2) of Frittella et al. (2014) and the fact that we admit heterogeneous turnstiles does not play any role.

Theorem 39 ***fD.LG*** *is a heterogeneous multi-type proper display calculus.*

Proof By inspecting the rules, it is immediate to check that **fD.LG** enjoys conditions C_1-C_5, C_9' and C_{10}'. Condition C_8'' can be easily checked considering all possible cases. Below we consider the case where the cut formula main connective is $\otimes$ (see (24.18)). All other LG connectives behave similarly.

$$
\dfrac{\dfrac{\begin{array}{c}\vdots\,\pi_{1.1}\\ \mathring{X}\ \overset{\circ}{\vdash}{}^{+}\ \mathring{P}\end{array} \qquad \begin{array}{c}\vdots\,\pi_{1.2}\\ \mathring{Y}\ \overset{\circ}{\vdash}{}^{+}\ \mathring{Q}\end{array}}{\mathring{X}\ \hat{\otimes}\ \mathring{Y}\ \vdash^{+}\ \mathring{P}\otimes\mathring{Q}}\ \otimes_R
\qquad
\dfrac{\begin{array}{c}\vdots\,\pi_2\\ \mathring{P}\ \hat{\otimes}\ \mathring{Q}\ \overset{\circ}{\vdash}\ \mathring{\Delta}\end{array}}{\mathring{P}\otimes\mathring{Q}\ \overset{\circ}{\vdash}\ \mathring{\Delta}}\ \otimes_L}{\mathring{X}\ \hat{\otimes}\ \mathring{Y}\ \overset{\circ}{\vdash}\ \mathring{\Delta}}\ \text{Pn-cut}
\quad\rightsquigarrow\quad
\dfrac{\dfrac{\begin{array}{c}\vdots\,\pi_{1.2}\\ \mathring{Y}\ \overset{\circ}{\vdash}{}^{+}\ \mathring{Q}\end{array}
\qquad
\dfrac{\dfrac{\dfrac{\begin{array}{c}\vdots\,\pi_{1.1}\\ \mathring{X}\ \overset{\circ}{\vdash}{}^{+}\ \mathring{P}\end{array} \qquad \dfrac{\begin{array}{c}\vdots\,\pi_2\\ \mathring{P}\ \hat{\otimes}\ \mathring{Q}\ \overset{\circ}{\vdash}\ \mathring{\Delta}\end{array}}{\mathring{P}\ \overset{\circ}{\vdash}\ \mathring{\Delta}\ \check{/}\ \mathring{Q}}\ \hat{\otimes}\dashv\check{/}}{\mathring{X}\ \overset{\circ}{\vdash}\ \mathring{\Delta}\ \check{/}\ \mathring{Q}}\ \text{Pn-Cut}}{\mathring{X}\ \hat{\otimes}\ \mathring{Q}\ \overset{\circ}{\vdash}\ \mathring{\Delta}}\ \hat{\otimes}\dashv\check{/}}{\mathring{Q}\ \overset{\circ}{\vdash}\ \mathring{X}\ \check{\backslash}\ \mathring{\Delta}}\ \hat{\otimes}\dashv\check{\backslash}}{\mathring{Y}\ \overset{\circ}{\vdash}\ \mathring{X}\ \check{\backslash}\ \mathring{\Delta}}\ \text{Pn-Cut}}{\mathring{X}\ \hat{\otimes}\ \mathring{Y}\ \overset{\circ}{\vdash}\ \mathring{\Delta}}\ \hat{\otimes}\dashv\check{\backslash}
\tag{24.18}
$$

The case where the cut formula main connective is $\downarrow$ and the precedent is pure is given in (24.19). When the precedent is shifted, we just need replace $\hat{\uparrow}$ by $\hat{\uparrow}$ in (24.19). $\uparrow$ behaves similarly.

$$
\dfrac{\dfrac{\dfrac{\vdots\ \pi_1}{X \vdash\!\cdot^{+} \check{\downarrow} N}}{X \vdash\!\cdot^{+} \downarrow N}\,\downarrow_R \qquad \dfrac{\dfrac{\vdots\ \pi_2}{N \vdash^{-} \Delta}}{\downarrow N \cdot\!\vdash\!\cdot^{+} \check{\downarrow}\Delta}\,\downarrow_L}{X \vdash\!\cdot^{+} \check{\downarrow}\Delta}\,\text{P-Cut}
\quad\rightsquigarrow\quad
\dfrac{\dfrac{\dfrac{\dfrac{\vdots\ \pi_1}{X \vdash\!\cdot^{+} \check{\downarrow} N}}{\hat{\upharpoonleft} X \vdash^{-} N}\,\hat{\upharpoonleft} \dashv \check{\downarrow} \qquad \dfrac{\vdots\ \pi_2}{N \vdash^{-} \Delta}}{\hat{\upharpoonleft} X \vdash^{-} \Delta}\,\text{N-Cut}}{X \vdash\!\cdot^{+} \check{\downarrow}\Delta}\,\hat{\upharpoonleft} \dashv \check{\downarrow}
\tag{24.19}
$$

We emphasize that the previous transformations are indeed instances of a standard, uniform reduction strategy always available in the case of display calculi and the fact that we admit heterogeneous connectives and heterogeneous turnstiles does not play any role. Indeed, it is immediate to see that the reduction strategy exemplified above is guaranteed by observing that (i) in principal cuts, one cut formula is introduced by a tonicity rule and the other cut formula is introduced by a translation rule, (ii) condition C_5 ensures that principal terms are always introduced in display (both in the case of tonicity and translation rules), moreover auxiliary terms are either already in display (in the case of tonicity rules) or can be displayed (in the case of translation rules), namely focused proof sections have the visibility property and non focused proof sections have the display property.

The mutation relation of **fD.LG** is $\mu = \mathrm{Id} \cup \{(\check{\downharpoonleft}, \check{\downarrow}), (\check{\downarrow}, \check{\downharpoonleft}), (\hat{\uparrow}, \hat{\upharpoonleft}), (\hat{\upharpoonleft}, \hat{\uparrow})\}$. It is routine to verify conditions C_6'' and C_7'' by inspecting the rules. We provide a few examples, the rest of the cases being analogous:

Introduction rules: For the meta-rule

$$
\otimes_L\ \dfrac{\mathring{P}\ \hat{\otimes}\ \mathring{Q} \ \mathring{\vdash}\ \mathring{\Delta}}{\mathring{P} \otimes \mathring{Q} \ \mathring{\vdash}\ \mathring{\Delta}}
$$

the only parametric variable is $\mathring{\Delta}$. By Proposition 14 the only possible heterogeneous cut is if $\mathring{\Delta}$ is of sort $\mathsf{ShiftedNegStr}$, $\dot{\Delta} \cdot\!\vdash^{-} \Gamma$. Substituting $\dot{\Delta}$ in the rule with Γ, is an instance of the same meta-rule. For the meta-rule

$$
\downarrow_L\ \dfrac{N \vdash^{-} \Delta}{\downarrow N \cdot\!\vdash\!\cdot^{+} \check{\downarrow}\,\Delta}
$$

the only parametric variable is Δ. Then by Proposition 14 no possible heterogeneous cuts exist. For the meta-rule

$$
\dfrac{\mathring{X} \ \mathring{\vdash}^{+} \check{\downarrow} N}{\mathring{X} \ \mathring{\vdash}^{+} \downarrow N}\,\downarrow_R
$$

the only parametric variable is $\mathring{X}$. By Proposition 14 the only possible heterogeneous cut is if $\mathring{X}$ is of sort $\mathsf{ShiftedPosStr}$, the mutation under which is again an instance of the same meta-rule.

Display postulates: The display rules

$$
\dfrac{\hat{\uparrow} X \cdot\!\vdash\!\cdot^{-} \dot{\Delta}}{X \vdash^{+} \check{\downharpoonleft}\,\dot{\Delta}}\,\hat{\uparrow} \dashv \check{\downharpoonleft}
\qquad
\dfrac{\hat{\uparrow} X \cdot\!\vdash^{-} \Delta}{X \vdash\!\cdot^{+} \check{\downarrow}\Delta}\,\hat{\uparrow} \dashv \check{\downarrow}
\qquad
\hat{\upharpoonleft} \dashv \check{\downarrow}\,\dfrac{\dot{X} \cdot\!\vdash\!\cdot^{+} \check{\downarrow}\,\Delta}{\hat{\upharpoonleft}\,\dot{X} \vdash^{-} \Delta}
$$

are the mutations of each other.
Structural rules: For the rule

$$\frac{\mathring{X} \overset{\circ}{\vdash} \Delta}{\mathring{X} \overset{\circ}{\vdash}^{+} \check{\downarrow} \Delta} \; \check{\downarrow}$$

by Proposition 14 there are no right heterogeneous cuts. The mutation under the only possible left heterogeneous cut is an instance of the same rule.

Corollary 40 *fD.LG enjoys canonical cut-elimination.*

24.6 Proof Translations

The rules of the minimal calculus **fLG** are provided in Sect. 3.1 of Moortgat and Moot (2012). Here are the translations between **fD.LG** and **fLG**. In the following, we assume that in **fLG**, axioms are only possible on atomic formulas. The results remain valid when axioms on every formula are allowed, but $\lceil\cdot\rceil$ would involve a structural normalization using $\llbracket\cdot\rrbracket$ and $[\![\lceil\cdot\rceil]\!]$ transformations.

24.6.1 From fLG to fD.LG

Definition 41 Given a **fLG** formula A its positive polarization $\lceil A\rceil^{+}_{\mathsf{F}}$ (resp. negative polariazation $\lceil A\rceil^{-}_{\mathsf{F}}$) is a positive (resp. negative) **fD.LG**-formula defined by (24.20). Moreover, we have that $\lceil A\rceil^{+}_{\mathsf{F}}$ (resp. $\lceil A\rceil^{-}_{\mathsf{F}}$) is pure iff A is **fLG**-positive (resp. **fLG**-negative).

$$\begin{array}{lcl@{\quad}lcl@{\quad}lcl@{\quad}lcl}
\lceil A\otimes B\rceil^{+}_{\mathsf{F}} &=& \lceil A\rceil^{+}_{\mathsf{F}}\otimes\lceil B\rceil^{+}_{\mathsf{F}} & \lceil A\oslash B\rceil^{+}_{\mathsf{F}} &=& \lceil A\rceil^{+}_{\mathsf{F}}\oslash\lceil B\rceil^{-}_{\mathsf{F}} & \lceil A\obslash B\rceil^{+}_{\mathsf{F}} &=& \lceil A\rceil^{-}_{\mathsf{F}}\obslash\lceil B\rceil^{+}_{\mathsf{F}} & \lceil p\rceil^{+}_{\mathsf{F}} &=& p\\
\lceil A\oplus B\rceil^{+}_{\mathsf{F}} &=& \downarrow(\lceil A\rceil^{-}_{\mathsf{F}}\oplus\lceil B\rceil^{-}_{\mathsf{F}}) & \lceil A\backslash B\rceil^{+}_{\mathsf{F}} &=& \downarrow(\lceil A\rceil^{+}_{\mathsf{F}}\backslash\lceil B\rceil^{-}_{\mathsf{F}}) & \lceil A/B\rceil^{+}_{\mathsf{F}} &=& \downarrow(\lceil A\rceil^{-}_{\mathsf{F}}/\lceil B\rceil^{+}_{\mathsf{F}}) & \lceil n\rceil^{+}_{\mathsf{F}} &=& \downarrow n\\[6pt]
\lceil A\otimes B\rceil^{-}_{\mathsf{F}} &=& \uparrow(\lceil A\rceil^{+}_{\mathsf{F}}\otimes\lceil B\rceil^{+}_{\mathsf{F}}) & \lceil A\oslash B\rceil^{-}_{\mathsf{F}} &=& \uparrow(\lceil A\rceil^{+}_{\mathsf{F}}\oslash\lceil B\rceil^{-}_{\mathsf{F}}) & \lceil A\obslash B\rceil^{-}_{\mathsf{F}} &=& \uparrow(\lceil A\rceil^{-}_{\mathsf{F}}\obslash\lceil B\rceil^{+}_{\mathsf{F}}) & \lceil p\rceil^{-}_{\mathsf{F}} &=& \uparrow p\\
\lceil A\oplus B\rceil^{-}_{\mathsf{F}} &=& \lceil A\rceil^{-}_{\mathsf{F}}\oplus\lceil B\rceil^{-}_{\mathsf{F}} & \lceil A\backslash B\rceil^{-}_{\mathsf{F}} &=& \lceil A\rceil^{+}_{\mathsf{F}}\backslash\lceil B\rceil^{-}_{\mathsf{F}} & \lceil A/B\rceil^{-}_{\mathsf{F}} &=& \lceil A\rceil^{-}_{\mathsf{F}}/\lceil B\rceil^{+}_{\mathsf{F}} & \lceil n\rceil^{-}_{\mathsf{F}} &=& n
\end{array} \tag{24.20}$$

Definition 42 Given an input structure X (resp. an output structure Δ), $\lceil X\rceil^{+}_{\mathsf{S}}$ (resp. $\lceil\Delta\rceil^{-}_{\mathsf{S}}$) is a positive structure (resp. negative structure) of **fD.LG** defined by (24.21) without structural shift. The translation of a **fLG**-sequent into a **fD.LG**-sequent is given by (24.22).

$$\begin{array}{lcl@{\;}lcl@{\;}lcl@{\;}lcl}
\lceil X\hat{\otimes} Y\rceil^{+}_{\mathsf{S}} &=& \lceil X\rceil^{+}_{\mathsf{S}}\hat{\otimes}\lceil Y\rceil^{+}_{\mathsf{S}} & \lceil X\hat{\oslash}\Delta\rceil^{+}_{\mathsf{S}} &=& \lceil X\rceil^{+}_{\mathsf{S}}\hat{\oslash}\lceil\Delta\rceil^{-}_{\mathsf{S}} & \lceil \Delta\hat{\obslash} Y\rceil^{+}_{\mathsf{S}} &=& \lceil \Delta\rceil^{-}_{\mathsf{S}}\hat{\obslash}\lceil Y\rceil^{+}_{\mathsf{S}} & \lceil A\rceil^{+}_{\mathsf{S}} &=& \lceil A\rceil^{+}_{\mathsf{F}}\\
\lceil \Delta\check{\oplus}\Gamma\rceil^{-}_{\mathsf{S}} &=& \lceil \Delta\rceil^{-}_{\mathsf{S}}\check{\oplus}\lceil \Gamma\rceil^{-}_{\mathsf{S}} & \lceil X\check{\backslash}\Delta\rceil^{-}_{\mathsf{S}} &=& \lceil X\rceil^{+}_{\mathsf{S}}\check{\backslash}\lceil\Delta\rceil^{-}_{\mathsf{S}} & \lceil \Delta\check{/} Y\rceil^{-}_{\mathsf{S}} &=& \lceil \Delta\rceil^{-}_{\mathsf{S}}\check{/}\lceil Y\rceil^{+}_{\mathsf{S}} & \lceil A\rceil^{-}_{\mathsf{S}} &=& \lceil A\rceil^{-}_{\mathsf{F}}
\end{array} \tag{24.21}$$

$$\lceil X\vdash \boxed{A}\rceil = \lceil X\rceil^{+}_{\mathsf{S}} \overset{\circ}{\vdash}^{+} \lceil A\rceil^{+}_{\mathsf{F}} \qquad \lceil \boxed{A}\vdash\Delta\rceil = \lceil A\rceil^{-}_{\mathsf{F}} \overset{\circ}{\vdash}^{-} \lceil\Delta\rceil^{-}_{\mathsf{S}} \qquad \lceil X\vdash\Delta\rceil = \lceil X\rceil^{+}_{\mathsf{S}} \overset{\circ}{\vdash} \lceil\Delta\rceil^{-}_{\mathsf{S}} \tag{24.22}$$

Definition 43 A **fD.LG**-sequent S is called **normal** if there exists a **fLG**-sequent S' such that $S = \lceil S' \rceil$.

By the remark in Definition 42, normal sequents do not contain structural shifts. Therefore, a normal sequent is either positive focused, negative focused or neutral (non-focused).

Theorem 44 *Every **fLG**-derivation of a sequent S can be translated into a **fD.LG**-derivation of $\lceil S \rceil$.*

Proof Set a proof π of a sequent S. We translate every rule R of π by a rule $\lceil R \rceil$ of **fD.LG** by induction on π. We only treat a sufficient sample of rules. The others are similar.

$$\left\lceil \frac{}{p \vdash \boxed{p}}\,\mathrm{Ax} \right\rceil = \frac{\dfrac{}{p \vdash^{+} p}\,Id}{\lceil p \rceil^{+}_{\mathsf{S}} \vdash^{+} \lceil p \rceil^{+}_{\mathsf{F}}}$$

$$\left\lceil \frac{X \vdash \boxed{A} \qquad Y \vdash \boxed{B}}{X \hat{\otimes} Y \vdash \boxed{A \otimes B}}\,\otimes_R \right\rceil = \frac{\dfrac{\lceil X \rceil^{+}_{\mathsf{S}} \overset{\circ}{\vdash}^{+} \lceil A \rceil^{+}_{\mathsf{F}} \qquad \lceil Y \rceil^{+}_{\mathsf{S}} \overset{\circ}{\vdash}^{+} \lceil B \rceil^{+}_{\mathsf{F}}}{\lceil X \rceil^{+}_{\mathsf{S}} \hat{\otimes} \lceil Y \rceil^{+}_{\mathsf{S}} \vdash^{+} \lceil A \rceil^{+}_{\mathsf{F}} \otimes \lceil B \rceil^{+}_{\mathsf{F}}}\,\otimes_R}{\lceil X \hat{\otimes} Y \rceil^{+}_{\mathsf{S}} \vdash^{+} \lceil A \otimes B \rceil^{+}_{\mathsf{F}}}$$

$$\left\lceil \frac{A \hat{\otimes} B \vdash \Delta}{A \otimes B \vdash \Delta}\,\otimes_L \right\rceil = \frac{\dfrac{\dfrac{\lceil A \hat{\otimes} B \rceil^{+}_{\mathsf{S}} \overset{\circ}{\vdash} \lceil \Delta \rceil^{-}_{\mathsf{S}}}{\lceil A \rceil^{+}_{\mathsf{F}} \hat{\otimes} \lceil B \rceil^{+}_{\mathsf{F}} \overset{\circ}{\vdash} \lceil \Delta \rceil^{-}_{\mathsf{S}}}}{\lceil A \rceil^{+}_{\mathsf{F}} \otimes \lceil B \rceil^{+}_{\mathsf{F}} \overset{\circ}{\vdash} \lceil \Delta \rceil^{-}_{\mathsf{S}}}\,\otimes_R}{\lceil A \otimes B \rceil^{+}_{\mathsf{S}} \overset{\circ}{\vdash} \lceil \Delta \rceil^{-}_{\mathsf{S}}}$$

In the following, we assume that A is **fLG**-positive, i.e. begins by $\otimes$, $\oslash$, $\obslash$ or a positive atom. From Definition 41, we clearly have that $\lceil A \rceil^{-}_{\mathsf{F}} = \uparrow \lceil A \rceil^{+}_{\mathsf{F}}$ with $\lceil A \rceil^{+}_{\mathsf{F}}$ being positive pure.

$$\left\lceil \frac{A \vdash \Delta}{\boxed{A} \vdash \Delta}\,\tilde{\mu} \right\rceil = \frac{\dfrac{\dfrac{\lceil A \rceil^{+}_{\mathsf{F}} \overset{\circ}{\vdash} \lceil \Delta \rceil^{-}_{\mathsf{S}}}{\hat{\uparrow} \lceil A \rceil^{+}_{\mathsf{F}} \overset{\circ}{\vdash}^{-} \lceil \Delta \rceil^{-}_{\mathsf{S}}}\,\hat{\uparrow}}{\uparrow \lceil A \rceil^{+}_{\mathsf{F}} \overset{\circ}{\vdash}^{-} \lceil \Delta \rceil^{-}_{\mathsf{S}}}\,\uparrow_L}{\lceil A \rceil^{-}_{\mathsf{F}} \overset{\circ}{\vdash}^{-} \lceil \Delta \rceil^{-}_{\mathsf{S}}}$$

$$\left\lceil \frac{X \vdash \boxed{A}}{X \vdash A}\,\mu^{*} \right\rceil = \frac{\dfrac{\dfrac{\lceil X \rceil^{+}_{\mathsf{S}} \vdash^{+} \lceil A \rceil^{+}_{\mathsf{F}}}{\hat{\uparrow} \lceil X \rceil^{+}_{\mathsf{S}} \cdot\!\vdash\!\cdot^{-} \uparrow \lceil A \rceil^{+}_{\mathsf{F}}}\,\uparrow_R}{\lceil X \rceil^{+}_{\mathsf{S}} \vdash\cdot \uparrow \lceil A \rceil^{+}_{\mathsf{F}}}\,\hat{\uparrow}}{\lceil X \rceil^{+}_{\mathsf{S}} \vdash\cdot \lceil A \rceil^{-}_{\mathsf{S}}}$$

Display postulates are translated by themselves. Note that translation $\lceil\cdot\rceil$ is injective on derivations.

24.6.2 From fD.LG to fLG

Definition 45 If S is an **fD.LG**-derivable sequent, we call S *fDLG-normal* or simply *normal* provided that no structural shift symbol occurs in S (cfr. the yellow and white areas of Fig. 24.3 and Definition 43), and *non-normal* otherwise (cfr. the grey area of Fig. 24.3).

We identify proof-sections containing non-normal sequents in Proposition 50 and give their translation back to **fLG** in Theorem 51.

Definition 46 The depolarization $\lfloor A \rfloor$ of a **fD.LG**-formula A is the **fLG**-formula obtained by removing the shifts of A.

Definition 47 Given a **fD.LG**-structure Ψ without structural shift, its translation $\lfloor \Psi \rfloor$, obtained by depolarizing each formula occurring in Ψ, is a **fLG**-structure. Moreover, if Ψ is positive (resp. negative), then $\lfloor \Psi \rfloor$ is an input (resp. output) structure.

Definition 48 Given a normal sequent S, its translation $\lfloor S \rfloor$ is given by (24.23).

$$\lfloor \mathring{X} \overset{\circ}{\vdash}{}^{+} \mathring{P} \rfloor = \lfloor \mathring{X} \rfloor \vdash \boxed{\lfloor \mathring{P} \rfloor} \qquad \lfloor \mathring{N} \overset{\circ}{\vdash}{}^{-} \mathring{\Delta} \rfloor = \boxed{\lfloor \mathring{N} \rfloor} \vdash \lfloor \mathring{\Delta} \rfloor \qquad \lfloor \mathring{X} \overset{\circ}{\vdash} \mathring{\Delta} \rfloor = \lfloor \mathring{X} \rfloor \vdash \lfloor \mathring{\Delta} \rfloor \tag{24.23}$$

Definition 49 A processing proof-section of a **fD.LG**-proof π is a proof-section of π where the leaves and the root are normal sequents and there is at least one internal non-normal sequent.

Proposition 50 *In a minimal proof, every processing proof-section is of the form of one proof-section of (24.24).*

$$\begin{array}{ll}
\leftharpoonup \left\{ \begin{array}{c} N \vdash^{-} \Delta \\ \hline \downarrow N \ \cdot\!\vdash\!\cdot^{+} \check{\downarrow}\Delta \\ \hline \downarrow N \cdot\!\vdash \Delta \end{array} \right. \begin{array}{l} \downarrow L \\ \check{\downarrow} \end{array}
&
\rightharpoonup \left\{ \begin{array}{c} X \vdash^{+} P \\ \hline \hat{\uparrow} X \ \cdot\!\vdash\!\cdot^{-} \uparrow P \\ \hline X \vdash\!\cdot \uparrow P \end{array} \right. \begin{array}{l} \uparrow R \\ \hat{\uparrow} \end{array}
\\[2em]
\rightharpoondown \left\{ \begin{array}{c} X \vdash N \\ \hline X \vdash\!\cdot^{+} \check{\downarrow} N \\ \hline X \vdash\!\cdot^{+} \downarrow N \end{array} \right. \begin{array}{l} \check{\downarrow} \\ \downarrow R \end{array}
\quad
\dot{\rightharpoondown} \left\{ \begin{array}{c} \dot{X} \cdot\!\vdash N \\ \hline \dot{X} \ \cdot\!\vdash\!\cdot^{+} \check{\downarrow} N \\ \hline \dot{X} \ \cdot\!\vdash\!\cdot^{+} \downarrow N \end{array} \right. \begin{array}{l} \check{\downarrow} \\ \downarrow R \end{array}
&
\leftharpoondown \left\{ \begin{array}{c} P \vdash \Delta \\ \hline \hat{\uparrow} P \ \cdot\!\vdash^{-} \Delta \\ \hline \uparrow P \ \cdot\!\vdash^{-} \Delta \end{array} \right. \begin{array}{l} \hat{\uparrow} \\ \uparrow L \end{array}
\quad
\dot{\leftharpoondown} \left\{ \begin{array}{c} P \vdash\!\cdot \dot{\Delta} \\ \hline \hat{\uparrow} P \ \cdot\!\vdash\!\cdot^{-} \dot{\Delta} \\ \hline \uparrow P \ \cdot\!\vdash\!\cdot^{-} \dot{\Delta} \end{array} \right. \begin{array}{l} \hat{\uparrow} \\ \uparrow L \end{array}
\\[2em]
\leftrightharpoons \left\{ \begin{array}{c} n \vdash^{-} n \\ \hline \downarrow n \ \cdot\!\vdash\!\cdot^{+} \check{\downarrow} n \\ \hline \downarrow n \ \cdot\!\vdash\!\cdot^{+} \downarrow n \end{array} \right. \begin{array}{l} \downarrow L \\ \downarrow R \end{array}
&
\rightleftharpoons \left\{ \begin{array}{c} p \vdash^{+} p \\ \hline \hat{\uparrow} p \ \cdot\!\vdash\!\cdot^{-} \uparrow p \\ \hline \uparrow p \ \cdot\!\vdash\!\cdot^{-} \uparrow p \end{array} \right. \begin{array}{l} \uparrow R \\ \uparrow L \end{array}
\end{array} \tag{24.24}$$

Proof By looking at the rules of **fD.LG**, the rules where the premise is normal but not the conclusion are $\downarrow_L$, $\downarrow_R$, $\uparrow_L$, $\uparrow_R$, $\check{\downarrow}$ and $\hat{\uparrow}$. Minimality of the proof forbids to stay in positive / negative non-focused phase for two sequents or more. It is easy to check that the cases $\leftharpoondown$ and $\rightleftharpoons$ are only possible when both precedent and succedent are a formula, which only happens when that formula is an atom.

Theorem 51 *Every **fD.LG**-derivation of a normal sequent S can be transformed into a derivation of $\lfloor S \rfloor$.*

Proof Set a derivation of a normal sequent S and call π its minimal proof. We translate every rule R of π by a rule $\lfloor R \rfloor$ of **fLG** by induction on π. As previously, we only treat a sufficient sample of the cases.

• Normal rules (i.e. both premises and conclusions are normal):

$$\left\lfloor \dfrac{}{p \vdash^{+} p}\, p\text{-Id} \right\rfloor \quad = \quad \dfrac{\dfrac{}{p \vdash \boxed{p}}\, Ax}{\lfloor p \rfloor \vdash \boxed{\lfloor p \rfloor}}$$

$$\left\lfloor \dfrac{\mathring{X} \overset{\circ}{\vdash}^{+} \mathring{P} \qquad \mathring{Y} \overset{\circ}{\vdash}^{+} \mathring{Q}}{\mathring{X} \hat{\otimes} \mathring{Y} \vdash^{+} \mathring{P} \otimes \mathring{Q}}\, \otimes_R \right\rfloor = \dfrac{\dfrac{\lfloor \mathring{X} \rfloor \vdash \boxed{\lfloor \mathring{P} \rfloor} \qquad \lfloor \mathring{Y} \rfloor \vdash \boxed{\lfloor \mathring{Q} \rfloor}}{\lfloor \mathring{X} \rfloor \hat{\otimes} \lfloor \mathring{Y} \rfloor \vdash \boxed{\lfloor \mathring{P} \rfloor \otimes \lfloor \mathring{Q} \rfloor}}\, \otimes_R}{\lfloor \mathring{X} \hat{\otimes} \mathring{Y} \rfloor \vdash \boxed{\lfloor \mathring{P} \otimes \mathring{Q} \rfloor}}$$

$$\left\lfloor \dfrac{\mathring{P} \hat{\otimes} \mathring{Q} \overset{\circ}{\vdash} \mathring{\Delta}}{\mathring{P} \otimes \mathring{Q} \overset{\circ}{\vdash} \mathring{\Delta}}\, \otimes_L \right\rfloor \quad = \quad \dfrac{\dfrac{\lfloor \mathring{P} \rfloor \hat{\otimes} \lfloor \mathring{Q} \rfloor \vdash \lfloor \mathring{\Delta} \rfloor}{\lfloor \mathring{P} \rfloor \otimes \lfloor \mathring{Q} \rfloor \vdash \lfloor \mathring{\Delta} \rfloor}\, \otimes_L}{\lfloor \mathring{P} \otimes \mathring{Q} \rfloor \vdash \lfloor \mathring{\Delta} \rfloor}$$

Display postulates of LG connectives are translated by themselves too.

• We translate processing proof-sections by one or two (de)focusing rules. Here, we take the case P is pure positive, so $\lfloor P \rfloor$ is **fLG**-positive and the application of μ^* and $\tilde{\mu}$ is allowed.

$$\left\lfloor \rightharpoonup \left\{ \begin{array}{c} X \vdash^{+} P \\ \hline \hat{\uparrow} X \ \cdot\!\vdash\!\cdot^{-} \uparrow P \\ \hline X \vdash\cdot \uparrow P \end{array} \begin{array}{l} \uparrow_R \\ \hat{\uparrow} \end{array} \right. \right\rfloor = \begin{array}{c} \lfloor X \rfloor \vdash \boxed{\lfloor P \rfloor} \\ \hline \lfloor X \rfloor \vdash \lfloor P \rfloor \\ \hdashline \lfloor X \rfloor \vdash \lfloor \uparrow P \rfloor \end{array} \ \mu^{*}$$

$$\left\lfloor \leftharpoondown \left\{ \begin{array}{c} P \vdash \Delta \\ \hline \hat{\uparrow} P \ \cdot\!\vdash^{-} \Delta \\ \hline \uparrow P \ \cdot\!\vdash^{-} \Delta \end{array} \begin{array}{l} \hat{\uparrow} \\ \uparrow_L \end{array} \right. \right\rfloor = \begin{array}{c} \lfloor P \rfloor \vdash \lfloor \Delta \rfloor \\ \hline \boxed{\lfloor P \rfloor} \vdash \lfloor \Delta \rfloor \\ \hdashline \boxed{\lfloor \uparrow P \rfloor} \vdash \lfloor \Delta \rfloor \end{array} \ \tilde{\mu}$$

$$\left\lfloor \rightleftharpoons \left\{ \begin{array}{c} p \vdash^{+} p \\ \hline \hat{\uparrow} p \ \cdot\!\vdash\!\cdot^{-} \uparrow p \\ \hline \uparrow p \ \cdot\!\vdash\!\cdot^{-} \uparrow p \end{array} \begin{array}{l} \uparrow_R \\ \uparrow_L \end{array} \right. \right\rfloor = \begin{array}{c} \lfloor p \rfloor \vdash \boxed{\lfloor p \rfloor} \\ \hline \lfloor p \rfloor \vdash \lfloor p \rfloor \\ \hline \boxed{\lfloor p \rfloor} \vdash \lfloor p \rfloor \end{array} \begin{array}{l} \mu^{*} \\ \tilde{\mu} \end{array}$$

Although translation $\lfloor \cdot \rfloor$ is not injective on structures, it is injective on derivations.

24.7 Conclusions

We observe that every connective in the language of **fD.LG** exhibits a core of minimal properties in any sub-algebra, namely it has finite arity and it is isotone or antitone in each coordinate. This leaves open the option that additional properties hold in some sub-algebras. Special sub-classes of $\mathbb{FP}.\mathbb{LG}$ algebras could then be captured by expanding the minimal calculus with opportune structural rules. If the minimal calculus is expanded with structural rules (see Greco et al., 2016 for the notion of *analytic axioms* in arbitrary signatures and a procedure to transform them into structural rules), the canonical cut-elimination provided in Sect. 24.5 is preserved. In future works We plan to provide a fully fledged account of the following natural conjecture: every displayable logic (with or without a lattice reduct) expanded with analytic structural rules closed under mutations has a focalized heterogenous multi-type display calculus complete w.r.t. a fully polarized algebraic semantics.

Funding Giuseppe Greco: NWO grant under the scope of the project "A composition calculus for vector-based semantic modelling with a localization for Dutch" (360-89-070).

Valentin D. Richard: UPS grant *Bourse de mobilité internationale de stage de l'Université Paris-Salcay*.

Appendix 1: Proof Complements

Proof (Proof of Proposition 5) Set $A_1, A_2 \in \mathcal{A}$, and $B_1', B_2' \in \mathcal{B}'$ such that $A_2 \leq_{\mathcal{A}} A_1$, $B_1' \leq_{\mathcal{B}'} B_2'$ and $A_1 \preccurlyeq B_1'$. Equation $A_1 \preccurlyeq B_1'$ is equivalent to $L(A_1) \preccurlyeq_{\mathcal{B}} B_1'$, and as $\preccurlyeq_{\mathcal{B}}$ is a weakening relation, we have $L(A_1) \preccurlyeq_{\mathcal{B}} B_2'$. This last equation is equivalent (by adjunction) to $A_1 \preccurlyeq_{\mathcal{A}} R(B_2')$ and as $\preccurlyeq_{\mathcal{A}}$ is a weakening relation, we have $A_2 \preccurlyeq_{\mathcal{A}} R(B_2')$, which is equivalent to $A_2 \preccurlyeq B_2'$. Therefore, $\preccurlyeq$ is a weakening relation.

Proof (Proof that $\mathring{\precsim}$ is a weakening relation (Definition 7)) We have $\mathring{\precsim} = \cdot\!\preceq \sqcup \preceq \sqcup \preceq\!\cdot$ a relation on $(\mathcal{A} \sqcup \mathcal{A}') \times (\mathcal{B} \sqcup \mathcal{B}')$. Set $A_1, A_2 \in \mathcal{A} \sqcup \mathcal{A}'$ and $B_1, B_2 \in \mathcal{B} \sqcup \mathcal{B}'$ such that $A_2 \leq_{\mathcal{A} \sqcup \mathcal{A}'} A_1, B_1 \leq_{\mathcal{B} \sqcup \mathcal{B}'} B_2$ and $A_1 \mathring{\precsim} B_1$. We only show how to get $A_2 \mathring{\precsim} B_1$, the other side is similar by symmetry of the problem.

- If A_1 and A_2 are both in $\mathcal{A}$ (resp. $\mathcal{A}'$), then $A_2 \leq_{\mathcal{A} \sqcup \mathcal{A}'} A_1$ is equivalent to $A_2 \leq_{\mathcal{A}} A_1$ (resp. $A_2 \leq_{\mathcal{A}'} A_1$) and $A_1 \mathring{\precsim} B_1$ is equivalent to $A_1 \preceq\!\cdot B_1$ or $A_1 \preceq B_1$ (resp. $A_1 \cdot\!\preceq B_1$). In all cases, because $\cdot\!\preceq$, $\preceq$ and $\preceq\!\cdot$ are all weakening relations, we have $A_2 \cdot\!\preceq B_1$, $A_2 \preceq B_1$ or $A_2 \preceq\!\cdot B_1$, hence $A_2 \mathring{\precsim} B_1$.
- If $A_2 \in \mathcal{A}$ and $A_1 \in \mathcal{A}'$, then we have $A_2 \preccurlyeq_{\mathcal{A}} A_1$ and $A_1 \cdot\!\preceq B_1$. Therefore $A_2 \preceq B_1$, because $\preccurlyeq_{\mathcal{A}} \cdot\!\preceq \subseteq \preceq$ by hypothesis, hence $A_2 \mathring{\precsim} B_1$.
- The case $A_2 \in \mathcal{A}'$ and $A_1 \in \mathcal{A}$ is impossible because the union is disjoint and $\preccurlyeq_{\mathcal{A}}$ is only from $\mathcal{A}$ to $\mathcal{A}'$.

The following four lemmas provide detailed complements to the proof of completeness by showing that the equivalence relations $\approx_s$ for $s \in \{\mathbb{P}, \dot{\mathbb{P}}, \mathbb{N}, \dot{\mathbb{N}}\}$ are congruences are thus that the operations and weakening relations/orders defined on them are well-defined.

Lemma 52 *Given a formula A and a structure Ψ, we write $\mathsf{Str}(A)$ the structure obtained by turning the connectives of A into their structural counterparts, and $\mathsf{Fm}(\Psi)$ the formula obtained by turning the connectives of Ψ into their operational counterpart. Transformations $\lfloor\!\lfloor\ \rfloor\!\rfloor$ and $\lceil\!\lceil\ \rceil\!\rceil$ enjoy the following property:*

$$\begin{gathered}
\lfloor\!\lfloor A \rfloor\!\rfloor = \lfloor\!\lfloor \mathsf{Str}(A) \rfloor\!\rfloor \quad \text{if } A \text{ is a } \mathcal{F}\text{-formula} \\
\lfloor\!\lfloor \Psi \rfloor\!\rfloor = \mathsf{Fm}(\Psi) \quad \text{if } \Psi \text{ is a } \mathcal{G}\text{-structure} \\
\lfloor\!\lfloor \mathring{X} \,\hat{\otimes}\, \mathring{Y} \rfloor\!\rfloor = \lfloor\!\lfloor \mathring{X} \rfloor\!\rfloor \otimes \lfloor\!\lfloor \mathring{Y} \rfloor\!\rfloor \quad \lfloor\!\lfloor \mathring{X} \,\hat{\oslash}\, \mathring{\Delta} \rfloor\!\rfloor = \lfloor\!\lfloor \mathring{X} \rfloor\!\rfloor \,\hat{\oslash}\, \lceil\!\lceil \mathring{\Delta} \rceil\!\rceil \quad \lfloor\!\lfloor \mathring{\Delta} \,\hat{\obslash}\, \mathring{Y} \rfloor\!\rfloor = \lceil\!\lceil \mathring{\Delta} \rceil\!\rceil \,\hat{\obslash}\, \lfloor\!\lfloor \mathring{Y} \rfloor\!\rfloor \\
\lfloor\!\lfloor \hat{\uparrow} X \rfloor\!\rfloor = \hat{\uparrow} \lfloor\!\lfloor X \rfloor\!\rfloor \qquad \lfloor\!\lfloor \hat{\upharpoonleft} \dot{X} \rfloor\!\rfloor = \hat{\upharpoonleft} \lfloor\!\lfloor \dot{X} \rfloor\!\rfloor \\
\lceil\!\lceil A \rceil\!\rceil = \lceil\!\lceil \mathsf{Str}(A) \rceil\!\rceil \quad \text{if } A \text{ is a } \mathcal{G}\text{-formula} \\
\lceil\!\lceil \Psi \rceil\!\rceil = \mathsf{Fm}(\Psi) \quad \text{if } \Psi \text{ is a } \mathcal{F}\text{-structure} \\
\lceil\!\lceil \mathring{\Delta} \,\check{\oplus}\, \mathring{\Gamma} \rceil\!\rceil = \lceil\!\lceil \mathring{\Delta} \rceil\!\rceil \,\check{\oplus}\, \lceil\!\lceil \mathring{\Gamma} \rceil\!\rceil \quad \lceil\!\lceil \mathring{X} \,\check{\backslash}\, \mathring{\Delta} \rceil\!\rceil = \lfloor\!\lfloor \mathring{X} \rfloor\!\rfloor \,\check{\backslash}\, \lceil\!\lceil \mathring{\Delta} \rceil\!\rceil \quad \lceil\!\lceil \mathring{\Delta} \,\check{/}\, \mathring{Y} \rceil\!\rceil = \lceil\!\lceil \mathring{\Delta} \rceil\!\rceil \,\check{/}\, \lfloor\!\lfloor \mathring{Y} \rfloor\!\rfloor \\
\lceil\!\lceil \check{\downarrow} \Delta \rceil\!\rceil = \check{\downarrow} \lceil\!\lceil \Delta \rceil\!\rceil \qquad \lceil\!\lceil \check{\downharpoonleft} \dot{\Delta} \rceil\!\rceil = \check{\downharpoonleft} \lceil\!\lceil \dot{\Delta} \rceil\!\rceil \\
\lfloor\!\lfloor p \rfloor\!\rfloor = \lceil\!\lceil p \rceil\!\rceil = p \qquad \lfloor\!\lfloor n \rfloor\!\rfloor = \lceil\!\lceil n \rceil\!\rceil = n
\end{gathered}$$

Proof Unfolding Definition 29 directly gives these results.

Lemma 53 *If $\mathring{X} \overset{\circ}{\vdash} \Delta$ (resp. $X \overset{\circ}{\vdash} \mathring{\Delta}$) is derivable, then* $\dfrac{\mathring{X} \overset{\circ}{\vdash} \Delta}{\mathring{X} \overset{\circ}{\vdash} \mathsf{Fm}(\Delta)}$ $\left(\textit{resp.}\ \dfrac{X \overset{\circ}{\vdash} \mathring{\Delta}}{\mathsf{Fm}(X) \overset{\circ}{\vdash} \mathring{\Delta}}\right)$ *is derivable.*

Proof By induction on Δ (resp. X), by successively applying translation rules on all structural connectives of Δ (resp. X).

Lemma 54 *For every structure Ψ of sort s, the sequent $\lfloor\!\lfloor \Psi \rfloor\!\rfloor \; t_s \; \lceil\!\lceil \Psi \rceil\!\rceil$ is derivable, if it is defined.*

Proof By induction on Ψ. If Ψ is an atomic formula, $\lfloor\!\lfloor \Psi \rfloor\!\rfloor = \lceil\!\lceil \Psi \rceil\!\rceil$ so p-Id or n-Id is applicable.

If $\Psi = \mathring{X} \mathbin{\hat{\otimes}} \mathring{Y}$, let us develop the case $\mathring{X} = X$ and $\mathring{Y} = \dot{Y}$, the others being similar. We use the induction hypothesis on X and $\dot{Y}$. By assumption, $\lfloor\!\lfloor \dot{Y} \rfloor\!\rfloor$ exists, so $\dot{Y}$ begins with a shift: $\dot{Y} = \check{\downarrow}\Delta$ and we can then derive

$$
\dfrac{
\dfrac{\dfrac{}{\lfloor\!\lfloor X \rfloor\!\rfloor \vdash^{+} \lceil\!\lceil X \rceil\!\rceil}\,(\mathrm{IH})_X}{\lfloor\!\lfloor X \rfloor\!\rfloor \vdash^{+} \mathsf{Fm}(X)}\,\text{Lemma 52}
\qquad
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\lfloor\!\lfloor \dot{Y} \rfloor\!\rfloor \cdot\!\vdash\!\cdot^{+} \lceil\!\lceil \check{\downarrow}\Delta \rceil\!\rceil}\,(\mathrm{IH})_{\dot{Y}}}{\lfloor\!\lfloor \dot{Y} \rfloor\!\rfloor \cdot\!\vdash\!\cdot^{+} \check{\downarrow}\lceil\!\lceil \Delta \rceil\!\rceil}\,\text{Lemma 52}}{\lfloor\!\lfloor \dot{Y} \rfloor\!\rfloor \cdot\!\vdash \lceil\!\lceil \Delta \rceil\!\rceil}\,\check{\downarrow}}{\lfloor\!\lfloor \dot{Y} \rfloor\!\rfloor \cdot\!\vdash \mathsf{Fm}(\Delta)}\,\text{Lemma 53}}{\lfloor\!\lfloor \dot{Y} \rfloor\!\rfloor \cdot\!\vdash\!\cdot^{+} \check{\downarrow}\mathsf{Fm}(\Delta)}\,\check{\downarrow}}{\lfloor\!\lfloor \dot{Y} \rfloor\!\rfloor \cdot\!\vdash\!\cdot^{+} \downarrow\mathsf{Fm}(\Delta)}\,\downarrow_R}{\lfloor\!\lfloor \dot{Y} \rfloor\!\rfloor \cdot\!\vdash\!\cdot^{+} \mathsf{Fm}(\check{\downarrow}\Delta)}\,\text{Lemma 52}
}{
\dfrac{\lfloor\!\lfloor X \rfloor\!\rfloor \mathbin{\hat{\otimes}} \lfloor\!\lfloor \dot{Y} \rfloor\!\rfloor \vdash^{+} \mathsf{Fm}(X) \otimes \mathsf{Fm}(\check{\downarrow}\Delta)}{\lfloor\!\lfloor X \mathbin{\hat{\otimes}} \dot{Y} \rfloor\!\rfloor \vdash^{+} \lceil\!\lceil X \mathbin{\hat{\otimes}} \check{\downarrow}\Delta \rceil\!\rceil}\,\text{Lemma 52}
}\,\otimes_R
$$

The other LG-connectives work similarly.

If $\Psi = \check{\downarrow}\Delta$, we use the induction hypothesis on Δ to get

$$
\dfrac{\dfrac{\dfrac{\lfloor\!\lfloor \Delta \rfloor\!\rfloor \vdash^{-} \lceil\!\lceil \Delta \rceil\!\rceil}{\mathsf{Fm}(\Delta) \vdash^{-} \lceil\!\lceil \Delta \rceil\!\rceil}\,\text{Lemma 52}}{\downarrow\mathsf{Fm}(\Delta) \cdot\!\vdash\!\cdot^{+} \check{\downarrow}\lceil\!\lceil \Delta \rceil\!\rceil}\,\downarrow_L}{\lfloor\!\lfloor \check{\downarrow}\Delta \rfloor\!\rfloor \cdot\!\vdash\!\cdot^{+} \lceil\!\lceil \check{\downarrow}\Delta \rceil\!\rceil}\,\text{Lemma 52}
$$

$\hat{\uparrow}$ works dually.

If Ψ begins with a shift adjoint, Ψ is not in the domain of both $\lceil\!\lceil\;\rceil\!\rceil$ and $\lceil\!\lceil\;\rceil\!\rceil$.

Lemma 55 *For every derivable sequents $\lfloor\!\lfloor \Psi \rfloor\!\rfloor \; t \; \lceil\!\lceil \Phi \rceil\!\rceil$ and $\lfloor\!\lfloor \Phi \rfloor\!\rfloor \; t' \; \lceil\!\lceil \Psi' \rceil\!\rceil$, we can derive the cut*

$$
\dfrac{\lfloor\!\lfloor \Psi \rfloor\!\rfloor \; t \; \lceil\!\lceil \Phi \rceil\!\rceil \qquad \lfloor\!\lfloor \Phi \rfloor\!\rfloor \; t' \; \lceil\!\lceil \Psi' \rceil\!\rceil}{\lfloor\!\lfloor \Psi \rfloor\!\rfloor \; tt' \; \lceil\!\lceil \Psi' \rceil\!\rceil}
$$

where the composition tt' is determined by the sort of Ψ and Ψ'.

Proof The proof mimics the cut-elimination proof of Theorem 38. We proceed by double induction on $\omega \times \omega$ ordered lexicographically where the first coordinate denotes the complexity of the cut structure Φ, and the second denotes the depth of the cut.

If Φ is an atomic formula, $\lfloor\!\lfloor \Phi \rfloor\!\rfloor = \lceil\!\lceil \Phi \rceil\!\rceil$ is a formula, so we can proceed to a tt' cut of (24.8) (i.e. P-Cut, N-Cut, Pn-Cut or nN-Cut).

If $\Phi = \mathring{X} \hat{\otimes} \mathring{Y}$, we know that at the introduction of $\lceil\lceil \Psi \rceil\rceil = \mathsf{Fm}(\mathring{X}) \otimes \mathsf{Fm}(\mathring{Y})$, we have some sequent $\mathring{X}' \hat{\otimes} \mathring{Y}' \vdash^{+} \mathsf{Fm}(\mathring{X}) \otimes \mathsf{Fm}(\mathring{Y})$ and the proof

$$
\cfrac{\cfrac{\begin{array}{c}\vdots\, \pi_1 \\ \mathring{X}' \overset{\circ}{\vdash}{}^{+} \lceil\lceil \mathring{X} \rceil\rceil\end{array} \qquad \begin{array}{c}\vdots\, \pi_2 \\ \mathring{Y}' \overset{\circ}{\vdash}{}^{+} \lceil\lceil \mathring{Y} \rceil\rceil\end{array}}{\mathring{X}' \hat{\otimes} \mathring{Y}' \vdash^{+} \lceil\lceil \mathring{X} \rceil\rceil \otimes \lceil\lceil \mathring{Y} \rceil\rceil}\ \otimes_R}{\begin{array}{c}\vdots\, \pi \\ \lfloor\lfloor \Psi \rfloor\rfloor \ t\ \lceil\lceil \Phi \rceil\rceil\end{array}}
$$

We work in cases. If $t' = \overset{\circ}{\vdash}$ then We can then apply the induction hypothesis on $\mathring{X}$ and $\mathring{Y}$:

$$
\cfrac{\begin{array}{c}\vdots \vdash \pi_1 \\ \mathring{X}' \overset{\circ}{\vdash}{}^{+} \vdash \lceil\lceil \mathring{X} \rceil\rceil\end{array} \qquad \cfrac{\cfrac{\cfrac{\begin{array}{c}\vdots \vdash \pi_2 \\ \mathring{Y}' \overset{\circ}{\vdash}{}^{+} \vdash \lceil\lceil \mathring{Y} \rceil\rceil\end{array} \qquad \cfrac{\cfrac{\begin{array}{c}\vdots \vdash \\ \lfloor\lfloor \mathring{X} \hat{\otimes} \mathring{Y} \rfloor\rfloor \overset{\circ}{\vdash} \vdash \lceil\lceil \Psi' \rceil\rceil\end{array}}{\lfloor\lfloor \mathring{X} \rfloor\rfloor \hat{\otimes} \lfloor\lfloor \mathring{Y} \rfloor\rfloor \overset{\circ}{\vdash} \vdash \lceil\lceil \Psi' \rceil\rceil}\ \text{Lemma 52}}{\lfloor\lfloor \mathring{Y} \rfloor\rfloor \overset{\circ}{\vdash} \vdash \lfloor\lfloor \mathring{X} \rfloor\rfloor \check{\backslash} \lceil\lceil \Psi' \rceil\rceil}\ \hat{\otimes} \dashv \check{\backslash}}{\mathring{Y}' \overset{\circ}{\vdash} \vdash \lfloor\lfloor \mathring{X} \rfloor\rfloor \check{\backslash} \lceil\lceil \Psi' \rceil\rceil}\ (\text{IH})_{\mathring{Y}}}{\lfloor\lfloor \mathring{X} \rfloor\rfloor \hat{\otimes} \lfloor\lfloor \mathring{Y} \rfloor\rfloor \overset{\circ}{\vdash} \vdash \lceil\lceil \Psi' \rceil\rceil}\ \hat{\otimes} \dashv \check{\backslash}}{\lfloor\lfloor \mathring{X} \rfloor\rfloor \overset{\circ}{\vdash} \vdash \lceil\lceil \Psi' \rceil\rceil \check{/} \mathring{Y}'}\ \hat{\otimes} \dashv \check{/}}{\cfrac{\mathring{X}' \overset{\circ}{\vdash} \vdash \lceil\lceil \Psi' \rceil\rceil \check{/} \mathring{Y}'}{\begin{array}{c}\mathring{X}' \hat{\otimes} \mathring{Y}' \overset{\circ}{\vdash} \vdash \lceil\lceil \Psi' \rceil\rceil \\ \vdots \vdash \mu(\pi) \\ \lfloor\lfloor \Psi \rfloor\rfloor \ tt' \vdash \lceil\lceil \Psi' \rceil\rceil\end{array}}\ \hat{\otimes} \dashv \check{/}}\ (\text{IH})_{\mathring{X}}
$$

If $t' = \overset{\circ}{\vdash}{}^{+}$, i.e. $\lfloor\lfloor \mathring{X} \hat{\otimes} \mathring{Y} \rfloor\rfloor \overset{\circ}{\vdash}{}^{+} \vdash \lceil\lceil \Psi' \rceil\rceil$, notice that the last rule applied is one of the following: $\otimes_R$, $\downarrow_R$, $\hat{\uparrow} \dashv \check{\downharpoonright}$, $\hat{\uparrow} \dashv \check{\downarrow}$, $\hat{\upharpoonright} \dashv \check{\downarrow}$ and $\check{\downarrow}$.

If it is $\otimes_R$ we proceed again using the induction hypotheses $\mathring{X}$ and $\mathring{Y}$ in a straightforward way. If it is $\downarrow_R$ and $\check{\downarrow}$ it is immediate that we can reduce the depth of the cut by 1. Assume now that the rule used was $\hat{\uparrow} \dashv \check{\downarrow}$ or $\hat{\upharpoonright} \dashv \check{\downarrow}$ and so $\lceil\lceil \Psi' \rceil\rceil$ is of the form $\check{\downarrow} \lceil\lceil \Psi'' \rceil\rceil$. In that case we work as follows:

$$
\dfrac{\dfrac{\begin{matrix}\vdots\vdash\pi_1\\ \mathring{X}' \;\mathring{\vdash}^{+} \vdash \lceil\!\lceil \mathring{X} \rceil\!\rceil\end{matrix} \qquad \dfrac{\dfrac{\begin{matrix}\vdots\vdash\pi_2\\ \mathring{Y}' \;\mathring{\vdash}^{+} \vdash \lceil\!\lceil \mathring{Y} \rceil\!\rceil\end{matrix} \qquad \dfrac{\dfrac{\dfrac{\begin{matrix}\vdots\vdash\\ \lfloor\!\lfloor \mathring{X} \hat{\otimes} \mathring{Y} \rfloor\!\rfloor \;\mathring{\vdash}^{+} \vdash \check{\downarrow}\lceil\!\lceil \Psi'' \rceil\!\rceil\end{matrix}}{\lfloor\!\lfloor \mathring{X} \hat{\otimes} \mathring{Y} \rfloor\!\rfloor \;\mathring{\vdash} \vdash \lceil\!\lceil \Psi'' \rceil\!\rceil}\,\check{\downarrow}}{\lfloor\!\lfloor \mathring{X} \rfloor\!\rfloor \hat{\otimes} \lfloor\!\lfloor \mathring{Y} \rfloor\!\rfloor \;\mathring{\vdash} \vdash \lceil\!\lceil \Psi'' \rceil\!\rceil}\,\text{Lemma 52}}{\lfloor\!\lfloor \mathring{Y} \rfloor\!\rfloor \;\mathring{\vdash} \vdash \lfloor\!\lfloor \mathring{X} \rfloor\!\rfloor \check{\backslash} \lceil\!\lceil \Psi'' \rceil\!\rceil}\,\hat{\otimes} \dashv \check{\backslash}}{\mathring{Y}' \;\mathring{\vdash} \vdash \lfloor\!\lfloor \mathring{X} \rfloor\!\rfloor \check{\backslash} \lceil\!\lceil \Psi'' \rceil\!\rceil}\,(\mathrm{IH})_{\mathring{Y}}}{\dfrac{\lfloor\!\lfloor \mathring{X} \rfloor\!\rfloor \hat{\otimes} \lfloor\!\lfloor \mathring{Y} \rfloor\!\rfloor \;\mathring{\vdash} \vdash \lceil\!\lceil \Psi'' \rceil\!\rceil}{\lfloor\!\lfloor \mathring{X} \rfloor\!\rfloor \;\mathring{\vdash} \vdash \lceil\!\lceil \Psi'' \rceil\!\rceil \check{/} \mathring{Y}'}\,\hat{\otimes} \dashv \check{/}}\,\hat{\otimes} \dashv \check{\backslash}}{\dfrac{\dfrac{\mathring{X}' \;\mathring{\vdash} \vdash \lceil\!\lceil \Psi'' \rceil\!\rceil \check{/} \mathring{Y}'}{\mathring{X}' \hat{\otimes} \mathring{Y}' \;\mathring{\vdash} \vdash \lceil\!\lceil \Psi'' \rceil\!\rceil}\,\hat{\otimes} \dashv \check{/}}{\mathring{X}' \hat{\otimes} \mathring{Y}' \;\mathring{\vdash}^{+} \vdash \check{\downarrow}\lceil\!\lceil \Psi'' \rceil\!\rceil}\,\check{\downarrow}}\,(\mathrm{IH})_{\mathring{X}}
$$

$$
\begin{matrix}\vdots\vdash \mu(\pi)\\ \lfloor\!\lfloor \Psi \rfloor\!\rfloor \; tt' \vdash \check{\downarrow}\lceil\!\lceil \Psi'' \rceil\!\rceil\end{matrix}
$$

Finally if the rule was $\hat{\uparrow} \dashv \check{\downharpoonleft}$ then $\lceil\!\lceil \Psi' \rceil\!\rceil$ is of the form $\check{\downharpoonleft}\lceil\!\lceil \Psi'' \rceil\!\rceil$ and work as follows:

$$
\dfrac{\dfrac{\begin{matrix}\vdots\vdash\pi_1\\ \mathring{X}' \;\mathring{\vdash}^{+} \vdash \lceil\!\lceil \mathring{X} \rceil\!\rceil\end{matrix} \qquad \dfrac{\dfrac{\begin{matrix}\vdots\vdash\pi_2\\ \mathring{Y}' \;\mathring{\vdash}^{+} \vdash \lceil\!\lceil \mathring{Y} \rceil\!\rceil\end{matrix} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\begin{matrix}\vdots\vdash\\ \lfloor\!\lfloor \mathring{X} \hat{\otimes} \mathring{Y} \rfloor\!\rfloor \;\mathring{\vdash}^{+} \vdash \check{\downharpoonleft}\lceil\!\lceil \Psi'' \rceil\!\rceil\end{matrix}}{\hat{\uparrow}\lfloor\!\lfloor \mathring{X} \hat{\otimes} \mathring{Y} \rfloor\!\rfloor \;\mathring{\vdash}^{-} \vdash \lceil\!\lceil \Psi'' \rceil\!\rceil}\,\hat{\uparrow} \dashv \check{\downharpoonleft}}{\lfloor\!\lfloor \mathring{X} \hat{\otimes} \mathring{Y} \rfloor\!\rfloor \;\mathring{\vdash} \vdash \lceil\!\lceil \Psi'' \rceil\!\rceil}\,\hat{\uparrow}}{\lfloor\!\lfloor \mathring{X} \rfloor\!\rfloor \hat{\otimes} \lfloor\!\lfloor \mathring{Y} \rfloor\!\rfloor \;\mathring{\vdash} \vdash \lceil\!\lceil \Psi'' \rceil\!\rceil}\,\text{Lemma 52}}{\lfloor\!\lfloor \mathring{Y} \rfloor\!\rfloor \;\mathring{\vdash} \vdash \lfloor\!\lfloor \mathring{X} \rfloor\!\rfloor \check{\backslash} \lceil\!\lceil \Psi'' \rceil\!\rceil}\,\hat{\otimes} \dashv \check{\backslash}}{\mathring{Y}' \;\mathring{\vdash} \vdash \lfloor\!\lfloor \mathring{X} \rfloor\!\rfloor \check{\backslash} \lceil\!\lceil \Psi'' \rceil\!\rceil}\,(\mathrm{IH})_{\mathring{Y}}}{\dfrac{\lfloor\!\lfloor \mathring{X} \rfloor\!\rfloor \hat{\otimes} \lfloor\!\lfloor \mathring{Y} \rfloor\!\rfloor \;\mathring{\vdash} \vdash \lceil\!\lceil \Psi'' \rceil\!\rceil}{\lfloor\!\lfloor \mathring{X} \rfloor\!\rfloor \;\mathring{\vdash} \vdash \lceil\!\lceil \Psi'' \rceil\!\rceil \check{/} \mathring{Y}'}\,\hat{\otimes} \dashv \check{/}}\,\hat{\otimes} \dashv \check{\backslash}}{\dfrac{\dfrac{\dfrac{\mathring{X}' \;\mathring{\vdash} \vdash \lceil\!\lceil \Psi'' \rceil\!\rceil \check{/} \mathring{Y}'}{\mathring{X}' \hat{\otimes} \mathring{Y}' \;\mathring{\vdash} \vdash \lceil\!\lceil \Psi'' \rceil\!\rceil}\,\hat{\otimes} \dashv \check{/}}{\hat{\uparrow}\mathring{X}' \hat{\otimes} \mathring{Y}' \;\mathring{\vdash}^{-} \vdash \lceil\!\lceil \Psi'' \rceil\!\rceil}\,\hat{\uparrow}}{\mathring{X}' \hat{\otimes} \mathring{Y}' \;\mathring{\vdash}^{+} \vdash \check{\downharpoonleft}\lceil\!\lceil \Psi'' \rceil\!\rceil}\,\hat{\uparrow} \dashv \check{\downharpoonleft}}\,(\mathrm{IH})_{\mathring{X}}
$$

$$
\begin{matrix}\vdots\vdash \mu(\pi)\\ \lfloor\!\lfloor \Psi \rfloor\!\rfloor \; tt' \vdash \check{\downharpoonleft}\lceil\!\lceil \Psi'' \rceil\!\rceil\end{matrix}
$$

If $\Phi = \hat{\uparrow}X$, we use the same procedure, by induction hypothesis on X. The turnstile t' can only be $\mathring{\vdash}^{-}$. Here we develop the case where Ψ' is shifted.

$$
\dfrac{\dfrac{\vdots \vdash \pi'}{X' \vdash^{+} \vdash \lceil\!\lceil X \rceil\!\rceil}}{\hat{\uparrow} X' \cdot\!\vdash\!\cdot^{-} \vdash \uparrow \lceil\!\lceil X \rceil\!\rceil}\,\uparrow_R
\qquad \rightsquigarrow
\qquad
\dfrac{\vdots \vdash \pi}{\lfloor\!\lfloor \Psi \rfloor\!\rfloor\ t \vdash \lceil\!\lceil \Phi \rceil\!\rceil}
$$

$$
\dfrac{\dfrac{\dfrac{\vdots \vdash \pi'}{X' \overset{\circ}{\vdash}{}^{+} \vdash \lceil\!\lceil X \rceil\!\rceil}
\quad
\dfrac{\dfrac{\dfrac{\vdots \vdash}{\lfloor\!\lfloor \hat{\uparrow} X \rfloor\!\rfloor \cdot\!\vdash\!\cdot^{-} \vdash \lceil\!\lceil \Psi' \rceil\!\rceil}}{\hat{\uparrow} \lfloor\!\lfloor X \rfloor\!\rfloor \cdot\!\vdash\!\cdot^{-} \vdash \lceil\!\lceil \Psi' \rceil\!\rceil}\,\text{Lemma 52}}{\lfloor\!\lfloor X \rfloor\!\rfloor \vdash^{+} \vdash \check{\downharpoonleft} \lceil\!\lceil \Psi' \rceil\!\rceil}\,\hat{\uparrow} \dashv \check{\downharpoonleft}}{X' \vdash^{+} \vdash \check{\downharpoonleft} \lceil\!\lceil \Psi' \rceil\!\rceil}\,(\mathrm{IH})_X}{\hat{\uparrow} X' \cdot\!\vdash\!\cdot^{-} \vdash \lceil\!\lceil \Psi' \rceil\!\rceil}\,\hat{\uparrow} \dashv \check{\downharpoonleft}
$$

$$
\dfrac{\vdots \vdash \pi[\lceil\!\lceil \Psi' \rceil\!\rceil / \lceil\!\lceil \Phi \rceil\!\rceil]}{\lfloor\!\lfloor \Psi \rfloor\!\rfloor\ tt' \vdash \lceil\!\lceil \Psi' \rceil\!\rceil}
$$

The other cases are treated similarly.

Complements for the proof of completeness (Theorem 30) We prove that the equivalence relations $\approx_s$ are congruences, i.e. that they respect the operations and the orders. It will follow that the operations and weakening relations defined on these equivalence classes are well-defined. We only detail the case of $\otimes$ with one pure and one shifted premise, and $\preceq$, the rest being similar.

$$
\dfrac{\dfrac{\dfrac{\dfrac{X \approx_{\mathbb{P}} Y}{\lfloor\!\lfloor X \rfloor\!\rfloor \vdash^{+} \lceil\!\lceil Y \rceil\!\rceil} \quad \dfrac{\dot{X} \approx_{\dot{\mathbb{P}}} \dot{Y}}{\lfloor\!\lfloor \dot{X} \rfloor\!\rfloor \cdot\!\vdash\!\cdot^{+} \lceil\!\lceil \dot{Y} \rceil\!\rceil}}{\lfloor\!\lfloor X \rfloor\!\rfloor \hat{\otimes} \lfloor\!\lfloor \dot{X} \rfloor\!\rfloor \vdash^{+} \lceil\!\lceil Y \rceil\!\rceil \otimes \lceil\!\lceil \dot{Y} \rceil\!\rceil}\,\otimes_R}{\lfloor\!\lfloor X \hat{\otimes} \dot{X} \rfloor\!\rfloor \vdash^{+} \lceil\!\lceil Y \hat{\otimes} \dot{Y} \rceil\!\rceil}\,\text{Lemma 52}
\quad
\dfrac{\dfrac{\dfrac{X \approx_{\mathbb{P}} Y}{\lfloor\!\lfloor Y \rfloor\!\rfloor \vdash^{+} \lceil\!\lceil X \rceil\!\rceil} \quad \dfrac{\dot{X} \approx_{\dot{\mathbb{P}}} \dot{Y}}{\lfloor\!\lfloor \dot{Y} \rfloor\!\rfloor \cdot\!\vdash\!\cdot^{+} \lceil\!\lceil \dot{X} \rceil\!\rceil}}{\lfloor\!\lfloor Y \rfloor\!\rfloor \hat{\otimes} \lfloor\!\lfloor \dot{Y} \rfloor\!\rfloor \vdash^{+} \lceil\!\lceil X \rceil\!\rceil \otimes \lceil\!\lceil \dot{X} \rceil\!\rceil}\,\otimes_R}{\lfloor\!\lfloor Y \hat{\otimes} \dot{Y} \rfloor\!\rfloor \vdash^{+} \lceil\!\lceil X \hat{\otimes} \dot{X} \rceil\!\rceil}\,\text{Lemma 52}}{X \hat{\otimes} \dot{X} \approx_{\mathbb{P}} Y \hat{\otimes} \dot{Y}}
$$

$$
\dfrac{\dfrac{X \approx_{\mathbb{P}} Y}{\lfloor\!\lfloor X \rfloor\!\rfloor \vdash^{+} \lceil\!\lceil Y \rceil\!\rceil}
\quad
\dfrac{\lfloor\!\lfloor Y \rfloor\!\rfloor \vdash \lceil\!\lceil \Gamma \rceil\!\rceil \quad \dfrac{\Delta \approx_{\mathbb{N}} \Gamma}{\lfloor\!\lfloor \Gamma \rfloor\!\rfloor \vdash^{-} \lceil\!\lceil \Delta \rceil\!\rceil}}{\lfloor\!\lfloor Y \rfloor\!\rfloor \vdash \lceil\!\lceil \Delta \rceil\!\rceil}\,\text{Corollary 40}}{\lfloor\!\lfloor X \rfloor\!\rfloor \vdash \lceil\!\lceil \Delta \rceil\!\rceil}\,\text{Corollary 40}
$$

For every homogeneous turnstile t, reflexivity,[7] transitivity and antisymmetry of $t^{\mathbb{A}}$ are a consequent of Lemma 54, Corollary 40 and definition of $\approx_s$ respectively. For heterogeneous turnstiles t, the weakening property of $t^{\mathbb{A}}$ is a consequence of Corollary 40. The property (24.4) of Definition 8 is due to to rules $\hat{\uparrow} \dashv \check{\downharpoonleft}$ and $\hat{\upharpoonleft} \dashv \check{\downarrow}$ and (24.4) to rules $\check{\downarrow}$ and $\hat{\uparrow}$. The adjunction (24.6) straightforwardly hold thanks to the corresponding rules in (24.10). We only develop the example of property $\uparrow^{\mathbb{A}} \dashv \downharpoonleft^{\mathbb{A}}$:

[7] Reflexivity on structures Ψ such that $\lfloor\!\lfloor \Psi \rfloor\!\rfloor\ t\ \lceil\!\lceil \Psi \rceil\!\rceil$ is not defined is explicitly added.

$$
\begin{array}{cl}
\uparrow^{\mathbb{A}}[X]_{\approx_{\mathbb{P}}} \ \cdot\!\vdash\!\cdot^{-\mathbb{A}} \ [\dot{\Delta}]_{\approx_{\dot{\mathbb{N}}}} & \\
\hdashline
\lfloor\!\lfloor \hat{\uparrow} X \rfloor\!\rfloor \ \cdot\!\vdash\!\cdot^{-} \ \lceil\!\lceil \dot{\Delta} \rceil\!\rceil & \\
\hdashline
\hat{\uparrow}\lfloor\!\lfloor X \rfloor\!\rfloor \ \cdot\!\vdash\!\cdot^{-} \ \lceil\!\lceil \dot{\Delta} \rceil\!\rceil & \text{Lemma 52} \\
\hline
\lfloor\!\lfloor X \rfloor\!\rfloor \ \vdash^{+} \ \check{\downharpoonleft}\lceil\!\lceil \dot{\Delta} \rceil\!\rceil & \hat{\uparrow} \dashv \check{\downharpoonleft} \\
\hdashline
\lfloor\!\lfloor X \rfloor\!\rfloor \ \vdash^{+} \ \lceil\!\lceil \check{\downharpoonleft} \dot{\Delta} \rceil\!\rceil & \text{Lemma 52} \\
\hdashline
[X]_{\approx_{\mathbb{P}}} \ \vdash^{+\mathbb{A}} \ \downharpoonleft^{\mathbb{A}}[\dot{\Delta}]_{\approx_{\dot{\mathbb{N}}}} &
\end{array}
$$

Appendix 2: Symmetries

Lambek–Grishin calculus exhibits two main symmetries (Moortgat, 2009): an order-preserving left-right symmetry $\cdot^{\bowtie}$ and an order-reversing dual symmetry $\cdot^{\infty}$ represented in (24.25)[8]. We extend them to $\mathbb{FP.LG}$ and **fD.LG** by (24.26). The dual of a turnstile t is given by (24.26) through the turnstile interpretation of (24.12): $t^{\infty} = (t^{\mathbb{A}})^{\infty}$, e.g. $\vdash^{+\infty} = \vdash^{-}$.

$$
\bowtie \ \begin{array}{cccc} A \backslash C & A \otimes B & A \oplus B & C \oslash B \\ \hline\hline C / A & B \otimes A & B \oplus A & B \obslash C \end{array}
\qquad
\infty \ \begin{array}{ccc} A \backslash C & A \otimes B & C / B \\ \hline\hline C \oslash A & B \oplus A & B \obslash C \end{array}
\tag{24.25}
$$

$$
\infty \ \begin{array}{ccccc} \underline{\leq}^{+} & \underline{\preceq}\cdot^{+} & \cdot\underline{\leq}\cdot^{+} & \underline{\preceq}\cdot & \underline{\preceq} \\ \hline\hline \underline{\leq}^{-} & \cdot\underline{\preceq}^{-} & \cdot\underline{\leq}\cdot^{-} & \cdot\underline{\preceq} & \underline{\preceq} \end{array}
\tag{24.26}
$$

The presentation of **fD.LG** rules in Sect. 24.3.1 also reflects the dual symmetry. In Eq. (24.9) the dual of rule R is the one displayed on the opposite side of the page w.r.t. the vertical axis. Therefore we have the following property.

Proposition 56 *If $\Phi \ t \ \Psi$ is a derivable sequent, then $\Phi^{\bowtie} \ t \ \Psi^{\bowtie}$ and $\Psi^{\infty} \ t^{\infty} \ \Phi^{\infty}$ are also derivable.*

References

Abramsky, S., & Jagadeesan, R. (1994a). Games and full completeness for multiplicative linear logic. *Journal of Symbolic Logic, 59*(2), 543–574.

Abramsky, S., & Jagadeesan, R. (1994b). New foundations for the geometry of interaction. *Information and Computation, 111*(1), 53–119.

Abramsky, S., & Melliès, P.-A. (1999). Concurrent games and full completeness. In *14th Annual IEEE Symposium on Logic in Computer Science, Trento, Italy, July 2–5, 1999* (pp. 431–442). IEEE Computer Society.

[8] These definitions should be understood as $(A \otimes B)^{\bowtie} = B^{\bowtie} \otimes A^{\bowtie}$, $(A \otimes B)^{\infty} = B^{\infty} \oplus A^{\infty}$, etc.

Andreoli, J.-M. (1992). Logic programming with focusing proofs in linear logic. *Journal of Logic and Computation, 2*(3), 297–347.

Andreoli, J.-M. (2001). Focussing and proof construction. *Annals of Pure and Applied Logic, 107*(1), 131–163.

Bastenhof, A. (2012). Polarized Montagovian semantics for the Lambek-Grishin calculus. In P. de Groote, & M. J. Nederhof (Eds.), *Formal grammar*. *Lecture Notes in Computer Science* (Vol. 7395). Berlin: Springer.

Battilotti, G., Faggian, C., & Sambin, G. (2000). Basic logic: Reflection, symmetry, visibility. *Journal of Symbolic Logic,65*.

Benabou, J. (1973). *Les distributeurs: d'après le cours de questions spéciales de mathématique*. Rapport n. 33 du Séminaire de Mathématique Pure. Institut de mathématique pure et appliquée, Université Catholique de Louvain.

Bernardi, R., & Moortgat, M. (2007). Continuation semantics for symmetric categorial grammar. In D. Leivant & R. de Queiroz (Eds.), *Logic, language, information and computation* (pp. 53–71). Berlin: Springer.

Bilkova, M., Kurz, A., Petrisan, D., & Velebil, J. (2013). Relation lifting, with an application to the many-valued cover modality. *Logical Methods in Computer Science,9*(4:8), 1–48.

Birkhoff, G., & Lipson, J. D. (1970). Heterogeneous algebras. *Journal of Combinatorial Theory, 8*(1), 115–133.

Cockett, J. R. B., & Seely, R. A. G. (2007). Polarized category theory, modules, and game semantics. In *Theory and applications of categories No. 2* (Vol. 18, pp. 4–101).

Frittella, S., Greco, G., Kurz, A., Palmigiano, A., & Sikimić, V. (2014). Multi-type sequent calculi. In A. Indrzejczak, J. Kaczmarek, & M. Zawidski (Eds.), *Proceedings trends in logic XIII* (Vol. 13, pp. 81–93).

Galatos, N., & Jipsen, P. (2016). Weakening relation algebras and FL^2-algebras. In U. Fahrenberg, P. Jipsen, & M. Winter, (Eds.), *Relational and algebraic methods in computer science. RAMiCS 2020*. Lecture notes in computer science (No. 12062, pp. 117–133). Lódź University Press.

Galatos, N., & Jipsen, P. (2019). The structure of generalized bi-algebras and weakening relation algebras. Submitted.

Goré, R. (1998). Substructural logics on display. *Logic Journal of IGPL, 6*(3), 451–504.

Greco, G., Jipsen, P., & Liang, F., Palmigiano, A., & Tzimoulis, A. (2018). Algebraic proof theory for LE-logics. arXiv:1808.04642

Greco, G., Ma, M., Palmigiano, A., Tzimoulis, A., & Zhao, Z. (2016). Unified correspondence as a proof-theoretic tool. *Journal of Logic and Computation, 28*(7), 1367–1442.

Hamano, M., & Scott, P. (2007). A categorical semantics for polarized MALL. *Annals of Pure and Applied Logic, 145*(3), 276–313.

Hamano, M., & Takemura, R. (2010). A phase semantics for polarized linear logic and second order conservativity. *Journal of Symbolic Logic, 75*(77–102), 2010.

Jung, A., Kegelmann, M., & Moshier, A. M. (1999). Multi lingual sequent calculus and coherent spaces. *Fundamenta Informaticae, 37*(4), 369–412.

Kurz, A., Moshier, A. M., & Jung, A. (2019). Stone duality for relations. arXiv:1912.08418

Lambek, J. (1958). The mathematics of sentence structure. *The American Mathematical Monthly, 65*(3), 154–170.

Lambek, J. (1961). On the calculus of syntactic types. In R. Jakobson (Ed.), *Structure of language and its mathematical aspects*. *Proceedings of Symposia in Applied Mathematics* (Vol. XII, pp. 166–178). American Mathematical Society.

Laurent, O. (2004). A proof of the focusing property of linear logic. Unpublished note. Revised 2017.

Miller, D. (2004). *An overview of linear logic programming*. London mathematical society lecture note series (pp. 119–150). Cambridge: Cambridge University Press.

Moortgat, M. (2009). Symmetric categorial grammar. *Journal of Philosophical Logic, 38*(6), 681–710.

Moortgat, M., & Moot, R. (2012). Proof nets and the categorial flow of information. In A. Baltag, D. Grossi, A. Marcoci, B. Rodenhäuser, & S. Smets (Eds.), *Logic and Interactive RAtionality, Yearbook 2011, ILLC* (pp. 270–302). Amsterdam: University of Amsterdam.

Pratt, V. (1999). Chu spaces. School on category theory and applications, Coimbra (Vol. 21, 39–100).

Street, R. (1980). Fibrations in bicategories. *Cahiers de Topologie et Géométrie Différentielle Catégoriques, 21*(2), 111–160.

Wansing, H. (2002). *Sequent systems for modal logics* (Vol. 8, pp. 61–145). Dordrecht: Springer.

Chapter 25
On Strictifying Extensional Reflexivity in Compact Closed Categories

Peter Hines

Abstract This chapter has two related aims. The first is to study the categorical setting of Abramsky, Haghverdi, & Scott's untyped linear combinatory algebras (Abramsky et al., 2002), and the second is to relate this to the much more recent work by Abramsky & Heunen on Frobenius algebras in the infinitary setting (Abramsky & Heunen, 2010). The key to this is (extensional) reflexivity (i.e. the property of an object being isomorphic to its own internal hom $R \cong [R \to R]$). We first characterise extensionally reflexive objects in compact closed categories, then consider when & how this property may be 'strictified'—how we may give a monoidally equivalent category where the isomorphisms exhibiting reflexivity are in fact identity arrows. This results in small two-object compact closed categories consisting of a unit object and a single (non-unit) strictly reflexive object. We then move on to studying the endomorphism monoids of such objects from an algebraic rather than logical or categorical viewpoint. We demonstrate that these necessarily contain an interesting inverse monoid that may be thought of as Richard Thompson's iconic group $\mathcal{F}$ together with the equally iconic bicyclic monoid $\mathcal{B}$ of semigroup theory, with non-trivial interactions between the two derived from the Frobenius algebra identity—and claim this as a particularly significant example of the (unitless) Frobenius algebras of Abramsky and Heunen (2010). We first develop the theory from a purely theoretical point of view, then move on to develop concrete examples, based on the algebra and category theory behind (Girard, 1988a, b; Abramsky et al., 2002). The concrete examples we give are based on the traced monoidal category of partial injections, and reflexive objects in the compact closed category that results from applying the **Int** or **GoI** construction. We then give compact closed categories, monoidally equivalent to compact closed subcategories of **Int**(**pInj**), where this reflexivity is exhibited by identity arrows, and show how the above algebraic structures (Thompson's $\mathcal{F}$, the bicyclic monoid, and Frobenius algebras) arise in a fundamental manner.

This work is of course !( ) dedicated to Samson Abramsky.

P. Hines (✉)
Y.C.C.S.A., University of York, York, U.K.
e-mail: peter.hines@york.ac.uk

A. Palmigiano and M. Sadrzadeh (eds.), *Samson Abramsky on Logic and Structure in Computer Science and Beyond,* Outstanding Contributions to Logic 25,
https://doi.org/10.1007/978-3-031-24117-8_25

Keywords Compact closure · Categorical traces · Untyped logical systems · Reflexivity · Coherence · Strictification · Geometry of interaction · Frobenius algebras · Inverse monoids

25.1 Historical Background

The starting point for this chapter is Girard's Geometry of Interaction program—in particular the first two parts. It is by now well-established that compact closed categories model key aspects of this (see Abramsky et al., 2002, Haghverdi & Scott, 2005 for a good account), and this observation motivated the name of Abramsky's categorical **GoI** construction (Abramsky, 1996) (see Sect. 25.7).

However, this immediately poses an interesting puzzle. In (1988a), Girard makes the rather cryptic comment that his system 'forgets types', even though the stated aim was to produce a model of the polymorphically typed System $\mathcal{F}$, rather than a purely untyped system. The explanation seems to be that the GoI system moves from a rigidly typed system to an entirely untyped system, in order to build a more flexible (polymorphic) type system on top of this.[1] The claim that there is an untyped logical system at the core of Girard's GOI was borne out in Abramsky, Haghverdi, and Scott's paper (Abramsky et al., 2002) that gave an untyped combinatory logic (precisely, linear-exponential combinatory logic) based on the primitives from Girard's first two Geometry of Interaction papers (Girard, 1988a, b).

It is natural to wish to study this type-freeness via the 'objects as types' paradigm of categorical logic (Lambek & Scott, 1986). In the classical/Cartesian world, Lambek & Scott introduced, as models of untyped lambda calculus, the C-monoids of Lambek and Scott (1986), which may reasonably be viewed as monoids satisfying all of the axioms for Cartesian closure apart from the existence of a terminal object.[2]

Based on similar ideas, analogues of compact closure within monoids were studied in Hines (1997, 1999) (again motivated by J.-Y. Girard's first three Geometry of Interaction papers Girard, 1988a, b, 1995) and the claim was made that certain monoids derived from the GOI program satisfy, 'unitless analogues of compact closure'. This work was carried out independently of Abramsky et al. (2002); however, the underlying algebraic structures—based on Girard's original system—are identical.

[1] This is commonly studied via has become known as Hyland-Ong types (Hyland & Ong, 2000). E. Haghverdi & P. Scott also presented a Geometry of Interaction system that is decidedly typed in Haghverdi and Scott (2005); however, this current chapter concentrates on the purely untyped aspects of the Geometry of Interaction program, and (re-)introducing types would be a non-trivial subsequent step.

[2] The complete situation is slightly more subtle than this description suggests. There is a good case that C-monoids do indeed account for units, but these are implicit—a passage to the Karoubi envelope then makes them explicit. This is beyond the scope of this paper, but covered in Lambek and Scott (1986).

There are significant subtleties associated with this claim.[3] With Cartesian closure, we are left with the essential concepts even without the terminal object. The same—at least for the usual axiomatisation—does not apply to compact closure.

25.2 Compact Closure

In Kelly and Laplaza (1980), the abstract 2-categorical definition of a compact closed category is shown to have a concrete characterisation in terms of the existence of a duality and distinguished arrows. It is by now standard to take this as fundamental.

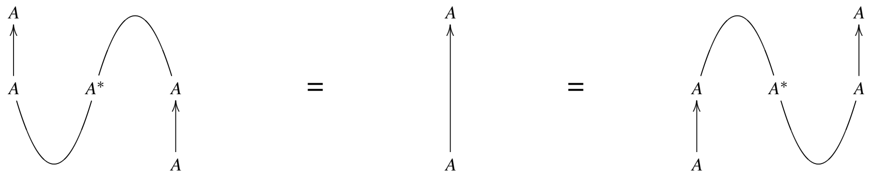
Definition 1 A symmetric monoidal category $(\mathcal{C}, \otimes, \sigma_{\_,\_}, I)$ is **compact closed** when it is equipped with:

- a **dual**—a contravariant monoidal functor $(\_)^* : \mathcal{C}^{op} \to \mathcal{C}$ satisfying $((\_)^*)^* = Id_{\mathcal{C}}$
- for all objects $A \in Ob(\mathcal{C})$, distinguished **unit** & **co-unit** arrows $\eta_A : I \to A \otimes A^*$ and $\epsilon_A : A^* \otimes A \to I$ that satisfy the **yanking axiom**

$$(1_A \otimes \epsilon)(\eta \otimes 1_A) \;=\; 1_A \;=\; (\epsilon_{A^*} \otimes 1_A)(1_A \otimes \eta_{A^*})$$

Using the usual diagrammatic conventions from Joyal and Street (1993a, b), the unit/co-unit arrows are drawn as 'cups' and 'caps'

giving the yanking axiom as



Remark 1 (*Interpretations and Examples*) The neat diagrammatic representation of the unit and co-unit arrows, and yanking, readily lead to numerous distinct interpretations. One of the earliest and most natural was for the unit and co-unit arrows to be interpreted as the **Axiom** and **Cut** rules of various forms of linear logic

[3] To complicate matters, the axiomatisation in Hines (1999) work was intended & stated to be equivalent to that of Hines (1997). Unfortunately, as recently pointed out by C. Heunen, several severe typo.s made their way into the final published version—this axiomatisation, at least, unambiguously does not capture compact closure!

$$\text{Axiom}: \frac{}{A \quad A^{\perp}} \qquad\qquad \text{Cut}: \frac{A^{\perp} \quad A}{}$$

This is the basis for the close connection between compact closure and the Geometry of Interaction discussed throughout, and the interpretation of yanking as cut-elimination is intuitively natural. The concrete examples we develop in Sect. 25.11 onwards are based on Girard's GoI system, and derived from the Int or GoI construction described in Sect. 25.7.

An alternative interpretation was found in Hines (2003, 2008), where algebraic models of dynamics of Turing machines were observed to be compact closed; the unit and co-unit model the changes of direction of the read-write head of a Turing machine as it moves over the tape, and the dual corresponds to interchanging the role of 'left' and 'right' in the definition of the dynamics.

A further interpretation may be found in the Categorical Quantum Mechanics program of Abramsky and Coecke (2005), where the unit and co-unit respectively correspond to the production of a maximally entangled state, and a (post-selected) measurement that results in the observation of this maximally entangled state.

Numerous other interpretations have been given—in particular, the pregroups of Lambek (1999) arise from dropping the requirement of symmetry (implicitly, considering non-commutative multiplicative linear logic) to give linguistic models with a distinctly logical flavour. See Buszkowski (2001, 2003) and Lambek (1999) for the original motivation from linear logic and the identification as 'non-symmetric compact closure'.

25.2.1 On the Definition(s) of Compact Closed Monoids

It is not immediate how a compact closed monoid should be defined, apart from in the highly degenerate case where the unique object is the unit object. In this setting, we recall the folklore that the subcategory of *any* monoidal category generated by its unit object is compact closed, with MacLane's distinguished unit arrows & their inverses trivially satisfying the axioms for the unit/co-unit of compact closure.

With Cartesian closure, we are generally happy to accept that the existence of a terminal object is a relatively minor part of the definition (and indeed may even be hidden in the structure of C-monoids—see Footnote 2), and consider that a Cartesian closed monoid is one satisfying all the other parts of the definition. By contrast, erasing the unit object from the definition of compact closure leaves us with nothing—not even the duality (some authors, notably Joyal et al., 1996, define the duality in terms of the unit/co-unit arrows).

The approach taken in Hines (1997, 1999) was to give an axiomatisation of compact closure for semi-monoidal categories (i.e. categories satisfying all MacLane's axioms except for those relating to the unit object—see Definition 6) that

1. does not mention the unit, and
2. is equivalent to to the usual definition in the presence of a unit.

Regardless of whether the logical interpretation of working without a unit is desirable, a fundamental question needs to be asked:

How do we know this definition of *unitless compact closure* is the 'correct' definition? Is it sufficient for it to reduce to the standard definition in the presence of a unit?

Expanding on this question, we may ask what the implications would be of two such axiomatisations of unitless compact closure that both reduce to the usual definition in the presence of a unit, but are provably inequivalent? How should we decide between them?

Such an pair of provably distinct axiomatisations has recently been established in joint (currently unpublished) work by the author and C. Heunen. The concrete examples of Hines (1997, 1999) satisfy both sets of axioms, and the defining arrows of these axiomatisations correspond in each case to distinct interesting logical, computational, or categorical features that we would be unhappy to exclude in the general case; there is simply no reason to prefer one axiomatisation over the other, and every reason to consider structures satisfying both axiomatisations.

The existence of concrete examples implies these are compatible, so may follow from a single set of axioms. However, this illustrates that reduction to the usual axioms in the presence of a unit is therefore not sufficient. What is also needed is some notion of completeness—that any other compatible axiom scheme that is equivalent to compact closure in the presence of a unit is a consequence of it.

The axioms given in Hines (1997) do not have such a 'completeness' property and, at best, cannot be the whole story. The project described above remains ongoing.

25.3 From Closed Monoids to Reflexive Objects

Nothing in the above discussion precludes compact closed categories where all non-unit objects are isomorphic (concrete examples given in part 4. of Proposition 10), or even (in a small category) identical—giving two-object compact closed categories, with a single non-unit object: see Corollary 2 and the concrete examples of Definition 24. However, it is also worthwhile to take a step back and again ask the obvious question,

"What was the purpose of Lambek & Scott introducing Cartesian Closed Monoids?"

The immediate answer of course is, 'to model untyped lambda calculus'. Looking slightly deeper (Lambek & Scott, 1986) observes that Cartesian closure, combined with the fact that there is only one object, means that the unique object c is necessarily isomorphic to its own function space c^c (i.e. is a *reflexive object*), and this is the key (Scott, 1980) to unrestricted application & abstraction.

Remark 2 (*Untyped systems—a change of emphasis*) From a categorical logic 'objects as types' perspective, it seems natural that an 'untyped version of $\mathcal{X}$' should be modeled by an "$\mathcal{X}$–monoid", provided this can be defined in a satisfactory manner.

When the intention is to model the unrestricted application/abstraction of an untyped lambda calculus, from precisely the same perspective the notion of reflexivity is the desirable property. These two concepts—although closely related—are not identical.

We therefore take as fundamental the notion of objects that are 'isomorphic to their own internal hom'. The most general setting in which reflexivity may be defined is that of the 'non-monoidal closed categories' of Laplaza (1977) (see Remark 7 for the justification for such a general setting). These are defined simply as categories with an internal hom functor (i.e. without explicit reference to any monoidal tensor or adjunction between hom and tensor) that satisfies some fairly intricate coherence conditions. Laplaza's definition, of course, includes the more familiar monoidal closed categories as a special case.

The following is taken from Scott (1980) (see also Hyland, 2017), although we make some necessary (see Remark 3) changes in terminology.

Definition 2 Let $(C, [\_ \to \_])$ be a closed category, in the sense of Laplaza (1977) (this includes monoidal closed categories as a special case).

An **(extensionally) reflexive object**, or simply **reflexive object** $R \in Ob(C)$ is one that is isomorphic to its own internal hom, so $R \cong [R \to R]$.

Explicitly, reflexive objects are equipped with mutually inverse isomorphisms:

- The **app** isomorphism $\rhd : R \to [R \to R]$
- The **lam** isomorphism $\lhd : [R \to R] \to R$

satisfying $\lhd \rhd = 1_R$ and $\rhd \lhd = 1_{[R \to R]}$.

We say that R is **weakly** or **intensionally reflexive** when $[R \to R]$ is a retract of R, so $\rhd \lhd = 1_{[R \to R]}$, but $\lhd \rhd = e^2 = e$ is an idempotent of $C(R, R)$. Note that this breaks with convention somewhat, by allowing for extensional reflexivity to be a very special case of weak reflexivity.

Finally, we say a reflexive object is **strictly reflexive** when $\lhd$ and $\rhd$ are identity arrows. Strictly reflexive objects are of course extensional.

Remark 3 (*A conflict of terminology*) It is more common (e.g. Hyland, 2017; Scott, 1980) to refer to (extensional) reflexivity as *"strict reflexivity"*, and to what we call "weak reflexivity" simply as *"reflexivity"*. This immediately sets us up for a strong and probably unavoidable conflict of notation; we are interested in 'strictifying' reflexivity in the sense of strictification within categorical coherence—i.e. giving a suitable equivalence of categories under which the isomorphisms exhibiting (extensional) reflexivity become identity arrows.

The usage of the term "strict" is very well-established in both fields, so this conflict of notation is unavoidable—all we can do is point out the conventions we use!

Remark 4 (*Weak, strong, & strict reflexivity in models of combinatory logic*) A key aim of this chapter is to study the categorical/algebraic setting for the untyped combinatory logic of Abramsky, Haghverdi & Scott, which we claim is that of strict reflexivity. In some sense, compact closed monoids would be too extravagant a setting. By contrast to the logical interpretation of Girard (1988a, b), the system of

Abramsky et al. (2002) has no connectives (& hence no need for tensors), and no negation (& hence no non-degenerate duality is required);[4] in a λ calculus, the key notions are those of application & abstraction. From that viewpoint, it is appropriate to concentrate on reflexivity.

However, combinatory logics are 'lower-level' systems than λ calculii. The notion of abstraction itself is derived from more primitive operations—an operation that acts like lambda-abstraction is built up using combinators. Our claim is that in models of untyped combinatory logic, we will not only observe reflexivity, but reflexivity that is *strict*—the isomorphisms that exhibit reflexivity are identities.

Finally, although reflexivity already features heavily in Abramsky et al. (2002), it is intensional, or weak, reflexivity. Girard's original system (implicitly) considered both—we are therefore 'plugging a gap'. It is also difficult to see how reflexivity that is not exhibited by isomorphisms may be strictified, unless we pass to the Karoubi envelope (i.e. idempotent splitting). This will then provide reflexive objects that are extensionally reflexive—the setting we consider.

25.3.1 Reflexive Objects of Compact Closed Categories

Reflexive objects of compact closed categories have a particularly simple characterisation.

Definition 3 Let $(\mathcal{C}, \_ \otimes \_, I)$ be a monoidal category. An object $N \in Ob(\mathcal{C})$ is called **self-similar** or **pseudo-idempotent** when it satisfies $N \cong N \otimes N$. The isomorphisms exhibiting this self-similarity are unique up to unique isomorphism (Hines, 2016), and commonly referred to as the **code** and **decode** arrows $\lhd \in \mathcal{C}(N \otimes N, N)$ and $\rhd \in \mathcal{C}(N, N \otimes N)$ respectively.

Similarly, let $(\mathcal{D}, (\_)^*)$ be a category with a dual. An object $S \in Ob(\mathcal{D})$ is called **self-dual** when it satisfies $S \cong S^*$; there is no standard notation or terminology for arrows exhibiting self-duality.

Examples of self-similar objects are given in Sect. 25.14, and self-similar objects of compact closed categories in Lemma 6, Proposition 10, and Definition 24. A characterisation of self-dual objects in a large class of compact closed categories is given in Corollary 5, with the example of Lemma 6 being particularly relevant.

Remark 5 The notions 'pseudo-idempotency' and 'self-similarity' are precisely equivalent; the different terminology simply arises from different fields. Some authors (e.g. Fiore & Leinster, 2010) use the term 'idempotent' for what we call 'pseudo-idempotent', although Joyal and Kock (2013) defines an idempotent to satisfy the much stricter condition $\lhd \otimes I_N = 1_N \otimes \lhd$. For consistency, we use the term *self-similar object* rather than *pseudo-idempotent*, unless usage is very well-established.

[4] However, units do play a rôle in Abramsky et al. (2002), so reflexivity within a setting including these is essential.

Self-duality and self-similarity together are enough to characterise reflexive objects of compact closed categories.

Lemma 1 *Let* $(C, \_ \otimes \_, \sigma_{\_,\_}, I, (\ )^*)$ *be a compact closed category. The reflexive objects of* C *are precisely the self-dual self-similar objects.*

Proof Both parts of this proof are based on the very special form the internal hom takes in a compact closed category, as $[A \to B] \stackrel{def.}{=} A^* \otimes B$.
($\Rightarrow$) Let $R \in Ob(C)$ be a reflexive object, so there exist mutually inverse arrows $\mathrel{|\!\rhd} \in C(R, [R \to R])$ and $\mathrel{\lhd\!|} \in C([R \to R], R)$. As the dual is a contravariant monoidal functor, the following diagram commutes:

$$\begin{tikzcd}
R^* & {[R \to R]^*} \arrow[l, "\mathrel{|\!\rhd}^*"'] \arrow[d, "1_{[R\to R]^*}"] \\
R^* \arrow[u, "1^*_R"] \arrow[r, "\mathrel{\lhd\!|}^*"'] & {[R \to R]^*}
\end{tikzcd}$$

Appealing to the identity $[R \to R] = R^* \otimes R$, we then derive isomorphisms exhibiting the self-duality of R.

$$\begin{tikzcd}
R \arrow[r, "\mathrel{|\!\rhd}"] \arrow[d, "1_R"'] & R^* \otimes R \arrow[r, "\sigma_{R^*,R}"] & R \otimes R^* \arrow[r, "\mathrel{|\!\rhd}^*"] & R^* \arrow[d, "1_{R^*}"] \\
R & R^* \otimes R \arrow[l, "\mathrel{\lhd\!|}"] & R \otimes R^* \arrow[l, "\sigma_{R,R^*}"] & R^* \arrow[l, "\mathrel{\lhd\!|}^*"]
\end{tikzcd}$$

Once we have established that $R \cong R^*$ is self-dual, self-similarity or pseudo-idempotency of R follows as $R \cong [R \to R] = R^* \otimes R \cong R \otimes R$. Thus reflexive objects are self-dual and self-similar.
($\Leftarrow$) This direction is even simpler. Let R be both self-dual and self-similar. Then $R \ \cong\ R \otimes R \ \cong\ R^* \otimes R = [R \to R]$ and so R is reflexive. $\square$

Remark 6 (*Examples of self-similar and reflexive objects*) It is not difficult to give examples of either self-similar or reflexive objects. From a general viewpoint, self-similarity dates back at least as far as Hilbert's parable of the Grand Hotel and Cantor's work on foundations of set theory (see Yanofsky, 2013 for a good overview), and is significantly prefigured by Galileo's 'infinity paradox'. As observed in Hines (1997) it is also a key feature of Girard's Geometry of Interaction program, and the term 'self-similar' is used in a wide range of algebraic fields (e.g. Elder, 2013) motivated by a connection with fractal structures.

Relevant concrete examples, along with Hilbert-hotel style bijections based on the natural numbers, are given in Sect. 25.14 and Definitions 20.

For any self-similar object X of a compact closed category, its tensor with its dual, $X \otimes X^*$ is not only self-similar, but self-dual. Thus, provided a compact closed category does indeed contain self-similar objects, we have instant access to reflexive objects.[5]

Concrete examples of reflexive objects are given in Corollary 11. However, at this point we cannot simply declare that we are done, and have established a suitable categorical setting for the type-freeness evident in Girard (1988a, b) and Abramsky et al. (2002). It is notable that all the action takes place within a single algebraic structure. Instead, we treat reflexivity like many other categorical properties and consider how reflexivity may be 'strictified' so that the isomorphisms exhibiting it become, under a suitable equivalence of categories, identity arrows.

25.4 Strictification of Reflexivity

What we now need is a notion of 'strictifying' reflexivity; moving—via a categorical equivalence—from a setting in which an object is reflexive up to some pair of isomorphisms, to a setting in which reflexivity may be exhibited by identity arrows.

Definition 4 Given a reflexive object R of a closed category $(\mathcal{C}, [\ \to\ ])$, we define a **strictification** of the reflexivity of R to be:

- a small closed category $\mathcal{S}$ and a faithful functor of closed categories $\Gamma : \mathcal{S} \to \mathcal{C}$,
- a reflexive object $N \in Ob(\mathcal{S})$ where $\Gamma(N) = R$,
- a closed category $\mathcal{D}$, equivalent to $\mathcal{S}$, in which this reflexivity is exhibited by identity arrows.

The above slightly convoluted description is required to avoid violating the principle of equivalence; in practice, we will simply refer to 'a small closed subcategory of $\mathcal{C}$ that contains R', along with an equivalent closed subcategory where reflexivity is exhibited by identities.

When R is a reflexive object of a *monoidal* closed category $(\mathcal{C}, \otimes, [\_ \to \_], I)$ we say that such a strictification is a **monoidal strictification** when the relevant equivalence of categories is a monoidal equivalence, and the faithful functor $\Gamma : \mathcal{S} \to \mathcal{C}$ is a monoidal functor.

Remark 7 (*Closure, and monoidal closure*) Closure within the 'non-monoidal closed categories' found in the somewhat obscure paper (Laplaza, 1977) of M.

[5] It is perhaps easier to point out compact closed categories that *do not* have reflexive objects. The compact closed category $(\mathbf{Hilb_{fd}}, \otimes)$ of finite-dimensional Hilbert spaces with tensor product, as studied in the categorical quantum mechanics program, is a notable example. We refer to Abramsky et al. (1999) for the obstacles to defining compact closure in the infinite-dimensional (& hence reflexive) case. It is tempting, although highly speculative, to relate this to the problems early pioneers of quantum computing had in attempting to define a 'fully quantum' universal computer (Deutsch, 1985; Myers, 1997).

Laplaza is undoubtedly the most 'pure' form of closure available. Additional categorical features associated with closure (such as the Cartesian product used in Lambek & Scott, 1986) often correspond to additional logical or computational structures (in Lambek & Scott, 1986, the product re-appears in logical form, as 'surjective pairing'), and a notion of closure defined without reference to any other operations beyond an internal hom. does not impose any additional notions on a logical or lambda-calculus interpretation.

Despite this, we consider (monoidal) strictification of reflexivity within compact closed categories. Thus, we require an equivalence of categories preserves the unit object, elements of objects and names of arrows, and indeed the notion of monoidal well-pointedness.

This is partly for theoretical reasons; we are studying the setting of Abramsky et al. (2002) and Girard (1988a, b), rather than some more abstract notion of combinatory logic or lambda model. It is also partly for practical reasons; Laplaza (1977) requires what have been referred to as 'monstrous' coherence conditions, which would make any such strictification a decidedly non-trivial task—we refer to Hyland (2014) for what is possibly the closest approach.

We are therefore considering what is possibly the easiest case: monoidal strictification for a class of reflexive objects in compact closed categories, including those used in Girard (1988a, b) and Abramsky et al. (2002).

25.5 Strict Duality, and Strict Self-similarity

As the notion of reflexivity in compact closed categories simply splits into the notions of self-duality and self-similarity, we first consider what it means to have strict versions of both of these.

25.5.1 Strict Self-duality

The notion of self-duality in the strict setting is well-studied:

Definition 5 A **dagger** on a (monoidal) category C is defined in Selinger (2007) to be a contravariant (monoidal) functor $(\ )^\dagger : C^{op} \to C$ that is the identity on objects and satisfies $\left((\ )^\dagger\right)^\dagger = Id_C$. Thus, a dagger can be thought of as a strict version of self-duality. Simply as notation, we extend this definition slightly, and say that a dual $(\ )^* : C^{op} \to C$ is a **dagger at an object** $X \in Ob(C)$ when X is strictly self-dual. This is precisely equivalent to stating that the monoidal subcategory generated by X is a dagger category.

Remark 8 The best-known examples of daggers on compact closed categories are undoubtedly those found in the 'categorical quantum mechanics' program (Abramsky & Coecke, 2005), where the dual in the compact closed category of finite-dimensional Hilbert spaces is a dagger. However, we are not able to use this setting to provide reflexive objects (see Footnote 5).

The notion of a dagger has occasionally been criticised as an 'evil' concept (i.e. breaking the principle of equivalence by relying on a notion of equality between objects). Leaving aside the possibility of the purist leveling the same criticism at the identity functor, we take an alternative viewpoint: self-duality is simply a categorical property for which we may consider—in the appropriate setting—a strict form.

The following is then a simple corollary of Lemma 1 above:

Corollary 1 *Let N be a reflexive strictly self-dual object of a compact closed category $(C, \otimes, (\ )^*)$, so $(\ )^*$ is a dagger at N. Then $N \in Ob(C)$ is self-similar, and the lam and app arrows exhibiting reflexivity*

$$\lhd\!\!\!| : [N \to N] \to N \;\; and \;\; |\!\!\!\rhd : N \to [N \to N]$$

are also code/decode arrows exhibiting self-similarity

$$\lhd : N \otimes N \to N \;\; and \;\; \rhd : N \otimes N \to N$$

Proof The internal hom in a compact closed category is given by $[N \to N] = N^* \otimes N$. As the dual is a dagger at N, we derive $[N \to N] = N \otimes N$. The lam and app isomorphisms are then mutually inverse bijections exhibiting $N \cong N \otimes N$. However, these are unique up to unique isomorphism (Hines, 2016), so our result follows. □

Corollary 2 *Let N be a reflexive strictly self-dual object of a compact closed category. Then*

1. *The monoidal subcategory generated by N is compact closed,*
2. *All non-unit objects of this subcategory are isomorphic,*
3. *When N is strictly reflexive, this subcategory has precisely two objects, one of which is the unit object.*

(Concrete examples of 1. and 2. are given in Part 4. of Proposition 10, and of Part 3. in Definition 24).

This convergence of the elements of reflexivity is particularly relevant for compact closed categories arising from the **Int** or **GoI** construction, where all self-dual objects are by construction isomorphic to some strictly self-dual object.

25.5.2 Strict Self-similarity

It is now inevitable that we consider how self-similarity may be strictified. We are fortunate to be able to rely on a pre-existing coherence theorem and strictification procedure (Hines, 2016). However, somewhat inconveniently for our aim of finding a *monoidal* strictification of reflexivity, this firmly and unavoidably lives within the theory of *semi-monoidal* categories.

25.5.3 Monoidal and Semi-monoidal Categories

A semi-monoidal category is simply one that satisfies all MacLane's axioms for a monoidal category, except for those concerning the unit object. The following definitions may be found in Kock (2008).

Definition 6 A **semi-monoidal category** is a category $\mathcal{C}$ with a functor $\_ \otimes \_ : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$ that is associative up to an object-indexed family of natural isomorphisms $\tau_{X,Y,Z} : X \otimes (Y \otimes Z) \to (X \otimes Y) \otimes Z$ satisfying MacLane's pentagon condition

$$(\tau_{W,X,Y} \otimes 1_Z) \, \tau_{W,X\otimes Y,Z} \, (1_W \otimes \tau_{X,Y,Z}) = \tau_{W\otimes X,Y,Z} \, \tau_{W,X,Y\otimes Z}$$

A functor between semi-monoidal categories that (strictly) preserves the tensor is a **(strict) semi-monoidal functor**. We assume the obvious definition of **semi-monoidal equivalence of categories**.

A **semi-monoidal monoid** is simply a small semi-monoidal category with precisely one object. These lie on the border of algebra and category theory, and interesting algebraic structures frequently arise from core categorical ones in this setting (e.g. Sect. 25.5.4).

What is lost in the passage from monoidal to semi-monoidal categories is the following notions:

Definition 7 Let $(\mathcal{C}, \otimes, I)$ be a monoidal category. The **elements** of X are members of the homset $\mathcal{C}(I, X)$, and the elements of the unit object $\mathcal{C}(I, I)$ are known as **abstract scalars** (Abramsky, 2005), by analogy with the case of vector spaces and linear maps.

When the abelian monoid of abstract scalars is the singleton, we say that I is a **trivial unit**, or that $(\mathcal{C}, \otimes, I)$ has **trivial scalars**.

When $\mathcal{C}$ is a monoidal closed category, the elements of $[X \to Y]$ are known as **names** and are in 1:1 correspondence with the members of the homset $\mathcal{C}(X, Y)$. Finally, when $(\mathcal{C}, \otimes, I)$ is *compact closed*, there is also a 1:1 correspondence between $\mathcal{C}(X, Y)$ and the elements $\mathcal{C}([X \to Y], I)$; these are known as **co-names**.

We treat monoidal categories as a special class of semi-monoidal categories. When discussing potential unit objects in semi-monoidal categories, it is common to rely on A. Saavedra's characterisation[6] of units (Rivano, 1972), as laid out in Kock (2008)

[6] It is notable that Saavedra never explicitly stated that this precisely characterised units in the sense of MacLane-Kelly. This observation was made by J. Kock who also laid out the basics of the theory of semi-monoidal categories (Kock, 2008). He later extended this to a more general theory of 'weak units' in collaboration with A. Joyal (Joyal and Kock, 2013).

and Joyal and Kock (2013). Thus ‘being a unit’ is a property that an object may have, rather than a part of the definition.

Definition 8 A **(Saavedra) unit** in a semi-monoidal category $(\mathcal{C}, \otimes, \tau)$ is an object $U \in Ob(\mathcal{C})$ that is both **pseudo-idempotent** and **cancellable** i.e. it is self-similar, and the functors $(U \otimes \_), (\_ \otimes U) : \mathcal{C} \to \mathcal{C}$ are fully faithful.

The theory of Saavedra units is particularly relevant for semi-monoidal monoids, where it allows us to characterise those semi-monoidal monoids whose unique object is a unit object. The following is taken from Hines (2016).

Proposition 1 *Let $(M, \star, \alpha)$ be a semi-monoidal monoid. Then $\star$ is strictly associative (i.e. $\alpha = 1_M$) iff the unique object of M is a unit object.*

25.5.4 An Algebraic Interlude

Proposition 1 raises an obvious question: the canonical associativity isomorphisms for a (non-degenerate) semi-monoidal monoid form a non-trivial group—is it the same in every case, and if so, which group this is?

One of the most familiar and well-studied objects in group theory is Richard Thompson’s group $\mathcal{F}$. This was originally defined in terms of a representation, as the group of homeomorphisms of the unit interval that are piece-wise linear and order-preserving, non-differentiable only at a finite number of dyadic rationals, and have slope of the form 2^k, $k \in \mathbb{N}$ on differentiable sections.

We use the definition as a group presentation:

Definition 9 **Thompson’s group** $\mathcal{F}$ is defined by

$$\mathcal{F} = \langle x_0, x_1, x_2, \ldots \ : \ x_i^{-1} x_j x_i = x_{j+1} \ \forall \ i < j \rangle$$

Note that this is not a *minimal* presentation; it is well-established that $\{x_0, x_1\}$ generates the whole of $\mathcal{F}$. However, the required relators are significantly less intuitive.

It is by now folklore that group of canonical associativity isomorphisms in a (non-unit) semi-monoidal monoid is precisely Thompson’s $\mathcal{F}$. This—or at least statements equivalent to this claim—have been presented and rediscovered many times, and the following list is not exhaustive!

Remark 9 (*Connections between $\mathcal{F}$ and coherence for associativity*) As early as 1973, R. Thompson and J. McKenzie noted (McKenzie and Thompson, 1973) a connection with ‘*associativity laws*’. Lawson (2007) considered a class of semi-monoidal monoids in the special case where the tensor $\_ \star \_$ admits projection/injection arrows (as studied in Hines, 1997, 1999; Lawson, 1998), and demonstrated that the group of canonical isomorphisms[7] is precisely $\mathcal{F}$. Dehornoy (1996) considered $\mathcal{F}$ abstractly, and noted that, 'The only [non-trivial] relations in this presentation of $\mathcal{F}$ correspond to the well-known MacLane-Stasheff pentagon'. Fiore and Leinster (2010) considered the strict monoidal category freely generated by a generic (pseudo-)idempotent and proved that its symmetry group is precisely $\mathcal{F}$. Brin (2005) talks about, 'the resemblance of the usual coherence theorems with Thompson's group $\mathcal{F}$', and this observation was used in Hines (2016) to note that—at least in the 'free' case, canonical associativity isomorphisms for a semi-monoidal monoid are precisely a copy of $\mathcal{F}$. M. Lawson recently updated his paper (Lawson, 2007) in Lawson (2020), and gave a construction of $\mathcal{F}$ based on (finite) maximal binary prefix codes; any categorically-minded reader will identify this construction as functorial, and the relevant prefix codes as a representation of MacLane's monogenic category W (excluding the unit object).

The following theorem and outline proof is presented with no claim to originality; it is given simply because the monoidal category theory is often implicit, rather than explicit, in several of the references of Remark 9 above. We also wish to connect the generators of the presentation given in Definition 9 above with the category theory.

Theorem 1 *Let $(M, \_ \star \_)$ be a (non-unit) semi-monoidal monoid. The canonical associativity isomorphisms for $\_ \star \_$ form a copy of Thompson's group $\mathcal{F}$.*

Proof (*OUTLINE*)
Let us denote the canonical associator for $\_ \star \_$ by $\alpha \in M$, and define $\{x_j\}_{j \in \mathbb{N}}$ inductively by $x_0 = \alpha$, and $x_{i+1} = 1 \star x_i$. Functoriality of $\_ \star \_$ and MacLane's pentagon then immediately give the defining relations of Thompson's $\mathcal{F}$, as $x_i^{-1} x_j x_i = x_{j+1} \ \forall \ i < j$. Thus the group of canonical associativity isomorphisms contains a homomorphic image of $\mathcal{F}$. However, a standard fact (e.g. Brin, 1996) about $\mathcal{F}$ is that it has no non-abelian quotients, and so this is precisely a copy of $\mathcal{F}$. Finally, we may appeal to MacLane's pentagon to demonstrate that all canonical associativity isomorphisms for $\_ \star \_$ are generated by the set $\{x_j\}_{j \in \mathbb{N}}$. □

Remark 10 Thompson's group $\mathcal{F}$ is itself, of course, a semi-monoidal monoid. This observation was made—in non-categorical terms—by Brown (2006) where he describes a group homomorphism $\mu : \mathcal{F} \times \mathcal{F} \to \mathcal{F}$ that is "associative up to conjugation by the generator x_0". It is also remarkable that—despite Brown (2006) being phrased in entirely non-categorical terms—K. Brown also proves that $\mu(1, \_)$ and $\mu(\_, 1)$ are injective, but cannot be surjective. In our terms, he is establishing precisely that J. Kock's conditions for a Saavedra unit are *not* satisfied by the unique object of $\mathcal{F}$.

[7] A curious feature of Lawson (2007) is that the link between the given representation of $\mathcal{F}$ and associativity isomorphisms is not made explicit by the author, despite having (along with other authors) described and used the same operations as associativity isomorphisms in previous work (Lawson, 1998; Hines, 1997).

25.5.5 Functors Between Monoidal & Semi-monoidal Categories

We return to category theory, and define several functors between (large) categories of monoidal and semi-monoidal categories.

Definition 10 We define the large category **MonCat** to have, as objects, all monoidal categories. Arrows $\Gamma \in \mathbf{MonCat}((\mathcal{C}, \otimes_{\mathcal{C}}, I_{\mathcal{C}}), (\mathcal{D}, \otimes_{\mathcal{D}}, I_{\mathcal{D}}))$ are 'lax non-collapsing' monoidal functors $\Gamma : \mathcal{C} \rightarrow \mathcal{D}$ satisfying, for all $X, Y \in Ob(\mathcal{C})$,

$$\Gamma(X \otimes_{\mathcal{C}} Y) \cong \Gamma(X) \otimes_{\mathcal{D}} \Gamma(Y) \quad \text{and} \quad \Gamma(X) \cong I_{\mathcal{D}} \text{ iff } X \cong I_{\mathcal{C}}$$

We define the large category **SemiMonCat** to have, as objects, all semi-monoidal categories. An arrow $\Delta \in \mathbf{SemiMonCat}((\mathcal{G}, \otimes_{\mathcal{G}}), (\mathcal{H}, \otimes_{\mathcal{H}}))$ is a functor $\Delta : \mathcal{G} \rightarrow \mathcal{H}$ satisfying $\Delta(A \otimes_{\mathcal{G}} B) \cong \Delta(A) \otimes_{\mathcal{H}} \Delta(B)$, for all $A, B \in Ob(\mathcal{G})$. There is an obvious faithful functor $\iota : \mathbf{MonCat} \rightarrow \mathbf{SemiMonCat}$ corresponding to the triviality that every monoidal category is also semi-monoidal.

Given a semi-monoidal category $(\mathcal{C}, \otimes)$, we may adjoint a strict unit object simply by taking the categorical coproduct with the terminal monoidal category, and extending the tensor by strictness. This process is functorial; we denote this functor by $(\_)_{+I} : \mathbf{SemiMonCat} \rightarrow \mathbf{MonCat}$. Note that by construction, $(\mathcal{C}, \otimes)_{+I}$ has trivial scalars, and no non-trivial elements.

Going in the other direction, given a monoidal category $(\mathcal{M}, \_ \otimes \_, I)$, let us denote by $(\mathcal{M}, \_ \otimes \_)_{-I}$ the full subcategory consisting of all non-unit objects. This 'forgetting the unit' process is also functorial; we denote this by $(\_)_{-I} : \mathbf{MonCat} \rightarrow \mathbf{SemiMonCat}$. The composite $((\ )_{+I})_{-I} = Id_{\mathbf{SemiMonCat}}$. This is, of course, a one-sided inverse; $((\ )_{-I})_{+I} : \mathbf{MonCat} \rightarrow \mathbf{MonCat}$ is certainly not the identity functor. Rather, it has the effect of deleting elements (and therefore, when appropriate, names and co-names). This endofunctor on **MonCat** will prove important; we refer to it as the **de-element functor**, and denote it by $(\ )_{-\mathcal{E}\mathcal{F}\mathcal{A}\mathcal{W}} = \big((\ )_{-I}\big)_{+I} : \mathbf{MonCat} \rightarrow \mathbf{MonCat}$.

Remark 11 Given a compact closed category $(\mathcal{C}, \otimes)$, the de-element functor certainly does not result in a compact closed category; $(\mathcal{C}, \otimes)_{-\mathcal{E}\mathcal{F}\mathcal{A}\mathcal{W}}$ has lost elements, including abstract scalars, names, co-names, & the distinguished unit/co-unit maps. However, it may still be closed in the sense of Laplaza (1977).

Remark 12 A natural question is whether, given some abelian monoid U, we may 'extend' a monoidal category with trivial scalars to one where the abstract scalars are taken from this abelian monoid? This is a surprisingly non-trivial task; a procedure for doing so in the case of traced and compact closed categories is given in Abramsky (2005). We discuss this further in Sect. 25.17.

25.6 A Coherence Theorem for Self-similarity

We now describe the relevant points of the coherence theorem for self-similarity (Hines, 2016), which lives firmly within the category **SemiMon**.

The theory of semi-monoidal monoids is essentially interchangeable with the theory of self-similarity in semi-monoidal categories. Let $(M, \star)$ be a monoid with a semi-monoidal tensor; the unique object $m \in Ob(M)$ of this monoid is clearly self-similar, as $m \star m = m$. Similarly, the endomorphism monoid of any strictly self-similar object is clearly a semi-monoidal monoid.[8] Thus semi-monoidal monoids may be considered to be a 'strict' form of self-similar objects. This observation was formalised in the coherence theorem and strictification procedure of Hines (2016), where the following useful results may be found:

Theorem 2 *Let $N \in Ob(C)$ be a self-similar object of a semi-monoidal category $(C, \otimes)$, and let $\lhd \in C(N \otimes N, N)$ and $\rhd \in C(N, N \otimes N)$ be the unique (up to unique isomorphism)* code *and* decode *bijections exhibiting this self-similarity. Then*

1. *The operation defined on the endomorphism monoid of N by*

$$f \star g = \lhd(f \otimes g)\rhd \quad \forall f, g \in C(N, N)$$

   is a semi-monoidal tensor on $C(N, N)$.
2. *There is a semi-monoidal equivalence of categories between*
   a. *the semi-monoidal subcategory of $(C, \otimes)$ generated by N,*
   b. *the semi-monoidal monoid $(C(N, N), \_ \star \_)$.*
3. *As a consequence of Proposition 1, the equivalence of categories of Point 2 maps* strict *to* non-strict *associativity in the case where $\_ \otimes \_$ is strict but N is not a unit object.*

Proof We refer to Hines (2016) for proofs of the above. These proofs are based on giving necessary and sufficient conditions for the commutativity of diagrams over a certain class of primitives (the tensor $\otimes$ and its canonical isomorphisms, the tensor $\star$ and its canonical isomorphisms, and the code/decode arrows). □

Remark 13 (*On the choice of code/decode arrows*) It is natural to wonder whether the choice of code/decode arrows is significant—in particular, if we are intending to strictify *reflexivity*, should we not ensure that the distinguished $\mathrel{\rhd\!\!\!|}$ and $\mathrel{|\!\!\!\lhd}$ of Corollary 1 are chosen, as opposed to some other isomorphisms with the same source/target?

The answer to this lies in the observation of Hines (2016) that code/decode arrows are unique up to unique isomorphism. Thus, all strictifications of some self-similar object are semi-monoidally equivalent (algebraically, they are isomorphic

[8] A special case of this is strictly reflexive objects in compact closed categories. However, this is due to the special form that the internal hom takes in a compact closed category; the same need not be true in arbitrary monoidal closed categories, and is certainly not the case for non-monoidal closed categories.

semi-monoidal monoids), regardless of the choice of code/decode arrows. Changing the code/decode isomorphisms may be seen as moving to an isomorphic representation of the same structure—this is studied in more detail in Hines (2014), where an analogy between this and changes of basis in matrix representations is formalised.

Corollary 3 *The endomorphism monoid of every non-unit self-similar object contains a copy of Thompson's $\mathcal{F}$. (Note that this result holds even if the semi-monoidal category in question is strictly associative).*

The 'adjoining a strict unit' functor $(\ )_{+I} : \mathbf{SemiMonCat} \to \mathbf{MonCat}$ of Definition 10 does not give us, for free, a monoidal, rather than semi-monoidal, equivalence. Rather, we have the following simple corollary:

Corollary 4 *Let $(C, \otimes, I)$ be a monoidal category, and denote by $(N^{\otimes}, \otimes, I)$ the full monogenic monoidal subcategory generated by some self-similar object $N \cong N \otimes N$. Then there exists a monoidal equivalence of categories between $(C, \star)_{+I}$ and $(N^{\otimes}, \otimes, I)_{-\frac{\mathcal{E}}{\mathcal{F}}\frac{\mathcal{A}}{\mathcal{W}}}$.*

Remark 14 As Corollary 4 emphasises, the strictification process for self-similarity of Hines (2016) naturally lives within **SemiMonCat** rather than **MonCat**; applying it in the monoidal setting gives an equivalence of 'de-elemented' categories. Some work is needed in order to use it to strictify reflexivity in a compact closed category, even given a self-similar object with a dagger.

The key to doing this is the canonical construction of *compact closed categories* from *symmetric traced monoidal categories*, given by either the **Int** construction of Joyal et al. (1996), or the **GoI** construction of Abramsky (1996). Notably, this construction equips the compact closed category with a suitable range of elements (& hence names and co-names), even when the underlying traced category has none.

25.7 From Traced Categories to Compact Closure

It is well known that the two constructions of *compact closed categories* from *symmetric traced monoidal categories* (the **Int** construction of Joyal et al., 1996 and the **GoI** construction of Abramsky, 1996) are equivalent in the symmetric case, although Joyal et al. (1996) also considered the more general braided/tortile monoidal categories. Despite this, they used significantly different conventions, and the equivalence between the two was given by Haghverdi (2000) (see also Footnote 8.).

Somewhat perversely for a volume dedicated to the work of S. Abramsky, we will work with the conventions of Joyal et al. (1996) instead.[9]

[9] Although this has become convention, we do not simply act out of peer pressure. Rather, we consider the conventions of Joyal et al. (1996) and Abramsky (1996) to give two different, but isomorphic compositions (along with tensors, duals, etc.) on the same underlying structure. Our claim is that interesting category theory and algebra may arise out of taking a 2-category or bi-category

25.7.1 Categorical Traces

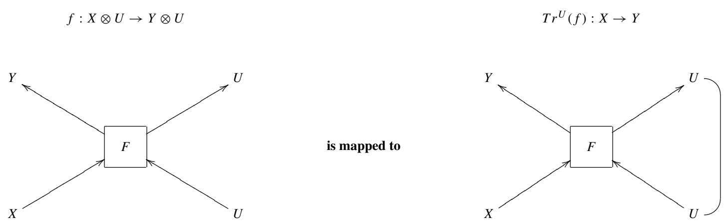
We start with the definition of a traced symmetric monoidal category, as a special case of the more general braided monoidal categories of Joyal et al. (1996).

Definition 11 A **trace** on a symmetric monoidal category $(\mathcal{C}, \_ \otimes \_, \sigma_{\_,\_}, I)$ is an object-indexed family of mappings of homsets $Tr^U_{X,Y} : \mathcal{C}(X \otimes U, Y \otimes U) \to \mathcal{C}(X, Y)$ that is natural in X and Y, dinatural in U, and satisfies the following axioms:

- **(Vanishing I)** $Tr^I_{X,Y}(\,) = Id_{\mathcal{C}(X,Y)}$, for all $X, Y \in Ob(\mathcal{C})$.
- **(Vanishing II)** $Tr^{U\otimes V}_{X,Y} = Tr^U_{X,Y}\left(Tr^V_{X\otimes U, Y\otimes U}(f)\right)$ for all $f : X \otimes U \otimes V \to Y \otimes U \otimes V$.
- **(Yanking)** $Tr^U_{U,U}(\sigma_{U,U}) = 1_U$.
- **(Superposing)** $Tr^U_{X,Y}(f) \otimes g = Tr^U_{X\otimes A, Y\otimes B}\left((1_Y \otimes \sigma_{B,U})(f \otimes g)(1_X \otimes \sigma_{A,U})\right)$ for all $f : X \otimes U \to Y \otimes U$, $g : A \to B$.

Remark 15 A consequence of the Vanishing I axiom is that when $(\mathcal{C}, \otimes, I)$ has trivial scalars, traces are uniquely determined by their action on non-unit objects. The Vanishing II axiom is also sometimes known as the ‘confluence axiom’, for obvious reasons.

The diagrammatic calculus for traced & compact closed categories is well-established in Joyal and Street (1993a, b) and Joyal et al. (1996), where traces appear as feedback loops:



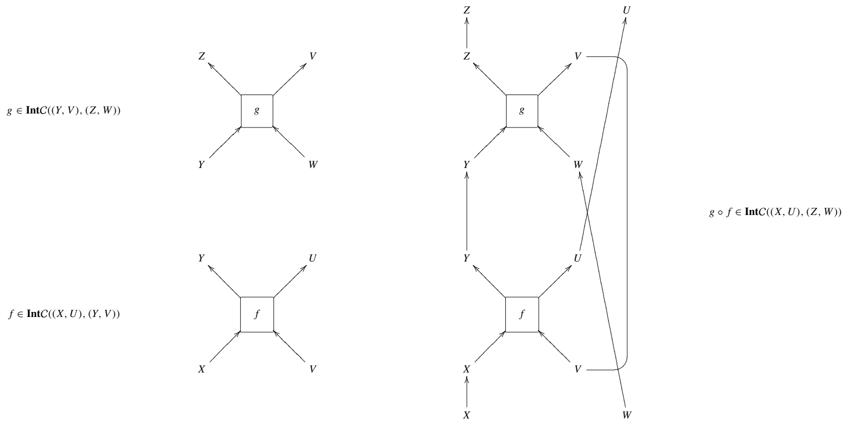
approach, and considering the interaction of the two. This was (implicitly) the approach taken in the identification in Hines (1997) of the cut/cut-elimination procedure in Girard’s Geometry of Interaction as compact closure. This was written when the author was unaware of the work of Abramsky (1996) and based on an early draft of Joyal et al. (1996). This referred to a second, isomorphic, ‘vertical’ composition on hom-sets of **Int**$(\mathcal{C})$, introduced in order to transform the ‘horizontal’ composition of Joyal et al. (1996) into something that directly matched the resolution formula and cut-elimination procedure within Girard’s Geometry of Interaction system. In retrospect, the (isomorphic) ‘horizontal’ and ‘vertical’ compositions are those derived from the respective **Int** and **GoI** constructions of Joyal et al. (1996) and Abramsky (1996). See Remark 23 for more details on this.

25.8 From Traces to Compact Closure

We now give an exposition of the construction of compact closed categories from traced symmetric monoidal categories

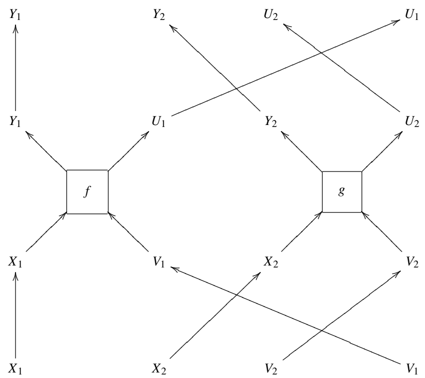
Definition 12 Let $(\mathcal{C}, \otimes, \sigma, I, Tr^{-}_{-,-}(\ ))$ be a traced symmetric monoidal category. The compact closed category $(\mathbf{Int}\mathcal{C}, \_\Box\_, (\ )^*, \epsilon_{\_}, \eta_{\_})$ is defined as follows:

1. **(Objects)** An object (X, U) of $\mathbf{Int}\mathcal{C}$ is a pair of objects of $\mathcal{C}$.
2. **(Arrows)** The homset $\mathbf{Int}\mathcal{C}((X, U), (Y, V))$ is precisely $\mathcal{C}(X \otimes V, Y \otimes U)$.
3. **(Composition)** Given arrows $f : (X, U) \to (Y, V)$ and $g : (Y, V) \to (Z, W)$, their composite $g \circ f \in \mathbf{Int}\mathcal{C}((X, U), (Z, W))$ is defined using the trace, symmetry, & composition of the underlying category $\mathcal{C}$ (which we denote by concatenation), as follows:



4. **(Identities)** The identity at an object $1_{(X,U)} \in \mathbf{Int}\mathcal{C}((X, U), (X, U))$ is simply $(1_X \otimes 1_U) \in \mathcal{C}(X \otimes U, X \otimes U)$.
5. **The Tensor** $\mathbf{Int}\mathcal{C}$ has a symmetric monoidal tensor, $\_\Box\_$, given by:

   - **(Objects)** $(X_1, U_1)\Box(X_2, U_2) = (X_1 \otimes X_2, U_2 \otimes U_1)$ for all (X_1, U_1), $(X_2, U_2) \in Ob(\mathbf{Int}\mathcal{C})$.
   - **(Arrows)** Given arrows $f : (X_1, U_1) \to (Y_1, V_1)$ and $g : (X_2, U_2) \to (Y_2, V_2)$, their tensor $f\Box g : (X_1, U_1)\Box(X_2, U_2) \to (Y_1, V_1)\Box(Y_2, V_2)$ is given diagrammatically, as



6. **(The unit object)** The unit object is simply (I, I), where I is the unit object of $\mathcal{C}$.
7. **(The dual on objects)** This is defined by $(X, U)^* = (U, X)$.
8. **(The dual on arrows)** This is defined in terms of the symmetry isomorphism of $\mathcal{C}$; given an arrow $f \in \mathcal{C}((X, U), (Y, V))$, then $(f)^* = \left(\sigma_{Y,U} f \sigma_{X,V}\right) \in$ **Int**$\mathcal{C}((U, X), (Y, V))$.
9. **(The unit and co-unit)** The distinguished *unit* and *co-unit* arrows $\eta : (I, I) \to (X, U)\Box(U, X)$ and $\epsilon : (U, X)\Box(X, U) \to (I, I)$ are specified by *symmetry arrows* in the underlying traced monoidal category $\mathcal{C}$, so

$$\eta_{(X,U)} = \left(\sigma_{I,X\otimes U}\right) \in \mathbf{Int}\mathcal{C}((I, I), (X, U)\Box(U, X))$$

and

$$\epsilon_{(U,X)} = \left(\sigma_{U\otimes X,I}\right) \in \mathbf{Int}\mathcal{C}((U, X)\Box(X, U), (I, I))$$

Although Joyal, Street, and Verity assumed strict associativity of the underlying traced monoidal category (and hence of the resulting compact closed category), and often left canonical isomorphisms implicit, it is nevertheless straightforward to write down the construction in the case where these canonical isomorphisms are made explicit. This rather thankless task was carried out in Hines (1997) where the following may be found:

Lemma 2 *Let* $(\mathcal{C}, \otimes, \sigma_{\_,\_}, \tau_{\_,\_,\_}, I, Tr())$ *be a traced symmetric monoidal category. Then the associativity and symmetry isomorphisms of the resulting compact closed category* (**Int**$\mathcal{C}$, $\_\Box\_$) *are given by*

Symmetry *For all* $(X, U), (X', U') \in Ob(\mathbf{Int}\mathcal{C})$,

$$S_{(X,U),(X',U')} = \sigma_{X,X'} \otimes \sigma_{U,U'}$$

Associativity *For all* $(X, U), (Y, V), (Z, W) \in Ob(\mathbf{Int}\mathcal{C})$,

$$T_{(X,U),(Y,V),(Z,W)} = \alpha_{X,Y,Z} \otimes \alpha^{-1}_{W,V,U}$$

Proof We refer to Hines (1997) for the details. The key point is to demonstrate that the sources and targets are correct.

Symmetry

$$C((X \otimes X') \otimes (U \otimes U'), (X' \otimes X) \otimes (U' \otimes U)) = \mathbf{Int}C((X,U)\Box(X',U'), (X',U')\Box(X,U))$$

Associativity

$$\begin{aligned} &C((X \otimes (Y \otimes Z)) \otimes ((W \otimes V) \otimes U), ((X \otimes Y) \otimes Z) \otimes (W \otimes (V \otimes U))) \\ = \ &\mathbf{Int}C((X,U)\Box((Y,V)\Box(Z,W)), ((X,U)\Box(Y,V))\Box(Z,W)) \end{aligned}$$

□

A common intuition is that **Int**C is a 'dualised' version of C that contains both C and C^{op}. This is apparent in the following, taken from Joyal et al. (1996).

Proposition 2 *There exist faithful traced monoidal functors $L_I, R_I : C \to \mathbf{Int}C$ that are covariant and contravariant respectively. These are given by, for all $X, Y \in Ob(C)$, and $f \in C(X,Y)$,*

Covariant $L_I(X) = (X, I)$ *and* $L_I(f) = (f \otimes 1_I) \in \mathbf{Int}C((X,I),(Y,I))$
Contravariant $R_I(X) = (I, X)$ *and* $R_I(f) = (1_I \otimes f) \in \mathbf{Int}C((I,X),(I,Y))$

Should we be prepared to consider semi-monoidal rather than monoidal functors, the above proposition generalises to arbitrary non-unit objects. In this setting, we have no single distinguished covariant and contravariant faithful semi-monoidal functors from $(C, \otimes)$ to $(\mathbf{Int}C, \Box)$; rather, we have such functors indexed by the objects of C.

Proposition 3 *Given an arbitrary object $U \in Ob(C)$ of some traced symmetric monoidal category, we may define both covariant and contravariant faithful semi-monoidal functors from $(C, \otimes)$ to $(\mathbf{Int}C, \Box)$ by, for all $X, Y \in Ob(C)$, and $f \in C(X,Y)$,*

Covariant $L_U(X) = (X, U)$ *and* $L_U(f) = (f \otimes 1_U) \in C((X,U),(Y,U))$
Contravariant $R_U(X) = (U, X)$ *and* $R_U(f) = (1_U \otimes f) \in \mathbf{Int}C((U,X),(U,Y))$

Proof This follows in precisely the same way as the standard proofs of Proposition 2; we are simply not insisting that our functors preserve a unit object. □

Corollary 5 *As a well-established corollary, objects of the form $(N, N) \in Ob(\mathbf{Int}C)$ are self-dual, for arbitrary $N \in Ob(C)$, and all self-dual objects are isomorphic to some object of this form.*

Corollary 6 *A self-dual object $(N, N) \in Ob(\mathbf{Int}C)$ is self-similar iff $N \in Ob(C)$ is self-similar.*

Proof
($\Leftarrow$) This is immediate, and well-established (e.g. Hines, 1997). Given code/decode arrows $\lhd \in C(N \otimes N, N)$ and $\rhd \in C(N, N \otimes N)$, the self-similarity of $(N, N) \in Ob(\mathbf{Int}C)$ is exhibited by

$$(\lhd \otimes \rhd) \in \mathbf{Int}C((N, N)\Box(N, N), (N, N)) \quad \text{and} \quad (\rhd \otimes \lhd) \in \mathbf{Int}C((N, N), (N, N)\Box(N, N))$$

($\Rightarrow$) Consider some $(N, N) \cong (N, N)\Box(N, N)$. then

$$(N, N) \;\cong\; (N, I)\Box(I, N) \;\cong\; ((N, I)\Box(I, N))\,\Box\,((N, I)\Box(I, N))$$

As the tensor of $\mathbf{Int}C$ is symmetric, this implies

$$(N, I)\Box(I, N) \;\cong\; ((N, I)\Box(N, I))\,\Box\,((I, N)\Box(I, N)) \;\cong\; (N \otimes N, I)\Box(I, N \otimes N)$$

and our result follows as the functors $L_I, R_I : C \to \mathbf{Int}C$ are faithful. $\Box$

As a corollary of the above two results, we observe that is is straightforward, at least in principle, to exhibit strictly reflexive objects of compact closed categories derived from the **Int** construction.

Corollary 7 *Let $N \in Ob(C)$ be a strictly self-similar object of a symmetric traced monoidal category. Then $(N, N) \in Ob(\mathbf{Int}C)$ is both strictly self-dual and strictly self-similar, and hence strictly reflexive.*

Proof Strict self-duality is immediate. The self-similarity of (N, N) is, by Corollary 6, exhibited by

$$(\lhd \otimes \rhd) \in \mathbf{Int}C((N, N)\Box(N, N), (N, N)) \quad \text{and} \quad (\rhd \otimes \lhd) \in \mathbf{Int}C((N, N), (NN)\Box(N, N))$$

However, when $\lhd = 1_N = 1_{N\otimes N} = \rhd$ these are both identity maps, so (N, N) is strictly self-similar. $\Box$

The above result is not as helpful as it first appears; it is not immediate how to find strictly self-similar objects of traced monoidal categories; we do not have a consistent method of strictifying self-similarity in a monoidal (rather than semi-monoidal) setting, so some work remains before we can achieve our goal of strictifying reflexivity in compact closed categories.

25.8.1 A Naming of Parts

A useful aspect of the **Int** construction is that it simply creates names (& indeed elements generally) *ex nihilo*. Less dramatically, we observe that even when the underlying traced monoidal category $(C, \otimes)$ has no elements, the category $\mathbf{Int}C$ is, by construction, fully equipped with names for all its arrows (& hence a wide range of elements).

Lemma 3 *Let $(\mathcal{C}, \otimes, \sigma, I, Tr^{-}_{\_,\_})$ be a traced symmetric monoidal category. Then:*

1. *For arbitrary $(U, V) \in Ob(\mathbf{Int}\mathcal{C})$, the elements of (U, V) are in 1:1 correspondence with the homset $\mathcal{C}(V, U)$.*
2. *For arbitrary $X \in Ob(\mathcal{C})$, the following are in 1:1 correspondence:*
   - *elements of $X \in Ob(\mathcal{C})$,*
   - *elements of $L_I(X) \in Ob(\mathbf{Int}\mathcal{C})$,*
   - *elements of $R_I(X) \in Ob(\mathbf{Int}\mathcal{C})$.*
3. *For an arbitrary self-dual object $(N, N) \in Ob(\mathbf{Int}\mathcal{C})$, the elements of (N, N) are in 1:1 correspondence with the endomorphism monoid $\mathcal{C}(N, N)$.*

Proof

1. From the definition of homsets, $\mathbf{Int}\mathcal{C}((I, I), (U, V)) \overset{def.}{=} \mathcal{C}(I \otimes V, U \otimes I) \cong \mathcal{C}(V, U)$.
2. As a special case of 1., $\mathbf{Int}\mathcal{C}((I, I), L_I(X)) \overset{def.}{=} \mathcal{C}(I \otimes I, X \otimes I) \cong \mathcal{C}(I, X)$ and $\mathbf{Int}\mathcal{C}((I, I), R_I(X)) \overset{def.}{=} \mathcal{C}(I \otimes I, I \otimes X) \cong \mathcal{C}(I, X)$.
3. This is again a special case of 1. □

25.9 Strictifying Reflexivity in a Compact Closed Category

The observations of Lemma 3, although straightforward, provide a route to the monoidal equivalences of compact closed categories we need in order to give a monoidal strictification of reflexivity. The following preliminary results are needed:

Proposition 4 *Let $(\mathcal{C}, \otimes)$ be a symmetric traced monoidal category with trivial scalars. Then*

1. *The de-elemented version $(\mathcal{C}, \otimes)_{-\substack{\mathcal{E}\mathcal{A}\\ \mathcal{F}\mathcal{W}}}$ is also traced.*
2. *For arbitrary non-unit $N \cong N \otimes N \in Ob(\mathcal{C})$ there is a †—isomorphism[10] of †-compact closed categories between*
   a. *The monoidal subcategory of $\mathbf{Int}\mathcal{C}$ generated by (N, N).*
   b. *The monoidal subcategory of $\mathbf{Int}\left(\mathcal{C}_{-\substack{\mathcal{E}\mathcal{A}\\ \mathcal{F}\mathcal{W}}}\right)$ generated by (N, N).*
3. *When N is self-similar, the monoidal subcategory of $\mathbf{Int}((\mathcal{C}(N, N), \star)_{+I})$ generated by (N, N) is † monoidal equivalent to a. and b. above.*

[10] The terminology ‘isomorphic’ rather than ‘equivalent’ is deliberate; these are both small categories, and there is a bijection of sets of objects as well as homsets.

Proof

1. This is a straightforward consequence of the Vanishing 1 axiom; as observed in Remark 15, tracing out the unit object is the identity on homsets. Conversely, as there is only one abstract scalar of $(\mathcal{C}, \otimes)_{-\frac{\mathcal{E}}{\mathcal{F}}\frac{\mathcal{A}}{\mathcal{W}}}$, traces of the form $Tr^U_{I,I}(\ )$ are also uniquely determined.
2. Parts a. and b. follows from Part 3. of Lemma 3 above; the only homsets of either of these compact closed categories determined by elements of the underlying traced category are the abstract scalars, which are trivial in both cases. It makes no difference whether or not we start with the 'de-elemented' version.
3. This follows from the monoidal equivalence of categories noted in Corollary 4 (and indeed the fact that the **Int** construction is not 'evil'; given monoidally equivalent traced monoidal categories $\mathcal{C}$ and $\mathcal{D}$, there exists a monoidal equivalence of categories between $\mathbf{Int}(\mathcal{C})$ and $\mathbf{Int}(\mathcal{D})$).
   We may also exhibit the trace on $(\mathcal{C}(N, N), \star)_{+I}$ explicitly; in Hines (1997) it is observed that for a self-similar object N of a symmetric traced monoidal category, there exists an operation $trace$ on $(\mathcal{C}(N, N), \star)$ given by
   $$trace(f) = Tr^N_{N,N}(\rhd f \lhd) \ \forall f \in \mathcal{C}(N, N)$$
   that 'satisfies all of the axioms of Joyal et al. (1996) apart from Vanishing I'. As $\mathcal{C}$ has trivial scalars, this extends uniquely (as described in Remark 15) to a categorical trace on the symmetric monoidal category $(\mathcal{C}(N, N), \star)_{+I}$, and coincides, up to monoidal equivalence, with the trace on the subcategory of $\mathcal{C}$ monoidally generated by N. $\square$

This now gives the desired monoidal strictification of extensional reflexivity.

Theorem 3 *Let $(\mathcal{C}, \otimes, \sigma, I, Tr(\ ))$ be a symmetric traced monoidal category with trivial scalars, and let $X \in Ob(\mathbf{Int}\mathcal{C})$ be a reflexive object. Then there exists a monoidal strictification of the reflexivity of X.*

Proof By Lemma 1, X is self-dual, and from Corollary 5 it is therefore isomorphic to $(N, N) \in Ob(\mathbf{Int}\mathcal{C})$ for some $N \in Ob(\mathcal{C})$. Without loss of generality, we therefore work with the strictly self-dual reflexive object $(N, N) \in Ob(\mathbf{Int}\mathcal{C})$. As X is self-similar, N is therefore a self-similar object of $(\mathcal{C}, \otimes)$, by Corollary 6.

By Proposition 4 above, the monoidal subcategory of $\mathbf{Int}((\mathcal{C}(N, N), \star)_{+I})$ generated by (N, N) then gives a monoidal strictification of this reflexivity, as axiomatised in Definition 4. $\square$

25.9.1 Discussion

We have exhibited a process that will strictify the reflexivity of a large class—although not all—reflexive objects in compact closed categories. There are several notable points.

1. The two restrictions on this process are:

   a. The compact closed category itself needs to arise from applying the **Int** or **GoI** construction, to a traced monoidal category.
   b. The underlying traced monoidal category (and hence the compact closed category itself) must have trivial scalars.

   The first condition rules out compact closed categories such as finite-dimensional Hilbert spaces with tensor product (which does not, in any case, have reflexive objects), or relations with Cartesian product (which certainly does have reflexive objects). However, it is satisfied by the categories on which the original Geometry of Interaction system was based.

   The second condition is also satisfied by the setting of the original Geometry of Interaction system that we will analyse in Sect. 25.11 onwards. It is harder to think of 'natural' examples that are ruled out by this restriction, although we may certainly construct examples by using the techniques of Abramsky (2005) to 'adjoin' a non-trivial involutive monoid of abstract scalars. This is discussed further in Sect. 25.17.
2. The object (N, N) of the compact closed category **Int** $((C(N, N), \star)_{+I})$ is strictly self-similar, and therefore its endomorphism monoid is a semi-monoidal monoid. At this point, it is tempting to axiomatise away all the problems of Sect. 25.2.1, and simply *define* a 'compact closed monoid' to be the result of such a process. This would be inaccurate, and on a par with simply defining a compact closed category to be some subcategory of one that arises from (a restricted version of) the **Int** or **GoI** construction.
3. A curiosity of this setting is that the endomorphism monoid of N in the traced monoidal category $(C(N, N), \star)_{+I}$ has the same underlying set as the endomorphism monoid of (N, N) in the compact closed category **Int** $(C(N, N), \star)_{+I})$, since N is *strictly* self-similar. This is a very useful property when we come to study such endomorphism monoids in a more algebraic setting.

Conjecture 1 The restrictions of Point 1. above are not essential, and reflexivity may be strictified for arbitrary reflexive objects in compact closed categories.

25.10 Properties of Strictly Reflexive Objects

We now move on from demonstrating that—at least in certain cases—reflexivity may be strictified, and consider the structure of strictly reflexive objects in compact closed categories more generally. Our setting includes, but is not restricted to, objects derived from the strictification of reflexivity given in Theorem 3.

25.10.1 Notation & An Algebraic Perspective

As observed in Point 3 of Sect. 25.9.1, for a strictly self-similar object N of a traced symmetric monoidal category $\mathcal{C}$, the endomorphism monoids $\mathcal{C}(N, N)$ and $\mathbf{Int}\mathcal{C}((N, N), (N, N))$have the same underlying set. The **Int** construction provides a new composition, tensor, etc. on this set. With suitable notational discipline, we may treat the 'old' and 'new' compositions & tensors as operations on the same set, and even consider their interaction.

This is not done simply in order to horrify categorically-minded readers (although it will undoubtedly have that effect), but neither is it just a notational trick.

The historical precedent and motivation for this was not originally phrased categorically. In Girard's system, it is notable that the dynamics (i.e. cut and cut-elimination) is modeled by the composition of an endomorphism monoid of a *compact closed* category, whereas the model of conjunction is the tensor $\_ \star \_$ of the underlying *traced* category. The models of the exponentials are different again, and the bang $!(\ )$ is best seen as a right fixed-point semi-monoidal endofunctor $f \star !(f) = !(f)$ for the tensor modeling the conjunction—but is derived from yet another monoidal tensor that distributes over the tensor modeling conjunction.

The system as a whole relies on treating all these as operations on the same underlying set, and indeed considering their interaction. Outside of models of untyped systems, it is hard to account for all this; within the untyped (or strictly reflexive) setting it is more algebraically natural, but still category-theoretically disturbing.

Such notational abuse allows us to write down the following

Proposition 5 *Let* $(\mathcal{C}, \otimes, s_{\_,\_}, a_{\_,\_,\_}, Tr^{-}_{\_,\_})$ *be a symmetric traced monoidal category, and let* $N = N \otimes N \in Ob(\mathcal{C})$ *be a (non-unit) strictly self-similar object.* $(\mathcal{C}(N, N), \otimes)$ *is then a semi-monoidal monoid; for simplicity, let us denote its underlying set by M, its composition by $\cdot$, its tensor by $\otimes$, and its canonical associativity isomorphism & inverse by*

$$\tau = a_{N,N,N} \in \mathcal{C}(N \otimes (N \otimes N), (N \otimes N) \otimes N) = \mathcal{C}(N, N)$$

and

$$\tau' = a^{-1}_{N,N,N} \in \mathcal{C}((N \otimes N) \otimes N, N \otimes (N \otimes N)) = \mathcal{C}(N, N)$$

and its canonical symmetry isomorphism by

$$\sigma = s_{N,N} \in C(N \otimes N, N \otimes N) = C(N, N)$$

In the compact closed category $(\mathbf{Int}C, \Box)$*, the object* (N, N) *is strictly reflexive, and its endomorphism monoid is a semi-monoidal monoid with underlying set* M*. Let us denote its composition by* $\circ$*, its tensor by* $\Box$*, its canonical associativity isomorphism & inverse by* $T, T' \in M$ *and its symmetry isomorphism by* $S \in M$*.*

The following are then immediate:

1. $(M, \circ)$ *and* $(M, \cdot)$ *share the same identity element,* 1_M*.*
2. $T = \tau \otimes \tau^{-1}$
3. $S = \sigma \otimes \sigma$
4. *The functions* $(1 \otimes \_), (\_ \otimes 1) : (M, \cdot) \to (M, \cdot)$ *are homomorphic self-embeddings of the monoid* $(M, \cdot)$*.*
5. *The functions* $(1\Box\_), (\_\Box 1) : (M, \circ) \to (M, \circ)$ *are homomorphic self-embeddings of the monoid* $(M, \circ)$*.*
6. *The functions* $(1 \otimes \_), (\_ \otimes 1) : (M, \cdot) \to (M, \circ)$ *are homomorphic and anti-homomorphic monoid embeddings respectively.*
7. *The subset* $\{\tau, \tau'\} \subseteq M$ *generates, by closure under the composition* $\cdot$ *and the tensor* $\otimes$*, a subgroup of* $(M, \cdot)$ *isomorphic to Thompson's group* $\mathcal{F}$*.*
8. *The subset* $\{T, T'\} = \{\tau \otimes \tau', \tau' \otimes \tau\} \subseteq M$ *generates, by closure under the composition* $\circ$ *and the tensor* $\Box$*, a subgroup of* $(M, \circ)$ *isomorphic to Thompson's group* $\mathcal{F}$*.*

Proof Part 1. is immediate from the definition of the **Int** construction, and parts 2.–3. are similarly immediate from Lemma 2. Parts 4. and 5. are general results on semi-monoidal monoids found in Hines (2016). Part 6. is derived from Proposition 3. Finally, parts 7. and 8. are immediate from Theorem 1. $\Box$

We now move on to horrify further the categorically-minded reader, and consider some interactions between the canonical isomorphisms for $\star$, and the tensor $\Box$ and composition $\circ$. Doing so leads us to the (rather categorically respectable) theory of Frobenius algebras, as well as monoids derived from Frobenius algebras that have close connections with classic structures from both group and semigroup theory.

Remark 16 As indicated in Footnote 9, we may consider the composition, tensor, dual, etc. derived from Abramsky's **GoI** construction, rather than Joyal, Street, & Verity's **Int** construction, as yet another collection of operations on the same underlying set—together with a non-trivial bijection of the underlying set that will map between the two.

In the following sections, we consider the interactions between the composition, tensor, and canonical isomorphisms of the underlying traced monoidal category with those of the category derived from the **Int** construction. We could of course do the same with the operations derived from the **GoI** construction instead, and derive an entirely distinct (& similarly interesting) set of non-trivial interactions.

This brings us to another reason for using the conventions of Joyal et al. (1996) instead of those of Abramsky (1996)—we derive a more direct route to an interesting class of Frobenius algebras & monoids introduced by Abramsky and Heunen (2010).

25.10.2 Reflexivity and (Unitless) Frobenius Algebras & Monoids

As well as the fundamental rôle of compact closure in Abramsky & Coecke's categorical quantum mechanics program (Abramsky & Coecke, 2005), another key building block is the (closely connected—see Theorem 4) notion of a *Frobenius algebra*. Beyond their well-known applications in quantum field theory (Abramsky & Heunen, 2010), in quantum information they model fan-out operations (Høyer & Špalek, 2005) in quantum circuits, and play the rôle of orthonormal bases in purely categorical formulations (Coecke & Pavlovic, 2007; Coecke et al., 2013) of quantum foundations.

In stark contrast to compact closure itself, the notion of a 'unitless Frobenius algebra' is well-established, both in terms of theory and examples. These were defined and studied by Abramsky and Heunen (2010), from where the following definition (but not terminology) is taken:

Definition 13 Let $(C, \otimes, \alpha_{\_,\_,\_})$ be a semi-monoidal category. An **Abramsky-Heunen** or **A-H Frobenius algebra** $\mathcal{F} = (S, \Delta, \nabla)$ in C consists of an object $S \in Ob(C)$, equipped with an associative **split arrow** $\Delta : S \to S \otimes S$ and a co-associative **merge arrow** $\nabla : S \otimes S \to S$. These are required to satisfy the **Frobenius condition**

$$(1_S \otimes \nabla)\alpha^{-1}_{S,S,S}(\Delta \otimes 1_S) = \Delta\nabla = (\nabla \otimes 1_S)\alpha_{S,S,S}(1_S \otimes \Delta)$$

Explicitly, **associativity** and **co-associativity** are the requirements that

- $\nabla(1_S \otimes \nabla) = \nabla(\nabla \otimes 1_S)\alpha_{S,S,S}$
- $(1_S \otimes \Delta)\Delta = \alpha_{S,S,S}((\Delta \otimes 1_S)\Delta)$.

Every Frobenius algebra is—by definition—an A-H Frobenius algebra, but the converse is not true. In particular, as the above definition does not mention the unit object, there are therefore no obstacles to considering A-H Frobenius algebras whose distinguished object is the unique (non-unit) object of a semi-monoidal monoid.

Definition 14 Let $(M, \star, \alpha)$ be a semi-monoidal monoid, and let Δ and ∇ be the split and merge arrows of an A-H Frobenius algebra at the unique object of this monoid. We define its **A-H Frobenius** or **A-H F monoid** to be the semi-monoidal submonoid of $(M, \star)$ generated by the closure of $\{\alpha, \alpha^{-1}, \Delta, \nabla\}$ under composition and tensor.

Remark 17 (*Another variation of terminology*) It is more standard to refer to the 'split' and 'merge' arrows of a Frobenius algebra as the *co-monoid* and *monoid* arrows respectively. We do not do this, in order to avoid the potentially fatal confusion of terminology that would result.

An interesting class of examples of Frobenius algebras is given by the following well-known result:

Theorem 4 *For every object X of a compact closed category $(\mathcal{C}, \_ \otimes \_, \sigma_{\_,\_}, I, \epsilon_{\_}, \eta_{\_})$, the object $X^* \otimes X$ is self-dual and there is a Frobenius algebra (the* **canonical Frobenius algebra***) at $X \otimes X^*$ with distinguished arrows Δ and ∇ given by the following composites:*

$$X^* \otimes X \xrightarrow{\cong} X^* \otimes I \otimes X \xrightarrow{1_X^* \otimes \eta_X \otimes 1_X} X \otimes X^* \otimes X^* \otimes X$$

and

$$X^* \otimes X \otimes X^* \otimes X \xrightarrow{1_{X^*} \otimes \epsilon_X \otimes 1_X} X^* \otimes I \otimes X \xrightarrow{\cong} X^* \otimes X$$

Proof We refer to, for example, Vicary (2011) for a direct proof of this, but note that it follows abstractly from a more general result of Lauda (2005), where it is proved for any object with an ambidextrous adjunction—with the self-dual compact closed case following as a special case. □

Corollary 8 *In Theorem 4 above, $\nabla\Delta \cong 1_{X^* \otimes X} \otimes \mu$ for some abstract scalar $\mu \in \mathcal{C}(I, I)$, and hence when $(\mathcal{C}, \otimes, I)$ has trivial scalars $\nabla\Delta = 1_{X^* \otimes X}$. However, this is a one-sided, rather than two-sided inverse.*

Proof The composite $\nabla\Delta \cong 1_{X^* \otimes X}$ differs from the identity on $X^* \otimes X$ by the composite of a unit and a co-unit (i.e. a closed loop) in a compact closed category. Therefore, by Abramsky (2005), it is the identity up to an abstract scalar, and is precisely the identity when the monoid of scalars is trivial. To see why it is not a two-sided inverse, note that were this to be the case, every self-dual object of any compact closed category would be self-similar & hence reflexive. □

Definition 15 Let $N \cong X \otimes X^*$ be a self-dual object of a compact closed category $\mathcal{C}$. We refer to the A-H Frobenius algebra at N given by Theorem 4 above as the **standard A-H Frobenius algebra** at N.

In the case where N is a strictly reflexive object of $\mathcal{C}$, we observe that the endomorphism monoid of N is a semi-monoidal monoid, and therefore (following Definition 14) contains an A-H Frobenius monoid. We refer to this as the **standard A-H F. monoid** at N. When $\mathcal{C}$ also has trivial scalars, we refer to this as the **simple A-H F monoid**.

Conjecture 2 We conjecture that "non-degenerate standard A-H Frobenius monoids differ only by their scalars". More precisely, but less generally, all (non-abelian) simple A-H F. monoids are isomorphic.

The simple A-H F monoid is worth studying as much for its algebra as its category theory. The following places it in the mainstream of semigroup theory and group theory, as we demonstrate in Sect. 25.10.3.

Lemma 4 *Let N be a strictly reflexive object of a compact closed category $(C, \otimes, t_{\_,\_,\_})$ with trivial units. Then the split & merge maps $\Delta, \nabla \in C(N, N)$ of the simple A-H F monoid satisfy:*

1. $\nabla\Delta = 1_N$
2. $\Delta\nabla = 1_N$ *iff N is the unit object.*

Proof Part 1. is immediate from Corollary 8 above. For part 2., let denote $t_{N,N,N} \in C(N, N)$ by α, and assume that the inverse of part 1. is a two-sided inverse. In this case, the associativity axiom, $\nabla(1 \otimes \nabla) = \nabla(\nabla \otimes 1)\alpha$ implies $(1 \otimes \nabla) = (\nabla \otimes 1)\alpha$, giving $\alpha = \Delta \otimes \nabla$. Similarly, the co-associativity axiom implies $\alpha = \nabla \otimes \Delta$ and so $\alpha = \alpha^{-1}$. Therefore, associativity at this semi-monoidal monoid is strict. By Proposition 1 this is the case iff N is the unit object. $\square$

In certain cases (including those arising from the strictification of reflexivity procedure described in Theorem 3), we are able to give an explicit description of the standard A-H F monoid at a strictly reflexive object. The key to this is that in the case where the compact closed category of Theorem 4 above arises from the **Int** construction, the *split* and *merge* arrows have a particularly neat form:

Proposition 6 *Let $(C, \otimes, \alpha_{\_,\_,\_}, \sigma_{\_,\_})$ be a traced symmetric monoidal category. Then the defining* split *and* merge *arrows of the standard Frobenius algebra at a self-dual object $(N, N) \cong (N, I)\Box(I, N) \in Ob(\mathbf{Int}C)$,*

$$\Delta \in \mathbf{Int}C((N, N), (N, N)\Box(N, N)) \;=\; C(N \otimes (N \otimes N), (N \otimes N) \otimes N)$$

and

$$\nabla \in \mathbf{Int}C((N, N)\Box(N, N), (N, N)) \;=\; C((N \otimes N) \otimes N, N \otimes (N \otimes N))$$

are given by the symmetry and associativity isomorphisms of the underlying traced category, as

- $\Delta \;=\; \alpha_{N,N,N}(1 \otimes \sigma_{N,N})$
- $\nabla \;=\; (1 \otimes \sigma_{N,N})\alpha^{-1}_{N,N,N}$

respectively.

Proof This is immediate, and simply follows from substituting in the definitions of the unit/co-unit maps of **Int**C into the definitions of Theorem 4. $\square$

Remark 18 The above observation, although simple, breaks the connection between the standard Frobenius algebra at a self-dual object of **Int**C and the distinguished unit/co-unit maps—and hence, the unit object. Instead, the standard Frobenius algebra is simply derived from the canonical isomorphisms of the underlying traced symmetric monoidal category C. This will of course be key to writing down the generators of the standard A-H F monoid(s) of Definition 15.

Theorem 5 *Let* $(C, \otimes, s_{\_,\_}, a_{\_,\_,\_}, Tr^{-}_{\_,\_})$ *be a symmetric traced monoidal category, and let* $N = N \otimes N \in Ob(C)$ *be a (non-unit) strictly self-similar object. We follow the conventions & notation of Proposition 5, and denote the underlying set of* $C(N, N) = \mathbf{Int}C((N, N), (N, N))$ *by* M*, along with the two semi-monoidal monoid structures on it given by*

- *(From the underlying traced category)* $(M, \cdot, \otimes, \tau, \sigma)$
- *(From the compact closed category)* $(M, \circ, \Box, T, S)$.

Let us denote the inverse of $\tau \in (M, \cdot)$ *by* $\tau' \in (M, \cdot)$*. Then the standard A-H Frobenius monoid is the monoid generated by the closure of the set*

$$\{\tau \cdot (1 \otimes \sigma)\ ,\ (1 \otimes \sigma) \cdot \tau'\ ,\ \tau \otimes \tau'\ ,\ \tau' \otimes \tau\}$$

under the composition $\circ$ *and tensor* $\Box$.

Proof From Proposition 6, the split and merge arrows are given by $\Delta = \tau \cdot (1 \otimes \sigma)$ and $\nabla = (1 \otimes \sigma) \cdot \tau'$ respectively. The associator for $\_\Box\_$ and its inverse are given by $T = \tau \otimes \tau'$ and $T' = \tau' \otimes \tau$ respectively. These primitives then, by definition, generate the standard A-H Frobenius monoid, and satisfy the *Frobenius condition*

$$(1\Box\nabla) \circ T' \circ (\Delta\Box 1) \ = \ \Delta \circ \nabla \ = \ (\nabla\Box 1) \circ T(1\Box\Delta)$$

along with *associativity* and *co-associativity*

- $\nabla \circ (1\Box\nabla) = \nabla \circ (\nabla\Box 1) \circ T$
- $(1\Box\Delta) \circ \Delta = ((\Delta\Box 1) \circ \Delta) \circ T$. $\Box$

25.10.3 Algebraic Aspects

The standard A-H F monoid at a strictly reflexive object contains, by construction, all canonical associativity isomorphisms of the semi-monoidal endomorphism monoid; it therefore (by Theorem 1) contains a copy of Thompson's iconic group $\mathcal{F}$. We now demonstrate that the simple case also contains a copy of one of the most iconic structures from semigroup theory—the bicyclic monoid. Further, there is a very natural way in which the simple A-H F monoid may be thought of as, "interacting copies of the bicyclic monoid, and Thompson's $\mathcal{F}$".

The following was first published in Lyapin (1953), but appears previously to have been known by Clifford, Preston, & Rees (see Hollings, 2014 for a historical overview).

Definition 16 The **bicyclic monoid B** is the monoid with two generators, and a single relation

$$\mathcal{B} = \langle s, r \ : \ rs = 1 \rangle$$

(Note that this is a one-sided inverse, and certainly does not imply that $sr = 1$)!

It is remarkably well-studied, and appears in a wide range of algebraic and computational settings. Its theory is very well-established; the following results may be found in, for example, Lawson (1998).

Theorem 6

1. *$\mathcal{B}$ is an inverse monoid, and is isomorphic to the inverse monoid of partial injections on the natural numbers generated by the successor function and its (partially defined) inverse.*
2. *Given an arbitrary monoid M, and a pair of elements $r, s \in M$ satisfying $rs = 1 \neq sr$, then the submonoid generated by $\{r, s\}$ is isomorphic to $\mathcal{B}$.*
3. *The elements of $\mathcal{B}$ may be given a normal form as pairs of natural numbers, with composition given by, for all $(d, c), (b, a) \in \mathbb{N} \times \mathbb{N}$*

$$(d, c)(b, a) = \left(d + [b \stackrel{\bullet}{-} c], [c \stackrel{\bullet}{-} b] + a\right)$$

where the **monus** *operation $\stackrel{\bullet}{-}$ is defined by* $y \stackrel{\bullet}{-} x = \begin{cases} y - x & x \leq y \\ 0, & \textit{otherwise.} \end{cases}$

Remark 19 It is hard to avoid seeing the bicyclic monoid itself as a categorical structure. It is well-known that the natural numbers with the usual ordering forms a (posetal) category, and addition is a (strictly symmetric & associative) monoidal tensor on this category, with the unit object simply being $0 \in \mathbb{N}$. Further, it is straightforward that $\mathbb{N}$ is a *closed* monoidal category, with the internal hom given by the above monus operation, $y \stackrel{\bullet}{-} x = \begin{cases} y - x & x \leq y \\ 0, & \text{otherwise.} \end{cases}$

The bicyclic monoid then has all the appearance of being the result of a 'dualising' construction on this monoidal closed category. It may also be relevant that $(\mathbb{N}, +, 0)$ is traced, with the trace given on objects by subtraction.

Theorem 7 *Let N be a (non-unit) strictly reflexive object of a compact closed category $(C, \otimes, t_{\_,\_,\_}, s_{\_,\_}, I)$ with trivial scalars. Then $C(N, N)$ is a semi-monoidal monoid containing a copy of the simple A-H F monoid, which we denote $\mathbb{A} \subseteq C(N, N)$. Then*

1. *$\mathbb{A}$ is generated by the closure under composition and tensor of:*
   - *The associator $\alpha = t_{N,N,N}$ and its inverse α^{-1}.*
   - *The split map Δ and its generalised inverse ∇.*

2. *The associator and its inverse generate (by closure under composition & tensor), a copy of Thompson's $\mathcal{F}$ within $\mathbb{A}$.*
3. *The split element Δ and its generalised inverse ∇ generate (by closure under composition) a copy of the bicyclic monoid $\mathcal{B}$ within $\mathbb{A}$.*

Proof Part 1. is simply the definition of the simple A-H F monoid at N. Part 2. follows directly from Theorem 1. From Corollary 8, the split and merge arrows of the simple A-H F monoid satisfy $\nabla\Delta = 1$ and $\Delta\nabla \neq 1$, and from Part 2. of Theorem 6, they therefore generate a copy of the bicyclic monoid. □

25.11 Concrete Examples

We now move on from the abstract theory of strict extensional reflexivity to concrete examples. These have been separated out from the theoretical constructions, in order to emphasise that the theory is generally applicable, and not tied to any specific example.

The concrete setting we now consider it that of Girard's original two Geometry of Interaction papers (Girard, 1988a, b) (and indeed Abramsky, Haghverdi, & Scott's linear combinatory logic Abramsky et al., 2002). The starting point for this is the (inverse) traced monoidal category of partial injections, considered as a traced monoidal subcategory of an illustrative example of Joyal, Street, and Verity.

25.11.1 The Category of Relations, Its Matrix Calculus, and Its Trace

Joyal et al. (1996) used the monoidal category of relations with disjoint union as a canonical example of a traced, but not compact closed, category. Their treatment was based on writing relations in matrix form.

Definition 17 The category **Rel** of relations on sets has as objects all sets. An arrow $R \in \mathbf{Rel}(X, Y)$ is a subset $R \subseteq Y \times X$. Given an arrow $S \in \mathbf{Rel}(Y, Z)$, composition is given by the usual formula for relational composition,

$$(z, x) \in SR \quad \text{iff}\ \exists y \in Y \ :\ (z, y) \in S \ \text{ and } \ (y, x) \in R$$

The category of relations also has a dagger $(\ )^c : \mathbf{Rel}^{op} \to \mathbf{Rel}$, given by relational converse $R^c = \{(x, y) : (y, x) \in R\}$.

A partial function is a relation satisfying $(y, x), (z, x) \in R \Rightarrow y = z$, and a partial injection is a partial function whose converse is also a partial function. It is standard to write partial functions and partial injections in functional form, as $f(x) = y$, rather than $(y, x) \in f$. Partial functions and partial injections form wide subcategories of **Rel**, denoted **pFun** and **pInj** respectively.

The category **Rel** also has a biproduct, the disjoint union $\_ \uplus \_ : \mathbf{Rel} \times \mathbf{Rel} \to \mathbf{Rel}$ given by on objects by disjoint union $A \uplus B = A \times \{0\} \cup B \times \{1\}$ and extended to arrows in the obvious manner.

The biproduct structure implies the existence of projection and injection arrows. For all $A, B \in Ob(\mathbf{Rel})$, the projection arrows are the following partial injections

- $\pi_0 = \{(a, (a, 0)) : a \in A\} \in \mathbf{Rel}(A \uplus B, A)$
- $\pi_1 = \{(b, (b, 1)) : b \in B\} \in \mathbf{Rel}(A \uplus B, B)$

and the injection arrows are their relational converses

- $\iota_0 = \{((a, 0), a) : a \in A\} \in \mathbf{Rel}(A, A \uplus B)$
- $\iota_1 = \{((b, 1), b) : b \in B\} \in \mathbf{Rel}(B, A \uplus B)$

Both **pFun** and **pInj** are closed under $\_ \uplus \_$; however it is simply a symmetric monoidal tensor on these subcategories and neither a product nor a coproduct. In all three settings, the disjoint union has as unit object the empty set $I = \{\}$, and hence a trivial monoid of scalars.

Remark 20 (*Units, scalars, and strict associativity*) The unit object of **Rel** is not a strict unit; rather, $A \uplus I = A \times \{0\} \cup \{\} \times \{1\} = A \times \{0\} \cong A$.

The unique arrow of the endomorphism monoid of I is the nowhere-defined function on the empty set, thus **Rel** and the distinguished monoidal subcategories described above provide good examples of unit objects with trivial scalars. The elements (i.e. arrows of the form $f : I \to X$) are similarly uninteresting; they are the nowhere-defined partial injections whose domain is the empty set. These categories will therefore illustrate how the **Int** or **GoI** construction builds a rich structure of elements and names that is essentially unrelated to that of the underlying traced category.

Note that disjoint union is associative up to canonical isomorphism, but is not strictly associative. Joyal et al. (1996) implicitly appeal to MacLane's strictification procedure for associativity & units (MacLane, 1998), and work within a suitably strictified version of $(\mathbf{Rel}, \uplus)$. As well as eliminating repeated re-bracketings, this also provides the justification for working with arbitrary (finite) matrices, as discussed below.

In our setting we need to be cautious of the result of Hines (2016), that we cannot simultaneously strictify associativity and self-similarity (see also Sect. 25.5.2). Hence, following Sect. 25.9, we are not able to assume strict associativity in a setting where we wish to find strictly reflexive objects—at least, in the compact closed setting.

In the following sections we therefore do not assume strict associativity, although canonical isomorphisms may occasionally be omitted for reasons of clarity. We also restrict ourselves to (2×2) matrices, and consider larger matrices as "matrices of matrices", with the precise interpretation determined by the source/target objects.

25.11.2 Matrices of Relations

The following is well-established, and is a corollary of the biproduct structure described above. It is also heavily used in Joyal et al. (1996).

Proposition 7 *Given arbitrary* $F \in \mathbf{Rel}(A \uplus B, P \uplus Q)$*, then* F *uniquely determines & is uniquely determined by a* 2×2 *matrix of relations* $[f_{ij}]_{i,j\in\{0,1\}}$ *whose entries are given in terms of the projection/injection arrows by* $f_{ij} = \pi_i F \iota_j$*. It is standard to abuse notation and write* $F = \begin{pmatrix} f_{00} & f_{01} \\ f_{10} & f_{11} \end{pmatrix}$*. The composite of relations in matrix form is then given by the usual formula for matrix composition, with addition and multiplication interpreted by union and relational composition respectively:*

$$\begin{pmatrix} g_{00} & g_{01} \\ g_{10} & g_{11} \end{pmatrix} \begin{pmatrix} f_{00} & f_{01} \\ f_{10} & f_{11} \end{pmatrix} = \begin{pmatrix} g_{00} f_{00} \cup g_{01} f_{10} & g_{11} f_{11} \cup g_{10} f_{01} \\ g_{10} f_{00} \cup g_{11} f_{10} & g_{11} f_{11} \cup g_{10} f_{01} \end{pmatrix}$$

This may also be drawn via the usual 'summing over paths' description of matrix composition where these matrices are drawn as digraphs:

$$\begin{tikzcd}
A \arrow[r, "f_{00}"] \arrow[dr, "f_{10}" near start] & P \arrow[r, "g_{00}"] \arrow[dr, "g_{10}" near start] & X \\
B \arrow[r, "f_{11}"'] \arrow[ur, "f_{01}"' near start] & Q \arrow[r, "g_{11}"'] \arrow[ur, "g_{01}"' near start] & Y
\end{tikzcd}$$

and matrix composition interprets as 'summing over paths from source to target':

$$\begin{tikzcd}
A \arrow[r, "g_{00} f_{00} \cup g_{01} f_{10}"] \arrow[dr, "g_{10} f_{00} \cup g_{11} f_{10}" near end] & X \\
B \arrow[r, "g_{11} f_{11} \cup g_{10} f_{01}"'] \arrow[ur, "g_{00} f_{01} \cup g_{01} f_{11}" near end] & Y
\end{tikzcd}$$

Proof (*Outline*) This is very well-established, and the formula for matrix composition may be derived from the observation that the composite of a projection and an injection acts like a Kronecker delta, so $\pi_j \iota_i = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}$ □

The category **Rel** was used as an illustration of a traced symmetric monoidal category in Joyal et al. (1996), and the structure of the resulting compact closed category **Int**(**Rel**) was also described in detail. The following, from this reference, is key:

Theorem 8 *The category* (**Rel**, $\uplus$) *is traced, with the trace defined in terms of the reflexive transitive closure of relations. Given* $F = \begin{pmatrix} f_{00} & f_{01} \\ f_{10} & f_{11} \end{pmatrix} : X \uplus U \to Y \uplus U$, *then*

$$Tr^U_{X,Y}(F) \;=\; f_{00} \cup f_{01} \left(\bigcup_{j=0}^{\infty} f_{11}^{j} \right) f_{10}$$

Proof The proof that this operation satisfies the axioms of Definition 11 is given in Joyal et al. (1996), and relies heavily on the properties of the Kleene star (i.e. reflexive transitive closure) of relations in endomorphism monoids. □

Remark 21 In Abramsky (1996), concrete examples of categorical traces were described as belonging to one of two classes: either 'particle-style' (based on iteration or feedback), or 'wave-style' (based on fixed-points). The above trace on (**Rel**, $\uplus$) is the canonical example of a particle-style trace.

25.12 A Matrix Formalism for Partial Injections

As they are monoidal subcategories of (**Rel**, $\uplus$), both (**pFun**, $\uplus$) and (**pInj**, $\uplus$) admit matrix representations of arrows. However, unlike **Rel**, their homsets are not closed under arbitrary unions, so some care is needed when using a matrix formalism. The case of partial functions was covered by Manes & Arbib in their study of algebraic program semantics (Arbib & Manes, 1979), and based on this, necessary and sufficient conditions for a matrix of partial injections to represent a partial injection was given in Hines (1997).

Proposition 8 *Given* $X, V, Y, U \in Ob(\mathbf{pInj})$*, the matrix representations of partial injections in* $\mathbf{pInj}(X \uplus V, Y \uplus U)$ *are precisely those matrices* $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ *whose entries are partial injections such that the following diagrams commute:*

$$\begin{tikzcd} Y \arrow[d, "c^{\ddagger}"'] & X \arrow[l, "a"'] \arrow[d, "b"] \arrow[dl, "0_{XV}"] \\ V & U \arrow[l, "d^{\ddagger}"] \end{tikzcd} \qquad \begin{tikzcd} Y & X \arrow[l, "a"'] \\ Y \arrow[u, "c"] & U \arrow[l, "d^{\ddagger}"] \arrow[u, "b^{\ddagger}"'] \arrow[ul, "0_{UY}"'] \end{tikzcd}$$

Proof (*Outline*) *This was first proved in* Hines (1997). *The key points of this proof are given below, in order to give some insight into the structure of matrix representations within* **pInj**.

The starting point is the observation that, given a family of partial injections $\{f_j\}_{j \in J} \in \mathbf{pInj}(X, Y)$, their union $\bigcup_{j \in J} f_j \in \mathbf{Rel}(X, Y)$ is a partial injection iff $f_i^{\ddagger} f_j$ and $f_i g_j^{\ddagger}$ are idempotent, for all $i, j \in J$. An important special case is when $f_i^{\ddagger} f_j =$

0_X and $f_i f_j^\ddagger = 0_Y$, so $f_i \cap f_j = \emptyset$, and hence the union $\bigcup_{j \in J} f_j$ is trivially a partial injection.

Now consider the above matrix of partial injections. The commutativity of the above two diagrams is equivalent (up to the appropriate inclusions) to this condition. In the other direction, the projection/injection arrows impose this condition when writing down the matrix form of a partial injection. □

Definition 18 The above condition is sometimes known as the **rook matrix condition**, since it states that elements in the same row of a matrix have disjoint images, and elements in the same column have disjoint domains. This description is more useful for matrix calculations within **pInj**, but the characterisation as commuting squares is more useful when working with the compact closed category **Int**(**pInj**).

We therefore refer to a square $\begin{tikzcd} Y & X \arrow[l, "a"'] \arrow[d, "b"] \\ V \arrow[u, "c"] \arrow[r, "d"'] & U \end{tikzcd}$ of arrows in **pInj** as a **rook square** when the two diagrams of Proposition 8 above commute.

As a corollary of the above characterisation, it was shown in Hines (1997) that (**pInj**, $\uplus$) is closed under the trace described in Theorem 8.

Theorem 9 *The symmetric monoidal category* (**pInj**, $\uplus$) *is traced, with the trace given by*

$$Tr^U_{X,Y}\begin{pmatrix} f_{00} & f_{01} \\ f_{10} & f_{11} \end{pmatrix} = f_{00} \cup \bigcup_{j=0}^{\infty} \left(f_{01} f_{11}^j f_{10} \right)$$

Proof This was first given in Hines (1997), but also proved independently in Haghverdi (2000) and Abramsky et al. (2002). □

Remark 22 Unlike the case of **Rel**, it is inaccurate to write the above trace in **pInj** in terms of the Kleene star, as $f_{00} \cup f_{01} \left(\bigcup_{j=0}^{\infty} f_{11}^j \right) f_{10}$. The reflexive transitive closure $\left(\bigcup_{j=0}^{\infty} f_{11}^j \right)$ is in general simply a relation, and not a partial injection.

25.13 The Compact Closed Category Int(pInj)

We now describe the compact closed category that results from applying Joyal, Street, & Verity's **Int** construction to the traced category (**pInj**, $\uplus$) of partial injections. We emphasise that this is simply the explicit description of **Int**(**Rel**), as found in Joyal et al. (1996), restricted to the traced symmetric monoidal subcategory of partial injections.

Definition 19 Plugging in the traced monoidal category (**pInj**, $\uplus$) into the **Int** construction of Joyal et al. (1996) gives the following:

Objects These are pairs of objects of **pInj** (i.e. pairs of sets).

Arrows Homsets are given by $\mathbf{Int}(\mathbf{pInj})((X,U),(Y,V)) = \mathbf{pInj}(X \uplus V, Y \uplus U)$. Arrows of $\mathbf{pInj}(X \uplus V, Y \uplus U)$ are given as (2×2) matrices of partial injections satisfying the rook matrix condition such as $\begin{pmatrix} f_{00} & f_{01} \\ f_{10} & f_{11} \end{pmatrix} \in \mathbf{Rel}(X \uplus V, Y \uplus U)$. Following Joyal et al., 1996, we draw the corresponding arrow of $\mathbf{Int}(\mathbf{pInj})((X,U),(Y,V))$ as a (planar) graphical 4-tuple of partial injections satisfying the rook square condition (Definition 18), as follows: $\begin{tikzcd} Y & X \arrow[l, "f_{00}"'] \arrow[d, "f_{10}"] \\ V \arrow[u, "f_{01}"] \arrow[r, "f_{11}"'] & U \end{tikzcd}$.

Composition To compose such digraphs, we simply glue then along their common edge, followed by taking the union over all paths with identical source & target:

$$\begin{tikzcd} Z & Y \arrow[l, "g_{00}"'] \arrow[d, bend right, "g_{10}"'] & X \arrow[l, "f_{00}"'] \arrow[d, bend right, "f_{10}"] \\ W \arrow[u, bend left, "g_{01}"] \arrow[r, "g_{11}"'] & V \arrow[u, bend right, "f_{01}"'] \arrow[r, "f_{11}"'] & U \end{tikzcd} \qquad \begin{tikzcd} Z & & X \arrow[ll, "\bigcup_{j=0}^{\infty} g_{00}(f_{01}g_{10})^j f_{00}"'] \arrow[d, bend right, "f_{10} \cup \bigcup_{j=0}^{\infty} f_{11}g_{10}\left(f_{01}g_{10}^j\right)f_{00}"] \\ W \arrow[u, bend left, "g_{01} \cup \bigcup_{j=0}^{\infty} g_{00}f_{01}\left(g_{10}f_{01}^j\right)g_{11}"] \arrow[rr, "\bigcup_{j=0}^{\infty} f_{11}(g_{01}f_{10})^j g_{11}"'] & & U \end{tikzcd}$$

Monoidal tensor The tensor $\_\Box\_$ is defined on objects by $(X,U)\Box(X',U') = (X \uplus X', U' \uplus U)$ and on arrows by:

$$\begin{tikzcd} Y & X \arrow[l, "a"'] \arrow[d, "c"] \\ V \arrow[u, "b"] \arrow[r, "d"'] & U \end{tikzcd} \Box \begin{tikzcd} Y' & X' \arrow[l, "a'"'] \arrow[d, "c'"] \\ V' \arrow[u, "b'"] \arrow[r, "d'"'] & U' \end{tikzcd} = \begin{tikzcd} Y \uplus Y' & & X \uplus X' \arrow[ll, "\begin{pmatrix} a & 0 \\ 0 & a' \end{pmatrix}"'] \arrow[d, "\begin{pmatrix} 0 & c' \\ c & 0 \end{pmatrix}"] \\ V' \uplus V \arrow[u, "\begin{pmatrix} 0 & b \\ b' & 0 \end{pmatrix}"] \arrow[rr, "\begin{pmatrix} d' & 0 \\ 0 & d \end{pmatrix}"'] & & U' \uplus U \end{tikzcd}$$

The unit object This is simply the object $(\emptyset, \emptyset)$.

The dual The contravariant dual $(\ )^* : \mathbf{Int}(\mathbf{pInj})^{op} \to \mathbf{Int}(\mathbf{pInj})$ is defined on objects by $(X,U)^* = (U,X)$, and on arrows by $\left(\begin{tikzcd} Y & X \arrow[l, "a"'] \arrow[d, "c"] \\ V \arrow[u, "b"] \arrow[r, "d"'] & U \end{tikzcd}\right)^* = \begin{tikzcd} U & V \arrow[l, "d"'] \arrow[d, "b"] \\ X \arrow[u, "c"] \arrow[r, "a"'] & Y \end{tikzcd}$.

The compact closed structure The unit and co-unit arrows for the compact closed structure,

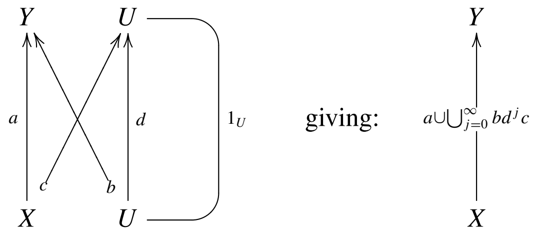
$$\eta_{(X,U)} : (\emptyset, \emptyset) \to (X,U)\Box(X,U)^* \text{ and } \epsilon_{(X,U)} : (X,U)^*\Box(X,U) \to (\emptyset, \emptyset)$$

are given by, respectively:

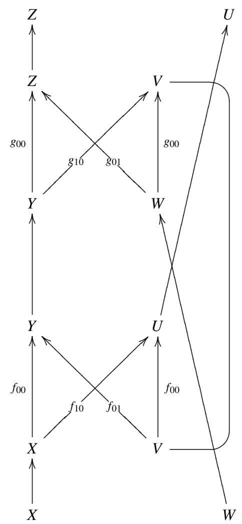
$$\begin{tikzcd} X \uplus U & \emptyset \arrow[l, "0"'] \arrow[d, "0"] \\ U \uplus X \arrow[u, "\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}"] \arrow[r, "0"'] & \emptyset \end{tikzcd} \quad \text{and} \quad \begin{tikzcd} \emptyset & U \uplus X \arrow[l, "0"'] \arrow[d, "\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}"] \\ \emptyset \arrow[u, "0"] \arrow[r, "0"'] & X \uplus U \end{tikzcd}$$

Remark 23 (*On the formulæ for composition*) The seemingly intricate formula for the composition of two rook squares arises in a very natural way from the diagrammatic representation of matrix composition, described in Proposition 7.

Taking the 'summing over paths' intuition seriously, the formula for the trace of $(\mathbf{Rel}, \uplus)$, and hence $(\mathbf{pInj}, \uplus)$, arises by introducing a feedback loop to the matrix representation of an arrow, and again summing over paths as shown:



In **pInj**, both composition and trace are given by 'summing over paths' constructions. Bringing these together gives the formula, and formalism, for composition in $\mathbf{Int}(\mathbf{pInj})$; consider arrows $F : (X, U) \to (Y, V)$ and $G : (Y, V) \to (Z, W)$ determined by rook matrices $\left(\begin{smallmatrix} f_{00} & f_{01} \\ f_{10} & f_{11} \end{smallmatrix}\right)$ and $\left(\begin{smallmatrix} g_{00} & g_{01} \\ g_{10} & g_{11} \end{smallmatrix}\right)$ respectively. The composite $GF \in \mathbf{Int}(\mathbf{pInj})((X, U), (Z, W))$ is then given by summing over paths in the following diagram:



Untangling this(!) but keeping the overall directed graph topology gives precisely the 'planar squares' formalism and composition described above. Thus, the rook squares formalism arises from taking the digraph representation of matrices of partial injections and re-drawing it in a planar manner. Similarly, the composition of IP is the natural 'summing over paths' operation that—unlike matrix composition—preserves planarity.[11]

Readers familiar with the conventions of Abramsky (1996), as well as those of Joyal et al. (1996) may wish to verify that Abramsky's composition within **GoI**(**pInj**) may be drawn in similar terms, and corresponds to *vertical* rather than *horizontal* pasting of rook squares, as illustrated below:

$$
\begin{tikzcd}
Y & X \arrow[l, "a"'] \arrow[d, "b"] \\
V \arrow[u, "c"] \arrow[r, bend left, "d"] & U \arrow[l, bend left, "f"] \arrow[d, "g"] \\
Q \arrow[u, "h"] \arrow[r, "k"'] & P
\end{tikzcd}
\qquad
\begin{tikzcd}
Y & & X \arrow[ll, "a\cup\bigcup_{j=0}^{\infty} c(fd)^j fb"'] \arrow[dd, "\bigcup_{j=0}^{\infty} g(df)^j b"] \\
 & & \\
Q \arrow[uu, "\bigcup_{j=0}^{\infty} c(fd)^j h"] \arrow[rr, "k\cup\bigcup_{j=0}^{\infty} g(df)^j dh"'] & & P
\end{tikzcd}
$$

Thus, at least at endomorphism monoids of self-dual objects (such as the reflexive, or strictly reflexive objects we discuss), it is reasonable to consider the compositions of Joyal et al. (1996) and Abramsky (1996) as distinct, but interacting, operations on the same underlying set (see Sect. 25.17).

25.14 Self-similarity, and Strict Self-similarity in pInj

We now exhibit self-similar objects of (**pInj**, $\uplus$), and demonstrate the semi-monoidal strictification of this self-similarity, as a first step towards exhibiting reflexive objects of **Int**(**pInj**), and strictifying this reflexivity.

It is easy to find self-similar objects of (**pInj**, $\uplus$); any countably infinite set D will suffice, and appropriate bijections between D and $D \uplus D$ are well-illustrated by the Hilbert's familiar parable of the Grand Hotel. Following the conventions of Girard (1988a, b), we will take the natural numbers as our canonical example, with self-similarity exhibited by the usual (bijective) Cantor pairing.

Definition 20 The **Cantor pairing** is the bijection $\lhd : \mathbb{N} \uplus \mathbb{N} \rightarrow \mathbb{N}$ given by $\lhd(n, i) = 2n + 1$. We denote its inverse by $\rhd : \mathbb{N} \rightarrow \mathbb{N} \uplus \mathbb{N}$; explicitly,

$$\rhd(n) = \begin{cases} \left(\frac{n}{2}, 0\right) & n \text{ even}, \\ \left(\frac{n-1}{2}, 1\right) & n \text{ odd}. \end{cases}$$

[11] An interesting open question is how much of the **Int** construction (at least in the 'particle-style' setting) may be thought of as 'imposing planarity' on some matrix calculus. We refer to Abramsky (2008) for further connections between planarity and (particle-style) compact closure.

This is closely related to the **dynamical algebra** of Danos and Regnier (1993) and Girard (1988a, b), which is the (inverse) submonoid of **pInj**$(\mathbb{N}, \mathbb{N})$ generated by the following partial injections

$$p(n) = \begin{cases} \frac{n}{2} & n \text{ even} \\ \text{undefined} & n \text{ odd} \end{cases} \quad , \quad q(n) = \begin{cases} \text{undefined} & n \text{ even} \\ \frac{n-1}{2} & n \text{ odd} \end{cases}$$

together with their generalised inverses $p^\ddagger(n) = 2n$ and $q^\ddagger(n) = 2n + 1$. These satisfy the following key conditions:

1. $pp^\ddagger = 1 = qq^\ddagger$
2. $pq^\ddagger = 0 = qp^\ddagger$
3. $p^\ddagger p \cup q^\ddagger q = 1$.

Remark 24 Conditions 1. and 2. above are the defining relations for the (two-generator) polycyclic monoid of Nivat and Perrot (1970), and condition 3. is a natural condition on concrete representations of polycyclic monoids. This observation was made in Hines (1997) and Lawson (1998) and polycyclic monoids form a significant and active research area in inverse semigroup theory generally.

Proposition 9 *The generators of the dynamical algebra, and their generalised inverses, arise as composites of the Cantor pairing and the canonical projection/injection arrows associated with the disjoint union, as*

$$p = \lhd \iota_0 \ , \ q = \lhd \iota_1 \ , \ p^\ddagger = \pi_0 \rhd \ , \ q^\ddagger = \pi_1 \rhd$$

Proof This may be verified almost instantly by direct calculation. It is also a special case of a more general connection between polycyclic monoids and categorical projections/injections described in Hines (1999). □

Lemma 5 *The endomorphism monoids of* $\mathbb{N}$ *and* $\mathbb{N} \uplus \mathbb{N}$ *in* **pInj** *are isomorphic.*

Proof The isomorphisms between them are given by conjugation by the code/decode arrows, as:

- $\lhd(\_)\rhd$: **pInj**$(\mathbb{N} \uplus \mathbb{N}, \mathbb{N} \uplus \mathbb{N}) \rightarrow$ **pInj**$(\mathbb{N} \uplus \mathbb{N}, \mathbb{N} \uplus \mathbb{N})$
- $\rhd(\_)\lhd$: **pInj**$(\mathbb{N}, \mathbb{N}) \rightarrow$ **pInj**$(\mathbb{N} \uplus \mathbb{N}, \mathbb{N} \uplus \mathbb{N})$. □

The interpretation of the dynamical algebra as composites of the Cantor pairing and projections/injections then allows us to make the link with matrix representations of arrows.

Corollary 9 *Let* $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \in$ **pInj**$(\mathbb{N} \uplus \mathbb{N}, \mathbb{N} \uplus \mathbb{N})$ *be a rook matrix. Then*

$$\lhd \begin{pmatrix} a & b \\ c & d \end{pmatrix} \rhd = p^\ddagger ap \cup p^\ddagger bq \cup q^\ddagger cp \cup q^\ddagger dq \in \mathbf{pInj}(\mathbb{N}, \mathbb{N})$$

Conversely, given $f \in \mathbf{pInj}(\mathbb{N},\mathbb{N})$*, then its inverse image under the monoid isomorphism described above is the following rook matrix:*

$$\rhd f \lhd = \begin{pmatrix} pfp^\ddagger & pfq^\ddagger \\ qfp^\ddagger & qfq^\ddagger \end{pmatrix} \in \mathbf{pInj}(\mathbb{N} \uplus \mathbb{N}, \mathbb{N} \uplus \mathbb{N})$$

Remark 25 As well as the above monoid isomorphism between $\mathbf{pInj}(\mathbb{N},\mathbb{N})$ and $\mathbf{pInj}(\mathbb{N} \uplus \mathbb{N}, \mathbb{N} \uplus \mathbb{N})$, there is also, by construction, a bijection between the rook squares of $\mathbf{Int}(\mathbf{pInj})((\mathbb{N},\mathbb{N}),(\mathbb{N},\mathbb{N}))$, and the rook matrices of $\mathbf{pInj}(\mathbb{N} \uplus \mathbb{N}, \mathbb{N} \uplus \mathbb{N})$. Thus, every arrow $f \in \mathbf{pInj}(\mathbb{N},\mathbb{N})$ uniquely determines and is determined by a rook square, with the correspondence given by $f \mapsto$
$$\begin{tikzcd} \mathbb{N} & \mathbb{N} \arrow[l, "pfp^\ddagger"'] \arrow[d, "qfp^\ddagger"] \\ \mathbb{N} \arrow[u, "pfq^\ddagger"] \arrow[r, "qfq^\ddagger"'] & \mathbb{N} \end{tikzcd}.$$

25.14.1 Strictifying Self-similarity within pInj

We first establish some notation & preliminary results:

Definition 21 Let us denote by $(\mathbf{pNat}, \uplus, \emptyset)$ the full symmetric monoidal subcategory of **pInj** generated by the natural numbers, together with disjoint union. Note that $(\mathbf{pNat}, \uplus)$ is also traced, with trace given by the trace of $(\mathbf{pInj}, \uplus)$.

We now give an explicit description of the semi-monoidal strictification of self-similarity within **pNat**. We take the code and decode arrows $\lhd : \mathbb{N} \uplus \mathbb{N} \to \mathbb{N}$ and $\rhd : \mathbb{N} \to \mathbb{N} \uplus \mathbb{N}$ to be as given in Definition 20, and similarly for the dynamical algebra.

Our starting point is a symmetric semi-monoidal tensor on the endomorphism monoid of $\mathbb{N}$; this is well-known as Girard's representation of multiplicative conjunction within (Girard, 1988a, b).

Definition 22 Given $f, g \in \mathbf{pNat}(\mathbb{N},\mathbb{N})$, we define their tensor $f \star g \in \mathbf{pNat}(\mathbb{N},\mathbb{N})$ by $f \star g = \lhd(f \uplus g)\rhd$. This may alternatively and equivalently be given in terms of the dynamical algebra, as $f \star g = p^\ddagger fp \cup q^\ddagger gq$. Explicitly,

$$(f \star g)(n) = \begin{cases} 2f\left(\frac{n}{2}\right) & n \text{ even, and } \frac{n}{2} \in dom(f) \\ 2g\left(\frac{n-1}{2}\right)+1 & n \text{ odd, and } \frac{n-1}{2} \in dom(g) \\ \perp & \text{otherwise} \end{cases}$$

The symmetry and associativity isomorphisms $\sigma, \tau \in \mathbf{pNat}(\mathbb{N}, \mathbb{N})$ for this tensor are given by $\sigma(n) = \begin{cases} n+1 & n \text{ even} \\ n-1 & n \text{ odd} \end{cases}$ and $\tau(n) = \begin{cases} 2n & n\ (mod\ 2) = 0 \\ n+1 & n\ (mod\ 4) = 1 \\ \frac{n-1}{2} & n\ (mod\ 4) = 3 \end{cases}$

These also may be given explicitly, in terms of the dynamical algebra, as

$$\sigma = p^\ddagger q \cup q^\ddagger p \quad \text{and} \quad \tau = \left(p^\ddagger\right)^2 p \cup p^\ddagger q^\ddagger pq \cup q^\ddagger q^2$$

Theorem 10 *The above operation $\_ \star \_$ is indeed a symmetric semi-monoidal tensor on the endomorphism monoid of* $\mathbb{N}$, *with associativity and symmetry maps as given above.*

Proof This is well-established (Hines, 1997; 1999), and the interpretation as a semi-monoidal strictification of self-similarity is given in Hines (2016). Explicit elementary arithmetic proofs of MacLane's pentagon and hexagon conditions are also given in Hines (2013). □

Now consider the semi-monoidal category $(\mathbf{pNat}_{-I}, \uplus)$ given by the 'forgetting the unit' functor of Definition 10. As well as the above tensor, the strictification procedure of Hines (2016) gives, as described in Sect. 25.6, a semi-monoidal equivalence of categories between $(\mathbf{pNat}_{-I}, \uplus)$ and $(\mathbf{pNat}(\mathbb{N}, \mathbb{N}), \star)$. This proceeds as follows:

Definition 23 For all objects $X \in Ob(\mathbf{pNat}_{-I})$, we define mutually inverse isomorphisms $C_X : X \to \mathbb{N}$ and $D_X : \mathbb{N} \to X$ inductively, by $C_\mathbb{N} = 1_\mathbb{N}$, and for all $A, B \in Ob(\mathbf{pNat}_{-I})$,

$$C_{A \uplus B} = \lhd(C_A \uplus C_B) \quad , \quad D_{A \uplus B} = (D_A \uplus D_B)\rhd$$

We then define a semi-monoidal functor $\Phi : (\mathbf{pNat}_{-I}, \uplus) \to (\mathbf{pNat}(\mathbb{N}, \mathbb{N}), \star)$ by:

Objects $\Phi(X) = \mathbb{N}$ for all $X \in Ob(\mathbf{pNat})$.
Arrows $\Phi(f) = C_B f D_A \in \mathbf{pNat}(\mathbb{N}, \mathbb{N})$, for all $f \in \mathbf{pNat}(A, B)$

Theorem 11 *The functor* $\Phi : (\mathbf{pNat}_{-I}, \uplus) \to (\mathbf{pNat}(\mathbb{N}, \mathbb{N}), \star)$ *is a semi-monoidal equivalence of categories.*

Proof This is immediate from the strictification procedure of Hines (2016) and heavily prefigured (albeit without the interpretation as a semi-monoidal equivalence of categories) in Hines (1997, 1999). □

We may now adjoint a strict unit to the above categories, as described in Definition 10. This results in the the following categories:

- $(\mathbf{pNat}(\mathbb{N}, \mathbb{N}), \star)_{+I}$, the above semi-monoidal category with a strict unit adjoined.
- $(\mathbf{pNat}, \uplus)_{-\frac{\mathcal{E}}{\mathcal{F}}\frac{\mathcal{A}}{\mathcal{W}}}$, the de-elemented version of $(\mathbf{pNat}, \uplus)$.

Corollary 10

1. *$(\mathbf{pNat}, \uplus)_{-\substack{\mathcal{E}\,\mathcal{A}\\ \mathcal{F}\,\mathcal{W}}}$ is monoidally equivalent to $(\mathbf{pNat}(\mathbb{N}, \mathbb{N}), \star)_{+I}$.*
2. *Both $(\mathbf{pNat}, \uplus)_{-\substack{\mathcal{E}\,\mathcal{A}\\ \mathcal{F}\,\mathcal{W}}}$ and $(\mathbf{pNat}(\mathbb{N}, \mathbb{N}), \star)_{+I}$ are traced.*

Proof

1. This follows immediately from Corollary 4, and the monoidal equivalence is given by a trivial extension of the Φ semi-monoidal functor of Definition 23 above to categories with strict units adjoined.
2. This follows from Proposition 4. The trace on $(\mathbf{pNat}, \uplus)_{-\substack{\mathcal{E}\,\mathcal{A}\\ \mathcal{F}\,\mathcal{W}}}$ is simply that of $(\mathbf{pNat}, \uplus)$, restricted to non-element arrows. The trace of $(\mathbf{pNat}(\mathbb{N}, \mathbb{N}), \star)_{+I}$ is given by, for all $f \in \mathbf{pNat}(\mathbb{N}, \mathbb{N})$,
   - $Tr^I_{I,I}(1_I) = 1_I$
   - $Tr^I_{\mathbb{N},\mathbb{N}}(f) = f$
   - $Tr^{\mathbb{N}}_{\mathbb{N},\mathbb{N}}(f) = f_{00} \cup \bigcup_{j=0}^{\infty} f_{01} f_{11}^j f_{10}$ where the components $f_{\_,\_}$ are as given in Corollary 9, as

$$f_{00} = pfp^\dagger \ \ , \quad f_{01} = pfq^\ddagger \ \ , \quad f_{10} = qfp^\ddagger \ \ , \quad f_{11} = qfq^\ddagger$$

□

25.15 From Self-similarity in pInj to (Strict) Reflexivity in Int(pInj)

As an immediate consequence of the self-similarity of the natural numbers, we may give self-dual self-similar objects in a compact closed category:

Lemma 6 *The (strictly) self-dual object $(N, N) = (N, N)^* \in Ob(\mathbf{Int}(\mathbf{pInj}))$ is self-similar.*

Proof From Corollary 6, as $\mathbb{N} \in Ob(\mathbf{pInj})$ is self-similar, so is $(\mathbb{N}, \mathbb{N}) \in Ob(\mathbf{Int}(\mathbf{pInj}))$, with the code/decode arrows for $(\mathbb{N}, \mathbb{N})$ given by the following rook squares:

$$\begin{tikzcd}
\mathbb{N} \uplus \mathbb{N} & \mathbb{N} \arrow[l, "\rhd"'] \arrow[d, "0"] \\
\mathbb{N} \uplus \mathbb{N} \arrow[u, "0"] \arrow[r, "\lhd"'] & \mathbb{N}
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
\mathbb{N} & \mathbb{N} \uplus \mathbb{N} \arrow[l, "\lhd"'] \arrow[d, "0"] \\
\mathbb{N} \arrow[u, "0"] \arrow[r, "\rhd"'] & \mathbb{N} \uplus \mathbb{N}
\end{tikzcd}$$

□

Corollary 11 $(\mathbb{N}, \mathbb{N}) \in Ob(\mathbf{Int}(\mathbf{pInj}))$ *is an extensionally reflexive object i.e. it is isomorphic to its own internal hom., so* $[(\mathbb{N}, \mathbb{N}) \to (\mathbb{N}, \mathbb{N})] \cong (\mathbb{N}, \mathbb{N})$.

Proof This is immediate from the characterisation of extensionally reflexive objects given in Lemma 1. □

Our stated aim is to provide concrete examples of how reflexivity may be strictified in a compact closed category—how we may give a compact closed subcategory containing the specified reflexive object, together with a monoidal equivalence to another compact closed category in which this reflexivity is exhibited by identity arrows. We do so by applying the abstract procedures laid out in Sect. 25.9 to the above natural example of an extensionally reflexive object in $\mathbf{Int}(\mathbf{pInj})$.

The following, although individually straightforward, will prove powerful:

Proposition 10

1. $\mathbf{Int}(\mathbf{pNat})$ *is a compact closed category where all objects* $(X, U) \in Ob(\mathbf{Int}(\mathbf{pNat}))$ *satisfying* $X \not\cong \emptyset \not\cong U$ *are isomorphic.*
2. *Let us denote by* $\mathbf{IpN}$ *the full monoidal subcategory of* $\mathbf{Int}(\mathbf{pNat})$ *generated by the self-dual object* $(\mathbb{N}, \mathbb{N})$*. Then* $\mathbf{IpN}$ *is a compact closed category where all non-unit objects are isomorphic.*

Proof

1. By construction, arbitrary non-unit objects $X, Y, U, V \in Ob(\mathbf{pNat})$ are all isomorphic. Let us fix isomorphisms $\phi \in \mathbf{pNat}(X, Y)$ and $\psi \in \mathbf{pNat}(U, V)$. Then the following rook squares give an isomorphism in $\mathbf{Int}(\mathbf{pNat})((X, U), (Y, V))$, together with its inverse:

$$\begin{tikzcd} Y & X \arrow[l, "\phi"'] \arrow[d, "0"] \\ V \arrow[u, "0"] \arrow[r, "\psi^{-1}"'] & U \end{tikzcd} \qquad \begin{tikzcd} X & Y \arrow[l, "\phi^{-1}"'] \arrow[d, "0"] \\ U \arrow[u, "0"] \arrow[r, "\psi"'] & V \end{tikzcd}$$

2. Note that point 1. above does *not* imply that all non-unit objects of $\mathbf{Int}(\mathbf{pNat})$ are isomorphic; counterexamples are provided by (X, I) and (I, X), where $X \neq \emptyset \in Ob(\mathbf{pNat})$. However, by construction, the full monoidal subcategory $\mathbf{IpN}$ generated by $(\mathbb{N}, \mathbb{N})$ is closed under tensor and dual, and by point 3., all non-unit objects are isomorphic. As it is a full subcategory, it also contains the relevant unit/co-unit arrows. □

Following the program laid out in above, we are of course moving towards a monoidal equivalence between the category $\mathbf{IpN}$ of part 4. above, and a compact closed category with a single non-unit object.

25.15.1 A Monoidal Strictification of Extensional Reflexivity

Corollary 10 above gives us precisely what we require for the strictification of reflexivity at a reflexive object of a compact closed category. We have a two-object traced monoidal category (i.e. $(\mathbf{pNat}(\mathbb{N}, \mathbb{N}), \star)_{+I}$) that is monoidally equivalent to a small subcategory of a (de-elemented version of) **pInj**, generated by a reflexive object.

Remark 26 The next step is to apply the **Int** construction to this two-object traced monoidal category; this will result in a four-object compact closed category that contains a strictly reflexive object, $(\mathbb{N}, \mathbb{N})$. We then consider the two-object compact closed monoidal subcategory generated by this distinguished object.

Definition 24 As our notation is in danger of becoming unwieldy at this point, let us simply denote by $\mathfrak{G}$ the two-object compact closed category resulting from the steps of Remark 26 above.

Expanding out the definitions results in the following:

Objects $Ob(\mathfrak{G}) = \{(I, I), (\mathbb{N}, \mathbb{N})\}$

Hom-sets By construction, all homsets (excluding the endomorphism monoid of the units) are equal:

Scalars $\mathfrak{G}((I, I), (I, I)) = \{1_I\}$
Elements $\mathfrak{G}((I, I), (\mathbb{N}, \mathbb{N})) = \mathbf{pNat}(I \star \mathbb{N}, \mathbb{N} \star I) = \mathbf{pNat}(\mathbb{N}, \mathbb{N})$
Co-Elements $\mathfrak{G}((\mathbb{N}, \mathbb{N}), (I, I)) = \mathbf{pNat}(\mathbb{N} \star I, I \star \mathbb{N}) = \mathbf{pNat}(\mathbb{N}, \mathbb{N})$
Endomorphisms $\mathfrak{G}((\mathbb{N}, \mathbb{N}), (\mathbb{N}, \mathbb{N})) = \mathbf{pNat}(\mathbb{N} \star \mathbb{N}, \mathbb{N} \star \mathbb{N}) = \mathbf{pNat}(\mathbb{N}, \mathbb{N})$

It will be convenient to describe arrows of homsets using the correspondence between members of $\mathbf{pNat}(\mathbb{N}, \mathbb{N})$ and rook squares over $\mathbf{pNat}(\mathbb{N}, \mathbb{N})$ of Remark 25. Given the same endomorphism $f \in \mathbf{pNat}(\mathbb{N}, \mathbb{N})$, we draw it in different ways as a rook square[12] in different homsets of $\mathfrak{G}$.

Elements In $\mathfrak{G}((I, I), (\mathbb{N}, \mathbb{N}))$, we draw f as
$$\begin{tikzcd} \mathbb{N} & I \arrow[l, "0"'] \arrow[d, "1_I"] \\ \mathbb{N} \arrow[u, "f"] \arrow[r, "0"'] & I \end{tikzcd}$$

Co-Elements In $\mathfrak{G}((\mathbb{N}, \mathbb{N}), (I, I))$, we draw f as
$$\begin{tikzcd} I & \mathbb{N} \arrow[l, "0"'] \arrow[d, "f"] \\ I \arrow[u, "1_I"] \arrow[r, "0"'] & \mathbb{N} \end{tikzcd}$$

[12] Note the use of a formal zero arrow, to denote an empty homset, in the rook squares for elements/co-elements. This is simply a notational convenience—we are not claiming the existence of a zero arrow between the formal unit object and other objects in the same category. Thus we have not quite arrived back in the category **pInj**, although the difference between an empty homset, and a homset containing the nowhere-defined function on the empty set, is subtle!

Endomorphisms In $\mathfrak{G}((\mathbb{N},\mathbb{N}),(\mathbb{N},\mathbb{N}))$, we draw f as
$$\begin{tikzcd} \mathbb{N} & \mathbb{N} \arrow[l, "pfp^\ddagger"'] \arrow[d, "qfp^\ddagger"] \\ \mathbb{N} \arrow[u, "pfq^\ddagger"] \arrow[r, "qfq^\ddagger"'] & \mathbb{N} \end{tikzcd}$$
and refer to Remark 25 for the observation that this rook square uniquely determines and is determined by $f \in \mathbf{pNat}(\mathbb{N},\mathbb{N})$.

Composition This is given by the 'pasting rook squares and summing over paths' described in Definition 19.

The tensor Tensors with the unit object are defined by strictness, so $(I,I)\Box\_$ and $\_\Box(I,I)$ are identity functors. At the non-unit object, we have $(\mathbb{N},\mathbb{N})\Box(\mathbb{N},\mathbb{N}) = (\mathbb{N},\mathbb{N})$, and for arrows, the tensor is defined on rook square representations, as

$$\begin{tikzcd} \mathbb{N} & \mathbb{N} \arrow[l, "f_{00}"'] \arrow[d, "f_{10}"] \\ \mathbb{N} \arrow[u, "f_{01}"] \arrow[r, "f_{11}"'] & \mathbb{N} \end{tikzcd} \ \Box\ \begin{tikzcd} \mathbb{N} & \mathbb{N} \arrow[l, "g_{00}"'] \arrow[d, "g_{10}"] \\ \mathbb{N} \arrow[u, "g_{01}"] \arrow[r, "g_{11}"'] & \mathbb{N} \end{tikzcd} = \begin{tikzcd} \mathbb{N} & & \mathbb{N} \arrow[ll, "p^\ddagger f_{00} p \cup q^\ddagger g_{00} q"'] \arrow[d, "q^\ddagger g_{10} p \cup p^\ddagger f_{10} q"] \\ \mathbb{N} \arrow[u, "p^\ddagger f_{01} q \cup q^\ddagger g_{01} p"] \arrow[rr, "p^\ddagger g_{11} p \cup q^\ddagger f_{11} q"'] & & \mathbb{N} \end{tikzcd}$$

The canonical isomorphisms

Associativity From Lemma 2, the associator T for the above tensor is $\tau \star \tau^{-1}$, where $\star$ is the tensor of $\mathbf{pNat}(\mathbb{N},\mathbb{N})_{+I}$ and $\left(p^\ddagger\right)^2 p \cup p^\ddagger q^\ddagger pq \cup q^\ddagger q^2$ is the corresponding associator. In rook square notation, this gives
$$\begin{tikzcd} \mathbb{N} & \mathbb{N} \arrow[l, "\tau"'] \arrow[d, "0"] \\ \mathbb{N} \arrow[u, "0"] \arrow[r, "\tau^{-1}"'] & \mathbb{N} \end{tikzcd}$$

Symmetry Also from Lemma 2, the commutativity isomorphism for $\_\Box\_$ is given by $\sigma \star \sigma$, where $\sigma = q^\dagger p \cup p^\dagger q$ is the commutativity isomorphism for $\_ \star \_$. In rook square notation this is simply
$$\begin{tikzcd} \mathbb{N} & \mathbb{N} \arrow[l, "\sigma"'] \arrow[d, "0"] \\ \mathbb{N} \arrow[u, "0"] \arrow[r, "\sigma"'] & \mathbb{N} \end{tikzcd}$$

The unit object By construction, we have a strict unit object, (I,I).

The compact closed structure

The dual The dual of $\mathfrak{G}$ is a dagger, which we nevertheless write as $(\ )^*$, to avoid confusion with the generalised inverse of **pInj**. Thus, on objects, $\mathbb{N}^* = \mathbb{N}$, and on arrows it is given by Definition 12. Explicitly, given an arrow in $\mathfrak{G}((X,U),(Y,V))$ represented as a rook square, its dual is given by
$$\left(\begin{tikzcd} Y & X \arrow[l, "a"'] \arrow[d, "c"] \\ V \arrow[u, "b"] \arrow[r, "d"'] & U \end{tikzcd}\right)^* = \begin{tikzcd} U & V \arrow[l, "d"'] \arrow[d, "b"] \\ X \arrow[u, "c"] \arrow[r, "a"'] & Y \end{tikzcd}$$

The unit and co-unit Recall that within $\mathfrak{G}$,

$$(\mathbb{N},\mathbb{N})\Box(\mathbb{N},\mathbb{N})^* \;=\; (\mathbb{N},\mathbb{N}) \;=\; (\mathbb{N},\mathbb{N})^*\Box(\mathbb{N},\mathbb{N})$$

We then have the two distinguished arrows for the compact closed structure, $\eta : (I, I) \to (\mathbb{N},\mathbb{N})$ and $\epsilon : (\mathbb{N},\mathbb{N}) \to (I, I)$ given by, respectively:

$$\begin{tikzcd} \mathbb{N} & I \arrow[l, "0"'] \arrow[d, "0"] \\ \mathbb{N} \arrow[u, "\sigma"] \arrow[r, "0"'] & I \end{tikzcd} \quad \text{and} \quad \begin{tikzcd} I & \mathbb{N} \arrow[l, "0"'] \arrow[d, "\sigma"] \\ I \arrow[u, "0"] \arrow[r, "0"'] & \mathbb{N} \end{tikzcd}$$

where $\sigma = p^\ddagger q \cup q^\ddagger p$ is the symmetry map for $\_ \star \_$, the tensor of the underlying traced monoidal category.

Remark 27 The above two-object compact closed category provides an example of the strictification of extensional reflexivity described in abstract terms in the first half of this paper; we have already seen that the following key properties are satisfied:

1. The compact closed category $(\mathfrak{G}, \Box, (\ )^*, (I, I))$ is monoidally equivalent to the compact closed subcategory of $\mathbf{Int}(\mathbf{pInj})$ monoidally generated by $(\mathbb{N},\mathbb{N}) \in Ob(\mathbf{Int}(\mathbf{pInj}))$.
2. The object $(\mathbb{N},\mathbb{N}) \in Ob(\mathbf{Int}(\mathbf{pInj}))$ is extensionally reflexive.
3. The unique non-unit object $(\mathbb{N},\mathbb{N}) \in Ob(\mathfrak{G})$ is strictly extensionally reflexive.

25.16 Algebraic Aspects

Having established concrete examples of the relevant abstract category theory, we move on to considering the algebraic aspects—precisely, the embeddings of Thompson's $\mathcal{F}$ associated with coherence for associativity, and the simple Abramsky-Heunen Frobenius monoid associated with strict reflexivity in a compact closed category.

Definition 25 Following Point 3 of Sect. 25.9.1, we observe that the endomorphism monoids $\mathfrak{G}((\mathbb{N},\mathbb{N}),(\mathbb{N},\mathbb{N}))$ and $\mathbf{pInj}(\mathbb{N},\mathbb{N})$ have the same underlying set. Let us denote this by $\mathcal{H}$. It will be convenient to use rook matrix notation for elements of $\mathcal{H}$, and write a partial injection f as $\begin{pmatrix} pfp^\ddagger & pfq^\ddagger \\ qfp^\ddagger & qfq^\ddagger \end{pmatrix}$; entirely equivalently, we say that $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ is simply shorthand for the partial injection $p^\ddagger ap \cup p^\ddagger bq \cup q^\ddagger cp \cup q^\ddagger dq$. (As a notational device, we will also denote composition within $\mathbf{pInj}$ itself simply by concatenation).

Let us denote the composition on $\mathcal{H}$ arising from $\mathbf{pInj}$ as $\_ \cdot \_ : \mathcal{H} \times \mathcal{H} \to \mathcal{H}$, and that arising from $\mathfrak{G}$ as $\_ \circ \_ : \mathcal{H} \times \mathcal{H} \to \mathcal{H}$. Explicitly, these are given by:

- $\begin{pmatrix} e & f \\ g & h \end{pmatrix} \cdot \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} ea \cup fc & eb \cup fd \\ ga \cup hc & gb \cup hd \end{pmatrix}$.
- $\begin{pmatrix} e & f \\ g & h \end{pmatrix} \circ \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} \bigcup_{j=0}^{\infty} e\,(bg)^j\, a & f \cup \bigcup_{j=0}^{\infty} eb\left(gb^j\right) h \\ c \cup \bigcup_{j=0}^{\infty} dg\left(bg^j\right) a & \bigcup_{j=0}^{\infty} d\,(fc)^j\, h \end{pmatrix}$

By construction, we have two additional operations $\_ \star \_, \_\Box\_ : \mathcal{H} \times \mathcal{H} \to \mathcal{H}$ that are symmetric semi-monoidal tensors for $(\mathcal{H}, \cdot)$ and $(\mathcal{H}, \circ)$ respectively. These may be given explicitly, by:

- $\begin{pmatrix} e & f \\ g & h \end{pmatrix} \star \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} p^\ddagger ep \cup p^\ddagger fq \cup q^\ddagger gp \cup q^\ddagger hq & 0 \\ 0 & p^\ddagger ap \cup p^\ddagger bq \cup q^\ddagger cp \cup q^\ddagger dq \end{pmatrix}$.
- $\begin{pmatrix} e & f \\ g & h \end{pmatrix} \Box \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} p^\ddagger ap \cup q^\ddagger eq & p^\ddagger bq \cup q^\ddagger fp \\ p^\ddagger cq \cup q^\ddagger gp & p^\ddagger hp \cup q^\ddagger dq \end{pmatrix}$.

The symmetry and associativity isomorphisms for the semi-monoidal monoid $(\mathcal{H}, \cdot, \star)$ are given by, respectively,

$$\sigma = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \text{and} \quad \tau = \begin{pmatrix} p^\ddagger & q^\ddagger p \\ 0 & q \end{pmatrix}$$

The inverse of τ, with respect to the composition $\_ \cdot \_$ is given explicitly by $\tau' = \begin{pmatrix} p & 0 \\ p^\ddagger q & q^\ddagger \end{pmatrix}$, and σ is self-inverse w.r.t. the same composition.

Similarly, the associativity and symmetry isomorphisms for the semi-monoidal monoid $(\mathcal{H}, \circ, \Box)$ are given by, respectively, $S = \begin{pmatrix} \sigma & 0 \\ 0 & \sigma \end{pmatrix}$ and $T = \begin{pmatrix} \tau & 0 \\ 0 & \tau' \end{pmatrix}$.

Expanding out these definitions in terms of the dynamical algebra, we get $S = \begin{pmatrix} p^\ddagger q \cup q^\ddagger p & 0 \\ 0 & p^\ddagger p \cup q^\ddagger p \end{pmatrix}$ and

$$T = \begin{pmatrix} \left(p^\ddagger\right)^2 p \cup p^\ddagger q^\ddagger pq \cup q^\ddagger q^2 & 0 \\ 0 & p^\ddagger p^2 \cup q^\ddagger p^\ddagger qp \cup \left(q^\ddagger\right)^2 q \end{pmatrix}$$

S is then its own inverse, w.r.t. both compositions on $\mathcal{H}$. The inverse of T (again, w.r.t. both compositions) is given by

$$T' = \begin{pmatrix} p^\ddagger p^2 \cup q^\ddagger p^\ddagger qp \cup \left(q^\ddagger\right)^2 q & 0 \\ & \left(p^\ddagger\right)^2 p \cup p^\ddagger q^\ddagger pq \cup q^\ddagger q^2 \end{pmatrix}$$

The following identities are then immediate from both the algebraic description, and the details of the **Int** construction.

Lemma 7 *The above distinct associativity and symmetry elements are related as follows:*

- $S = \sigma \star \sigma = \sigma \cdot S \cdot \sigma$

- $T = \tau \star \tau'$
- $T' = \sigma \cdot T \cdot \sigma$

Proof These follow, simply by construction. $\square$

The above structure, consisting of two distinct monoid compositions and two distinct semi-monoidal structures, along with non-trivial interactions between them, then provides examples of the algebraic structures discussed in Sects. 25.5.4 and 25.10.2.

Theorem 12

1. *The submonoid of* $(\mathcal{H}, \cdot)$ *generated by* $\{\tau, \tau', 1 \star \tau, 1 \star \tau'\}$ *is a group isomorphic to Thompson's group* $\mathcal{F}$*, and is also generated by the closure of* $\{\tau, \tau'\}$ *under the composition* $\cdot$ *and the tensor* $\star$*.*
2. *The submonoid of* $(\mathcal{H}, \circ)$ *generated by* $\{T, T', 1\square T, 1\square T'\}$ *is again a subgroup isomorphic to Thompson's group* $\mathcal{F}$*, and is also generated by the closure of* $\{T, T'\}$ *under the composition* $\circ$ *and the tensor* $\square$*.*
3. *The elements*

   - $\Delta = \tau \cdot (1 \star \sigma)$
   - $\nabla = (1 \star \sigma) \cdot \tau'$

   satisfy $\Delta \cdot \nabla = 1 = \nabla \cdot \Delta$*, but*

   $$\nabla \circ \Delta = 1 \quad \textit{and} \quad \Delta \circ \nabla \neq 1$$

   Hence $\{\Delta, \nabla\}$ *generates a copy of the bicyclic monoid within* $(\mathcal{H}, \circ)$*.*
4. *The above elements satisfy*

   The Frobenius condition $\quad (1\square\nabla) \circ T' \circ (\Delta\square 1) = \Delta \circ \nabla = (\nabla\square 1) \circ T(1\square\Delta)$
   Associativity $\quad \nabla \circ (1\square\nabla) = \nabla \circ (\nabla\square 1) \circ T$
   Co-Associativity $\quad (1\square\Delta) \circ \Delta = ((\Delta\square 1) \circ \Delta) \circ T.$

   and hence generate a copy of the simple Abramsky-Heunen Frobenius algebra within the semi-monoidal monoid $(\mathcal{H}, \circ, \square)$*.*

Proof

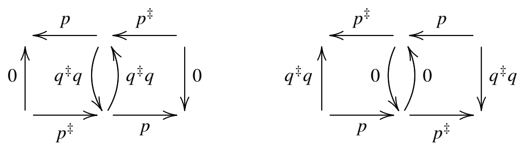
1. As $(\mathcal{H}, \cdot, \star)$ is a semi-monoidal monoid, this follows directly from Theorem 1.
2. Similarly, $(\mathcal{H}, \circ, \square)$ is a semi-monoidal monoid; this again follows directly from Theorem 1.
3. Although this is a corollary of the categorical reasoning of Part 3. of Theorem 7, it is also perhaps the last point at which a purely algebraic proof is readily accessible, so we prove this directly, as a check that our categorical reasoning is indeed correct. Expanding out the definition gives that

$$\Delta = \tau \cdot (1 \star \sigma) = \begin{pmatrix} p^\ddagger & q^\ddagger p \\ 0 & q \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & q^\ddagger p \cup p^\ddagger q \end{pmatrix} = \begin{pmatrix} p^\ddagger & q^\ddagger q \\ 0 & p \end{pmatrix}$$

Similarly,

$$\nabla = (1 \star \sigma) \cdot \tau' = \begin{pmatrix} 1 & 0 \\ 0 & q^\ddagger p \cup p^\ddagger q \end{pmatrix} \begin{pmatrix} p & 0 \\ p^\ddagger q & q^\ddagger p \end{pmatrix} = \begin{pmatrix} p & 0 \\ q^\ddagger q & p^\ddagger \end{pmatrix}$$

Direct matrix composition, together with the defining relations of the dynamical algebra, then show that $\Delta \cdot \nabla = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \nabla \cdot \Delta$. For the composite derived from the compact closed structure, a straightforward route to calculating the composites $\nabla \circ \Delta$ and $\Delta \circ \nabla$ is given by moving to the 'rook squares' formalism, and summing over paths within the following two diagrams:



Relying on the key identities $pq^\ddagger = 0 = qp^\ddagger$ simplifies this considerably, and gives the composites as:

$$\nabla \circ \Delta = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad \Delta \circ \nabla = \begin{pmatrix} p^\ddagger p & q^\ddagger q \\ q^\ddagger q & p^\ddagger p \end{pmatrix}$$

so $\nabla \circ \Delta = 1$ and $\Delta \circ \nabla \neq 1$ as required. (It is worth observing the curiosity that $(\Delta \circ \nabla) \cdot (\Delta \circ \nabla) = 1$. The categorical significance of this is currently unknown).

4. For this, we must simply appeal to the abstract category theory already developed, and claim it as a Corollary of Theorem 5. □

25.17 Future Directions

I would like to thank Samson Abramsky for the advice that I should never give a comprehensive account of a subject, but rather leave some 'low-hanging fruit' so that other authors have the opportunity to reference me. Although this was undoubtedly firmly tongue-in-cheek, there are nevertheless many directions that could be pursued.

Categorically It is hard to avoid the conclusion that this paper needs a good dose of the scalars. The strictification of reflexivity described in Sect. 25.9 relies on the underlying traced monoidal category having trivial scalars, and the interaction of Frobenius algebras, Thompson's $\mathcal{F}$ and the bicyclic monoid is also predicated on the monoid of scalars being trivial. It is presumably possible to get rid of this requirement in both cases, and give a somewhat more sophisticated procedure that can also deal with a non-trival monoid of scalars. Although this work remains to be carried out, the key to it is undoubtedly the methods of Abramsky (2005) of adjoining a non-trivial commutative monoid (with involution) of scalars to a traced or compact closed category.

Logically This paper has concentrated on categorical & algebraic aspects of the Geometry of Interaction system, rather than logical interpretations. A logical puzzle arises nevertheless; in the system of Girard (1988a, b), there is unavoidably a simple A-H Frobenius monoid, as studied in Sect. 25.10.2, and given explicitly in Sect. 25.16. The split/merge distinguished arrows of a Frobenius algebra have the interpretation with quantum-mechanical systems as 'fan-out'—a restricted form of copying that does not violate the no-cloning theorem (Wootters & Zurek, 1982), but does provide a great deal of computational power to quantum computational systems (Høyer & Špalek, 2005). The question is then simply, 'to which features of (resource-sensitive) linear logic do these arrows correspond?'. Possibly relevant is the fact that we have not, so far, given a treatment of how copying is treated within linear logic generally, or within the systems of Girard (1988a, b) and Abramsky et al. (2002) in particular. Logically, this is via the $!(\_)$ modality, which—as shown in Hines (1997)—appears in Girard (1988a, b) as a fixed-point functor on a semi-monoidal monoid $(M, \star)$, defined by $f \star !(f) = !(f)$, for all $f \in M$.

Algebraically Canonical coherence arrows of semi-monoidal monoids have a habit of appearing as interesting or well-known purely algebraic structures—the case of Thompson's $\mathcal{F}$ is a prime example. Similarly, the logicians' dynamical algebra has long been identified as not only Nivat and Perot's polycyclic monoids (Hines, 1997, Lawson, 1998), but also the algebra of projections/injections for a semi-monoidal monoid (Hines, 1997, 1999). When we consider symmetry as well as associativity arrows, we instead have to deal with Thompson's group $\mathcal{V}$ (Lawson, 2007). This immediately raises the natural question of the standard or simple A-H Frobenius monoid(s). It is entirely reasonable to expect these, considered simply as monoids rather than semi-monoidal monoids, to be well-known & well-studied for their algebraic properties.

Theorem 7 provides some justification for the intuition that the simple A-H F monoid is a combination of Thompson's $\mathcal{F}$ and the bicyclic monoid, with interactions between the two determined by the Frobenius condition—thus uniting three different iconic structures from different branches of algebra.

What is, however, missing from the above account is some subset of the A-H F monoid that generates it by closure under composition only (i.e. not closure under composition and tensor). This would undoubtedly make it more accessible to the algebra community, which is a natural setting in which it should also be studied. Their identification as well-known algebra must surely be close at hand.

Notation and diagrammatics The power of a good formalism for both making a subject accessible, and for developing new theory, can hardly be overestimated. This is shown by the utility of the string diagrams formalism for compact closed categories generally, and its application in the categorical quantum mechanics program in particular.

Unfortunately, it is also singularly inappropriate for working with strict reflexivity generally, and A-H Frobenius monoids in particular, due to its reliance on *strict* monoidal tensors. Recall from Hines (2016) that one cannot simultaneously have strict associativity and strict self-similarity (& hence strict reflexivity), except in the trivial case where everything collapses to the unit object. As the key building blocks of the A-H Frobenius monoids are (necessarily non-strict) associators, such a diagrammatic formalism is simply inapplicable.

A suitable formalism that illustrates the underlying concepts—which in many cases are actually quite simple—without becoming bogged down in syntax, is sorely needed.

Acknowledgements This paper owes a large debt to Chris Heunen (Edinburgh), both for the concrete results indicated in the text, and many stimulating discussions on the topics listed of this chapter. He has also been responsible for correcting several wrong turns I had taken, which is greatly appreciated. Thanks are due to Mark Lawson for numerous references and results on the inverse semigroup theoretic side, as well as for neat constructions that significantly simplify otherwise complex results. Phil Scott (Ottawa) has been very helpful, with discussions about linear logic and the Geometry of Interaction in general, how untyped logical and computational systems should be modeled, and in particular the rôle of units in untyped systems. I am also very grateful to Noson Yanofsky, for discussions on the structure and properties of Thompson's group $\mathcal{F}$ and related structures, and their interaction with categorical coherence generally, and Hilbert Hotel style operations in particular.

Finally, thanks are due to Samson Abramsky for uncountably many reasons; attempting to list them all would be futile.

References

Abramsky, S. (1996). Retracing some paths in process algebra. In U. Montanari & V. Sassone (Eds.), *CONCUR '96: Concurrency theory* (pp. 1–17). Berlin: Springer.

Abramsky, S. (2005). Abstract scalars, loops, and free traced and strongly compact closed categories. In J. Fiadeiro, N. Harman, M. Roggenbach, & J. Rutten (Eds.), *Algebra and coalgebra in computer science* (pp. 1–29). Berlin: Springer.

Abramsky, S. (2008). Temperley-lieb algebra: From knot theory to logic and computation via quantum mechanics. In *Mathematics of quantum computation and quantum technology*. Applied Mathematics and Nonlinear Sciences Series (pp. 515–558). Boca Raton: Chapman & Hall/CRC.

Abramsky, S., Blute, R., & Panangaden, P. (1999). Nuclear and trace ideals in tensored *-categories. *Journal of Pure and Applied Algebra, 143,* 3–47.

Abramsky, S., & Coecke, B. (2005). A categorical semantics of quantum protocols. In *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science (lics 2004)* (pp. 415–425). IEEE Computer Society Press.

Abramsky, S., Haghverdi, E., & Scott, P. (2002). Geometry of interaction and linear combinatory algebras. *Mathematical Structures in Computer Science,12*(5).

Abramsky, S., & Heunen, C. (2010). H*-algebras and nonunital frobenius algebras: First steps in infinitedimensional categorical quantum mechanics (Vol. 71, p. 11).

Arbib, M., & Manes, E. (1979). Partially-additive monoids, graph-growing, and the algebraic semantics of recursive calls. Lecture Notes in Computer Science. *Graph-grammars and their application to computer science and biology* (Vol. 73, pp. 127–138). Berlin: Springer.

Brin, M. G. (1996). The chameleon groups of Richard J. Thompson: Automorphisms and dynamics. *Publications Mathématiques de l'IHÉS,84*, 5–33.

Brin, M. G. (2005). Elementary amenable subgroups of R. Thompson's group F. *International Journal of Algebra and Computation, 15*(04), 619–642.

Brown, K. (2006). The homology of richard Thompson's group F. 394. In R. Grigorchuk, M. Mihalik, M. Sapir, & Z. Sunik (Eds.), *Topological and asymptotic aspects of group theory*. Contemporary mathematics (pp. 47–60).

Buszkowski, W. (2001). Lambek grammars based on pregroups. In *Logical Aspects of Computational Linguistics, 4th International Conference, LACL 2001, Le Croisic, France, June 27–29, 2001, Proceedings* (pp. 95–109).

Buszkowski, W. (2003). Sequent systems for compact bilinear logic. *Mathematical Logic Quarterly, 49*(5), 467–474.

Coecke, B., & Pavlovic, D. (2007). Quantum measurements without sums. In G. Chen, L. Kauffman, & S. Lamonaco (Eds.), *Mathematics of quantum computing and technology*. London: Chapman & Hall.

Coecke, B., Pavlovic, D., & Vicary, J. (2013). A new description of orthogonal bases. *Mathematical Structures in Computer Science, 23,* 555–567.

Danos, V., & Regnier, L. (1993). Local and asynchronous beta reduction. In *Proceedings of the Eighth Annual IEEE Symposium on Logic in Computer Science*.

Dehornoy, P. (1996). The structure group for the associativity identity. *Journal of Pure and Applied Algebra, 111*(1–3), 59–82.

Deutsch, D. (1985). Quantum theory, the Church-Turing principle and the universal quantum computer. *Proceedings of the Royal Society of London A, 400,* 97–117.

Elder, M. (2013). A short introduction to self-similar groups. *Asia Pacific Mathematics Newsletter,3*(1).

Fiore, M., & Leinster, T. (2010). An abstract characterization of Thompson's group F. *Semigroup Forum, 80*(2), 325–340.

Girard, J.-Y. (1988a). Geometry of interaction 1. In *Proceedings Logic Colloquium '88* (pp. 221–260). North-Holland.

Girard, J.-Y. (1988b). Geometry of interaction 2: Deadlock-free algorithms. Lecture notes in computer science. *Conference on computer logic* (Vol. 417, pp. 76–93). Berlin: Springer.

Girard, J.-Y. (1995). Geometry of interaction 3: Accommodating the additives. In *Advances in linear logic* (pp. 329–389). Cambridge: Cambridge University Press.

Haghverdi, E. (2000). *A categorical approach to linear logic, geometry of proofs and full completeness*. PhD thesis, University of Ottawa.

Haghverdi, E., & Scott, P. J. (2005). Towards a typed geometry of interaction. In *Computer science logic 2005*. Lecture notes in computer science (Vol. 3634). Berlin: Springer.

Hines, P. (1997). *The algebra of self-similarity and its applications*. PhD thesis, University of Wales, Bangor.

Hines, P. (1999). The categorical theory of self-similarity. *Theory and Applications of Categories, 6,* 33–46.

Hines, P. (2003). A categorical framework for finite state machines. *Mathematical Strucures in Computer Science, 13,* 451–480.

Hines, P. (2008). Machine semantics. *Theoretical Computer Science, 409,* 1–23.

Hines, P. (2013). Identities in modular arithmetic from untyped categorical coherence. In *Proceedings of the Reversible Computation 2013*, Springer LNCS.

Hines, P. (2016). Coherence and strictification for self-similarity. *Journal of Homotopy & Related Structures, 11*, 847–867.

Hines, P. (2014). Classical structures based on unitaries. Springer LNCS. In B. Coecke & P. Panangaden (Eds.), *Categories and types in logic, language, and physics* (Vol. 8222, pp. 188–210). Berlin: Springer.

Høyer, P., & Špalek, R. (2005). Quantum fan-out is powerful. *Theory of Computing, 1,* 81–103.

Hollings, C. (2014). *Mathematics across the iron curtain*. American Mathematical Society.

Hyland, J. M. E. (2014). Towards a notion of lambda monoid. *E.N.T.C.S., 303,* 59–77.

Hyland, J. M. E. (2017). Classical lambda calculus in modern dress. *Mathematical Structures in Computer Science, 27*(5), 762–781.

Hyland, J. M. E., & Ong, C.-H.L. (2000). On full abstraction for pcf: I, ii, and iii. *Information and Computation, 163*(2), 285–408.

Joyal, A., & Kock, J. (2013). Coherence for weak units. *Documenta Mathematica, 18,* 71–110.

Joyal, A., & Street, R. (1993a). The geometry of tensor calculus (i). *Advances in Mathematics, 102,* 20–78.

Joyal, A., & Street, R. (1993b). The geometry of tensor calculus (ii). *Manuscript*.

Joyal, A., Street, R., & Verity, D. (1996). Traced monoidal categories. *Mathematical Proceedings of the Cambridge Philosophical Society, 119*(3), 447–468.

Kelly, M., & Laplaza, M. (1980). Coherence for compact closed categories. *Journal of Pure and Applied Algebra, 19,* 193–213.

Kock, J. (2008). Elementary remarks on units in monoidal categories. *Mathematical Proceedings of the Cambridge Philosophical Society, 144,* 53–76.

Lambek, J. (1999). Type grammar revisited. In A. Lecomte, F. Lamarche, & G. Perrier (Eds.), *Logical aspects of computational linguistics* (pp. 1–27). Berlin: Springer.

Lambek, J., & Scott, P. (1986). *Introduction to higher order categorical logic*. Cambridge: Cambridge University Press.

Laplaza, M. (1977). Coherence in non-monoidal closed categories. *Transations of the American Mathematical Society, 230,* 293–311.

Lauda, A. (2005). Frobenius algebras and ambidextrous adjunctions. *Theory and Applications of Categories, 16,* 02.

Lawson, M. (2020). The polycyclic inverse monoids and the thompson groups revisited. arxiv.org/abs/2006.15338.

Lawson, M. V. (1998). *Inverse semigroups: The theory of partial symmetries*. Singapore: World Scientific.

Lawson, M. V. (2007). A class of subgroups of thompson's group v. *Semigroup Forum, 75,* 241–252.

Lyapin, E. (1953). Canonical form of elements of an associative system given by defining relations. *Leningradskii Gosudarstvennyii Pedagogicheskii Institut Uchenye Zapiski, 89,* 45–54.

MacLane, S. (1998). *Categories for the working mathematician* (2nd ed.). New York: Springer.

McKenzie, R., & Thompson, R. J. (1973). *An elementary construction of unsolvable word problem in group theory* (pp. 457–478). North Holland.

Myers, J. (1997). Can a universal quantum computer be fully quantum? *Physical Review Letters, 78,* 1823–1824.

Nivat, M., & Perrot, J. (1970). Une généralisation du monöide bicyclique. *Comptes Rendus de l'Académie des Sciences de Paris, 27,* 824–827.

Rivano, N. S. (1972). Catégories tannakiennes. *Bulletin de la Société Mathématique de France, 100,* 417–430.

Scott, D. S. (1980). Relating theories of the lambda calculus.

Selinger, P. (2007). Dagger compact closed categories and completely positive maps: (Extended abstract). In *Proceedings of the 3rd International Workshop on Quantum Programming Languages (QPL 2005)*. Electronic Notes in Theoretical Computer Science (Vol. 170, pp. 139–163).

Vicary, J. (2011). Categorical formulation of finite-dimensional c*-algebras. *Electronic Notes in Theoretical Computer Science,270*(1), 129–145. Proceeding of the Quantum Physics and Logic 2008.

Wootters, W. K., & Zurek, W. H. (1982). A single quantum cannot be cloned. *Nature, 299,* 802–803.

Yanofsky, N. (2013). *The outer limits of reason*. Cambridge: MIT Press.

Chapter 26
Semantics for a Lambda Calculus for String Diagrams

Bert Lindenhovius, Michael Mislove, and Vladimir Zamdzhiev

Abstract Linear/non-linear (LNL) models, as described by Benton, soundly model a LNL term calculus and LNL logic closely related to intuitionistic linear logic. Every such model induces a canonical enrichment that we show soundly models a LNL lambda calculus for string diagrams, introduced by Rios and Selinger (with primary application in quantum computing). Our abstract treatment of this language leads to simpler concrete models compared to those presented so far. We also extend the language with general recursion and prove soundness. Finally, we present an adequacy result for the diagram-free fragment of the language which corresponds to a modified version of Benton and Wadler's adjoint calculus with recursion. In keeping with the purpose of the special issue, we also describe the influence of Samson Abramsky's research on these results, and on the overall project of which this is a part.

26.1 Dedication

We are pleased to contribute to this volume honoring SAMSON ABRAMSKY's many contributions to logic and its applications in theoretical computer science. This contribution is an extended version of the paper Lindenhovius et al. (2018), which concerns the semantics of high-level functional quantum programming languages. This research is part of a project on the same general theme. The project itself would

B. Lindenhovius
Institute for Mathematical Methods in Medicine and Data Based Modeling, Johannes Kepler University Linz, Linz, Austria
e-mail: alindenh@tulane.edu

M. Mislove
Department of Computer Science, Tulane University, New Orleans, LA, USA
e-mail: mislove@tulane.edu

V. Zamdzhiev (✉)
Université de Lorraine, CNRS, Inria, LORIA, Lorraine, France
e-mail: vladimir.zamdzhiev@loria.fr

A. Palmigiano and M. Sadrzadeh (eds.), *Samson Abramsky on Logic and Structure in Computer Science and Beyond*, Outstanding Contributions to Logic 25,
https://doi.org/10.1007/978-3-031-24117-8_26

not exist if it weren't for Samson's support, guidance and participation. And, as we document in the final section of this contribution, Samson's research has had a direct impact on this work, and also points the way for further results along the general line we are pursuing. We are now in the fifth year of the project, and Samson has been a continual inspiration, both directly by his participation in the Tulane team, and indirectly by including us in a number of related activities he helped organize that exposed us to a variety of topics, all of which have been interesting and many of which have a direct connection to our research. We're happy to acknowledge his contributions, and we hope he enjoys reading about the research we report here.

26.1.1 Some Personal Comments

The first author first learned about the work of Samson via Abramsky & Coecke (2009). This work captures many features of quantum physics in a categorical framework, which highly impressed the first author, who soon discovered that this was just the tip of Samson's work. There are not many scientists as versatile as Samson, given the number of different subjects he made significant contributions to, and his courage to explore new areas is very inspiring. This author was of course a bit nervous when he finally met Samson in real life at the Simons Institute in Berkeley, but it turned out that Samson is also very friendly, and we had several enjoyable conversations about mathematics and computer science, but also about the life beyond science. The author hopes to learn much more from Samson in the future. Thank you for being such an inspiration!

The second author first met Samson in 1984 while spending a sabbatical at Oxford, when Samson was a member of the semantics group at Imperial College. Our paths have crossed often since then, at scientific meetings and during visits at each other's institutions. While we have never co-authored a paper together, we did edit a volume on Information Flow (Abramsky & Mislove, 2012), which grew out of Samson's Clifford Lectures at Tulane in 1998, and the follow-on meetings that we jointly helped organize. This author is reminded of the many important discussions that took place during such meetings, often occurring on the inevitable "conference outing". The first such chat this author can recall was during a walk in the dunes at Asilomar during the 1989 LiCS meeting, a meeting highlighted by Dana's Sunday morning sermon. But by far the most vivid memory is the night canopy tour on the outing following the Costa Rica Informatic Phenomena meeting, during which Samson let out a totally uncharacteristic yell as he threw himself into the dark clinging to a zip line. We also learned a lot about sloths on that trip. Those talks always were stimulating and informative, and often a lot of fun. Thank you, Samson!

The third author first met Samson at his Ph.D. interview in Oxford in 2012. The interview went very well and he then became a Ph.D. student of Samson for the next four years. Over the course of the Ph.D., Samson provided very valuable advice and support, which of course, the author appreciates very much. After the end of this author's Ph.D., we have met many times in numerous other scientific events which

was very enjoyable as always. Samson's many contributions to many different parts of logic are clearly very impressive and Samson himself has been a big inspirational figure throughout this author's career. Over the years, we have also spent a lot of time in social events and outings in many different places and in this way the third author got to know Samson outside of professional settings which was always very interesting and very fun. Thank you, Samson!

26.2 Introduction

In recent years string diagrams have found applications across a range of areas in computer science and related fields: in concurrency theory, where they are used to model Petri nets (Meseguer & Montanari, 1988); in systems theory, where they are used in a calculus of signal flow diagrams (Bonchi et al., 2015); and in quantum computing (Hadzihasanovic, 2015; Coecke & Duncan, 2008) where they represent quantum circuits and have been used to completely axiomatize the Clifford+T segment of quantum mechanics (Perdrix et al., 2017).

But as the size of a system grows, constructing string diagram representations by hand quickly becomes intractable, and more advanced tools are needed to accurately represent and reason about the associated diagrams. In fact, just generating large diagrams is a difficult problem. One area where this has been addressed is in the development of circuit description languages. For example, Verilog (2008) and VHDL (1997) are popular hardware description languages that are used to generate very large digital circuits. More recently, the PNBml language (Sobocinski & Stephens, 2014) was developed to generate Petri nets, and Quipper (2013) and QWIRE (2017) are quantum programming languages (among others) that are used to generate (and execute) quantum circuits.

In this paper we pursue a more abstract approach. We consider a lambda calculus for string diagrams whose primary purpose is to generate complicated diagrams from simpler components. However, we do not fix a particular application domain. Our development only assumes that the string diagrams we are working with enjoy a symmetric monoidal structure. Our goal is to help lay a foundation for programming languages that generate string diagrams, and that support the addition of extensions for specific application domains along with the necessary language features.

More generally, we believe the use of formal methods could aid us in obtaining a better conceptual understanding of how to design languages that can be used to construct and analyze large and complicated (families) of string diagrams.

Our Results. We study several calculi in this paper, beginning with the *combined LNL* (CLNL) calculus, which is the diagram-free fragment of our main language. The CLNL calculus, described in Sect. 26.3, can be seen as a modified version of Benton's LNL calculus, first defined in Benton (1995). The crucial difference is that in CLNL we allow the use of mixed contexts, so there is only one type of judgement. This reduces the number of typing rules, and allows us to extend the language to support the generation of string diagrams. We also present a categorical model for our language, which is given by an LNL model with finite coproducts, and prove its soundness.

Next, in Sect. 26.4, we describe our main language of interest, the *enriched CLNL* calculus, which we denote ECLNL. The ECLNL calculus adopts the syntax and operational semantics of Proto-Quipper-M, a circuit description language introduced by Rios & Selinger (2017), but we develop our own categorical model. Ours is the first *abstract* categorical model for the language, which is again given by an LNL model, but endowed with an additional *enrichment* structure. The enrichment is the reason we chose to rename the language. By design, ECLNL is an extension of the CLNL calculus that adds language features for manipulating string diagrams. We show that our abstract model satisfies the soundness and constructivity requirements (see Rios & Selinger (2017), Remark 4.1) of Rios and Selinger's original model. As special instances of our abstract model, we recover the original model of Rios and Selinger, and we also present a simpler concrete model, as well as one that is order enriched.

In Sect. 26.5 we resolve the open problem posed by Rios and Selinger of extending the language with general recursion. We show that all the relevant language properties are preserved, and then we prove soundness for both the CLNL and ECLNL calculi with recursion, after first extending our abstract models with some additional structure. We then present concrete models for the ECLNL calculus that support recursion and also support generating string diagrams from *any* symmetric monoidal category. We conclude the section with a concrete model for the CLNL calculus extended with recursion that we also prove is computationally adequate at non-linear types.

In Sect. 26.6, we conclude the paper and discuss further possible developments, such as adding inductive and recursive types, as well as a treatment of dependent types.

Related Work. Categorical models are fundamental for our results, and the ones we present rely on the LNL models first described by Benton in Benton (1995). Our work also is inspired by the language Proto-Quipper-M Rios & Selinger (2017) by Rios and Selinger, the latest of the circuit description languages Selinger and his group have been developing. Our ECLNL calculus has the same syntax and operational semantics as Proto-Quipper-M, but there are significant differences in the denotational models. Rios and Selinger start with a symmetric monoidal category $\mathbf{M}$, then they consider a fully faithful strong symmetric monoidal embedding of $\mathbf{M}$ into another category $\overline{\mathbf{M}}$ that has some suitable categorical structure (e.g. $\overline{\mathbf{M}} := [\mathbf{M}^{\mathrm{op}}, \mathbf{Set}]$), so that the category $\mathbf{Fam}(\overline{\mathbf{M}})$ is symmetric monoidal closed and contains $\mathbf{M}$. Their model is then given by the symmetric monoidal adjunction between $\mathbf{Set}$ and $\mathbf{Fam}(\overline{\mathbf{M}})$, which allows them to distinguish "parameter" (non-linear) terms and "state" (linear) terms. They show their language is type safe, their semantics is sound, and they remark that it also is computationally adequate at observable types (there is no recursion, so all programs terminate). The semantics for our ECLNL calculus enjoys the same properties, but we present both an abstract model and a simpler concrete model that doesn't involve a **Fam**$(-)$ construction. Moreover, we also describe an extension with recursion, based on ideas by Benton & Wadler (1996), and present an adequacy result for the diagram-free fragment of the language.

QWIRE (2017) also is a language for reasoning about quantum circuits. QWIRE is really two languages, a non-linear host language and a quantum circuits language. QWIRE led Rennela and Staton to consider a more general language Ewire (Rennela & Staton, 2017a, b), which can be used to describe circuits that are not necessarily quantum. Ewire supports dynamic lifting, and they prove a soundness result assuming the reduction system for the non-linear language is normalizing. They also discuss extending Ewire with conditional branching and inductive types over the $\otimes$- and $\oplus$-connectives (but not $\multimap$). However, these extensions require imposing additional structure on the diagrams, such as the existence of coproducts and fold/unfold gates. In our approach, we assume only that the diagrams enjoy a symmetric monoidal structure. In addition, our language also supports general recursion, whereas Ewire does not. An important similarity is that Ewire also makes use of enriched category theory to describe the denotational model.

Aside from Ewire and Proto-Quipper-M, the other languages we mentioned cannot generate arbitrary string diagrams, and some of them do not have a formal denotational semantics.

26.3 An Alternative LNL Calculus

LNL models were introduced by Benton (1995) as a means to soundly model an interesting LNL calculus together with a corresponding logic. The goal was to understand the relationship between intuitionistic logic and intuitionistic linear logic. In this section, we show that LNL models also soundly model a variant of the LNL calculus where, instead of having two distinct typing judgements (linear and non-linear), there is a single type of judgement whose context is allowed to be mixed. A similar idea was briefly discussed by Benton in his original paper Benton (1995). The syntax and operational semantics for this language are derived as a special case of the language of Rios & Selinger (2017). We denote the resulting language by CLNL, which we call the "Combined LNL" calculus.

As with the other calculi we consider, we begin our discussion by first describing a categorical model for CLNL. This makes the presentation of the language easier to follow. A categorical model of the CLNL calculus is given by an LNL model with finite coproducts, as the next definition shows.

Definition 1 (Benton (1995)) A *model of the CLNL calculus* (CLNL model) is given by the following data: a cartesian closed category (CCC) with finite coproducts $(\mathbf{V}, \times, \to, 1, \amalg, \emptyset)$; a symmetric monoidal closed category (SMCC) with finite coproducts $(\mathbf{C}, \otimes, \multimap, I, +, 0)$; and a symmetric monoidal adjunction:

$$\begin{tikzcd} \mathbf{V} \arrow[r, bend left, "F"] \arrow[r, phantom, "\perp"] & \mathbf{C} \arrow[l, bend left, "G"] \end{tikzcd}$$

We also adopt the following notation:

- The comonad-endofunctor is $! := F \circ G$.
- The unit of the adjunction $F \dashv G$ is $\eta : \mathrm{Id} \overset{\cdot}{\longrightarrow} G \circ F$.
- The counit of the adjunction $F \dashv G$ is $\epsilon : ! \overset{\cdot}{\longrightarrow} \mathrm{Id}$.

Throughout the remainder of this section, we consider an arbitrary, but fixed, CLNL model. The CLNL calculus, which we introduce next, is interpreted in the category **C**.

The syntax of the CLNL calculus is presented in Fig. 26.1. It is exactly the diagram-free fragment of the ECLNL calculus, and because of space reasons, we only show the typing rules for ECLNL. However, the typing rules of the CLNL calculus can be easily derived from those for ECLNL by ignoring the Q label contexts (see the (pair) rule example below). Of course, ECLNL has some additional terms not in CLNL, so the corresponding typing rules should be ignored as well.

Observe that the non-linear types are a subset of the types of our language. Note also that there is no grammar which defines linear types. We say that a type that is not non-linear is *linear*. This definition is strictly speaking not necessary, but it helps to illustrate some concepts. In particular, any type $A \multimap B$ is therefore considered to be linear, even if A and B are non-linear. The interpretation of a type A is an object $[\![A]\!]$ of **C**, defined by induction in the usual way (Fig. 26.2).

Recall that in an LNL model with coproducts, we have:

$$I \cong F(1); \qquad 0 \cong F(\emptyset);$$

$$F(X) \otimes F(Y) \cong F(X \times Y); \qquad F(X) + F(Y) \cong F(X \amalg Y)$$

because F is strong (symmetric) monoidal and also a left adjoint. Based on these observations, we can define a non-linear interpretation $(\!|P|\!) \in \mathbf{V}$ of non-linear types P by induction in the following way:

$$\begin{aligned} (\!|I|\!) &:= 1; \\ (\!|0|\!) &:= \emptyset; \\ (\!|P \otimes Q|\!) &:= (\!|P|\!) \times (\!|Q|\!); \\ (\!|P + Q|\!) &:= (\!|P|\!) \amalg (\!|Q|\!); \\ (\!|!A|\!) &:= G[\![A]\!]. \end{aligned}$$

The CLNL Calculus

Variables	x, y, z		
Types	A, B, C	$::=$	$0 \mid A+B \mid I \mid A \otimes B \mid A \multimap B \mid !A$
Non-linear types	P, R	$::=$	$0 \mid P+R \mid I \mid P \otimes R \mid !A$
Variable contexts	Γ	$::=$	$x_1 : A_1, x_2 : A_2, \ldots, x_n : A_n$
Non-linear variable contexts	Φ	$::=$	$x_1 : P_1, x_2 : P_2, \ldots, x_n : P_n$
Terms	m, n, p	$::=$	$x \mid c \mid \text{let } x = m \text{ in } n \mid \Box_C m \mid \text{left}_{A,B} m \mid \text{right}_{A,B} m \mid$ $\text{case } m \text{ of } \{\text{left } x \to n \mid \text{right } y \to p\} \mid * \mid m;n \mid \langle m, n \rangle \mid$ $\text{let } \langle x, y \rangle = m \text{ in } n \mid \lambda x^A.m \mid mn \mid \text{lift } m \mid \text{force } m$
Values	v, w	$::=$	$x \mid c \mid \text{left}_{A,B} v \mid \text{right}_{A,B} v \mid * \mid \langle v, w \rangle \mid \lambda x^A.m \mid \text{lift } m$
Term Judgements	$\Gamma \vdash m : A$		(typing rules below - ignore Q contexts)

The ECLNL Calculus

Extend the CLNL syntax with:

Labels	ℓ, k		
Labelled string diagrams	S, D		
Types	A, B, C	$::=$	$\cdots \mid \alpha \mid \text{Diag}(T, U)$
Non-linear types	P, R	$::=$	$\cdots \mid \text{Diag}(T, U)$
M-types	T, U	$::=$	$\alpha \mid I \mid T \otimes U$
Label contexts	Q	$::=$	$\ell_1 : \alpha_1, \ell_2 : \alpha_2, \ldots, \ell_n : \alpha_n$
Terms	m, n, p	$::=$	$\cdots \mid \ell \mid \text{box}_T m \mid \text{apply}(m, n) \mid (\vec{\ell}, S, \vec{\ell}')$
Label tuples	$\vec{\ell}, \vec{k}$	$::=$	$\ell \mid * \mid \langle \vec{\ell}, \vec{k} \rangle$
Values	v, w	$::=$	$\cdots \mid \ell \mid (\vec{\ell}, S, \vec{\ell}')$
Configurations	(S, m)		
Term Judgements	$\Gamma; Q \vdash m : A$		
Configuration Judgements	$Q \vdash (S, m) : A; Q'$		(see Definition 4.6)

The Typing Rules

$$\frac{}{\Phi, x : A; \emptyset \vdash x : A}\ (\text{var}) \qquad \frac{}{\Phi; \ell : \alpha \vdash \ell : \alpha}\ (\text{label}) \qquad \frac{}{\Phi; \emptyset \vdash c : A_c}\ (\text{const}) \qquad \frac{}{\Phi; \emptyset \vdash * : I}\ (*)$$

$$\frac{\Gamma; Q \vdash m : 0}{\Gamma; Q \vdash \Box_C m : C}\ (\text{initial}) \qquad \frac{\Gamma; Q \vdash m : A}{\Gamma; Q \vdash \text{left}_{A,B} m : A+B}\ (\text{left}) \qquad \frac{\Gamma; Q \vdash m : B}{\Gamma; Q \vdash \text{right}_{A,B} m : A+B}\ (\text{right})$$

$$\frac{\Phi, \Gamma_1; Q_1 \vdash m : A+B \qquad \Phi, \Gamma_2, x : A; Q_2 \vdash n : C \qquad \Phi, \Gamma_2, y : B; Q_2 \vdash p : C}{\Phi, \Gamma_1, \Gamma_2; Q_1, Q_2 \vdash \text{case } m \text{ of } \{\text{left } x \to n \mid \text{right } y \to p\} : C}\ (\text{case})$$

$$\frac{\Phi, \Gamma_1; Q_1 \vdash m : I \qquad \Phi, \Gamma_2; Q_2 \vdash n : C}{\Phi, \Gamma_1, \Gamma_2; Q_1, Q_2 \vdash m;n : C}\ (\text{seq}) \qquad \frac{\Phi, \Gamma_1; Q_1 \vdash m : A \qquad \Phi, \Gamma_2, x : A; Q_2 \vdash n : B}{\Phi, \Gamma_1, \Gamma_2; Q_1, Q_2 \vdash \text{let } x = m \text{ in } n : B}\ (\text{let})$$

$$\frac{\Phi, \Gamma_1; Q_1 \vdash m : A \qquad \Phi, \Gamma_2; Q_2 \vdash n : B}{\Phi, \Gamma_1, \Gamma_2; Q_1, Q_2 \vdash \langle m, n \rangle : A \otimes B}\ (\text{pair}) \qquad \frac{\Phi, \Gamma_1; Q_1 \vdash m : A \otimes B \qquad \Phi, \Gamma_2, x : A, y : B; Q_2 \vdash n : C}{\Phi, \Gamma_1, \Gamma_2; Q_1, Q_2 \vdash \text{let } \langle x, y \rangle = m \text{ in } n : C}\ (\text{let-pair})$$

$$\frac{\Gamma, x : A; Q \vdash m : B}{\Gamma; Q \vdash \lambda x^A.m : A \multimap B}\ (\text{abs}) \qquad \frac{\Phi, \Gamma_1; Q_1 \vdash m : A \multimap B \qquad \Phi, \Gamma_2; Q_2 \vdash n : A}{\Phi, \Gamma_1, \Gamma_2; Q_1, Q_2 \vdash mn : B}\ (\text{app}) \qquad \frac{\Phi; \emptyset \vdash m : A}{\Phi; \emptyset \vdash \text{lift } m : !A}\ (\text{lift})$$

$$\frac{\Gamma; Q \vdash m : !A}{\Gamma; Q \vdash \text{force } m : A}\ (\text{force}) \qquad \frac{\Gamma; Q \vdash m : !(T \multimap U)}{\Gamma; Q \vdash \text{box}_T m : \text{Diag}(T, U)}\ (\text{box})$$

$$\frac{\Phi, \Gamma_1; Q_1 \vdash m : \text{Diag}(T, U) \quad \Phi, \Gamma_2; Q_2 \vdash n : T}{\Phi, \Gamma_1, \Gamma_2; Q_1, Q_2 \vdash \text{apply}(m, n) : U}\ (\text{apply}) \qquad \frac{\emptyset; Q \vdash \vec{\ell} : T \quad \emptyset; Q' \vdash \vec{\ell}' : U \quad S \in \mathbf{M}_L(Q, Q')}{\Phi; \emptyset \vdash (\vec{\ell}, S, \vec{\ell}') : \text{Diag}(T, U)}\ (\text{diag})$$

Fig. 26.1 Syntax of the CLNL and ECLNL calculi

$$[\![\alpha]\!] := E([\![\alpha]\!]_{\mathbf{M}}); \qquad [\![0]\!] := 0; \qquad [\![A+B]\!] := [\![A]\!] + [\![B]\!] \qquad [\![I]\!] := I \qquad [\![A \otimes B]\!] := [\![A]\!] \otimes [\![B]\!]$$

$$[\![A \multimap B]\!] := [\![A]\!] \multimap [\![B]\!] \qquad [\![!A]\!] := ![\![A]\!] \qquad [\![\mathrm{Diag}(T,U)]\!] := F(\mathscr{C}([\![T]\!],[\![U]\!]))$$

$$[\![\Phi, x : A; \emptyset \vdash x : A]\!] := [\![\Phi]\!] \otimes [\![A]\!] \xrightarrow{\diamond \otimes \mathrm{id}} I \otimes [\![A]\!] \xrightarrow{\cong} [\![A]\!]$$

$$[\![\Phi; \ell : \alpha \vdash \ell : \alpha]\!] := [\![\Phi]\!] \otimes [\![\alpha]\!] \xrightarrow{\diamond \otimes \mathrm{id}} I \otimes [\![\alpha]\!] \xrightarrow{\cong} [\![\alpha]\!]$$

$$[\![\Phi; \emptyset \vdash c : A_c]\!] := [\![\Phi]\!] \xrightarrow{\diamond} I \xrightarrow{[\![c]\!]} [\![A_c]\!]$$

$$\begin{aligned}
&[\![\Phi, \Gamma_1, \Gamma_2; Q_1, Q_2 \vdash \mathrm{let}\ x = m\ \mathrm{in}\ n : B]\!] := [\![\Phi]\!] \otimes [\![\Gamma_1]\!] \otimes [\![\Gamma_2]\!] \otimes [\![Q_1]\!] \otimes [\![Q_2]\!] \\
&\quad \xrightarrow{\Delta \otimes \mathrm{id}} [\![\Phi]\!] \otimes [\![\Phi]\!] \otimes [\![\Gamma_1]\!] \otimes [\![\Gamma_2]\!] \otimes [\![Q_1]\!] \otimes [\![Q_2]\!] \xrightarrow{\cong} [\![\Phi]\!] \otimes [\![\Gamma_2]\!] \otimes [\![\Phi]\!] \otimes [\![\Gamma_1]\!] \otimes [\![Q_1]\!] \otimes [\![Q_2]\!] \\
&\quad \xrightarrow{\mathrm{id} \otimes [\![m]\!] \otimes \mathrm{id}} [\![\Phi]\!] \otimes [\![\Gamma_2]\!] \otimes [\![A]\!] \otimes [\![Q_2]\!] \xrightarrow{[\![n]\!]} [\![B]\!]
\end{aligned}$$

$$[\![\Gamma; Q \vdash \Box_C m : C]\!] := [\![\Gamma]\!] \otimes [\![Q]\!] \xrightarrow{[\![m]\!]} 0 \xrightarrow{!} C$$

$$[\![\Gamma; Q \vdash \mathrm{left}_{A,B} m : A+B]\!] := [\![\Gamma]\!] \otimes [\![Q]\!] \xrightarrow{[\![m]\!]} [\![A]\!] \xrightarrow{\mathrm{left}} [\![A]\!] + [\![B]\!] \xrightarrow{=} [\![A+B]\!]$$

$$[\![\Gamma; Q \vdash \mathrm{right}_{A,B} m : A+B]\!] := [\![\Gamma]\!] \otimes [\![Q]\!] \xrightarrow{[\![m]\!]} [\![B]\!] \xrightarrow{\mathrm{right}} [\![A]\!] + [\![B]\!] \xrightarrow{=} [\![A+B]\!]$$

$$[\![\Phi; \emptyset \vdash * : I]\!] := [\![\Phi]\!] \xrightarrow{\diamond} I$$

$$\begin{aligned}
&[\![\Phi, \Gamma_1, \Gamma_2; Q_1, Q_2 \vdash \mathrm{case}\ m\ \mathrm{of}\ \{\mathrm{left}\ x \to n \mid \mathrm{right}\ y \to p\} : C]\!] := [\![\Phi]\!] \otimes [\![\Gamma_1]\!] \otimes [\![\Gamma_2]\!] \otimes [\![Q_1]\!] \otimes [\![Q_2]\!] \\
&\quad \xrightarrow{\Delta \otimes \mathrm{id}} [\![\Phi]\!] \otimes [\![\Phi]\!] \otimes [\![\Gamma_1]\!] \otimes [\![\Gamma_2]\!] \otimes [\![Q_1]\!] \otimes [\![Q_2]\!] \xrightarrow{\cong} [\![\Phi]\!] \otimes [\![\Gamma_2]\!] \otimes [\![\Phi]\!] \otimes [\![\Gamma_1]\!] \otimes [\![Q_1]\!] \otimes [\![Q_2]\!] \\
&\quad \xrightarrow{\mathrm{id} \otimes [\![m]\!] \otimes \mathrm{id}} [\![\Phi]\!] \otimes [\![\Gamma_2]\!] \otimes [\![A+B]\!] \otimes [\![Q_2]\!] \xrightarrow{\cong} ([\![\Phi]\!] \otimes [\![\Gamma_2]\!] \otimes [\![A]\!] \otimes [\![Q_2]\!]) + ([\![\Phi]\!] \otimes [\![\Gamma_2]\!] \otimes [\![B]\!] \otimes [\![Q_2]\!]) \\
&\quad \xrightarrow{[[\![n]\!],[\![p]\!]]} [\![C]\!]
\end{aligned}$$

$$\begin{aligned}
&[\![\Phi, \Gamma_1, \Gamma_2; Q_1, Q_2 \vdash m; n : C]\!] := [\![\Phi]\!] \otimes [\![\Gamma_1]\!] \otimes [\![\Gamma_2]\!] \otimes [\![Q_1]\!] \otimes [\![Q_2]\!] \\
&\quad \xrightarrow{\Delta \otimes \mathrm{id}} [\![\Phi]\!] \otimes [\![\Phi]\!] \otimes [\![\Gamma_1]\!] \otimes [\![\Gamma_2]\!] \otimes [\![Q_1]\!] \otimes [\![Q_2]\!] \xrightarrow{\cong} [\![\Phi]\!] \otimes [\![\Gamma_1]\!] \otimes [\![Q_1]\!] \otimes [\![\Phi]\!] \otimes [\![\Gamma_2]\!] \otimes [\![Q_2]\!] \\
&\quad \xrightarrow{[\![m]\!] \otimes \mathrm{id}} I \otimes [\![\Phi]\!] \otimes [\![\Gamma_2]\!] \otimes [\![Q_2]\!] \\
&\quad \xrightarrow{\cong} [\![\Phi]\!] \otimes [\![\Gamma_2]\!] \otimes [\![Q_2]\!] \xrightarrow{[\![n]\!]} [\![C]\!]
\end{aligned}$$

$$\begin{aligned}
&[\![\Phi, \Gamma_1, \Gamma_2; Q_1, Q_2 \vdash \langle m, n \rangle : A \otimes B]\!] := [\![\Phi]\!] \otimes [\![\Gamma_1]\!] \otimes [\![\Gamma_2]\!] \otimes [\![Q_1]\!] \otimes [\![Q_2]\!] \\
&\quad \xrightarrow{\Delta \otimes \mathrm{id}} [\![\Phi]\!] \otimes [\![\Phi]\!] \otimes [\![\Gamma_1]\!] \otimes [\![\Gamma_2]\!] \otimes [\![Q_1]\!] \otimes [\![Q_2]\!] \xrightarrow{\cong} [\![\Phi]\!] \otimes [\![\Gamma_1]\!] \otimes [\![Q_1]\!] \otimes [\![\Phi]\!] \otimes [\![\Gamma_2]\!] \otimes [\![Q_2]\!] \\
&\quad \xrightarrow{[\![m]\!] \otimes [\![n]\!]} [\![A]\!] \otimes [\![B]\!]
\end{aligned}$$

$$\begin{aligned}
&[\![\Phi, \Gamma_1, \Gamma_2; Q_1, Q_2 \vdash \mathrm{let}\ \langle x, y \rangle = m\ \mathrm{in}\ n : C]\!] := [\![\Phi]\!] \otimes [\![\Gamma_1]\!] \otimes [\![\Gamma_2]\!] \otimes [\![Q_1]\!] \otimes [\![Q_2]\!] \\
&\quad \xrightarrow{\Delta \otimes \mathrm{id}} [\![\Phi]\!] \otimes [\![\Phi]\!] \otimes [\![\Gamma_1]\!] \otimes [\![\Gamma_2]\!] \otimes [\![Q_1]\!] \otimes [\![Q_2]\!] \xrightarrow{\cong} [\![\Phi]\!] \otimes [\![\Gamma_1]\!] \otimes [\![Q_1]\!] \otimes [\![\Phi]\!] \otimes [\![\Gamma_2]\!] \otimes [\![Q_2]\!] \\
&\quad \xrightarrow{[\![m]\!] \otimes \mathrm{id}} [\![A \otimes B]\!] \otimes [\![\Phi]\!] \otimes [\![\Gamma_2]\!] \otimes [\![Q_2]\!] \xrightarrow{\cong} [\![\Phi]\!] \otimes [\![\Gamma_2]\!] \otimes [\![A]\!] \otimes [\![B]\!] \otimes [\![Q_2]\!] \xrightarrow{[\![n]\!]} [\![C]\!]
\end{aligned}$$

$$[\![\Gamma; Q \vdash \lambda x^A . m : A \multimap B]\!] := [\![\Gamma]\!] \otimes [\![Q]\!] \xrightarrow{\mathbf{curry}([\![m]\!] \circ \cong)} [\![A]\!] \multimap [\![B]\!]$$

$$\begin{aligned}
&[\![\Phi, \Gamma_1, \Gamma_2; Q_1, Q_2 \vdash mn : B]\!] := [\![\Phi]\!] \otimes [\![\Gamma_1]\!] \otimes [\![\Gamma_2]\!] \otimes [\![Q_1]\!] \otimes [\![Q_2]\!] \\
&\quad \xrightarrow{\Delta \otimes \mathrm{id}} [\![\Phi]\!] \otimes [\![\Phi]\!] \otimes [\![\Gamma_1]\!] \otimes [\![\Gamma_2]\!] \otimes [\![Q_1]\!] \otimes [\![Q_2]\!] \xrightarrow{\cong} [\![\Phi]\!] \otimes [\![\Gamma_1]\!] \otimes [\![Q_1]\!] \otimes [\![\Phi]\!] \otimes [\![\Gamma_2]\!] \otimes [\![Q_2]\!] \\
&\quad \xrightarrow{[\![m]\!] \otimes [\![n]\!]} [\![A \multimap B]\!] \otimes [\![A]\!] \xrightarrow{\mathrm{ev}} [\![B]\!]
\end{aligned}$$

$$[\![\Phi; \emptyset \vdash \mathrm{lift}\ m : !A]\!] := [\![\Phi]\!] \xrightarrow{\mathbf{lift}} ![\![\Phi]\!] \xrightarrow{![\![m]\!]} ![\![A]\!]$$

$$[\![\Gamma; Q \vdash \mathrm{force}\ m : A]\!] := [\![\Gamma]\!] \otimes [\![Q]\!] \xrightarrow{[\![m]\!]} ![\![A]\!] \xrightarrow{\varepsilon} [\![A]\!]$$

$$[\![\Gamma; Q \vdash \mathrm{box}_T m : \mathrm{Diag}(T,U)]\!] := [\![\Gamma]\!] \otimes [\![Q]\!] \xrightarrow{[\![m]\!]} !([\![T]\!] \multimap [\![U]\!]) \xrightarrow{\cong} [\![\mathrm{Diag}(T,U)]\!]$$

$$\begin{aligned}
&[\![\Phi, \Gamma_1, \Gamma_2; Q_1, Q_2 \vdash \mathrm{apply}(m,n) : U]\!] := [\![\Phi]\!] \otimes [\![\Gamma_1]\!] \otimes [\![\Gamma_2]\!] \otimes [\![Q_1]\!] \otimes [\![Q_2]\!] \\
&\quad \xrightarrow{\Delta \otimes \mathrm{id}} [\![\Phi]\!] \otimes [\![\Phi]\!] \otimes [\![\Gamma_1]\!] \otimes [\![\Gamma_2]\!] \otimes [\![Q_1]\!] \otimes [\![Q_2]\!] \xrightarrow{\cong} [\![\Phi]\!] \otimes [\![\Gamma_1]\!] \otimes [\![Q_1]\!] \otimes [\![\Phi]\!] \otimes [\![\Gamma_2]\!] \otimes [\![Q_2]\!] \\
&\quad \xrightarrow{[\![m]\!] \otimes [\![n]\!]} [\![\mathrm{Diag}(T,U)]\!] \otimes [\![T]\!] \xrightarrow{\cong} !([\![T]\!] \multimap [\![U]\!]) \otimes [\![T]\!] \xrightarrow{\varepsilon \otimes \mathrm{id}} ([\![T]\!] \multimap [\![U]\!]) \otimes [\![T]\!] \xrightarrow{\mathrm{ev}} [\![U]\!]
\end{aligned}$$

$$[\![\Phi; \emptyset \vdash (\vec{\ell}, S, \vec{\ell'}) : \mathrm{Diag}(T,U)]\!] := [\![\Phi]\!] \xrightarrow{\diamond} I \xrightarrow{\cong} F(1) \xrightarrow{F(\Psi(\phi(\vec{\ell}, S, \vec{\ell'})))} [\![\mathrm{Diag}(T,U)]\!]$$

Fig. 26.2 Denotational semantics of the ECLNL calculus

Then a simple induction argument shows the next proposition.

Proposition 1 *For every non-linear type P, there is a canonical isomorphism $\iota_P : [\![P]\!] \cong F(\!|P|\!)$.*

A *context* is a function from a finite set of variables to types. We write contexts as $\Gamma = x_1 : A_1, x_2 : A_2, \dots, x_n : A_n$, where the x_i are variables and A_i are types. Its interpretation is as usual $[\![\Gamma]\!] = [\![A_1]\!] \otimes \cdots \otimes [\![A_n]\!]$. A variable in a context is non-linear (linear) if it is assigned a non-linear (linear) type. A context that contains only non-linear variables is called a *non-linear context*. Note, that we do not define linear contexts, because our typing rules refer only to contexts that either are non-linear or arbitrary (mixed).

A typing judgement has the form $\Gamma \vdash m : A$, where Γ is an (arbitrary) context, m is a term and A is a type. Its interpretation is a morphism $[\![\Gamma \vdash m : A]\!] : [\![\Gamma]\!] \to [\![A]\!]$ in $\mathbf{C}$, defined by induction on the derivation. For the typing rules of CLNL, the label contexts Q, Q', etc. from Fig. 26.1 should be ignored. For example, the (pair) rule in CLNL becomes:

$$\frac{\Phi, \Gamma_1 \vdash m : A \qquad \Phi, \Gamma_2 \vdash n : B}{\Phi, \Gamma_1, \Gamma_2 \vdash \langle m, n \rangle : A \otimes B}\ \text{(pair)}$$

The type system enforces that a linear variable is used exactly once, whereas a non-linear variable may be used any number of times, including zero. Unlike Benton's LNL calculus, derivations in CLNL are in general not unique, because non-linear variables may be part of an arbitrary context Γ. For example, if P_1 and P_2 are non-linear types, then:

$$\frac{x : P_1 \vdash x : P_1 \qquad y : P_2 \vdash y : P_2}{x : P_1, y : P_2 \vdash \langle x, y \rangle : P_1 \otimes P_2}\ \text{(pair)}$$

$$\frac{x : P_1 \vdash x : P_1 \qquad x : P_1, y : P_2 \vdash y : P_2}{x : P_1, y : P_2 \vdash \langle x, y \rangle : P_1 \otimes P_2}\ \text{(pair)}$$

are two different derivations of the same judgement. While this might seem to be a disadvantage, it leads to a reduction in the number of rules, it allows a language extension that supports describing string diagrams (see Sect. 26.4), and it allows us to easily add general recursion (see Sect. 26.5). Moreover, the interpretation of any two derivations of the same judgement are equal (see Theorem 3).

Definition 2 A morphism $f : [\![P_1]\!] \to [\![P_2]\!]$ is called *non-linear*, if

$$f = [\![P_1]\!] \xrightarrow{\iota} F(\!|P_1|\!) \xrightarrow{F(f')} F(\!|P_2|\!) \xrightarrow{\iota^{-1}} [\![P_2]\!],$$

for some $f' \in \mathbf{V}((\!|P_1|\!), (\!|P_2|\!))$.

Definition 3 We define maps on non-linear types as follows:

Discard: $\diamond_P := [\![P]\!] \xrightarrow{\iota} F(\!|P|\!) \xrightarrow{F1} F1 \xrightarrow{\cong} I;$

Copy: $\Delta_P := [\![P]\!] \xrightarrow{\iota} F(\!|P|\!) \xrightarrow{F(\langle \mathrm{id},\mathrm{id}\rangle)} F((\!|P|\!) \times (\!|P|\!)) \xrightarrow{\cong} F(\!|P|\!) \otimes F(\!|P|\!) \xrightarrow{\iota^{-1}\otimes\iota^{-1}} [\![P]\!] \otimes [\![P]\!];$

Lift: $\mathbf{lift}_P := [\![P]\!] \xrightarrow{\iota} F(\!|P|\!) \xrightarrow{F\eta} !F(\!|P|\!) \xrightarrow{!\iota^{-1}} ![\![P]\!].$

Proposition 2 *If $f : [\![P_1]\!] \to [\![P_2]\!]$ is non-linear, then:*

- $\diamond_{P_2} \circ f = \diamond_{P_1};$
- $\Delta_{P_2} \circ f = (f \otimes f) \circ \Delta_{P_1};$
- $\boldsymbol{lift}_{P_2} \circ f = !f \circ \boldsymbol{lift}_{P_1}.$

The operational and denotational semantics for the languages we discuss are presented in Figs. 26.2 and 26.3. The rules for CLNL are obvious special cases of those for ECLNL (which we discuss in the next section). The evaluation rules for CLNL can be derived from those of ECLNL (Fig. 26.3) by ignoring the diagram components. For example, the evaluation rule for (pair) is given by:

$$\frac{m \Downarrow v \qquad n \Downarrow v'}{\langle m, n\rangle \Downarrow \langle v, v'\rangle}$$

Similarly, the denotational interpretations of terms in CLNL can be derived from those of ECLNL (Fig. 26.2) by ignoring the Q label contexts. For example, the interpretation of $[\![\Phi, \Gamma_1, \Gamma_2 \vdash \langle m, n\rangle : A \otimes B]\!]$ is given by the composition:

$$[\![\Phi]\!] \otimes [\![\Gamma_1]\!] \otimes [\![\Gamma_2]\!] \xrightarrow{\Delta\otimes\mathrm{id}} [\![\Phi]\!] \otimes [\![\Phi]\!] \otimes [\![\Gamma_1]\!] \otimes [\![\Gamma_2]\!] \xrightarrow{\cong} [\![\Phi]\!] \otimes [\![\Gamma_1]\!] \otimes [\![\Phi]\!] \otimes [\![\Gamma_2]\!] \xrightarrow{[\![m]\!]\otimes[\![n]\!]} [\![A]\!] \otimes [\![B]\!].$$

Theorem 1 *Theorems 3–6 also hold true when restricted to the CLNL calculus in the obvious way.*

26.4 Enriching the CLNL Calculus

In this section we introduce the *enriched* CLNL calculus, ECLNL, whose syntax and operational semantics coincide with those of Proto-Quipper-M Rios & Selinger (2017). We rename the language in order to emphasize its dependence on its abstract categorical model, an LNL model with an associated *enrichment*. The categorical enrichment provides a natural framework for formulating the models we use, and for stating the constructivity properties (see Sect. 26.4.3) that we want our concrete models to satisfy.

$$\frac{}{(S,x) \Downarrow \text{Error}} \quad \frac{}{(S,\ell) \Downarrow (S,\ell)} \quad \frac{}{(S,c) \Downarrow (S,c)} \quad \frac{(S,m) \Downarrow (S',v) \quad (S',n[v/x]) \Downarrow (S'',v')}{(S,\text{let } x = m \text{ in } n) \Downarrow (S'',v')}$$

$$\frac{(S,m) \Downarrow (S',v)}{(S,\Box m) \Downarrow \text{Error}} \quad \frac{(S,m) \Downarrow (S',v)}{(S,\text{left } m) \Downarrow (S',\text{left } v)} \quad \frac{(S,m) \Downarrow (S',v)}{(S,\text{right } m) \Downarrow (S',\text{right } v)}$$

$$\frac{(S,m) \Downarrow (S',\text{left } v) \quad (S',n[v/x]) \Downarrow (S'',v')}{(S,\text{case } m \text{ of } \{\text{left } x \to n | \text{right } y \to p\}) \Downarrow (S'',v')} \quad \frac{(S,m) \Downarrow (S',\text{right } v) \quad (S',n[v/y]) \Downarrow (S'',v')}{(S,\text{case } m \text{ of } \{\text{left } x \to n | \text{right } y \to p\}) \Downarrow (S'',v')}$$

$$\frac{(S,m) \Downarrow \text{otherwise}}{(S,\text{case } m \text{ of } \{\text{left } x \to n | \text{right } y \to p\}) \Downarrow \text{Error}}$$

$$\frac{}{(S,*) \Downarrow (S,*)} \quad \frac{(S,m) \Downarrow (S',*) \quad (S',n) \Downarrow (S'',v')}{(S,m;n) \Downarrow (S'',v')} \quad \frac{(S,m) \Downarrow \text{otherwise}}{(S,m;n) \Downarrow \text{Error}}$$

$$\frac{(S,m) \Downarrow (S',v) \quad (S',n) \Downarrow (S'',v')}{(S,\langle m,n\rangle) \Downarrow (S'',\langle v,v'\rangle)}$$

$$\frac{(S,m) \Downarrow (S',\langle v,v'\rangle) \quad (S',n[v\ /\ x,v'\ /\ y]) \Downarrow (S'',w)}{(S,\text{let } \langle x,y\rangle = m \text{ in } n) \Downarrow (S'',w)} \quad \frac{(S,m) \Downarrow \text{otherwise}}{(S,\text{let } \langle x,y\rangle = m \text{ in } n) \Downarrow \text{Error}}$$

$$\frac{}{(S,\lambda x.m) \Downarrow (S,\lambda x.m)}$$

$$\frac{(S,m) \Downarrow (S',\lambda x.m') \quad (S',n) \Downarrow (S'',v) \quad (S'',m'[v/x]) \Downarrow (S''',v')}{(S,mn) \Downarrow (S''',v')} \quad \frac{(S,m) \Downarrow \text{otherwise}}{(S,mn) \Downarrow \text{Error}}$$

$$\frac{}{(S,\text{lift } m) \Downarrow (S,\text{lift } m)} \quad \frac{(S,m) \Downarrow (S',\text{lift } m') \quad (S',m') \Downarrow (S'',v)}{(S,\text{force } m) \Downarrow (S'',v)} \quad \frac{(S,m) \Downarrow \text{otherwise}}{(S,\text{force } m) \Downarrow \text{Error}}$$

$$\frac{(S,m) \Downarrow (S',\text{lift } n) \quad \text{freshlabels}(T) = (Q,\vec{\ell}) \quad (\text{id}_Q, n\vec{\ell}) \Downarrow (D,\vec{\ell}')}{(S,\text{box}_T m) \Downarrow (S',(\vec{\ell},D,\vec{\ell}'))}$$

$$\frac{(S,m) \Downarrow (S',\text{lift } n) \quad \text{freshlabels}(T) = (Q,\vec{\ell}) \quad (\text{id}_Q, n\vec{\ell}) \Downarrow \text{otherwise}}{(S,\text{box}_T m) \Downarrow \text{Error}} \quad \frac{(S,m) \Downarrow \text{otherwise}}{(S,\text{box}_T m) \Downarrow \text{Error}}$$

$$\frac{(S,m) \Downarrow (S',(\vec{\ell},D,\vec{\ell}')) \quad (S',n) \Downarrow (S'',\vec{k}) \quad \text{append}(S'',\vec{k},\vec{\ell},D,\vec{\ell}') = (S''',\vec{k}')}{(S,\text{apply}(m,n)) \Downarrow (S''',\vec{k}')}$$

$$\frac{(S,m) \Downarrow (S',(\vec{\ell},D,\vec{\ell}')) \quad (S',n) \Downarrow (S'',\vec{k}) \quad \text{append}(S'',\vec{k},\vec{\ell},D,\vec{\ell}') \text{ undefined}}{(S,\text{apply}(m,n)) \Downarrow \text{Error}} \quad \frac{}{(S,(\vec{\ell},D,\vec{\ell}')) \Downarrow (S,(\vec{\ell},D,\vec{\ell}'))}$$

Fig. 26.3 Operational semantics of the ECLNL calculus

We begin by briefly recalling the main ingredients of categories enriched over a symmetric monoidal closed category $(\mathbf{V}, \otimes, \multimap, I)$:

- A $\mathbf{V}$-*enriched* category (briefly, a $\mathbf{V}$-category) $\mathscr{A}$ consists of a collection of objects; for each pair of objects A, B there is a 'hom' object $\mathscr{A}(A, B) \in \mathbf{V}$; for each object A, there is a 'unit' morphism $u_A : I \to \mathscr{A}(A, A)$ in $\mathbf{V}$; and given objects A, B, C, there is a 'composition' morphism $c_{ABC} : \mathscr{A}(A, B) \otimes \mathscr{A}(B, C) \to \mathscr{A}(A, C)$ in $\mathbf{V}$.
- A $\mathbf{V}$-*functor* $F : \mathscr{A} \to \mathscr{B}$ between $\mathbf{V}$-categories assigns to each object $A \in \mathscr{A}$ an object $FA \in \mathscr{B}$, and to each pair of objects $A, A' \in \mathscr{A}$ a $\mathbf{V}$-morphism $F_{AA'} : \mathscr{A}(A, A') \to \mathscr{B}(FA, FA')$;
- A $\mathbf{V}$-*natural transformation* between $\mathbf{V}$-functors $F, G : \mathscr{A} \to \mathscr{B}$ consists of $\mathbf{V}$-morphisms $\alpha_A : I \to \mathscr{B}(FA, GA)$ for each $A \in \mathscr{A}$;
- A $\mathbf{V}$-functor $F : \mathscr{A} \to \mathscr{B}$ has a right $\mathbf{V}$-*adjoint* $G : \mathscr{B} \to \mathscr{A}$ if there is a $\mathbf{V}$-isomorphism, $\mathscr{B}(FA, B) \cong \mathscr{A}(A, GB)$ that is $\mathbf{V}$-natural in both A and B;

The $\mathbf{V}$-morphisms that occur in these definitions are all subject to additional conditions expressed in terms of commuting diagrams in $\mathbf{V}$; for these we refer to (Borceux 1994, Chap. 6), which provides a detailed exposition on enriched category theory. We denote the category of $\mathbf{V}$-categories by $\mathbf{V}$-$\mathbf{Cat}$.

The first example of a $\mathbf{V}$-enriched category is the category $\mathscr{V}$ that has the same objects as $\mathbf{V}$ and whose hom objects are given by $\mathscr{V}(A, B) = A \multimap B$. We refer to this category as the *self-enrichment* of $\mathbf{V}$. If $\mathscr{A}$ is a $\mathbf{V}$-category, then the $\mathbf{V}$-*copower* of an object $A \in \mathscr{A}$ by an object $X \in \mathbf{V}$ is an object $X \odot A \in \mathscr{A}$ together with an isomorphism $\mathscr{A}(X \odot A, B) \cong \mathscr{V}(X, \mathscr{A}(A, B))$, which is $\mathbf{V}$-natural in B.

Any (lax) monoidal functor $G : \mathbf{C} \to \mathbf{V}$ between symmetric monoidal closed categories induces a *change of base* functor $G_* : \mathbf{C}\text{-}\mathbf{Cat} \to \mathbf{V}\text{-}\mathbf{Cat}$ assigning to each $\mathbf{C}$-category $\mathscr{A}$ a $\mathbf{V}$-category $G_*\mathscr{A}$ with the same objects as $\mathscr{A}$, but with hom objects given by $(G_*\mathscr{A})(A, B) = G\mathscr{A}(A, B)$. In particular, if $\mathbf{V}$ is locally small (which we always assume), then the functor $\mathbf{V}(I, -) : \mathbf{V} \to \mathbf{Set}$ is a monoidal functor; the corresponding change of base functor assigns to each $\mathbf{V}$-category $\mathscr{A}$ its *underlying category*, which we denote with $\mathbf{A}$, i.e., the same letter but in boldface. We note that the underlying category of $\mathscr{V}$ is isomorphic to $\mathbf{V}$. Moreover, if the monoidal functor G above has a strong monoidal left adjoint, then the corresponding change of base functor maps $\mathbf{C}$-categories to $\mathbf{V}$-categories with isomorphic underlying categories, and $\mathbf{C}$-functors to $\mathbf{V}$-functors with the same underlying functors (up to the isomorphisms between the underlying categories). If $\mathbf{V}$ has all coproducts, then $\mathbf{V}(I, -)$ has a left adjoint $V : \mathbf{Set} \to \mathbf{V}$ that is monoidal (Borceux 1994, Proposition 6.4.6). Applying the corresponding change of base functor to a locally small category equips this category with the *free* $\mathbf{V}$-enrichment.

Symmetric monoidal categories can be generalized to $\mathbf{V}$-*symmetric monoidal* categories, where the monoidal structure is also enriched over $\mathbf{V}$ (Lucyshyn-Wright 2016, Sect. 4). It follows from (Lucyshyn-Wright 2016, Proposition 6.3) that the functor G_* above maps $\mathbf{C}$-symmetric monoidal categories to $\mathbf{V}$-symmetric monoidal categories. If for each fixed $A \in \mathbf{V}$, the $\mathbf{V}$-functor $(- \otimes A)$ has a right $\mathbf{V}$-adjoint, denoted $(A \multimap -)$, then we call $\mathscr{A}$ a $\mathbf{V}$-symmetric monoidal *closed* category. We note that the $(- \otimes -)$ and $(- \multimap -)$ bifunctors on $\mathbf{V}$ can be *enriched* to $\mathbf{V}$-bifunctors on $\mathscr{V}$ (i.e., such that their underlying functors correspond to the original functors) such that $\mathscr{V}$ becomes a $\mathbf{V}$-symmetric monoidal closed category.

Finally, if $\mathbf{V}$ has finite products, a $\mathbf{V}$-category $\mathscr{A}$ is said to have $\mathbf{V}$-coproducts if it has an object 0 and for each $A, B \in \mathscr{A}$ there is an object $A + B \in \mathscr{A}$ together with isomorphisms

$$1 \cong \mathscr{A}(0, C), \quad \mathscr{A}(A, C) \times \mathscr{A}(B, C) \cong \mathscr{A}(A + B, C),$$

$\mathbf{V}$-natural in C.

Definition 4 An *enriched CLNL model* is given by the following data:

1. A cartesian closed category $\mathbf{V}$ together with its self-enrichment $\mathscr{V}$, such that $\mathscr{V}$ has finite $\mathbf{V}$-coproducts;
2. A $\mathbf{V}$-symmetric monoidal closed category $\mathscr{C}$ with underlying category $\mathbf{C}$ such that $\mathscr{C}$ has $\mathbf{V}$-copowers and finite $\mathbf{V}$-coproducts;
3. A $\mathbf{V}$-adjunction: $\begin{tikzcd} \mathscr{V} \arrow[rr, bend left, "- \odot I"] \arrow[rr, phantom, "\perp"] & & \mathscr{C} \arrow[ll, bend left, "\mathscr{C}(I,-)"] \end{tikzcd}$ together with a CLNL model on the underlying adjunction.

We also adopt the following notation: F and G are the underlying functors of $(- \odot I)$ and $\mathscr{C}(I, -)$ respectively and we use the same notation for the underlying CLNL model as in Definition 1.

By definition, every enriched CLNL model is a CLNL model with some additional (enriched) structure. But as the next theorem shows, every CLNL model induces the additional enriched structure as well. The CCC $\mathbf{V}$ can be equipped with its self-enrichment $\mathscr{V}$ in a canonical way. The symmetric monoidal structure of the adjunction then allows us to equip the SMCC $\mathbf{C}$ with a $\mathbf{V}$-enrichment by making use of the induced change-of-base functors which stem from the adjunction. Then one can show that the now constructed $\mathbf{V}$-enriched category $\mathscr{C}$ has $\mathbf{V}$-copowers and the original adjunction enriches to a $\mathbf{V}$-enriched one. We conclude:

Theorem 2 *Every CLNL model induces an enriched CLNL model.*

Proof Combine (Egger et al. 2014, Proposition 6.7) and (Lucyshyn-Wright 2016, Theorem 11.2). □

The following proposition will be useful when defining the semantics of our language.

Proposition 3 *In every enriched CLNL model:*

1. *There is a* $\mathbf{V}$*-natural isomorphism* $G(A \multimap B) \cong \mathscr{C}(A, B)$;
2. $!(A \multimap B) \cong F(\mathscr{C}(A, B))$.
3. *There is a natural isomorphism* $\Psi : \mathbf{C}(A, B) \cong \mathbf{V}(1, \mathscr{C}(A, B))$.

Proof

(1.) $G(A \multimap B) = \mathscr{C}(I, A \multimap B) \cong \mathscr{C}(A, B)$;

(2.) Apply F to (1.);

(3.) $\mathbf{C}(A, B) \cong \mathbf{C}(I, A \multimap B) \cong \mathbf{C}(F1, A \multimap B) \cong \mathbf{V}(1, G(A \multimap B)) \cong \mathbf{V}(1, \mathscr{C}(A, B))$. □

26.4.1 The String Diagram Model

The ECLNL calculus is designed to describe string diagrams. So we first explain exactly what kind of diagrams we have in mind. The morphisms of any symmetric monoidal category can be described using string diagrams (Selinger, 2011).[1] So, we choose an arbitrary symmetric monoidal category $\mathbf{M}$, and then the string diagrams we will be working with are exactly those that correspond to the morphisms of $\mathbf{M}$.

For example, if we set $\mathbf{M} = \mathbf{FdCStar}$, the category of finite-dimensional C*-algebras and completely positive maps, then we can use our calculus for quantum programming. Another interesting choice for quantum computing, in light of recent results Perdrix et al. (2017), is setting $\mathbf{M}$ to be a suitable category of ZX-calculus diagrams. If $\mathbf{M} = \mathbf{PNB}$, the category of Petri Nets with Boundaries (Owen, 2015), then our calculus may be used to generate such Petri nets.

As with CLNL, our discussion of ECLNL begins with its categorical model.

Definition 5 An *ECLNL model* is given by the following data:

- An enriched CLNL model (Definition 4);
- A symmetric monoidal category $(\mathbf{M}, \boxtimes, J)$ and a strong symmetric monoidal functor $E : \mathbf{M} \to \mathbf{C}$.

For the remainder of the section, we consider an arbitrary, but fixed, ECLNL model.

26.4.2 Syntax and Semantics

We first introduce new types in our syntax that correspond to the objects of $\mathbf{M}$. Using terminology introduced in Rios & Selinger (2017), where string diagrams are referred to as *circuits*, we let W be a fixed set of *wire types*, and we assume there is

[1] The interested reader can consult Selinger (2011) for more information on string diagrammatic representations of morphisms.

an interpretation $[\![-]\!]_{\mathbf{M}} : W \to \mathrm{Ob}(\mathbf{M})$. We use $\alpha, \beta, \ldots$ to range over the elements of W. For a wire type α, we define the interpretation of α in $\mathbf{C}$ to be $[\![\alpha]\!] = E([\![\alpha]\!]_M)$. The grammar for $\mathbf{M}$-types is given in Fig. 26.1, and we extend $[\![-]\!]_{\mathbf{M}}$ to $\mathbf{M}$-types in the obvious way.

To build more complicated string diagrams from simpler components, we need to refer to certain wires of the component diagrams, to specify how to compose them. This is accomplished by assigning *labels* to the wires of our string diagrams, as demonstrated in the following construction.

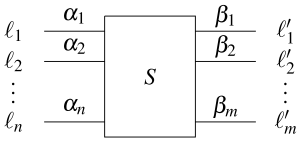
Let L be a countably infinite set of labels. We use letters ℓ, k to range over the elements of L. A *label context* is a function from a finite subset of L to W, which we write as $\ell_1 : \alpha_1, \ldots, \ell_n : \alpha_n$. We use $Q_1, Q_2, \ldots$ to refer to label contexts. To each label context $Q = \ell_1 : \alpha_1, \ldots, \ell_n : \alpha_n$, we assign an object of $\mathbf{M}$ given by $[\![Q]\!]_{\mathbf{M}} := [\![\alpha_1]\!]_{\mathbf{M}} \boxtimes \cdots \boxtimes [\![\alpha_n]\!]_{\mathbf{M}}$. If $Q = \emptyset$, then $[\![Q]\!]_{\mathbf{M}} = J$. We denote *label tuples* by $\boldsymbol{\ell}$ and $\boldsymbol{k}$; these are simply tuples of label terms built up using the (pair) rule.

We now define the category $\mathbf{M}_L$ of *labelled string diagrams*:

- The objects of $\mathbf{M}_L$ are label contexts Q.
- The morphisms of $\mathbf{M}_L(Q_1, Q_2)$ are exactly the morphisms of $\mathbf{M}([\![Q_1]\!]_{\mathbf{M}}, [\![Q_2]\!]_{\mathbf{M}})$.

So, by construction, $[\![-]\!]_{\mathbf{M}} : \mathbf{M}_L \to \mathbf{M}$ is a full and faithful functor. Observe that if Q and Q' are label contexts that differ only by a renaming of labels, then $Q \cong Q'$. Moreover, for any two label contexts Q_1 and Q_2, by renaming labels we can construct $Q_1' \cong Q_1$ such that Q_1' and Q_2 are disjoint.

We equip the category $\mathbf{M}_L$ with the unique (up to natural isomorphism) symmetric monoidal structure that makes $[\![-]\!]_{\mathbf{M}}$ a symmetric monoidal functor. We then have $Q \otimes Q' \cong Q \cup Q'$ for any pair of disjoint label contexts. We use S, D to range over the morphisms of $\mathbf{M}_L$ and we visualise them in the following way:



where $S : \{\ell_1 : \alpha_1, \ldots, \ell_n : \alpha_n\} \to \{\ell_1' : \beta_1, \ldots, \ell_m' : \beta_m\} \in \mathbf{M}_L$ and $[\![S]\!]_{\mathbf{M}} : [\![\alpha_1]\!]_{\mathbf{M}} \boxtimes \cdots \boxtimes [\![\alpha_n]\!]_{\mathbf{M}} \to [\![\beta_1]\!]_{\mathbf{M}} \boxtimes \cdots \boxtimes [\![\beta_m]\!]_{\mathbf{M}} \in \mathbf{M}$.

A label context $Q = \ell_1 : \alpha_1, \ldots, \ell_n : \alpha_n$ is interpreted in $\mathbf{C}$ as $[\![Q]\!] = [\![\alpha_1]\!] \otimes \cdots \otimes [\![\alpha_n]\!]$ or by $[\![Q]\!] = I$ if $Q = \emptyset$. A labelled string diagram $S : Q \to Q'$ is interpreted in $\mathbf{C}$ as the composition:

$$[\![S]\!] := [\![Q]\!] \xrightarrow{\cong} E([\![Q]\!]_{\mathbf{M}}) \xrightarrow{E([\![S]\!]_{\mathbf{M}})} E([\![Q']\!]_{\mathbf{M}}) \xrightarrow{\cong} [\![Q']\!].$$

We also add the type $\mathrm{Diag}(T, U)$ to the language (see Fig. 26.1); $\mathrm{Diag}(T, U)$ should be thought of as the type of string diagrams with inputs T and outputs U, where T and U are $\mathbf{M}$-types.

The term language is extended by adding the labels and label tuples just discussed, and the terms $\text{box}_T m$, $\text{apply}(m, n)$ and $(\boldsymbol{\ell}, S, \boldsymbol{\ell}')$. The term $\text{box}_T m$ should be thought of as "boxing up" an already completed diagram m; $\text{apply}(m, n)$ represents the application of the boxed diagram m to the state n; and the term $(\boldsymbol{\ell}, S, \boldsymbol{\ell}')$ is a value which represents a boxed diagram. Users of the ECLNL programming language are not expected to write labelled string diagrams S or terms such as $(\boldsymbol{\ell}, S, \boldsymbol{\ell}')$. Instead, these terms are computed by the programming language itself. Depending on the diagram model, the language should be extended with constants that are exposed to the user, for example, for quantum computing, a constant $H : \text{Diag}(\textbf{qubit}, \textbf{qubit})$ could be utilised by the user to build quantum circuits.

The term typing judgements from the previous section are now extended to include a label context as well, which is separated from the variable context using a semicolon; the new format of a term typing judgement is $\Gamma; Q \vdash m : A$. Its interpretation is a morphism $[\![\Gamma]\!] \otimes [\![Q]\!] \to [\![A]\!]$ in $\mathbf{C}$ that is defined by induction on the derivation as shown in Fig. 26.2.

In the definition of the (diag) rule in the denotational semantics, we use a function ϕ, which we now explain. From the premises of the rule, it follows that $[\![\boldsymbol{\ell}]\!] : [\![Q]\!] \to [\![T]\!]$ and $[\![\boldsymbol{\ell}']\!] : [\![Q']\!] \to [\![U]\!]$ are isomorphisms. Then, $\phi(\boldsymbol{\ell}, S, \boldsymbol{\ell}')$ is defined to be the morphism:

$$\phi(\boldsymbol{\ell}, S, \boldsymbol{\ell}') = [\![T]\!] \xrightarrow{[\![\boldsymbol{\ell}]\!]^{-1}} [\![Q]\!] \xrightarrow{[\![S]\!]} [\![Q']\!] \xrightarrow{[\![\boldsymbol{\ell}']\!]} [\![U]\!].$$

Theorem 3 *Let D_1 and D_2 be derivations of a judgement $\Gamma; Q \vdash m : A$. Then $[\![D_1]\!] = [\![D_2]\!]$.*

Because of this theorem, we write $[\![\Gamma; Q \vdash m : A]\!]$ instead of $[\![D]\!]$.

A *configuration* is a pair (S, m), where S is a labelled string diagram and m is a term. Operationally, we may think of S as the diagram that has been constructed so far, and m as the program which remains to be executed.

Definition 6 A configuration is said to be *well-typed* with inputs Q, outputs Q' and type A, which we write as $Q \vdash (S, m) : A; Q'$, if there exists Q'' disjoint from Q', s.t. $S : Q \to Q'' \cup Q'$ is a labelled string diagram and $\emptyset; Q'' \vdash m : A$.

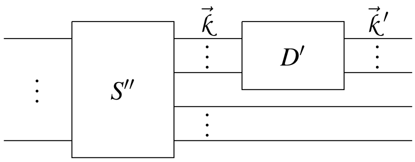
Thus, in a well-typed configuration, the term m has no free variables and its labels correspond to a subset of the outputs of S. We interpret a well-typed configuration $Q \vdash (S, m) : A; Q'$, by:

$$[\![(S, m)]\!] := [\![Q]\!] \xrightarrow{[\![S]\!]} [\![Q'']\!] \otimes [\![Q']\!] \xrightarrow{[\![\emptyset; Q'' \vdash m:A]\!] \otimes \text{id}} [\![A]\!] \otimes [\![Q']\!]$$

The big-step semantics is defined on configurations; because of space reasons, we only show an excerpt of the rules in Fig. 26.3. The rest of the rules are standard. A *configuration value* is a configuration (S, v), where v is a value. The evaluation relation $(S, m) \Downarrow (S', v)$ then relates configurations to configuration values. Intuitively, this can be interpreted in the following way: assuming a constructed diagram S, then evaluating term m results in a diagram S' (obtained from S by appending other subdiagrams described by m) and value v. There's also an error relation $(S, m) \Downarrow$ Error which indicates that a run-time error occurs when we execute term m from configuration S. There are many such Error rules, but they are uninteresting, so we omit some of them (also see Theorem 4).

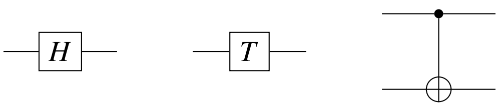
The operational semantics is presented in Fig. 26.3. The evaluation rule for $\text{box}_T m$ makes use of a function *freshlabels*. Given a **M**-type T, freshlabels(T) returns a pair $(Q, \boldsymbol{\ell})$ such that $\emptyset; Q \vdash \boldsymbol{\ell} : T$, where the labels in $\boldsymbol{\ell}$ are fresh in the sense that they do not occur anywhere else in the derivation. This can always be done, and the resulting Q and $\boldsymbol{\ell}$ are determined uniquely, up to a renaming of labels (which is inessential).

The evaluation rule for apply(m, n) makes use of a function *append*. Given a labelled string diagram S'' together with a label tuple $\vec{k}$ and term $(\boldsymbol{\ell}, D, \boldsymbol{\ell}')$, it is defined as follows. Assuming that $\boldsymbol{\ell}$ and $\vec{k}$ correspond exactly to the inputs of D and that $\boldsymbol{\ell}'$ contains exactly the outputs of D, then we may construct a term $(\vec{k}, D', \vec{k}')$ which is equivalent to $(\boldsymbol{\ell}, D, \boldsymbol{\ell}')$ in the sense that they only differ by a renaming of labels. Moreover, we may do so by choosing D' and $\vec{k}'$ such that the labels in $\vec{k}'$ are fresh. Then, assuming the labels in $\vec{k}$ correspond to a subset of the outputs of S'', we may construct the labelled string diagram S''' given by the composition:



Finally, append$(S'', \vec{k}, \boldsymbol{\ell}, D, \boldsymbol{\ell}')$ returns the pair $(S''', \vec{k}')$ if the above assumptions are met, and is undefined otherwise (which would result in a run-time error).

Example 1 Let us assume that our string diagram model is given by quantum circuits. More precisely, assume that **M** contains at least the following diagram generators:

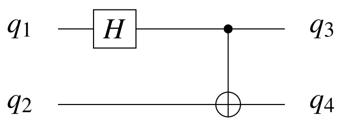


representing the diagrams that describe (from left to right): the Hadamard gate, the T gate and the CNOT gate of quantum computing. Assume further that the syntax of the language is extended with a built-in wire type **qubit** and term constants H : Diag(**qubit**, **qubit**), T : Diag(**qubit**, **qubit**) and $CNOT$: Diag(**qubit** $\otimes$ **qubit**, **qubit** $\otimes$ **qubit**) which are the user-exposed constants for constructing quantum circuits. Here we are overloading notation which hopefully should not lead to confusion.

Next, consider the term $m \equiv \text{apply}(CNOT, \langle \text{apply}(H, q_1), q_2 \rangle)$, where

$$\emptyset; q_1 : \textbf{qubit}, q_2 : \textbf{qubit} \vdash m : \textbf{qubit} \otimes \textbf{qubit}.$$

This represents a unitary operation on two qubits, where we first apply a Hadamard gate on the first qubit and then we apply a CNOT gate on the two qubits, with the first one being the control. If we add some (obvious) rules for dealing with constants in the operational semantics, then we may evaluate m when it appears as part of a configuration. For example, $(\text{id}_{\textbf{qubit}\otimes\textbf{qubit}}, m) \Downarrow (S, \langle q_3, q_4 \rangle)$, where S is the labelled string diagram given by



where we have omitted annotating the types of the wires (which are all **qubit**).

Theorem 4 (Error freeness Rios & Selinger (2017)) *If* $Q \vdash (S, m) : A; Q'$ *then* $(S, m) \not\Downarrow$ *Error.*

Theorem 5 (Subject reduction Rios & Selinger (2017)) *If* $Q \vdash (S, m) : A; Q'$ *and* $(S, m) \Downarrow (S', v)$, *then* $Q \vdash (S', v) : A; Q'$.

With this in place, we may now show our abstract model is sound. We remark that our abstract model is strictly more general than the one of Rios and Selinger (see Sect. 26.2, ***Related Work***).

Theorem 6 (Soundness) *If* $Q \vdash (S, m) : A; Q'$ *and* $(S, m) \Downarrow (S', v)$, *then* $[\![(S, m)]\!] = [\![(S', v)]\!]$.

26.4.3 A Constructive Property

If we assume, in addition, that $E : \mathbf{M} \to \mathbf{C}$ is fully faithful, then setting $\mathcal{M}(T, U) := \mathscr{C}(ET, EU)$ for $T, U \in \mathbf{M}$ defines a $\mathbf{V}$-enriched category $\mathcal{M}$ with the same objects as $\mathbf{M}$, and whose underlying category is isomorphic to $\mathbf{M}$. Moreover, E enriches to a fully faithful $\mathbf{V}$-functor $\underline{E} : \mathcal{M} \to \mathscr{C}$. As a consequence, our abstract model enjoys the following constructive property:

$$\begin{aligned}
&\mathbf{C}([\![\Phi]\!], [\![T]\!] \multimap [\![U]\!]) \cong \mathbf{C}(F(X), [\![T]\!] \multimap [\![U]\!]) \cong \\
&\mathbf{V}(X, G([\![T]\!] \multimap [\![U]\!])) \cong \mathbf{V}(X, \mathscr{C}([\![T]\!], [\![U]\!])) \cong \\
&\mathbf{V}(X, \mathscr{C}(\underline{E}[\![T]\!]_{\mathbf{M}}, \underline{E}[\![U]\!]_{\mathbf{M}})) = \mathbf{V}(X, \mathcal{M}([\![T]\!]_{\mathbf{M}}, [\![U]\!]_{\mathbf{M}}))
\end{aligned}$$

where we use the additional structure only in the last step. This means that any well-typed term $\Phi; \emptyset \vdash m : T \multimap U$ corresponds to a $\mathbf{V}$-parametrised family of string diagrams. For example, if $\mathbf{V} = \mathbf{Set}$ (or $\mathbf{V} = \mathbf{CPO}$), then we get precisely a (Scott-continuous) function from X to $\mathcal{M}(\llbracket T \rrbracket_{\mathbf{M}}, \llbracket U \rrbracket_{\mathbf{M}})$ or in other words, a (Scott-continuous) family of string diagrams from $\mathbf{M}$.

26.4.4 Concrete Models

The original concrete model of Rios and Selinger is now easily recovered as an instance of our abstract model (assuming $\mathbf{M}$ is small):

$$\begin{tikzcd} \mathbf{Set} \arrow[r, bend left, "- \odot I"] \arrow[r, phantom, "\perp"] & \mathbf{Fam}([\mathbf{M}^{\mathrm{op}}, \mathbf{Set}]) \arrow[l, bend left, "\mathbf{Fam}([\mathbf{M}^{\mathrm{op}}, \mathbf{Set}])(I,-)"] & {[\mathbf{M}^{\mathrm{op}}, \mathbf{Set}]} \arrow[l, hook'] & \mathbf{M} \arrow[l, hook', "Y"'] \end{tikzcd}$$

where $\mathbf{Fam}(-)$ is the well-known *families construction*. However, our abstract treatment of the language allows us to present a simpler sound model:

$$\begin{tikzcd} \mathbf{Set} \arrow[r, bend left, "- \odot I"] \arrow[r, phantom, "\perp"] & {[\mathbf{M}^{\mathrm{op}}, \mathbf{Set}]} \arrow[l, bend left, "{[\mathbf{M}^{\mathrm{op}}, \mathbf{Set}](I,-)}"] & \mathbf{M} \arrow[l, hook', "Y"'] \end{tikzcd}$$

And, an order-enriched model is given by:

$$\begin{tikzcd} \mathcal{CPO} \arrow[r, bend left, "- \odot I"] \arrow[r, phantom, "\perp"] & {[\mathcal{M}^{\mathrm{op}}, \mathcal{CPO}]} \arrow[l, bend left, "{[\mathcal{M}^{\mathrm{op}}, \mathcal{CPO}](I,-)}"] & \mathcal{M} \arrow[l, hook', "Y"'] \end{tikzcd}$$

where $\mathcal{M}$ is the free $\mathbf{CPO}$-enrichment of $\mathbf{M}$ (obtained by discretely ordering its homsets) and $\mathcal{CPO}$ is the self-enrichment of $\mathbf{CPO}$.

26.5 The ECLNL Calculus with Recursion

Additional structure for Benton's LNL models needed to support recursion was discussed by Benton and Wadler in Benton & Wadler (1996). This structure allows them to model recursion in related lambda calculi, and in the LNL calculus (renamed the "adjoint calculus") as well. However, they present no syntax or operational semantics for recursion in their LNL calculus and instead they "…*omit the rather messy details*". Here we extend both the CLNL and ECLNL calculi with recursion in a simple way by using similar additional semantic structure. We conjecture the simplicity of our extension is due to our use of a single type of judgement that employs mixed contexts; this is the main distinguishing feature of our CLNL calculus compared to the LNL calculus of Benton and Wadler. Furthermore, we also include a computational adequacy result for the CLNL calculus with recursion.

26.5.1 Extension with Recursion

We extend the ECLNL calculus by adding the term rec $x^{!A}.m$ and we add an additional typing rule (left) and an evaluation rule (right) as follows:

$$\frac{\Phi, x :!A; \emptyset \vdash m : A}{\Phi; \emptyset \vdash \text{rec } x^{!A}.m : A}\ (\text{rec}) \qquad \frac{(S, m[\text{lift rec } x^{!A}.m \;/\; x]) \Downarrow (S', v)}{(S, \text{rec } x^{!A}.m) \Downarrow (S', v)}$$

Notice that in the typing rule, the label contexts are empty and all free variables in m are non-linear. As a special case, the CLNL calculus also can be extended with recursion:

$$\frac{\Phi, x :!A \vdash m : A}{\Phi \vdash \text{rec } x^{!A}.m : A}\ (\text{rec}) \qquad \frac{m[\text{lift rec } x^{!A}.m \;/\; x] \Downarrow v}{\text{rec } x^{!A}.m \Downarrow v}$$

In both cases, (parametrised) algebraic compactness of the !-endofunctor is what is needed to soundly model the extension; Benton and Wadler make the same assumption.

Definition 7 An endofunctor $T : \mathbf{C} \to \mathbf{C}$ is *algebraically compact* if T has an initial T-algebra $T(\Omega) \xrightarrow{\omega} \Omega$ for which $\Omega \xrightarrow{\omega^{-1}} T(\Omega)$ is a final T-coalgebra. If the category $\mathbf{C}$ is monoidal, then an endofunctor $T : \mathbf{C} \to \mathbf{C}$ is *parametrically algebraically compact* if the endofunctor $A \otimes T(-)$ is algebraically compact for every $A \in \mathbf{C}$.

We note that this notion of parametrised algebraic compactness is weaker than Fiore's corresponding notion (Fiore, 1994), but it suffices for our purposes. This allows us to extend both ECLNL and CLNL models with recursion in the same way.

Definition 8 A model of the (E)CLNL calculus with recursion is given by a model of the (E)CLNL calculus for which the !-endofunctor is parametrically algebraically compact.

If $\Phi \in \mathbf{C}$ is a non-linear object, then the endofunctor $\Phi \otimes !(-)$ is algebraically compact. Let $\Phi \otimes !\Omega_\Phi \xrightarrow{\omega_\Phi} \Omega_\Phi$ be its initial algebra and let $m : \Phi \otimes !A \to A$ be an arbitrary morphism. We define γ_Φ and σ_m to be the unique anamorphism and catamorphism, respectively, such that the diagram in Fig. 26.4 commutes.

$$\begin{tikzcd}
\Phi \otimes !\Phi \arrow[d, dashed, "\mathrm{id} \otimes !\gamma_\Phi"'] & \Phi \otimes \Phi \arrow[l, "\mathrm{id} \otimes \mathbf{lift}"'] & \Phi \arrow[l, "\Delta"'] \arrow[d, dashed, "\gamma_\Phi"] \\
\Phi \otimes !\Omega_\Phi \arrow[d, "\mathrm{id}"'] & & \Omega_\Phi \arrow[ll, "\omega_\Phi^{-1}"'] \arrow[d, "\mathrm{id}"] \\
\Phi \otimes !\Omega_\Phi \arrow[rr, "\omega_\Phi"] \arrow[d, dashed, "\mathrm{id} \otimes !\sigma_m"'] & & \Omega_\Phi \arrow[d, dashed, "\sigma_m"] \\
\Phi \otimes !A \arrow[rr, "m"'] & & A
\end{tikzcd}$$

Fig. 26.4 Definition of σ_m and γ_Φ

Using this notation, we extend the denotational semantics to interpret recursion by adding the rule:

$$[\![\Phi; \emptyset \vdash \mathrm{rec}\ x^{!A}.m : A]\!] := \sigma_{[\![m]\!]} \circ \gamma_{[\![\Phi]\!]}.$$

Observe that when $\Phi = \emptyset$, we get:

$$[\![\mathrm{rec}\ x^{!A}.m]\!] = [\![m]\!] \circ ![\![\mathrm{rec}\ x^{!A}.m]\!] \circ \mathbf{lift} = [\![m]\!] \circ [\![\mathrm{lift}\ \mathrm{rec}\ x^{!A}.m]\!]$$

which is precisely a *linear fixpoint* in the sense of Braüner (1997).

Theorem 7 *Theorems 3–6 from the previous section remain true for the (E)CLNL calculus extended with recursion.*

26.5.2 Concrete Models

Let **CPO** be the category of cpo's (possibly without bottom) and Scott-continuous functions, and let $\mathbf{CPO}_{\perp!}$ be the category of *pointed* cpo's and *strict* Scott-continuous functions.

We present a concrete model for an arbitrary small symmetric monoidal **M**. Let $\mathcal{M}$ be the free **CPO**-enrichment of **M**. Then $\mathcal{M}$ has the same objects as **M** and hom-cpo's $\mathcal{M}(A, B)$ given by the hom-sets $\mathbf{M}(A, B)$ equipped with the discrete order. $\mathcal{M}$ is then a **CPO**-symmetric monoidal category with the same monoidal structure as **M**.

Let $\mathcal{M}_\perp$ be the free $\mathbf{CPO}_{\perp!}$-enrichment of **M**. Then, $\mathcal{M}_\perp$ has the same objects as **M** and hom-cpo's $\mathcal{M}_\perp(A, B) = \mathcal{M}(A, B)_\perp$, where $(-)_\perp : \mathcal{CPO} \to \mathcal{CPO}_{\perp!}$ is the domain-theoretic lifting functor. $\mathcal{M}_\perp$ is then a $\mathbf{CPO}_{\perp!}$-symmetric monoidal category with the same monoidal structure as that of $\mathcal{M}$ where, in addition, $\perp_{A,B}$ satisfies the conditions of Proposition 4 (see Sect. 26.5.3 below).

By using the enriched Yoneda lemma together with the Day convolution monoidal structure, we see that the enriched functor category $[\mathcal{M}_\perp^{\mathrm{op}}, \mathcal{CPO}_{\perp!}]$ is $\mathbf{CPO}_{\perp!}$-symmetric monoidal closed.

Theorem 8 *The following data:*

$$\begin{tikzcd} \mathcal{CPO} \arrow[r, bend left, "- \odot I"] \arrow[r, phantom, "\perp"] & {[\mathcal{M}_\perp^{op}, \mathcal{CPO}_{\perp!}]} \arrow[l, bend left, "{[\mathcal{M}_\perp^{op}, \mathcal{CPO}_{\perp!}](I,-)}"] & \mathcal{M}_\perp \arrow[l, hook', "Y"'] & \mathcal{M} \arrow[l, hook'] \end{tikzcd}$$

is a sound model of the ECLNL calculus extended with recursion.

Proof The subcategory inclusion $\mathcal{M} \hookrightarrow \mathcal{M}_\perp$ is **CPO**-enriched, faithful and strong symmetric monoidal, as is the enriched Yoneda embedding Y. The **CPO**-copower $(- \odot I)$ is given by:

$$(- \odot I) = (- \bullet I) \circ (-)_\perp,$$

where $(- \bullet I) : \mathcal{CPO}_{\perp!} \to [\mathcal{M}_\perp^{\mathrm{op}}, \mathcal{CPO}_{\perp!}]$ is the $\mathbf{CPO}_{\perp!}$-copower with the tensor unit (see Borceux (1994)). This follows because the right adjoint and the adjunction factor through $\mathcal{CPO}_{\perp!}$. Parametrised algebraic compactness of the !-endofunctor follows from (Fiore 1994, pp. 161–162). □

Moreover, the concrete model enjoys a constructive property similar to the one in Sect. 26.4.3. Using the same argument, if $\Phi; \emptyset \vdash m : T \multimap U$, then we obtain:

$$[\mathcal{M}_\perp^{\mathrm{op}}, \mathcal{CPO}_{\perp!}](\llbracket \Phi \rrbracket, \llbracket T \rrbracket \multimap \llbracket U \rrbracket) \cong \mathcal{CPO}(X, \mathcal{M}_\perp(\llbracket T \rrbracket_{\mathbf{M}}, \llbracket U \rrbracket_{\mathbf{M}}))$$

Therefore, the interpretation of m corresponds to a Scott-continuous function from X to $\mathcal{M}_\perp(\llbracket T \rrbracket_{\mathbf{M}}, \llbracket U \rrbracket_{\mathbf{M}})$. In other words, this is a family of *string diagram computations*, in the sense that every element is either a string diagram of **M** or a non-terminating computation.

Theorem 9 *The CLNL model*

$$\begin{tikzcd} \mathbf{CPO} \arrow[r, bend left, "(-)_\perp"] \arrow[r, phantom, "\perp"] & \mathbf{CPO}_{\perp!} \arrow[l, bend left, "U"] \end{tikzcd}$$

, where U is the forgetful functor, is a sound model for the CLNL calculus with recursion.

Proof Again, parametrised algebraic compactness of the !-endofunctor follows from (Fiore 1994, pp. 161–162). □

26.5.3 Computational Adequacy

In this subsection we show that computational adequacy holds at non-linear types for the concrete CLNL model given in the previous subsection.

We begin by showing that in any (E)CLNL model with recursion, the category **C** is pointed, which allows us to introduce a notion of undefinedness. Towards that end, we first introduce a slightly weaker notion, following Braüner (1997).

Definition 9 A symmetric monoidal closed category is *weakly pointed* if it is equipped with a morphism $\perp_A : I \to A$ for each object A, such that for every morphism $h : A \to B$, we have $h \circ \perp_A = \perp_B$. In this case, for each pair of objects A and B, there is a morphism $\perp_{A,B} = A \xrightarrow{\lambda_A^{-1}} I \otimes A \xrightarrow{\mathbf{uncurry}(\perp_{A \multimap B})} B$.

Proposition 4 (Braüner (1997)) *Let* **A** *be a weakly pointed category. Then:*

1. $f \circ \perp_{A,B} = \perp_{A,C}$ *for each morphism* $f : B \to C$*;*
2. $\perp_{B,C} \circ f = \perp_{A,C}$ *for each morphism* $f : A \to B$*;*
3. $\perp_{A,B} \otimes f = \perp_{A \otimes C, B \otimes D}$ *for each morphism* $f : C \to D$.
4. $f \otimes \perp_{A,B} = \perp_{C \otimes A, D \otimes B}$ *for each morphism* $f : C \to D$.

Lemma 1 *Any weakly pointed category with an initial object* 0 *is pointed. Moreover,* $\perp_A = \perp_{I,A}$ *and* $\perp_{A,B}$ *are zero morphisms.*

Theorem 10 *For every model of the (E)CLNL calculus with recursion,* **C** *is a pointed category with*

$$\perp_A = I \xrightarrow{\gamma_I} \Omega_I \xrightarrow{\sigma_{\epsilon_A}} A,$$

where Ω_I *is the carrier of the initial algebra for the* !*-endofunctor.*

Proof It suffices to show for any $h : A \to B$ that $h \circ \perp_A = \perp_B$ which follows from the naturality of ϵ and initiality of σ_ϵ. □

In particular, we have: $[\![\emptyset; \emptyset \vdash \text{rec } x^{!A}.\text{force } x : A]\!] = \perp_{[\![A]\!]}$. Thus, the interpretation of the simplest non-terminating program (of any type) is a zero morphism, as one would expect. Naturally, we use the zero morphisms of **C** to denote undefinedness in our adequacy result.

Assume that $\mathscr{C}$ is **CPO**-enriched and that $\perp_{A,B}$ is least in $\mathscr{C}(A, B)$. We shall use $\bigvee_i a_i$ to denote the supremum of the increasing chain $(a_i)_{i \in \mathbb{N}}$. For any Scott-continuous function $K : \mathscr{C}(A, B) \to \mathscr{C}(A, B)$, let $K^0 = \perp_{A,B}$ and $K^{i+1} = K(K^i)$, for $i \in \mathbb{N}$. Then $\bigvee_i K^i$ is the least fixpoint of K. Note that K isn't strict in general.

Lemma 2 *Consider an (E)CLNL model with recursion, where* $\mathbf{V} = \mathbf{CPO}$ *and where* $\perp_{A,B}$ *is least in* $\mathscr{C}(A, B)$*, for all objects A and B (or equivalently* $\mathscr{C}$ *is* $\mathbf{CPO}_{\perp!}$*-enriched). Let* $m : \Phi \otimes !A \to A$ *be a morphism in* **C**. *Let* K_m *be the Scott-continuous function* $K_m : \mathscr{C}(\Phi, A) \to \mathscr{C}(\Phi, A)$ *given by* $K_m(f) = m \circ (id \otimes !f) \circ (id \otimes \textbf{\textit{lift}}) \circ \Delta$*. Then:*

$$\sigma_m \circ \gamma_\Phi = \bigvee_i K_m^i.$$

The significance of this lemma is that it provides an equivalent semantic definition for the (rec) rule in terms of least fixpoints, provided we assume order-enrichment for our (E)CLNL models.

For the remainder of the section, we consider only the CLNL calculus which we interpret in the CLNL model of Theorem 9. Therefore, in what follows $\mathbf{C} = \mathbf{CPO}_{\perp !}$.

Lemma 3 *Let* $\emptyset \vdash v : P$ *be a well-typed value, where* P *is a non-linear type. Then* $[\![\emptyset \vdash v : P]\!] \neq \perp$.

Next, we prove adequacy using the standard method based on *formal approximation relations*, a notion first devised by Plotkin (1985).

Definition 10 For any type A, let:

$$\begin{aligned} V_A &:= \{v \mid v \text{ is a value and } \emptyset \vdash v : A\}; \\ T_A &:= \{m \mid \emptyset \vdash m : A\}. \end{aligned}$$

We define two families of *formal approximation relations:*

$$\begin{aligned} \trianglelefteq_A &\subseteq (\mathbf{C}(I, [\![A]\!]) - \{\perp\}) \times V_A \\ \sqsubseteq_A &\subseteq \mathbf{C}(I, [\![A]\!]) \times T_A \end{aligned}$$

by induction on the structure of A:

(A1) $f \trianglelefteq_I *$ iff $f = \mathrm{id}_I$;
(A2.1) $f \trianglelefteq_{A+B}$ left v iff $\exists f'.\ f = \text{left} \circ f'$ and $f' \trianglelefteq_A v$;
(A2.2) $f \trianglelefteq_{A+B}$ right v iff $\exists f'.\ f = \text{right} \circ f'$ and $f' \trianglelefteq_B v$;
(A3) $f \trianglelefteq_{A \otimes B} \langle v, w \rangle$ iff $\exists f', f''$, such that:
$f = f' \otimes f'' \circ \lambda_I^{-1}$ and $f' \trianglelefteq_A v$ and $f'' \trianglelefteq_B w$;
(A4) $f \trianglelefteq_{A \multimap B} \lambda x.\, m$ iff $\forall f' \in \mathbf{C}(I, [\![A]\!]), \forall v \in V_A$:

$$f' \trianglelefteq_A v \Rightarrow \text{eval} \circ (f \otimes f') \circ \lambda_I^{-1} \sqsubseteq_B m[v/x];$$

(A5) $f \trianglelefteq_{!A}$ lift m iff f is a non-linear morphism and
$\epsilon_A \circ f \sqsubseteq_A m$;
(B) $f \sqsubseteq_A m$ iff $f \neq \perp \Rightarrow\ \exists v \in V_A.\, m \Downarrow v$ and $f \trianglelefteq_A v$.

So, the relation $\trianglelefteq$ relates morphisms to values and $\sqsubseteq$ relates morphisms to terms.

Lemma 4 *If* $f \trianglelefteq_P v$*, where* P *is a non-linear type, then* f *is a non-linear morphism.*

Lemma 5 *For any* $m \in T_A$*, the property* $(- \sqsubseteq_A m)$ *is admissible for the (pointed) cpo* $\mathscr{C}(I, [\![A]\!])$ *in the sense that Scott fixpoint induction is sound.*

Proof One has to show $\perp \sqsubseteq_A m$, which is trivial, and also that $(- \sqsubseteq_A m)$ is closed under suprema of increasing chains of morphisms, which is easily proven by induction on A. □

Proposition 5 *Let $\Gamma \vdash m : A$, where $\Gamma = x_1 : A_1, \ldots, x_n : A_n$. Let $v_i \in V_{A_i}$ such that $f_i \trianglelefteq_{A_i} v_i$. If f is the composition:*

$$f := I \xrightarrow{\cong} I \otimes \cdots \otimes I \xrightarrow{f_1 \otimes \cdots \otimes f_n} \llbracket \Gamma \rrbracket \xrightarrow{\llbracket \Gamma \vdash m:A \rrbracket} \llbracket A \rrbracket,$$

then $f \sqsubseteq_A m[\overline{v} \mathbin{/} \overline{x}]$.

Proof By induction on the derivation of m. For the (rec) case, one should use Lemmas 5 and 2. □

Definition 11 We shall say that a well-typed term m *terminates*, in symbols $m \Downarrow$, iff there exists a value v, such that $m \Downarrow v$.

The next theorem establishes sufficient conditions for termination at *any* type.

Theorem 11 (Termination) *Let $\emptyset \vdash m : A$ be a well-typed term. If $\llbracket \emptyset \vdash m : A \rrbracket \neq \perp$, then $m \Downarrow$.*

Proof This is a special case of the previous proposition when $\Gamma = \emptyset$. We get $\llbracket \emptyset \vdash m : A \rrbracket \sqsubseteq_A m$, and thus $m \Downarrow$ by definition of $\sqsubseteq_A$. □

We can now finally state our adequacy result.

Theorem 12 (Adequacy) *Let $\emptyset \vdash m : P$ be a well-typed term, where P is a non-linear type. Then:*

$$m \Downarrow \;\; \textit{iff} \;\; \llbracket \emptyset \vdash m : P \rrbracket \neq \perp .$$

Proof The right-to-left direction follows from Theorem 11. The other direction follows from soundness and Lemma 3. □

The model of Theorem 9 was presented as an example by Benton & Wadler (1996) for their LNL calculus extended with recursion, however without stating an adequacy result. We have now shown that it is computationally adequate at non-linear types for our CLNL calculus.

26.6 Conclusion

We begin by briefly reviewing our results. Starting from Benton's LNL calculus (Benton, 1995), we defined the CLNL calculus, which adopts a single combined typing judgement, as opposed to the more traditional approach with separate non-linear and linear typing judgements. Our approach is inspired by the language Proto-Quipper-M and its initial model (Rios & Selinger, 2017), in which the combined typing judgement is meant to be a convenience for the programmer. We showed that these calculi—Benton's LNL and our CLNL—have the same categorical models, and we showed the CLNL calculus can be extended with recursion in the same categorical model, bringing rigor to the discussion in Benton & Wadler (1996). Next, we showed that the CLNL calculus can be extended with language features that turn it into a lambda calculus for string diagrams, which we named the ECLNL calculus (this is essentially Proto-Quipper-M Rios & Selinger (2017)). We then identified the abstract models of ECLNL by considering the categorical enrichment of LNL models. Our abstract approach allowed us to identify concrete models that are simpler than those previously considered, and it also allowed us to extend the language with general recursion, thereby solving an open problem posed by Rios and Selinger. The enrichment structure also made it possible to easily establish the constructivity properties that are expected to hold of a string diagram description language. Finally, we proved an adequacy result for the CLNL calculus, which is the diagram-free fragment of the ECLNL calculus.

There are a number of ingredients in the development of these results that deserve further discussion:

- Categorical quantum mechanics;
- Linear logic and linear/non-linear models;
- Domain theory and support for recursion;
- String diagrammatic languages and calculi.

Our aim is to document Samson's influence on these developments, and on the direction of future research along these lines.

The first item on the list has yet to be mentioned. The work of Abramsky & Coecke (2004) established the beginning of the program on *categorical quantum mechanics* which has played a crucial role in the development of logical methods for quantum information processing and has led to the development of quantum diagrammatic calculi, most notably, the ZX-calculus Bob & Ross (2008), which provide an alternative to the usual linear algebraic language for quantum computing. In this paper, we have also focused on a diagrammatic approach to quantum information processing where our language is used to generate (quantum) circuit diagrams. However, we only take into account the symmetric monoidal structure of the diagrams, whereas a large (and very important) part of categorical quantum mechanics focuses further on the compact closed structure of the diagrams which is a key component in the design of many quantum diagrammatic calculi (e.g. ZX-calculus). It will be interesting to see whether our work can be extended to also take into account the compact closed structure and we return to this point below when we discuss future work.

Soon after its introduction, Girard's linear logic Girard (1987) was hailed as the "logic behind logic" by logicians, and as a means for "controlling the use of resources" by theoretical computer scientists. Samson's seminal article Abramsky (1993) showed how to interpret linear logic from a computational perspective. In those early days, the focus was as much on the classical fragment, where the application was in the concurrent setting, as on the intuitionistic setting, where the, "...refinement of the lambda calculus" led to the linear lambda calculus. Samson showed the classical fragment yields a concurrent process paradigm with an operational semantics in the style of Berry and Boudol's chemical abstract machine Berry & Boudol

(1990). His interpretation of the intuitionistic fragment allowed, "finer control over the order of evaluation and storage allocation, while maintaining the logical content of programs as proofs, interpreting computation as cut-elimination". Samson predicted that the result would, "...open up a promising new approach to the parallel implementation of functional programming languages; and offers the prospect of typed concurrent programming in which correctness is guaranteed by the typing," predictions that have largely succeeded. Moreover, our work in this paper on our linear/non-linear lambda calculus has also been heavily influenced by his work on the linear lambda calculus (Abramsky, 1993).

Samson contrasts the simple, concrete computational interpretation driven by syntax presented in Abramsky (1993) with Girard's geometric view of computation (Girard, 2011, 1995). He then expressed the hope that the results in Abramsky (1993) would lead to connections with Girard's approach. His later work realized this hope: the Geometry of Interaction was employed in Abramsky et al. (2000) to give a game semantics solution to the longstanding full abstraction problem for PCF. In fact, this line of research had a profound effect on our understanding of computation as taking place between two equal partners: a process run by a user on some device, and "the environment", understood as what was happening outside the control of the individual user. While we have avoided the problem of full abstraction in this paper, this is still a very important property that should be considered as part of future work.

Early work on linear logic focused equally on its classical and its intuitionistic fragments, with each fragment having a non-linear analog. Over time, the computer science community focused more narrowly on the intuitionistic fragment, for obvious reasons. Benton & Wadler (1996) were the first to consider LNL models from a programming language perspective. Several authors followed this with models of languages featuring both linear and non-linear types within the same language, but it was Selinger & Valiron (2008) who first used LNL models in the context of quantum computing, albeit as a model for a call-by-value computational linear lambda calculus. That line of research includes the work on the quantum lambda calculus (Selinger & Valiron, 2009) and it runs up to Rios & Selinger (2017), which is a direct inspiration for our work.

One result of both logicians and computer scientists working in the same arena was an inevitable clash of terminology. Among computer scientists working on quantum computing, it is customary to use "classical" to refer to a computer relying on current technology, and "quantum" to refer to a device utilizing quantum systems to perform computations. In the logic community, "classical" is reserved for classical logic, as opposed to, e.g., intuitionistic logic. Likewise, when discussing linear/non-linear models and the associated lambda calculi, one sometimes hears "intuitionistic" used to refer to notions that are not linear. The resulting overloading of terminology can easily cause confusion. Our solution is to reserve "classical" to refer to physical notions from classical physics, and we use "linear" and "non-linear" to distinguish forms of lambda calculi and, likewise, primitives of programming languages. This works, of course, because we have virtually no need to refer to classical logic.

The use of domain theory for giving semantics to lambda calculi that feature both linear and non-linear fragments, and by extension, to programming languages that include both linear and non-linear features has been mainly in the form of category theory, and in particular, in categories enriched over directed complete partial orders. Indeed, little use has been made of continuous domains in this setting, and we doubt that any will emerge because of the eventual need to support monoidal closed categories that also are invariant under the valuations monad. The major impetus for semantic models has been type theory, where support for term recursion and recursive types depends heavily on the work of Freyd on algebraic completeness (Peter, 1991). We have chosen to focus on (ω-) CPOs—partial orders in which increasing sequences have least upper bounds—because these allow simpler arguments (as typified, e.g., in Lehmann & Smyth (1981)). Nevertheless, Samson has been a leader in the adoption of domain theory and category theory and their techniques in modeling logics and programming languages, and his Handbook chapter with Jung (1994) on domain theory is a standard reference for the area. Indeed, we used it many times in the preparation of this paper.

26.7 Future Work

To discuss future work, we first need to mention extensions of this work that already have appeared. First, the papers Lindenhovius et al. (2019, 2020) consider the question of adding type recursion to Proto-Quipper-M (PQM). The approach is based on Fiore (1994) on recursive types in FPC, and it presents an extension of FPC, the fixed point calculus originally due to Plotkin, in which recursive types are introduced to a linear/non-linear lambda calculus. The new metalanguage is called LNL-FPC, to indicate it is an extension of FPC to include linear primitives. The results include soundness and computational adequacy at non-linear types. The metalanguage LNL-FPC represents an extension of the circuit-free fragment of PQM.

A primary issue along this line is to extend the computational adequacy result to also include circuits. The problem has been the basic structure of models for PQM. It starts with a small symmetric monoidal category $\mathbf{M}$ whose morphisms are meant to represent quantum circuits (and perhaps additional morphisms). The next step is to form a certain categorical completion containing $\mathbf{M}$, which is accomplished via an enriched presheaf category. But this presheaf category uses the Day convolution to define the tensor product, and the resulting tensor product is not order reflecting, a key hypothesis for the proof of computational adequacy in Lindenhovius et al. (2019). Recent work offers some promise for overcoming this problem. As reported at a recent POPL workshop Kornell et al. (2020), the first two authors have been collaborating with a colleague on a new model whose linear category has a tensor product that is order reflecting. We believe this may lead to a model in which computational adequacy can be shown to hold for an extension of PQM that includes circuits, and that also supports general recursion.

Going further, recent work by the third author and his colleagues (Péchoux et al., 2020) shows that computational adequacy at all types can be achieved in a model consisting of W*-algebras. Since the model in Kornell et al. (2020) has many similar features to the former one, we are hopeful this result can transfer to that setting.

Returning to the work of Abramsky and Coecke on categorical quantum mechanics (Abramsky & Coecke, 2009), it is very important to consider an extension of our language which can accommodate compact closed structure. This paves the way for new applications of the language with quantum diagrammatic calculi, such as the ZX-calculus (Bob & Ross, 2008). The work in Kornell et al. (2020) seems like a good foundation for building up such models.

Finally, the work in Kornell et al. (2020) has succeeded in many aspects, and we hope that the model can be extended to also support the execution of quantum circuits, and not just their generation (which is the case for the present paper). To accomplish that goal, we first need to incorporate state preparation in the model, and this depends in turn on adding a monad of probability measures/valuations as a computational effect. Our efforts to accomplish that have led to an interesting new result in the realm of "classical" domain theory. In Jia & Mislove (2020) the second author and a colleague have devised a new monad of valuations on DCPOs for which the Fubini Theorem holds. The construction uses Keimel and Lawson's D-completion of the simple valuations (Keimel & Lawson, 2020), and relies on showing this family admits a barycenter map for the algebras of the monad. Our hope is that this monad will show the way for adding state preparation to the model (Kornell et al., 2020).

Acknowledgements We thank Francisco Rios and Peter Selinger for many conversations about their work on Proto-Quipper-M, as well as on our work. We also thank Sam Staton for raising the question of why the families construction is needed in a model of Proto-Quipper-M (it isn't). We also thank Samson Abramsky and Mathys Rennela for valuable conversations about this work during their recent visits to Tulane. We also thank the Simons Institute for the Theory of Computing where much of the initial portion of this work took place. This work was partially funded by the AFOSR under the MURI grant number FA9550-16-1-0082 entitled, "Semantics, Formal Reasoning, and Tool Support for Quantum Programming".

References

Abramsky, S., & Coecke, B. (2004). *A categorical semantics of quantum protocols*. In *19th IEEE symposium on logic in computer science (LICS 2004), 14-17 July 2004, Turku, Finland, proceedings* (pp. 415–425). IEEE Computer Society. https://doi.org/10.1109/LICS.2004.1319636.

Abramsky, S., & Jung, A. (1994). Domain theory. 3. Oxford: Clarendon Press.

Abramsky, S., & Mislove, M. (Eds.) (2012). Mathematical foundations of information flow. In *Proceedings of symposia in applied mathematics* (Vol. 71). Providence, RI: AMS.

Abramsky, S. (1993). Computational interpretations of linear logic. *Theoretical Computer Science, 111,* 3–57.

Abramsky, S., & Coecke, B. (2009). Categorical quantum mechanics. *Handbook of quantum logic and quantum structures, quantum logic* (pp. 261–324). Elsevier.

Abramsky, S., Jagadeesan, R., & Malacaria, P. (2000). Full abstraction for PCF. *Information and Computation, 163,* 409–470.

Benton, P. N. (1995). A mixed linear and non-linear logic: Proofs, terms and models. In *Computer science logic: 8th workshop, CSL '94, selected papers*.

Benton, P. N., & Wadler, P. (1996). Linear logic, monads and the lambda calculus. In *LICS 1996*.

Berry, G., & Boudol, G. (1990). The chemical abstract machine. In *17th ACM symposium on principles of programming semantics* (Vol. 111, pp. 81–94). ACM Press.

Bonchi, F., Sobocinski, P., & Zanasi, F. (2015). Full abstraction for signal flow graphs. In *POPL* (pp. 515–526). ACM.

Borceux, F. (1994). *Handbook of categorical algebra 2: Categories and structures*. Cambridge University Press.

Braüner, T. (1997). A general adequacy result for a linear functional language. *Theoretical Computer Science, 177,* 27–58.

Coecke, B., & Duncan, R. (2008). Interacting quantum observables. In Aceto, L., Damgård, I., Goldberg, L. A., Halldórsson, M. M., Ingólfsdóttir, A. et al. (Eds.), *Automata, languages and programming, 35th international colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part II - Track B: Logic, semantics, and theory of programming & track C: Security and cryptography foundations*, *Lecture notes in computer science* (Vol. 5126, pp. 298–310). Springer. https://doi.org/10.1007/978-3-540-70583-3_25.

Coecke, B., & Duncan, R. (2008). Interacting quantum observables. In *ICALP (2)*, *Lecture notes in computer science* (Vol. 5126, pp. 298–310). Springer.

Egger, J., Møgelberg, R. E., & Simpson, A. (2014). The enriched effect calculus: Syntax and semantics. *Journal of Logic and Computation, 24*(3), 615–654.

Fiore, M. P. (1994). *Axiomatic domain theory in categories of partial maps*. Ph.D. thesis, University of Edinburgh, UK.

Freyd, P. (1991). Algebraically complete categories. In *Category theory* (pp. 95–104). Springer.

Girard, J.-Y. (1995). On geometry of interaction. In H. Schwichtenberg (Ed.), *Proof and computation, NATO ASI series F: Computer and systems sciences* (Vol. 139, pp. 145–191). Springer.

Girard, J.-Y. (1987). Linear logic. *Theoretical Computer Science, 50,* 1–102.

Girard, J.-Y. (2011). Geometry of interaction V: Logic in the hyperfinite factor. *Theoretical Computer Science, 412*(20), 1860–1883.

Green, A. S., Lumsdaine, P. L., Ross, N. J., Selinger, P., & Valiron, B. (2013). Quipper: A scalable quantum programming language. In: *PLDI* (pp. 333–342). ACM.

Hadzihasanovic, A. (2015). A diagrammatic axiomatisation for qubit entanglement. In *LICS* (pp. 573–584). IEEE Computer Society.

Jia, X., Mislove, M. (2020). *Completing Simple Valuations in K-categories*. https://arxiv.org/abs/2002.01865.

Keimel, K., & Lawson, J. D. (2020). D-completions and the d-topology. *Annals of Pure and Applied Logic,159*, 292–306. https://doi.org/10.1016/j.apal.2008.06.019.

Kornell, A., Lindenhovius, B., & Mislove, M. (2020). *Quantum CPOs*. https://popl20.sigplan.org/home/planqc-2020.

Lehmann, D. J., & Smyth, M. B. (1981). *Algebraic specification of data types: A synthetic approach*. *Mathematical systems theory*.

Lindenhovius, B., Mislove, M., & Zamdzhiev, V. (2018). Enriching a linear/non-linear lambda calculus: A programming language for string diagrams. *Proceedings of the 33rd annual ACM/IEEE symposium on logic in computer science,* (pp. 659–668). https://doi.org/10.1145/3209108.3209196.

Lindenhovius, B., Mislove, M.,& Zamdzhiev, V. (2019). Mixed linear and non-linear recursive types. In *Proceedings of the ACM on programming languages* 3 Issue ICFP. New York, NY, USA: ACM.

Lindenhovius, B., Mislove, M. W.,& Zamdzhiev, V. (2020). LNL-FPC: The linear/non-linear fixpoint calculus. Accepted subject to minor revisions for the journal LMCS (Logical Methods in Computer Science). http://arxiv.org/abs/1906.09503.

Lucyshyn-Wright, R. B. B. (2016). Relative symmetric monoidal closed categories I: Autoenrichment and change of base. *Theory and applications of categories*.

Meseguer, J., & Montanari, U. (1988). Petri nets are monoids: A new algebraic foundation for net theory. In *LICS* (pp. 155–164). IEEE Computer Society.

Paykin, J., Rand, R., & Zdancewic, S. (2017). QWIRE: A core language for quantum circuits. In *POPL* (pp. 846–858). ACM.

Péchoux, R., Perdrix, S., Rennela, M., & Zamdzhiev, V. (2020): Quantum programming with inductive datatypes: Causality and affine type theory. In *Proceedings of FOSSACS 2020* to appear.

Perdrix, S., Jeandel, E., & Vilmart, R. (2017). *A complete axiomatisation of the ZX-calculus for clifford+t quantum mechanics*.

Plotkin, G. D. (1985): *Lectures on predomains and partial functions. Notes for a course given at CSLI Stanford University.*

Rennela, M., & Staton, S. (2017). *Classical control and quantum circuits in enriched category theory*. To appear in MFPS XXXIII.

Rennela, M., & Staton, S. (2017). *Classical Control, Quantum Circuits and Linear Logic in Enriched Category Theory*.

Rios, F., & Selinger, P. (2017). *A categorical model for a quantum circuit description language*. To appear in QPL 2017.

Selinger, P. (2011). A survey of graphical languages for monoidal categories. *New Structures for Physics*.

Selinger, P., & Valiron, B. (2008). A linear-non-linear model for a computational call-by-value lambda calculus. In *Proceedings of the eleventh international conference on foundations of software science and computation structures (FOSSACS 2008)*, *Lecture notes in computer science* (Vol. 4962, pp. 81–96). Springer.

Selinger, P., & Valiron, B. (2009). Quantum lambda calculus. In *Semantic techniques in quantum computation* (pp. 135–172). Cambridge University Press.

Sobocinski, P., & Stephens, O. (2014). A programming language for spatial distribution of net systems. In Petri Nets.

Stephens, O. (2015). *Compositional specification and reachability checking of net systems*. Ph.D. thesis, University of Southampton, UK.

Thomas, D., & Moorby, P. (2008). *The verilog hardware description language*. Springer Science & Business Media.

Zainalabedin, N. (1997). *VHDL: Analysis and modeling of digital systems*. McGraw-Hill, Inc.

Chapter 27
Retracing Some Paths in Categorical Semantics: From Process-Propositions-as-Types to Categorified Reals and Computers

Dusko Pavlovic

Abstract The logical parallelism of propositional connectives and type constructors extends beyond the static realm of predicates, to the dynamic realm of processes. Understanding the logical parallelism of *process* propositions and *dynamic* types was one of the central problems of the semantics of computation, albeit not always clear or explicit. It sprung into clarity through the early work of Samson Abramsky, where the central ideas of denotational semantics and process calculus were brought together and analyzed by categorical tools, e.g. in the structure of *interaction categories*. While some logical structures borne of dynamics of computation immediately started to emerge, others had to wait, be it because the underlying logical principles (mainly those arising from coinduction) were not yet sufficiently well-understood, or simply because the research community was more interested in other semantical tasks. Looking back, it seems that the process logic uncovered by those early semantical efforts might still be starting to emerge and that the vast field of results that have been obtained in the meantime might be a valley on a tip of an iceberg. In the present paper, I try to provide a logical overview of the gamut of interaction categories and to distinguish those that model computation from those that capture processes in general. The main coinductive constructions turn out to be of the latter kind, as illustrated towards the end of the paper by a compact category of all real numbers as processes, computable and uncomputable, with polarized bisimulations as morphisms. The operation of addition of the reals arises as the biproduct, real vector spaces are the enriched bicompletions, and linear algebra arises from the enriched kan extensions. At the final step, I sketch a structure that characterizes the computable fragment of categorical semantics.

Supported by NSF and AFOSR.

D. Pavlovic (✉)
University of Hawaii, Honolulu, Hi, USA
e-mail: dusko@hawaii.edu

A. Palmigiano and M. Sadrzadeh (eds.), *Samson Abramsky on Logic and Structure in Computer Science and Beyond,* Outstanding Contributions to Logic 25,
https://doi.org/10.1007/978-3-031-24117-8_27

Keywords Interaction categories · Propositions-as-types · Semantics of computation · Coalgebra · Coinduction · Process logics, Categorical semantics · Dynamic relations · Computability

Personal Introduction

I first learned about Samson Abramsky's work from his invited plenary lecture at the International Category Theory Meeting in Montreal in 1991. It was the golden age of category theory, and Montreal was at the heart of it, and I got to be a postdoc there. Just a few years earlier, I was a dropout freelance programmer, but had become a mathematician, and was uninterested in computers. I was told, however, that Abramsky had constructed some categories that no one had seen before, so I came to listen to his talk. I also had a talk to give later that day myself, but for some reason, I do not recall how that went. At the end of Abramsky's plenary lecture, Saunders Mac Lane stood up, one of the two fathers of category theory, high up near the ceiling of the amphitheater, and spoke for a long time. He criticized computer science in general. After that, Bill Lawvere stood up, and provided some friendly comments, suggesting directions for progress and improvement.

Two years later, I became an "*EU Human Capital Mobility*"fellow within the Theory Group at the Imperial College in London, led by Samson Abramsky. I started learning computer science and spent a lot of time trying to understand Samson's *interaction categories* (Abramsky, 1993). In the meantime, he had constructed more categories that no one had seen before. My fellowship ended after a year or two, and the human capital mobility turned out to be much greater than anyone could imagine, but I continued to think about interaction categories for years. Here I try to summarize some of that thinking.

27.1 Introduction: On Categorical Logics and Propositions-as-Types

The category of sets or types. This is a paper about categorical semantics. It is written for a collection intended for logicians. If you are reading this, then you are presumed to be interested in categorical logic, although you may not be interested in categories in general. To ease this tension, I will avoid abstract categories, and mostly stick with the category **S** of sets and functions. It is presented, however, as a universe of types, by specifying which type constructors are used in each construction. Initially, we just need the cartesian products, but later we need more. The naive set theory used to be presented incrementally. Nowadays most mathematicians think of types as sets, and most programmers think of sets as types, so it seems reasonable for logicians and computer scientists to identify the two. To keep the naive-set-theory flavor, we usually call the type inhabitants *elements*, where type theorists use the term *terms*.

When a set is constructed as a type, then it can be construed as a proposition: since its elements are constructions, they can be viewed as proofs (Martin-Löf, 1984). Such interpretations originate from logic, where the idea of propositions-as-types was first encountered along the paths of proofs-as-constructions (Church, 1940; Kolmogoroff, 1932; Martin-Löf, 1975). We retrace these paths first, and proceed throughout with propositions-as-types, types-as-sets, terms-as-elements, elements-as-morphisms (Lambek, 1980; Lawvere, 1969).

Naming names. While sets and types signal different approaches, many concepts are studied in different communities under different names even if there are no significant differences. This is useful to place narratives in their contexts or to authenticate speakers' allegiances. It is not easy to avoid such connotations when they are undesired. In some cases, I resorted to renaming. E.g., the *histories* from Sect. 27.2.3.1 onwards are known as non-empty lists, or words, or strings. There are other examples. I am not trying to reinvent them but to dissociate them from narrow contexts. I hope they will not be too distracting.

27.1.1 Logics of Types

Bertrand Russell proposed his *ramified theory of types* (Russell, 1908) as a logical framework for paradox prevention. Alonzo Church and Stephen Kleene advanced type theory into a model of computation (Church, 1940; Kleene, 1935). Dana Scott adopted type theory as the foundation for a mathematical approach to the semantics of computation (Scott, 1993). The semantics of programming languages were built steadily upon that foundation (Gunter, 1992; Reynolds, 1998). Process semantics also arose from that foundation (Milner, 1975), but had to undergo a substantive conceptual evolution before the types could capture dynamics. I followed these developments through Samson Abramsky's work.

The propositions-as-types paradigm was discovered many times. In logic and computer science, it is attributed to Haskell Curry and William Howard (Seldin & Hindley, 1980; Girard et al., 1989b, Chap. 3). Howard got the idea from Georg Kreisel (Wadler, 2015), and Kreisel's goal was to formalize Brouwer's concept of proofs-as-constructions (Kreisel, 1968). An early formalization of Brouwer's concept goes back to Kolmogorov (1932).

The structural reason why propositions and types obey analogous laws was offered by Lawvere (1969), who pointed out that the propositional and the typing rules are instances of analogous categorical *adjunctions*; and that the proof constructions and the term derivations arise from the adjunction units and counits. This gave rise to the idea of categorical proof theory, pursued by Lambek (1968a, 1969, 1972), and to the basic structures of categorical semantics, succinctly described in Lambek and Scott

(1986, and the references therein). In the preface to his seminal report Scott (1993), Dana Scott explained that

> a category represents the 'algebra of types', just as abstract rings give us the algebra of polynomials, originally understood to concern only integers or rationals. One can of course think only of particular type systems, but, for a full understanding, one needs also to take into account the general theory of types, and especially *translations* or *interpretations* of one system in another.

Samson Abramsky spearheaded the efforts towards expanding the categorical semantics of program abstraction, as formalized in type theory and merge it with a categorical semantics of process abstraction and interaction, as formalized in the theory of concurrency and process calculi. This led to interaction categories (Abramsky, 1993; Cockett and Spooner, 1995; Pavlovic, 1995, 1996), specification structures (Abramsky et al., 1995; Pavlovic and Abramsky, 1997b), and a step further to geometry of interaction (Abramsky and Jagadeesan, 1994) and game semantics (Abramsky, 1997; Abramsky & Jagadeesan, 1992; Abramsky et al., 1994, 2000, and many other publications). As the realm of program abstraction expanded, e.g. into quantum computation and protocols (Abramsky and Coecke, 2004), the semantical apparatus also expanded (Abramsky, 2010, 2012), the tree branched (Coecke and Pavlovic, 2007; Pavlovic, 2012), some branches crossed.[1] In the present paper, however, we are only concerned with the root. And even that might be overly ambitious.

27.1.2 Categorical Proof Theory

Proofs-as-constructions. The Curry-Howard isomorphism is one of the conceptual building blocks of type theory, built deep into the foundation of computer science and functional programming (Girard et al., 1989b, Chap. 3). The fact that it is an *isomorphism* means that the type constructors on one side obey the same laws as the propositional connectives on the other side; and these laws are expressed as a bijection between the terms and the proofs.

27.1.2.1 Entailments as Morphisms

In categorical proof theory, logical sequents are treated as arrows in a category (Lambek, 1968a, 1969, 1972; Lawvere, 1969). The reflexivity and the transitivity of the entailment relation then correspond to the main categorical structures: the identities and the composition.

[1] E.g., Pavlovic (2011) used the methods of Pavlovic and Abramsky (1997b) to expand the models of Abramsky and Coecke (2004).

$$\frac{}{A \vdash A} \qquad \begin{tikzcd} 1 \arrow[d, "\ulcorner \mathrm{id} \urcorner"] \\ \mathbf{S}(A, A) \end{tikzcd} \tag{27.1}$$

$$\frac{A \vdash B \qquad B \vdash C}{A \vdash C} \qquad \begin{tikzcd} \mathbf{S}(A, B) \times \mathbf{S}(B, C) \arrow[d, "(-;-)"] \\ \mathbf{S}(A, C) \end{tikzcd}$$

But while there is at most one sequent $A \vdash B$ for given A and B, there can be many arrows between A and B in a category. Categorical semantics of the logical entailment must therefore be imposed by equations:

$$\begin{tikzcd} & \mathbf{S}(A, B) \arrow[dl, "\langle \ulcorner \mathrm{id} \urcorner, \mathrm{id} \rangle"'] \arrow[dd, "\mathrm{id}"] \\ \mathbf{S}(A, A) \times \mathbf{S}(A, B) \arrow[dr, "(-;-)"'] & \\ & \mathbf{S}(A, B) \end{tikzcd} \qquad \begin{tikzcd} \mathbf{S}(A, B) \arrow[dr, "\langle \mathrm{id}, \ulcorner \mathrm{id} \urcorner \rangle"] \arrow[dd, "\mathrm{id}"'] & \\ & \mathbf{S}(A, B) \times \mathbf{S}(B, B) \arrow[dl, "(-;-)"] \\ \mathbf{S}(A, B) & \end{tikzcd}$$

$$\begin{tikzcd} \mathbf{S}(A, B) \times \mathbf{S}(B, C) \times \mathbf{S}(C, D) \arrow[r, "\mathrm{id} \times (-;-)"] \arrow[d, "(-;-) \times \mathrm{id}"'] & \mathbf{S}(A, B) \times \mathbf{S}(B, D) \arrow[d, "(-;-)"] \\ \mathbf{S}(A, C) \times \mathbf{S}(C, D) \arrow[r, "(-;-)"] & \mathbf{S}(A, D) \end{tikzcd}$$

27.1.2.2 Conjunction and Disjunction as Product and Coproduct

Algebraically, the conjunction and the disjunction are the meet and the join in the proposition lattice. Categorically, they are the product and the coproduct:

$$\frac{X \vdash A \qquad X \vdash B}{X \vdash A \wedge B} \qquad \begin{tikzcd} \mathbf{S}(X, A) \times \mathbf{S}(X, B) \arrow[d, shift right, "\langle -, - \rangle"'] \\ \mathbf{S}(X, A \times B) \arrow[u, shift right, "\langle \pi_A \circ -, \pi_B \circ - \rangle"'] \end{tikzcd} \tag{27.2}$$

$$\frac{A \vdash X \qquad B \vdash X}{\overline{\overline{A \vee B \vdash X}}} \qquad \begin{array}{c} \mathbf{S}(A, X) \times \mathbf{S}(B, X) \\ \left[-,-\right] \big\downarrow \ \big\uparrow \left\langle - \circ \iota_A, - \circ \iota_B \right\rangle \\ \mathbf{S}(A + B, X) \end{array} \tag{27.3}$$

Definition 27.1.1 A category with the product constructor $\times$ supporting the correspondence (27.2) is called *cartesian*. A category with the coproduct constructor $+$supporting the correspondence (27.3) is called *cocartesian*.

The difference between the algebraic and the categorical view is that in the first case there is at most one entailment $X \vdash A$, whereas in the second case there can be many arrows $X \to A$, usually labelled, and viewed as functions in the category $\mathbf{S}$. The mapping in (27.2) on the right establishes the bijection between the proofs or functions $X \to A \times B$ and the pairs of proofs or functions $X \to A$ and $X \to B$. The proof transformations thus become function manipulations. If the elements of sets or entries of data types, witness the corresponding propositions, then the logical operations are operations on data. E.g., proof of conjunction becomes a pair of data entries. It often comes as a surprise that such simple-minded analogies can be effective tools in functional programming (Pavlovic, 2020). They also have far-reaching logical consequences, some of which are pursued in the present paper.

27.1.2.3 Function Abstraction and Cartesian Closed Categories

The fact that the conjunction $A \wedge (-)$ is the right adjoint to the implication $A \supset (-)$ (Lawvere, 1969) means that the implication introduction and elimination can be expressed as the reversible logical rule in (27.4) on the left.

$$\frac{(A \wedge X) \vdash B}{\overline{\overline{X \vdash (A \supset B)}}} \supset \qquad \begin{array}{c} \mathbf{S}(A \times X, B) \\ (A \Rightarrow -) \circ \eta_X \big\downarrow \ \big\uparrow \varepsilon_X \circ (A \times -) \\ \mathbf{S}\big(X, (A \Rightarrow B)\big) \end{array} \tag{27.4}$$

The corresponding type-theoretic correspondence in (27.4) on the right was the first example of the propositions-as-types phenomenon. This bijection between two sets of proofs-as-terms was noticed by Haskell Curry back in the 1930s.s. The operation corresponding to the implication introduction, i.e. going down, is called the *abstraction*. The operation corresponding to the implication elimination, i.e. going up, is called the *application*. The categorical view of the resulting correspondence captures its uniformity with respect to all indexing types X, i.e. its *polymorphism*, as the *naturality* with respect to the type constructors $(A \times -)$ and $(A \Rightarrow -)$. A correspondence between two constructors is natural if it is preserved under all substitutions. For (27.4) every $f \in \mathbf{S}(X, Y)$ induces the two squares in (27.5), one formed by ηs, the other by εs.

$$\begin{tikzcd} \mathbf{S}(A \times Y, B) \arrow[r, "-\circ(A\times f)"] \arrow[d, bend right, "(A\Rightarrow -)\circ\eta_Y"'] & \mathbf{S}(A \times X, B) \arrow[d, bend right, "(A\Rightarrow -)\circ\eta_X"'] \\ \mathbf{S}\big(Y, (A\Rightarrow B)\big) \arrow[r, "-\circ f"] \arrow[u, bend right, "\varepsilon_Y\circ(A\times -)"'] & \mathbf{S}\big(X, (A\Rightarrow B)\big) \arrow[u, bend right, "\varepsilon_X\circ(A\times -)"'] \end{tikzcd} \tag{27.5}$$

The naturality of these squares means that they are commutative. The commutativity of these squares captures the type-theoretic *conversion rules* imposed on the abstraction operation and the application operation:

$$\begin{tikzcd} & A \times (A\Rightarrow(A\times X)) \arrow[dr, "\varepsilon_{A\times X}"] & \\ A\times X \arrow[ur, "A\times\eta_X"] \arrow[rr, "\mathrm{id}"] & & A\times X \end{tikzcd} \qquad \Big(\lambda a.\ f_x(a)\Big)\cdot b = f_x(b) \tag{β}$$

$$\begin{tikzcd} A \Rightarrow X \arrow[rr, "\mathrm{id}"] \arrow[dr, "\eta_{(A\Rightarrow X)}"'] & & A\Rightarrow X \\ & A \Rightarrow (A\times(A\Rightarrow X)) \arrow[ur, "(A\Rightarrow\varepsilon_X)"'] & \end{tikzcd} \qquad \lambda a.\ \Big(g_x\cdot a\Big) = g_x \tag{η}$$

The application operation is derived from the adjunction counit $\varepsilon_X : A\times(A\Rightarrow X)\longrightarrow X$ in the form $g\cdot a = \varepsilon(a, g)$. The abstraction operation is written in type theory it is written using the variables, like in the rules (β) and (η) above, but can also derived from the adjunction unit $\eta_X : X \longrightarrow (A\Rightarrow(A\times X))$, in the form $\lambda(f) = (A\Rightarrow f)\circ\eta_X$.

Definition 27.1.2 A *cartesian closed* category is a cartesian category $\mathbf{S}$ with the static implication $(A\Rightarrow B)$ for every pair of types A, B and the X-natural (function) abstraction operation

$$\mathbf{S}(A\times X, B) \xrightarrow[\sim]{\lambda_X^{AB}} \mathbf{S}(X, (A\Rightarrow B)) \tag{27.6}$$

Remark. Towards aligning with Definition 27.2.1, note that the function abstraction λ_{AB} is with respect to the functors $H_{AB}, \nabla_{AB} : \mathbf{S}^o \longrightarrow \mathbf{S}$ defined

$$H_{AB}X = \mathbf{S}(A\times X, B) \qquad\qquad \nabla_{AB}X = \mathbf{S}(X, (A\Rightarrow B))$$

The arrow parts are induced by precomposition.

27.1.3 Modalities as Monads and Comonads

27.1.3.1 Possibility and Side-Effects

A possibility modality can be introduced by the rules on the left.

$$\frac{}{A \vdash \Diamond A} \qquad \frac{A \wedge B \vdash \Diamond C}{\Diamond A \wedge \Diamond B \vdash \Diamond C} \qquad \begin{tikzcd} \mathbf{S}(A \times B, MC) \arrow[d, bend right, "\#\phi"'] \\ \mathbf{S}(MA \times MB, MC) \arrow[u, bend right, "\eta\eta"'] \end{tikzcd} \tag{27.7}$$

Each of the logical rules corresponds to one of the categorical transformations on the right, where $\eta\eta$ precomposes $A \times B \xrightarrow{\eta_A \times \eta_B} MA \times MB \longrightarrow MC$, whereas $\#\phi$ first #-lifts $A \times B \xrightarrow{g} MC$, and then precomposes to $MA \times MB \xrightarrow{\phi} M(A \times B) \xrightarrow{g^\#} MC$. The quadruple $(M, \eta, \#, \phi)$ is the structure of a *commutative monad* (Barr and Wells, 2005; Kock, 1971, 2012; Manes, 1976). The type constructor M, the unit $\eta : A \longrightarrow MA$ and the lifting # of $A \xrightarrow{f} MC$ to $MA \xrightarrow{f^\#} MB$ satisfy the equations

$$\eta_A^\# = \mathrm{id}_{MA} \qquad f^\# \circ \eta_{A \times B} = f \qquad (f^\# \circ t)^\# = f^\# \circ t^\# \tag{27.8}$$

This triple is one of the equivalent presentations of the structure of a monad (Manes, 1976). Most presentations (Barr and Wells, 2005; Lambek and Scott, 1986) define a monad as a triple (M, η, μ), where $\mu_A : MMA \longrightarrow MA$ are the (cochain) evaluators. The lifting # is derivable from the evaluators by setting $f^\# = \left(MA \xrightarrow{Mf} MMB \xrightarrow{\mu} MB\right)$, whereas the evaluators are derivable as the liftings of the identities, in the form $\mu_A = \left(MMA \xrightarrow{\mathrm{id}^\#} MA\right)$. The lifting operation # seems more convenient for programming. The last component ϕ of the structure $(M, \eta, \#, \phi)$ (or of the equivalent form (M, η, μ, ϕ)) is the *bilinearity* $\phi_{AB} : MA \times MB \longrightarrow M(A \times B)$, which makes the monad commutative (Kock, 1971, 2012). This natural family satisfies

$$\phi_{AC} \circ (\eta_A \times \eta_C) = \eta_{A \times C} \qquad \left(\phi_{BD} \circ (f \times g)\right)^\# \circ \phi_{AC} = \phi_{BD} \circ (f^\# \times g^\#) \tag{27.9}$$

for all pairs $f : A \longrightarrow MB$ and $g : C \longrightarrow MD$. Similar equations are valid for all tuples. The equations in (27.8) define the identities and the composition in the category of free algebras (in the Kleisli form)

$$|\mathbf{S}_M| = |\mathbf{S}|$$
$$\mathbf{S}_M(A, B) = \mathbf{S}(A, MB)$$

While correspondences (27.3) persist in $\mathbf{S}_M$, the natural bijection in (27.2) does not, and $\mathbf{S}_M$ is not a cartesian category any more. However, the equations in (27.9) assure that the product $\times$ from $\mathbf{S}$ persists as a monoidal structure in $\mathbf{S}_M$ (Kock, 2012). Intuitively, a function $A \longrightarrow MB$ produces not just the outputs in B, but also some *side-effects* (Milner, 1975), represented in the type MB. E.g., the fact that computations may not terminate means that they implement functions in the form $A \longrightarrow B_\perp$ where the monad

$$\begin{aligned} (-)_\perp : \mathbf{S} &\longrightarrow \mathbf{S} \\ X &\mapsto X \cup \{\perp\} \end{aligned} \tag{27.10}$$

adjoins to each set a fresh element $\perp$. This is the *maybe* monad, corresponding to the algebraic theory with a single constant and no operations or equations. The category $\mathbf{S}_\perp$ is (equivalent to) the category of sets and partial functions.

Another side-effect of interest is the *nondeterminism*. Some computations may depend on the states of the computer, which may depend on the environment. Running the same program on the same inputs may therefore produce different outputs at different times, for no unobservable reason. Such computations implement functions in the form $A \longrightarrow \wp B$, where the monad

$$\begin{aligned} \wp : \mathbf{S} &\longrightarrow \mathbf{S} \\ X &\mapsto \{V \subseteq X\} \end{aligned} \tag{27.11}$$

maps each set to the set of its subsets, a.k.a. its *powerset*. This is the *nondeterminism* (or powerset) monad. It maps to every function $X \xrightarrow{g} Y$ the function $\wp X \xrightarrow{\wp g} \wp Y$, which takes $V \subseteq X$ to $\wp g(V) = \{g(x) \in Y \mid x \in V\}$. The unit $X \xrightarrow{\eta} \wp X$ maps $x \in X$ to $\eta(x) = \{x\}$. The lifting maps a function $A \xrightarrow{f} \wp B$ to $\wp A \xrightarrow{f^\#} \wp B$ where $f^\#(V) = \cup_{v \in V} f(v)$. For reasons discussed in Appendix A, it satisfies

$$\mathbf{S}(A, \wp B) \cong \mathbf{S}(B, \wp A)$$

which makes the category $\mathbf{S}_\wp$ of nondeterminisic functions self-dual, along the natural bijection $\mathbf{S}_\wp(A, B) \cong \mathbf{S}_\wp(B, A)$. The idea is that, given a nondeterministic function $A \longrightarrow \wp B$, knowing all possible B-outputs for each A-input allows us to extract all possible A-inputs for each B-output, which yilelds just another nondeterministic function $B \longrightarrow \wp A$. See Appendix A for more.

Notation. Since they will be cast in leading roles, the above categories of functions with effects are abbreviated to:

- $\mathbf{P} = \mathbf{S}_\perp$ — category of partial functions, and
- $\mathbf{R} = \mathbf{S}_\wp$ — category of relations.

Background. In mathematics, monads emerged as a "standard construction" of free algebras involving topologies (Barr & Wells, 2005; Manes, 1976). The observation that the type constructors M that add side-effects also carry the monad structure goes back to Moggi (1991). Initially proposed as a semantical tool, monads turned out to be a powerful and convenient programming tool. Nowadays, monads' popularity among programmers drives interest in semantics. Mathematically, a monad M is a saturated view of an algebraic theory, presented not by operations and equations, but as a mapping from any set of generators B to the free algebra MB. The unit η maps each generator to its place in the free algebra. The lifting # expands the assignment $A \xrightarrow{f} MB$ from the generators A to the algebra homomorpism $MA \xrightarrow{f^\#} MB$. Any monad corresponds to an algebraic theory, albeit with infinitary operations. The semantical assumption that all computational side-effects can be captured by algebraic operations has deep repercussions on the concept of computation.

27.1.3.2 Necessity and Reductions

Dually, a necessity modality can be introduced by

$$\frac{}{\Box A \vdash A} \qquad \frac{\Box A \vdash B \vee C}{\Box A \vdash \Box B \vee \Box C} \qquad \begin{array}{c} \mathbf{S}(GA, B + C) \\ {}^{\#}\big\downarrow \ \big\uparrow{}^{-\circ \varepsilon_{B+C}} \\ \mathbf{S}(GA, GB + GC) \end{array} \tag{27.12}$$

This time the triple $(G, \varepsilon, \#)$ is made into a comonad by the equations:

$$\varepsilon_A^\# = \mathrm{id}_{GA} \qquad \varepsilon_{B+C} \circ f^\# = f \qquad (f \circ t^\#)^\# = f^\# \circ t^\#$$

The third equation defines the composition in the category of coffee coalgebras, in the *Kleisli* form again:

$$|\mathbf{S}_G| = |\mathbf{S}|$$
$$\mathbf{S}_G(A, B) = \mathbf{S}(GA, B)$$

Computational interpretations of comonads are less standard, but overviews can be found in Brookes and Geva (1992), Uustalu and Vene (2008). We will need a *history comonad* to capture the time extension of processes in Sect. 27.2.3.1. For the moment, let us just mention the *indexing* comonads

$$\begin{aligned} A \times (-) : \mathbf{S} &\longrightarrow \mathbf{S} \\ X &\mapsto A \times X \end{aligned} \tag{27.13}$$

which exist for each $A \in \mathbf{S}$, with the counits $A \times X \xrightarrow{\varepsilon} X$ realized by the projections, and the lifting $A \times X \xrightarrow{h} Y + Z$ defined to be $A \times X \xrightarrow{<\mathrm{id}_A, h>} A \times (Y + Z) \cong (A \times Y) + (A \times Z)$. The Kleisli category $\mathbf{S}_{A\times}$ freely adjoins an indeterminate arrow $1 \xrightarrow{x} A$ to $\mathbf{S}$, and plays the role of the polynomial extension $\mathbf{S}[x : A]$ (Lambek and Scott, 1986; Pavlovic, 1997a). Like any Kleisli category, $\mathbf{S}_{A\times}$ provides a *resolution* of its comonad, in the sense that it factors through the functors

$$A \times (-) = \left(\mathbf{S} \xrightarrow{-\circ\varepsilon} \mathbf{S}_{A\times} \xrightarrow{\#} \mathbf{S} \right)$$

as displayed in (27.12). While the Kleisli resolution is *initial* among the resolutions of the comonad $A \times (-)$, some of the constructions in this paper are built upon the fact that the resolution

$$A \times (-) = \left(\mathbf{S} \xrightarrow{\Pi_A} \mathbf{S}/A \xrightarrow{\mathrm{Dom}} \mathbf{S} \right)$$

is *final* among all resolutions. Here $\mathbf{S}/A$ is the category of $\mathbf{S}$-morphisms into A, the functor Π_A functor maps X to the projection $A \times X \xrightarrow{\pi_A} A$, whereas the Dom functor Dom takes the $\mathbf{S}/A$-objects, which are the $\mathbf{S}$-morphisms with the codomain A, to their domains $\mathrm{Dom}(X \longrightarrow A) = X$.

Lemma 27.1.3 *The domain functor* Dom : $\mathbf{S}/A \longrightarrow \mathbf{S}$ *is final among all functors* $F : \mathbf{C} \longrightarrow \mathbf{S}$ *which map the terminal object* 1 *into* A.

$$\begin{tikzcd} & \mathbf{C} \arrow[dl, "\forall F"'] \arrow[dd, dashed, "\exists! F'"] \\ \mathbf{S} & \\ & \mathbf{S}/A \arrow[ul, "\mathrm{Dom}"] \end{tikzcd}$$

Proof Given F with $F1 = A$, the unique F' with $\mathrm{Dom} \circ F' = F$ is $F'X = F(X \xrightarrow{!} 1)$. □

27.1.4 *Labelled Sequents, Commutative Monads, and Surjections*

In propositional logic, a sequent $X \vdash Y$ transforms proofs of X into proofs of Y. If there are several different ways to derive one from the other, the sequent $X \vdash Y$ identifies them all. This leads to a mismatch within the propositions-as-types interpretation because it implies that there is at most one proof $X \vdash A \supset B$, while there can be many different terms of type $X \xrightarrow{t} (A \Rightarrow B)$. This mismatch is resolved by labelling the sequents, writing $X \overset{t}{\mapsto} A \supset B$ for the former sequent. We use the symbol $\mapsto$ (and not $\vdash$) for labelled sequents, to be able to write $X \mapsto Y$ instead of $X \overset{f}{\mapsto} Y$ when the label f is irrelevant. The categorical proof theory originates from studies of labelled sequents in Lambek (1968a, 1969, 1972). A non-categorical theory of labelled sequents was developed in Gabbay (1996).

For a modality $\Diamond$, the sequents between the propositions $\Diamond A \wedge \Diamond B$ and $\Diamond(A \wedge B)$ are derivable both ways, and the two are considered equivalent. The proposition $\Diamond\top$

is also equivalent to the truth $\top$. For a monad M, the maps $M(A \times B) \xrightarrow{<M\pi_A, M\pi_B>} MA \times MB$ and $M1 \xrightarrow{!} 1$ are derivable from the cartesian structure, and the maps $MA \times MB \xrightarrow{\phi} M(A \times B)$ and $1 \xrightarrow{\eta} M1$ are given by the monad structure. However, these pairs of functions both ways are generally not inverse to one another. The type $M1$ is generally not final, and the type $M(A \times B)$ is generally not a product. The side-effects of type $M(A \times B)$ are different from the side-effects that arise when the outputs are received into MA and MB separately.

While this state of affairs is justified in algebra, where $M1$ is the free algebra over a single generator, it is not justified in the semantics of computation, where the trivial outputs of type 1 should not cause nontrivial side effects contained in type $M1$. Viewing the monad M as an algebraic theory shows that the nontrivial elements of $M1$ arise from the constants of the algebraic theory. This requirement is not satisfied either by the maybe monad, or by the nondeterminism monad, as the former gives the universe $\mathbf{P} = \mathbf{S}_\perp$ of sets and partial maps, the latter the universe $\mathbf{R} = \mathbf{S}_\wp$ of sets and relations. The former is the category of free algebras for the theory with a single constant $\perp$, and no other operations. The latter is the category of free join semilattices, where the lattice unit is a constant again.

Lemma 27.1.3 says that making 1 into the unit type (final object) in $\mathbf{R} = \mathbf{S}_\wp$ leads to the slice category $\mathbf{tR} = \mathbf{R}/1$, which boils down to

$$\begin{aligned}
|\mathbf{tR}| &= \coprod_{A\in|\mathbf{S}|} \wp A \\
\mathbf{tR}(S_{\subseteq A}, T_{\subseteq B}) &= \Big\{ R \in \mathbf{R}(A,B) \mid (x \in S \iff \exists y \in T.\, xRy) \ \wedge \\
&\qquad \wedge\ (y \in T \iff \exists x \in S.\, xRy) \Big\}
\end{aligned} \tag{27.14}$$

Since the $\Rightarrow$-direction of each of the conjuncts in (27.14) implies the $\Leftarrow$-direction of the other conjunct, the requirement boils down to $\forall x \in S \exists y \in T.\, xRy$ and $\forall y \in T \exists x \in S.\, xRy$. The category $\mathbf{tR}$ is thus equivalent to the subcategory of $\mathbf{R}$ comprised of the relations that are total in both directions. Proceeding in a similar way to make 1 into the final type in the category $\mathbf{S}_\perp = \mathbf{P}$ leads to the slice category $\mathbf{tP} = \mathbf{P}/1$, which is equivalent to the subcategory $\mathbf{tS}$ of $\mathbf{S}$ spanned by the surjective functions:

$$\begin{aligned}
|\mathbf{tP}| &= \coprod_{A\in|\mathbf{S}|} \wp A \\
\mathbf{tP}(S_{\subseteq A}, T_{\subseteq B}) &= \Big\{ f \in \mathbf{S}(S,T) \mid y \in T \Rightarrow \exists x \in S.\, f(x) = y \Big\}
\end{aligned} \tag{27.15}$$

Remark for the category theorist. The forgetful functor $\mathbf{tP} \longrightarrow \mathbf{tS}$, where $\mathbf{tS}$ is the category of sets and surjections, is an equivalence because it is surjective on the objects, and full and faithful on the morphisms. However, for each set $S \in \mathbf{S}$ there is a proper class of sets A such that $S_{\subseteq A} \in \mathbf{tP}$ is mapped to $S \in \mathbf{tS}$. Constructing the adjoint equivalence $\mathbf{tS} \longrightarrow \mathbf{tP}$ thus involves a choice from these proper classes of objects.

27.2 Process Logics

27.2.1 Idea of Process

The alignment of logic and type theory remains stable as long as the world is stable: the true propositions remain true, and the data types remain as given. The problems arise when processes need to be modeled, and their dynamic aspects need to be taken into account.

There are physical processes, chemical processes, mental processes, social processes. The common denominator is that they change state: a physical process changes the state of matter; a mental process changes the state of mind. Computation is also a process. Although already a local execution of a program changes the local states of a computer, it seems that the crucial aspects of processes of computation arise from their interleaving with the processes of communication, from the resulting computational interactions, and only emerge into light when the problem of concurrency is taken into account. That is why the semantics of computational processes, formalized in process calculi, initially forked off from the main branch of the semantics of programming languages. The main part of Samson Abramsky's work, which I am trying to reconstruct in logical terms, was concerned with bringing the two branches together.

27.2.2 Process Propositions and Implications

27.2.2.1 Process Sequents Must Be Labelled

Process logics involve modeling states. There are many different ways to model states, but within a propositions-as-types framework, state spaces occur together with the data types, subject to the same derivation rules. Although the two must be treated differently within the rules (as we shall see already in Sect. 27.2.4), both require *labelled* sequents. For state spaces, this is clearly unavoidable. As mentioned in Sect. 27.1.4, an unlabelled sequent $X \vdash Y$ identifies all different proofs that X entails Y. In particular, there is just one entailment $X \vdash X$, the trivial one. But if X is a state space, then modeling state transitions requires nontrivial sequents $X \overset{\xi}{\mapsto} X$. The labels allow distinguishing the nontrivial sequents, where the states change, from the trivial one, where they do not.

27.2.2.2 Machine Abstraction and Process-Closed Categories

A process implication $[A, B]$ asserts not just that A implies B, but also that A implies $[A, B]$. Under the propostions-as-types interpretation, the type $[A, B]$ thus comes with two functions

- $A \wedge [A, B] \overset{\varepsilon}{\mapsto} B$ $(\upsilon^\bullet)$
- $A \wedge [A, B] \overset{\zeta}{\mapsto} [A, B]$ (υ°)

The label says that the latter is not an instance of the propositional conjunction elimination, a.k.a. projection on the types side. The sequent ζ is a *coinductive* clause, saying that $[A, B]$ is true on its own whenever it is true together with A as a *guard* (Coquand, 1993; Pavlovic, 1998). This is a typical *impredicative* claim, of kind which is often used mathematical analysis (Pavlovic and Pratt, 1999, 2002; Pavlović and Escardó, 1998). The general idea is that, whenever a proposition X, guarded by a proposition A, entails a proposition B, *and moreover also itself*, i.e. whenever X comes with the sequents

- $A \wedge X \mapsto B$ $(\llbracket - \rrbracket^\bullet)$
- $A \wedge X \mapsto X$ $(\llbracket - \rrbracket^\circ)$

then X also entails the process implication $[A, B]$. Putting it all together, we get the following rules:

$$\frac{}{A \wedge [A, B] \overset{\upsilon}{\mapsto} B \wedge [A, B]}\,\upsilon \qquad \frac{A \wedge X \overset{\varphi}{\mapsto} B \wedge X}{X \overset{\llbracket\varphi\rrbracket}{\longmapsto} [A, B]}\,\llbracket - \rrbracket \qquad \begin{array}{c} \mathbf{S}(A \times X, B \times X) \\ \Big\downarrow {\scriptstyle \llbracket - \rrbracket_X} \\ \mathbf{S}\,(X, [A, B]) \end{array} \tag{27.16}$$

Terminology. A function in the form $\xi : A \times X \longrightarrow B \times X$ is often called a *machine*, and the set X is construed as its *state space*. The induced description $\llbracket \xi \rrbracket : X \longrightarrow [A, B]$ is called *anamorphism*.[2]

Naturality. Comparing the $\llbracket - \rrbracket$-rule with the $(\supset)$-rule in Sect. 27.1.2.3, shows the sense in which $[A, B]$ is a dynamic version of the implication $(A \supset B)$. But note that the rule $(\supset)$ is reversible, whereas the rule $\llbracket - \rrbracket$ is not; and that the X-natural bijection in (27.4) on the right boils down to an X-natural transformation on the right in (27.16). Moreover, since X occurs on both sides of the sequent $A \wedge X \overset{\varphi}{\mapsto} B \wedge X$, and thus in both covariant and contravariant position in $\mathbf{S}(A \times X, B \times X)$, the naturality of

[2] Anamorphisms are the coalgebra homomorphisms into final coalgebras. The name is due, I believe, to Lambert Meertens. It seems to have caught on without having been introduced in a publication. Many functional programmers call them *unfolds*, generalizing the special case of lists. A machine $A \times X \longrightarrow B \times X$ can be viewed as a coalgebra $X \longrightarrow (A \Rightarrow (B \times X))$.

$[\![-]\!]_X$ is not as simple as in (27.5), and it genuinely adds to the story. This time the naturality is in the form

$$\begin{tikzcd} \mathbf{S}(A \times Y, B \times Y) \arrow[d, "{[\![-]\!]_Y}"] & \arrow[l] \Theta_{AB} f \arrow[r] & \mathbf{S}(A \times X, B \times X) \arrow[d, "{[\![-]\!]_X}"] \\ \mathbf{S}(Y, [A, B]) \arrow[rr, "(-\circ f)"] & & \mathbf{S}(X, [A, B]) \end{tikzcd} \tag{27.17}$$

where Θ_{AB} is the functor

$$\begin{aligned} \Theta_{AB} \ : \ \mathbf{S}^o &\longrightarrow \mathbf{R} \\ X &\mapsto \mathbf{S}(A \times X, B \times X) \end{aligned} \tag{27.18}$$

where **R** is the category of sets and relations, described in Appendix A. The arrow part of this functor transforms a function $f \in \mathbf{S}(X, Y)$ into the relation $\Theta_{AB} f = (f)$ which is a subset of $\mathbf{S}(A \times Y, B \times Y) \times \mathbf{S}(A \times X, B \times X)$ defined by

$$\zeta(f)\xi \iff \begin{tikzcd} A \times Y \arrow[d, "\zeta"] & A \times X \arrow[l, "A\times f"'] \arrow[d, "\xi"] \\ B \times Y & B \times X \arrow[l, "B\times f"'] \end{tikzcd} \tag{27.19}$$

The relation $(- \circ f)$ in (27.17) is the arrow part of the functor

$$\begin{aligned} \nabla_{AB} : \mathbf{S}^o &\longrightarrow \mathbf{R} \\ X &\mapsto \mathbf{S}(X, [A, B]) \end{aligned} \tag{27.20}$$

where $\nabla_{AB} f = (- \circ f)$ is the subset of $\mathbf{S}(Y, [A, B]) \times \mathbf{S}(X, [A, B])$ defined

$$y(- \circ f)x \iff y \circ f = x \tag{27.21}$$

∇_{AB} is, of course, just homming into $[A, B]$, i.e. a functor to the category of sets **S** extended along the inclusion $\mathbf{S} \hookrightarrow \mathbf{R}$ of functions as special relations. The naturality of $[\![-]\!] : \Theta_{AB} \longrightarrow \nabla_{AB}$ genuinely depends on this casting. It says that $[\![-]\!]$ must preserve the machine (i.e. coalgebra) homomorphisms specified in (27.19). The concept of an AB-machine homomorphism has herewith been reconstructed logically, from the properties of the dynamic implication $[A, B]$ in (27.16).

To reconstruct the structure of final AB-machine, substitute $[A, B]$ for Y in (27.16), to get the outer square in

$$
\begin{tikzcd}
\mathbf{S}\left(A \times [A,B]\,,\, B\times [A,B]\right) \arrow[rrr, "(\llbracket\varphi\rrbracket)"] \arrow[ddd, "\llbracket - \rrbracket"'] & & & \mathbf{S}\left(A\times X, B\times X\right) \arrow[ddd, "\llbracket - \rrbracket"] \\
& \upsilon \arrow[r, mapsto] \arrow[d, mapsto] & \varphi \arrow[d, mapsto] & \\
& \mathrm{id}_{[A,B]} \arrow[r, mapsto] & \llbracket\varphi\rrbracket & \\
\mathbf{S}\left([A,B]\,,[A,B]\right) \arrow[rrr, "(-\circ\llbracket\varphi\rrbracket)"'] & & & \mathbf{S}\left(X, [A,B]\right)
\end{tikzcd}
\tag{27.22}
$$

The inner square says that, if we bind together the two left-hand rules in (27.16) by requiring that

$$\llbracket\upsilon\rrbracket_{[A,B]} = \mathrm{id}_{[A,B]}$$

then the naturality requirement in (27.16) implies that $A \times [A, B] \stackrel{\upsilon}{\to} B \times [A, B]$ is final among all AB-machines. This is conveniently summarized in the next definition, intended for the readers with categorical background.

Definition 27.2.1 A *process closed* category is a cartesian category $\mathbf{S}$ with a process implication $[A, B]$ for any pair of types A, B and the X-natural *machine abstraction* operaton

$$\mathbf{S}(A \times X, B \times X) \xrightarrow{\llbracket - \rrbracket_X^{AB}} \mathbf{S}(X, [A, B]) \tag{27.23}$$

The naturality of $\llbracket - \rrbracket^{AB}$ is with respect to the functors $\Theta_{AB}, \nabla_{AB} : \mathbf{S}^o \longrightarrow \mathbf{R}$ defined in (27.18)–(27.21).

Remark. Definition 27.2.1 can be viewed as a lifting of Definition 27.1.2 to process logics. While the latter is the categorical setting of the static propositions-as-types paradigm, the former recasts categories with final AB-machines in a logical form. The simple logical relation between the two structures will be spelled out in Proposition 27.2.3.3.

27.2.2.3 Process Propositions

A static proposition B is equivalent with the static implication $\top \supset X$, where $\top$ is the true proposition. All propositions can thus be viewed as special implications: namely the implications from the truth. A dynamic proposition $[B]$ can thus be defined in the form $[B] = [\top, B]$. Since the conjunctions $\top \wedge X$ are also equivalent with X, dynamic propositions can be defined by the rules

$$\frac{}{[B] \overset{\upsilon}{\mapsto} B \wedge [B]}\ \upsilon \qquad\qquad \frac{X \overset{\beta}{\mapsto} B \wedge X}{X \overset{[\![\beta]\!]}{\longmapsto} [B]}\ [\![-]\!] \qquad\qquad \begin{array}{c} \mathbf{S}(X, B \times X) \\ \Big| \\ [\![-]\!] \\ \downarrow \\ \mathbf{S}(X, [B]) \end{array}$$

Retracing the analysis from Sect. 27.2.2.2 now presents a proposition $[B]$ with a structure map $[B] \overset{\upsilon}{\to} B \times [B]$, as final among all maps in the form $X \longrightarrow B \times X$. The structure map is thus a pair $\upsilon =< \upsilon^\bullet, \upsilon^\circ >$, where $\upsilon^\bullet : [B] \longrightarrow B$ gives an output of the process proposition, or an action, and $\upsilon^\circ : [B] \longrightarrow [B]$ gives a resumption. It is thus a stream of elements in B.

27.2.3 Relating Process Implications and Static Implications

The static implication is defined by the rules and the correspondence in (27.4). The process implication is defined by the rules and the correspondence in (27.16). How are they related? Under which conditions are both sets of rules supported? Proposition 27.2.3.3 provides an answer. Sections 27.2.3.1 and 27.2.3.2, introduce the structures involved in the answer.

27.2.3.1 History Types

A *process of A-histories over a state space X* is a pair of functions $\kappa = \langle \kappa_{(-)}, \kappa_{(::)} \rangle$ typed

$$A \xrightarrow{\kappa_{(-)}} X \xleftarrow{\kappa_{(::)}} A \times X \tag{27.24}$$

The idea is that,

- $\kappa_{(-)}(a) \in X$ is the initial state of a process that starts with $a \in A$;
- $\kappa_{(::)}(x, a) \in X$ is the next state of a process after the state $x \in X$ and event or action $a \in A$.

A history $a^n = (\, a_1\, a_2 \cdots a_n\,)$ thus takes the process κ to the state

$$x_n = \kappa_{(::)}(a_n, \kappa_{(::)}(a_{n-1}, \ldots \kappa_{(::)}(a_1, \kappa_{(-)}(a_0)) \cdots))$$

Each string of n actions, construed as an A-history is thus mapped to a unique element of X. If the histories $(\, a_1 \cdots a_n\,)$ are viewed as the elements of A^n, then the disjoint union (coproduct)

$$A^+ = \coprod_{n=1}^{\infty} A^n$$

is the type of all A-histories. This is what we call a *history type*. For any process of A-histories κ over X there is a unique *banana-function* (a.k.a. *fold*, or *catamorphism*) $A^+ \xrightarrow{(\!|\kappa|\!)} X$ that makes following diagram commute.

$$\begin{tikzcd}
 & A^+ \arrow[d, dashed, "(\!|\kappa|\!)"] & A \times A^+ \arrow[l, "(::)"'] \arrow[d, dashed, "A \times (\!|\kappa|\!)"] \\
A \arrow[ur, "(-)"] \arrow[dr, "\kappa(-)"'] & & \\
 & X & A \times X \arrow[l, "\kappa(::)"]
\end{tikzcd}$$

Hence the history type constructor, the functor

$$\begin{aligned} (-)^+ : \mathbf{S} &\longrightarrow \mathbf{S} \\ A &\mapsto A^+ \end{aligned} \tag{27.25}$$

27.2.3.2 Retractions and Idempotents

A *retraction* is a pair of maps $A \underset{i}{\overset{q}{\rightleftarrows}} B$ such that $q \circ i = \mathrm{id}_B$. The type B *is a retract of* A when there is such a pair. It is easy to see that the composite $\varphi = i \circ q$ is then idempotent, and the retraction $A \underset{i}{\overset{q}{\rightleftarrows}} B$ is its *splitting*. The following diagram summarizes a retraction

$$\begin{tikzcd}
A \arrow[rr, "\varphi"] \arrow[dr, "\varphi"] \arrow[dd, two heads, "q"'] & & A \\
 & A \arrow[ur, "\varphi"] \arrow[dr, two heads, "q"] & \\
B \arrow[ur, tail, "i"] \arrow[rr, equal] & & B \arrow[uu, tail, "i"']
\end{tikzcd}$$

It is easy to see that i is then an equalizer of φ and the identity; and that q is a coequalizer of the same pair. Since any functor preserves retractions, they provide an exampe of an *absolute limit and colimit*. A categorical construction is *absolute* when it is preserved by all functors. It turns out that all absolute limits and colimits boil down to retractions (Paré, 1971). A category where all idempotents split into retractions is thus *absolutely complete*. The *absolute completion* of a category takes its idempotents as the objects, and a morphism f between the idempotents φ and ψ is required to preserve them, in the sense that $f = \psi \circ f \circ \varphi$. This is the weakest form of a categorical completion. Retractions, or idempotent splittings,[3] are thus an instance of the (co)limit operation.

The following proposition is a first step towards expanding the propositions-as-types paradigm to processes, promised in the title of this paper.

27.2.3.3 Proposition

Let $\mathbf{S}$ be a cartesian category. Then the static implications and the process implications induce each other in the presence of the history types and the retractions. More precisely,

(a) a cartesian closed category is process closed whenever it has the history types;
(b) a process closed category is cartesian closed whenever it has the retractions.

The **proof** is given in the Appendix.

Process abstraction is function abstraction over history types. Proposition 27.2.3.3 says that a cartesian closed category with history types has final AB-machines for all types A and B, and that their state spaces $[A, B] = \left(A^+ \Rightarrow B\right)$ support rules (27.16). A final AB-state machine can be constructed as a final coalgebras for the functor

$$\begin{aligned} E_{AB} : \mathbf{S} &\longrightarrow \mathbf{S} \\ X &\mapsto (A \Rightarrow (B \times X)) \end{aligned}$$

i.e. as a limit of the tower in the form

$$\begin{tikzcd} 1 & (A \Rightarrow B) \arrow[l, "!"'] & (A \Rightarrow (B \times (A \Rightarrow B))) \arrow[l, "(A \Rightarrow (B \times !))"'] & \cdots \arrow[l, dotted] \\ \cdots & E^n_{AB}(1) & E^{n+1}_{AB}(1) \arrow[l, "E^n_{AB}(!)"'] & \left(A^+ \Rightarrow B\right) \arrow[l, dotted] \end{tikzcd} \tag{27.26}$$

The process implications $[A, B]$ are thus modeled together with the static implications $(A \Rightarrow B)$, and both sets of rules (27.4) and (27.16) are supported. Processes can thus be modeled as machines. This was indeed the starting idea of process semantics (Milner, 1975). However, early on along this path, it becomes clear that many different machines implement indistinguishable processes. The problem of process equivalence arises (Milner, 1982). The input and the output types A and B of a process are observable, but the state space X may not be. In fact, any observable behavior can be realized over many different, unobservable state spaces.

---

[3] While the term *idempotent splitting* is well-established in category theory, the term *retraction* is familiar in mathematics at large. They refer to the same fundamental operation (Pavlovic and Hughes, 2020, Sect. 11).

27.2.4 The Problem of Cut in Process Logics

The fact that a process model may not support process composition is not just a conceptual shortcoming, but an obstacle to applications. Engineering tasks are in principle made tractable by decomposing the required processes into components, implementing the components, and composing the implementations. The process component models thus usually encapsulate and hide implementations, and display the interfaces. This methodology is conceptualized in *full abstraction*, one of the tenets of semantics of computation ever since Milner (1975, Sect. 4).

The first logical requirement of compositionality is that the state spaces must be factored out. This is necessary if the composition is to comply with a cut rule (27.1). If the process sequents are state machines in the form $X \times A \overset{\varphi}{\mapsto} X \times B$ and $Y \times B \overset{\psi}{\mapsto} Y \times C$, then the cut rule would be something like

$$\frac{X \wedge A \overset{\varphi}{\mapsto} X \wedge B \qquad\qquad Y \wedge B \overset{\psi}{\mapsto} Y \wedge C}{Z \wedge A \overset{(\varphi;\psi)}{\longmapsto} Z \wedge C} \tag{27.27}$$

The mismatch between the state spaces X and Y needs to be somehow resolved by composite state space Z. How should processes pass their internal states to one another?

The intuitive difference between data and states is that data are processed, whereas the states enable the processing. The structural difference is that data can be copied and sent in messages, whereas the states are internal, and may not be communicable. The problem of process composition is thus that the *observable* aspects of processes, that get passed in process composition from one process to another, need to be separated from the unobservable aspects, that remain hidden from the compositions. The same problem arises in applying processes as dynamic functions on sources as dynamic elements.The latter can, of course, be viewed as a special case of the former, just like process propositions are viewed as a special case of process implications.

The idea towards a solution is that the observable aspects are presented as data types, the unobservable aspects as state spaces. Processes should thus keep their internal states for themselves, as any dynamics aspects of their interactions can be communicated using the process implications. This follows from the fact, spelled out at the end of Sect. 27.2.2.2, that the process implications are the state spaces of the final state machines. Dispensing with the internal states, the process composition should thus be defined as a sequent in the form

$$[A, B] \wedge [B, C] \overset{\gamma}{\mapsto} [A, C] \tag{27.28}$$

In the static logics, the sequents that establish the transitivity of implication are equivalent with the cut rule from (27.1). In the process logics, the sequents like (27.28) solve the problem with (27.27). The final machine and coalgebra structures carried by the process implications have been used to define composition in a variety of final-coalgebra-enriched categories Abramsky (1993), Abramsky et al. (1995), Pavlovic and Abramsky (1997b), Krstić et al. (2001). The general derivation pattern behind the composition sequents in the form (27.28) is something like this:

$$\dfrac{\dfrac{\dfrac{}{A\wedge[A,B]\overset{\upsilon}{\mapsto}B\wedge[A,B]}\,\upsilon}{A\wedge[A,B]\wedge[B,C]\overset{\alpha}{\mapsto}B\wedge[A,B]\wedge[B,C]}\qquad\dfrac{\dfrac{}{B\wedge[B,C]\overset{\upsilon}{\mapsto}C\wedge[B,C]}\,\upsilon}{B\wedge[A,B]\wedge[B,C]\overset{\beta}{\mapsto}C\wedge[A,B]\wedge[B,C]}}{\dfrac{A\wedge[A,B]\wedge[B,C]\ \xmapsto{(\alpha;\beta)}\ C\wedge[A,B]\wedge[B,C]}{[A,B]\wedge[B,C]\ \xmapsto{\gamma=[\![\alpha;\beta]\!]}\ [A,C]}\,[\![-]\!]}\tag{27.29}$$

The task of composing processes thus boils down to interpreting the process implications $[A, B]$. The task of applying processes to sources boils down to interpreting the process propositions $[A] = [\top, A]$.

27.3 Functions Extended in Time

27.3.1 Dynamic Elements as Streams

The outputs of a machine $a = \left(X \xrightarrow{\langle a^\bullet, a^\circ\rangle} A \times X\right)$ are observable as a stream $a^\omega = (a_0\, a_1 \cdots a_n \cdots)$. Starting from an initial state $x_0 \in X$ the process

- outputs $a_0 = a^\bullet_{x_0}$ and updates the state to $x_1 = a^\circ_{x_0}$; then it
- outputs $a_1 = a^\bullet_{x_1}$ and updates the state to $x_2 = a^\circ_{x_1}$; after n steps, it
- outputs $a_n = a^\bullet_{x_n}$ and updates the state to $x_{n+1} = a^\circ_{x_n}$; and so on.

A dynamic[4] element can thus be construed as a stream of outcomes of a repeated measurement or count. Such data streams arise in science, and they are the subject of statistical inference (Fisher, 1973). If the outcomes are the truth values, then these streams are the process propositions. When the frequencies are counted, then they are the streams of random variables called *sources* in information theory (Ash, 1990, Chap. 6).

[4] We use the terms "*dynamic*" and "*extended in time*" interchangeably. A distinguishing aspect, that justifies keeping both in traffic, will emerge later.

27.3.2 Functions Extended in Time as Deterministic Channels

A dynamic function from A to B is generated by a machine in the form $f = \left(A \times X \xrightarrow{\langle f^{\bullet}, f^{\circ}\rangle} B \times X\right)$. Starting from an initial state $x_0 \in X$ the process consists of the following data maps and state updates:

$$\begin{array}{rlcrl}
a_0 \mapsto & b_0 = f^{\bullet}_{x_0}(a_0) & & a_0 \mapsto & x_1 = f^{\circ}_{x_0}(a_0) \\
a_0\, a_1 \mapsto & b_1 = f^{\bullet}_{x_1}(a_1) & & a_0\, a_1 \mapsto & x_2 = f^{\circ}_{x_1}(a_1) \\
\cdots & & & \cdots & \\
a_0\, a_1 \cdots a_n \mapsto & b_n = f^{\bullet}_{x_n}(a_n) & & a_0\, a_1 \cdots a_n \mapsto & x_{n+1} = f^{\circ}_{x_n}(a_n) \\
\cdots & & & \cdots &
\end{array}$$

A dynamic function can thus be viewed as a stream of functions in the form

$$f^{\omega} = \left(f_0\, f_1 \cdots f_n \cdots\right) \qquad \text{where} \qquad f_n = f^{\bullet}_{x_n} : A^n \longrightarrow B$$

The propositions-as-types interpretation of the process implication is based on such streams of functions. Streams of random functions are studied in information theory as channels. Those considered here correspond to the *deterministic* channels (Ash, 1990, Sect. 3.2).

27.3.3 History Monad and Comonad

The construction $(-)^+ : \mathbf{S} \longrightarrow \mathbf{S}$, described in Sect. 27.2.3.1, supports the monad structure

$$\begin{array}{ccccccc}
A & \xrightarrow{\ \eta\ } & A^+ & \qquad & A^+ & \xleftarrow{\ g^{\#}\ } & B^+ \\
a & \longmapsto & \left(a\right) & \qquad & g(b_1)\cdot g(b_2)\cdots g(b_n) & \longleftarrow\!\shortmid & \left(a_1\, a_2 \cdots a_n\right)
\end{array}$$

where g is a function from B to A^+, and $\cdot$ is the string concatenation. The algebras for this monad are semigroups: the set of finite A-sequences (words, nonempty lists) A^+ is the free semigroup over A.

For our concerns, it is more interesting that the construction $(-)^+ : \mathbf{S} \longrightarrow \mathbf{S}$ also supports the comonad structure

$$
\begin{array}{ccccc}
A & \xleftarrow{\quad\varepsilon\quad} & A^+ & \xrightarrow{\quad f^\# \quad} & B^+ \\
\\
a_n & \longleftarrow\!\!\!\shortmid & (a_1 a_2 \cdots a_n) & \longmapsto & \Big(f(a_1)\ f(a_1 a_2)\ \cdots\ f(a_1 \cdots a_n) \Big)
\end{array}
$$

Thinking of the sequences $(a_1 a_2 \cdots a_n)$ as sequences of events makes them into *histories*. The cumulative functions $f^\#$ thus capture the functions extended in time. Proposition 27.2.3.3 says that proofs of the process implications $[A, B]$ correspond to such functions. This correspondence makes process implications into hom-sets of a category.

27.3.4 Category of Functions Extended in Time

The category of free coalgebras for the comonad $(-)^+$ is

$$
|\mathbf{S}_+| = |\mathbf{S}|
$$
$$
\mathbf{S}_+(A, B) = \mathbf{S}(A^+, B)
$$

The lifting # gives rise to the composition in this category:

$$
\frac{\dfrac{A^+ \xrightarrow{f} B}{A^+ \xrightarrow{f^\#} B^+} \qquad B^+ \xrightarrow{g} C}{(f; g) = \left(A^+ \xrightarrow{f^\#} B^+ \xrightarrow{g} C\right)}
$$

The counit $A^+ \xrightarrow{\varepsilon} A$ plays the role of the identity for this composition. The correspondence

$$
\mathbf{S}_+(A, B) \cong \mathbf{S}(1, [A, B])
$$

means that $\mathbf{S}_+$, in a sense, externalizes the process implications as functions extended in time, and makes their proofs composable. The time extension of their composition unfolds in their cumulative form. Since A^+ is the disjoint union of $\coprod_{n=1}^{\infty} A^n$, a function $f : A^+ \longrightarrow B$ can be viewed as the stream $f^\omega = (f_1\ f_2 \cdots f_n \cdots)$ of functions $f_n : A^n \longrightarrow B$, like in Sect. 27.3.2. The corresponding cumulative function $f^\# : A^+ \longrightarrow B^+$ can then be viewed as the stream $f^\# = \left(f^1\ f^2 \cdots f^n \cdots\right)$ of functions $f^n : A^n \longrightarrow B^n$ which commute in the following diagram

$$\begin{tikzcd}
A \arrow[d, "f^1"] & A^2 \arrow[l, "\overleftarrow{\pi}"'] \arrow[d, "f^2"] & A^3 \arrow[l, "\overleftarrow{\pi}"'] \arrow[d, "f^3"] & A^4 \arrow[l, "\overleftarrow{\pi}"'] \arrow[d, "f^4"] & A^i \arrow[l, dotted] \arrow[d, "f^i"] & \cdots \arrow[l, dotted] \\
B & B^2 \arrow[l, "\overleftarrow{\pi}"] & B^3 \arrow[l, "\overleftarrow{\pi}"] & B^4 \arrow[l, "\overleftarrow{\pi}"] & B^i \arrow[l, dotted] & \cdots \arrow[l, dotted]
\end{tikzcd} \tag{27.30}$$

Each $\overleftarrow{\pi}$ projects away the rightmost component. The components f^n are:

$$f^1 = f_1 \qquad\qquad f^{i+1} = \left\langle f^i \circ \overleftarrow{\pi}, f_{i+1} \right\rangle$$

27.4 Partial Functions Extended in Time

27.4.1 Output Deletions and Process Deadlocks

Recall from Sect. 27.1.3.1 (27.10) that the partiality monad $(-)_\perp : \mathbf{S} \longrightarrow \mathbf{S}$ adjoins a fresh element $\perp$ to every type. A partial function $f : A \rightharpoonup B$ can be viewed as the total function $A \longrightarrow B_\perp$, which sends to $\perp$ the elements where f is undefined. There are two logically different ways to lift this to processes:

$$\frac{A \wedge X \mathrel{|}\!\!\rightarrow B_\perp \wedge X}{X \mathrel{|}\!\!\rightarrow [A, B_\perp]} \qquad\qquad \frac{A \wedge X \mathrel{|}\!\!\rightarrow (B \wedge X)_\perp}{X \mathrel{|}\!\!\rightarrow [A, B]_\perp} \tag{27.31}$$

On the left, the process may *delete* some of the outputs, but it always proceeds to the next state, whether if has produced the output or not. On the right, the process may *deadlock* and fail to produce either the output or the next state. The meanings of the two implications $[A, B_\perp]$ and $[A, B]_\perp$ are captured, respectively, by the final coalgebras of the two functors

$$\begin{array}{ll} D_{AB\perp} : \mathbf{S} \longrightarrow \mathbf{S} & \qquad D_{A\perp B} : \mathbf{S} \longrightarrow \mathbf{S} \\ \qquad X \mapsto (A \Rightarrow (B_\perp \times X)) & \qquad\qquad X \mapsto (A \Rightarrow (B \times X)_\perp) \end{array}$$

The state spaces of the final coalgebras of these two functors are then the hom-sets of the two categories of partial functions extended in time:

$$\begin{array}{cc} |\mathbf{S}_{+\perp}| = |\mathbf{S}| & |\mathbf{S}_{\perp+}| = |\mathbf{S}| \\ \mathbf{S}_{+\perp}(A, B) = \mathbf{S}(A^+, B_\perp) & \mathbf{S}_{\perp+}(A, B) = \coprod_{S \in \curlyvee A^+} \mathbf{S}(S, B) \end{array} \tag{27.32}$$

where $\curlyvee A^+$ is the set of safety specifications in A (Abramsky, 1993; Alpern and Schneider, 1987; Pavlovic and Abramsky, 1997b)

$$\curlyvee A^+ = \{S \subseteq A^+ \mid \vec{x} \preccurlyeq \vec{y} \in S \implies \vec{x} \in S\} \tag{27.33}$$

and where the prefix relation $\vec{x} \preccurlyeq \vec{y}$ means that there is $\vec{z}$ such that $\vec{x}\vec{z} = \vec{y}$. An $\mathbf{S}_{\perp+}$-morphism is a ladder like (27.34), but with partial functions f_i as rungs. The commutativity requirement imples that $f_i(\vec{s})$ must be defined whenever $f_{i+1}(\vec{s}a)$ is defined for some a. Hence $S \in \curlyvee A^+$ in (27.32).

27.4.2 *Safety and Synchronicity*

For $B = 1$, the right-hand part of (27.32) boils down to $\mathbf{S}_{\perp+}(A, 1) \cong \curlyvee A^+$. The safety properties in $\curlyvee A^+$ can thus be viewed as the objects of categories of *safe* dynamic functions. The morphisms may be *synchronous* or *asynchronous*, depending on whether the outputs are always observable. They become asynchronous if some outputs may be hidden or deleted.

27.4.2.1 Synchronous Safe Functions

The category SFun of *safe dynamic functions* has all safety specifications as its objects. Combining the $\mathbf{S}_+$-ladders from (27.30) with the $\mathbf{S}_\perp/1$-surjections from (27.15) shows that the safe dynamic functions are ladders in the form

$$\begin{tikzcd}
S_1 \arrow[d, two heads, "f^1" description] & S_2 \arrow[l, "\overleftarrow{\pi}"'] \arrow[d, two heads, "f^2" description] \arrow[dl, phantom, "\llcorner w"] & S_3 \arrow[l, "\overleftarrow{\pi}"'] \arrow[d, two heads, "f^3" description] \arrow[dl, phantom, "\llcorner w"] & S_4 \arrow[l, "\overleftarrow{\pi}"'] \arrow[d, two heads, "f^4" description] \arrow[dl, phantom, "\llcorner w"] & S_i \arrow[l, dotted] \arrow[d, two heads, "f^i" description] \arrow[dl, phantom, "\llcorner w"] & \cdots \arrow[l, dotted] \\
T_1 & T_2 \arrow[l, "\overleftarrow{\pi}"] & T_3 \arrow[l, "\overleftarrow{\pi}"] & T_4 \arrow[l, "\overleftarrow{\pi}"] & T_i \arrow[l, dotted] & \cdots \arrow[l, dotted]
\end{tikzcd} \tag{27.34}$$

The functions f^i are not mere surjections, in the sense that for every history $\vec{t} \in T$ there is a history $\vec{s} \in S$ such that $\vec{t} = f^\#(\vec{s})$. They are surjections *extended in time*, in the sense that the prefixes of $\vec{t}$ must have been the image of the prefixes of $\vec{s}$, i.e. $\overleftarrow{\pi}\left(\vec{t}\right) = f^\#\left(\overleftarrow{\pi}\,\vec{s}\right)$. Categorically, this amounts to saying that the squares in (27.34) are weak pullbacks. Logically, the commutativity of (27.34) uncovers a general coinductive pattern:

$$f^\#(\vec{s}) = \vec{t} \iff \forall b \in B \left(\vec{t}b \in T \implies \exists a \in A.\, \vec{s}a \in S \wedge f^\#(\vec{s}a) = \vec{t}b\right) \tag{27.35}$$

Such coinductive surjections lie at the heart of process theory as components of *bisimulations*, which we shall encounter in the next section. Before that, note that the dynamic surjections satisfying (27.35) must be *synchronous*, in the sense that they preserve the length of the histories: the time ticks steadily up the ladder. If there are silent actions, i.e. if functions may delete their outputs, this synchronicity may be breached.

27.4.2.2 Asynchronous Safe Functions

The functions extended in time *asynchronously* inhabit the category $\mathbf{S}_{\perp+}$. The element $\perp$ added to the outputs plays the role of the *silent*, unobservable action (Hennessy and Milner, 1980; Milner, 1989). In synchronous models, the observer is assumed to have global testing capabilities (Abramsky, 1987). The asynchrony arises when some of the actions of the Environment may not be observable for the System. Viewed as channels, the asynchronous functions extended in time become the deterministic *deletion* channels (Mitzenmacher, 2009). This leads to coarser process equivalences. Combining both of the constructions (27.32) allows capturing both forms of the partiality in

$$
\begin{aligned}
|\mathbf{S}_{\perp+\perp}| &= |\mathbf{S}| \\
\mathbf{S}_{\perp+\perp}(A, B) &= \coprod_{S \in \curlyvee A^{+}} \mathbf{S}(S, B_{\perp})
\end{aligned}
\tag{27.36}
$$

A function $f \in \mathbf{S}(S, B_{\perp})$ can be viewed as a stream of functions $f = \big(f_n : S_{\leq n} \longrightarrow B_{\perp}\big)_{i=1}^{\infty}$, where $S_{\leq n}$ are safe histories of length up to n, including the empty history, i.e.

$$
S_{\leq n} = (S \cap A^{\leq n}) + \big\{()\big\}
\tag{27.37}
$$

Here $A^{\leq n}$ is the disjoint union (coproduct) $\coprod_{i=0}^{n} A^{i}$. The cumulative form $f^{\#} = \big(f^{\leq n} : S_{\leq n} \longrightarrow B^{\leq n}\big)_{n=1}^{\infty}$ is now defined by

$$
\begin{aligned}
f^{\leq 1}() &= () & \qquad f^{\leq n+1}() &= () \\
f^{\leq 1}(a) &= \begin{cases} () & \text{if } f_1(a) = \perp \\ f_1(a) & \text{otherwise} \end{cases} & \qquad f^{\leq n+1}(a\vec{x}) &= \begin{cases} f^{\leq n}(\vec{x}) & \text{if } f_{n+1}(a\vec{x}) = \perp \\ f^{\leq n}(\vec{x}) :: f_{n+1}(a\vec{x}) & \text{otherwise} \end{cases}
\end{aligned}
$$

Its components are this time the rungs of the ladder

$$
\begin{tikzcd}
S_{\leq 1} \arrow[d, "f^{\leq 1}"] & S_{\leq 2} \arrow[l, "\overleftarrow{\pi}"'] \arrow[d, "f^{\leq 2}"] & S_{\leq 3} \arrow[l, "\overleftarrow{\pi}"'] \arrow[d, "f^{\leq 3}"] & S_{\leq 4} \arrow[l, "\overleftarrow{\pi}"'] \arrow[d, "f^{\leq 4}"] & S_{\leq i} \arrow[l, dotted] \arrow[d, "f^{\leq i}"] & \cdots \arrow[l, dotted] \\
B^{\leq 1} & B^{\leq 2} \arrow[l, "\overleftarrow{\pi}"] & B^{\leq 3} \arrow[l, "\overleftarrow{\pi}"] & B^{\leq 4} \arrow[l, "\overleftarrow{\pi}"] & B^{\leq i} \arrow[l, dotted] & \cdots \arrow[l, dotted]
\end{tikzcd}
\tag{27.38}
$$

where each $\overleftarrow{\pi}$ again projects away the last component. The category $\mathsf{ASFun} = \mathbf{S}_{\perp+\perp}/1$ of asynchronous safe functions has the safety specifications as its objects again, and a morphism $f \in \mathsf{ASFun}(\mathsf{S}_{\underline{\in}\mathsf{A}}, \mathsf{T}_{\underline{\in}\mathsf{B}})$ is a tower in the form

$$
\begin{tikzcd}
S_{\leq 1} \arrow[d, two heads, "f^{\leq 1}"] & S_{\leq 2} \arrow[l, "\overleftarrow{\pi}"'] \arrow[d, two heads, "f^{\leq 2}"] \arrow[dl, phantom, "\llcorner^{w}"] & S_{\leq 3} \arrow[l, "\overleftarrow{\pi}"'] \arrow[d, two heads, "f^{\leq 3}"] \arrow[dl, phantom, "\llcorner^{w}"] & S_{\leq 4} \arrow[l, "\overleftarrow{\pi}"'] \arrow[d, two heads, "f^{\leq 4}"] \arrow[dl, phantom, "\llcorner^{w}"] & S_{\leq i} \arrow[l, dotted] \arrow[d, two heads, "f^{\leq i}"] \arrow[dl, phantom, "\llcorner^{w}"] & \cdots \arrow[l, dotted] \\
T_{\leq 1} & T_{\leq 2} \arrow[l, "\overleftarrow{\pi}"] & T_{\leq 3} \arrow[l, "\overleftarrow{\pi}"] & T_{\leq 4} \arrow[l, "\overleftarrow{\pi}"] & T_{\leq i} \arrow[l, dotted] & \cdots \arrow[l, dotted]
\end{tikzcd}
\tag{27.39}
$$

This tower differs from (27.38) in that the squares are weak pullbacks, and the rungs of the ladder are surjective.[5] It shows that the *asynchronous* surjections exended in time satisfy the following condition:

$$
f^{\#}(\vec{s}) = \vec{t} \iff \left(\forall b \in B.\ \vec{t}b \in T \Rightarrow \exists \vec{a} \in A^{+}.\ \vec{s}\vec{a} \in S \wedge f^{\#}(\vec{s}\vec{a}) = \vec{t}b\right) \tag{27.40}
$$

This condition differs from (27.35) in that each step up the T-side by $b \in B$ may be followed on the S-side by a string of steps $\vec{a} \in A^{+}$, rather than just a single step $a \in A$.

27.5 Relations Extended in Time

27.5.1 External and Internal Nondeterminism

We saw in Sect. 27.1.3.1 that nondeterminism is modeled using the powerset monad $\wp : \mathbf{S} \longrightarrow \mathbf{S}$. Since a subset $U \subseteq A$ corresponds to an element $U \in \wp A$, a binary relation $R \subseteq A \times B$, viewed as a set of subsets $aR \subseteq B$ indexed over $a \in A$ corresponds to the function $\bullet R \colon A \longrightarrow \wp B$. The same relation, viewed as a set of subsets $Rb \subseteq A$, indexed over $b \in B$ also corresponds to the function $R\bullet \colon B \longrightarrow \wp A$. See Appendix A for more details.

There are two ways again in which the side-effect, this time nondeterminism, may affect processes. Internal nondeterminism affects the outputs, whereas external nondeterminism may also affect the states:

$$
\frac{A \times X \xrightarrow{\xi} (\wp B \times X)_{\perp}}{X \xrightarrow{[\![\xi]\!]} \left[A, \wp B\right]_{\perp}} \qquad\qquad \frac{A \times X \xrightarrow{\zeta} \wp(B \times X)}{X \xrightarrow{[\![\zeta]\!]_{\wp}} [A, B]_{\wp}} \tag{27.41}
$$

The external nondeterminism on the right incorporates the partiality as the empty outcome $\emptyset \in \wp(B \times X)$. The partiality monad $(-)_{\perp}$ is explicitly added to the internal nondeterminism on the left since they would otherwise never deadlock, which is

[5] Formally, in any regular category $\mathbf{S}$, the fact that all rungs are surjective can be derived from the assumption that the starting component is a surjection, and that the squares are weak pullbacks.

problematic both conceptually and technically. If a process ξ on the left, e.g. involving some guessing that leads to internal nondeterminism, never deadlocks at a state $x \in X$ and on an input $a \in A$, then it determines a unique next state $\xi^{\circ}(a, x) \in X$, and may produce an output from the set $\xi^{\bullet}(a, x) \in \wp B$. For an externally nondeterministic process ζ on the right, both the outputs and the state transitions are impacted by the nondeterminism, and any pair from $\zeta(a, x) \in \wp(B \times X)$ may be produced when the input a is consumed at state x. The intended meanings of the two process implications $\left[A, \wp B\right]_{\perp}$ and $[A, B]_{\wp}$ are captured, respectively, as the final coalgebras of the functors

$$\begin{aligned} P_{AB} &: \mathbf{S} \longrightarrow \mathbf{S} & \qquad Q_{AB} &: \mathbf{S} \longrightarrow \mathbf{S} \\ & X \mapsto \big(A \Rightarrow (\wp B \times X)\big)_{\perp} & & X \mapsto \wp(A \times B \times X) \end{aligned} \tag{27.42}$$

The expression on the right is based on the bijection $\wp(A \times B \times X) \cong \big(A \Rightarrow \wp(B \times X)\big)$. The state spaces of the final coalgebras of these two functors are quite different. We consider them separately, in the next two sections.

27.5.2 Internal Nondeterminism

27.5.2.1 Synchronous Safe Relations

The state space of the final coalgebra of the functor $P_{A\wp B}$ can be constructed within $\mathbf{S}$ as a limit of the tower like (27.26)

$$\begin{tikzcd} 1 & (A \Rightarrow \wp B)_{\perp} \arrow[l, "!"'] & \big(A \Rightarrow \big(\wp B \times (A \Rightarrow \wp B)_{\perp}\big)\big)_{\perp} \arrow[l, "(A\Rightarrow(\wp B\times !))"'] & \cdots \arrow[l, dotted] \\ \cdots & P_{AB}^{n}(1) \arrow[l, dotted] & P_{AB}^{n+1}(1) \arrow[l, "P_{A\wp B}^{n}(!)"'] & \left[A, \wp B\right]_{\perp} \arrow[l, dotted] \end{tikzcd} \tag{27.43}$$

or presented simply as

$$\begin{aligned} |\mathbf{S}_{+\wp}| &= |\mathbf{S}| \\ \mathbf{S}_{+\wp}(A, B) &= \coprod_{S \in \curlyvee A^{+}} \mathbf{S}(S, \wp B) \end{aligned} \tag{27.44}$$

A morphism from A to B in $\mathbf{S}_{+\wp}$ is thus a pair $\langle S, R\rangle$, where S is a safety specification, i.e. a prefix-closed set of A-histories from (27.33), and R is a stream of relations, presented as a stream of functions $\bullet R = \left(S_n \xrightarrow{\bullet R_n} \wp B\right)_{n=1}^{\infty}$, where $S_n = S \cap A^n$, or viewed cumulatively as

$$\bullet R^{\#} = \left(S_n \xrightarrow{\bullet R^{n}} (\wp B)^{n}\right)_{n=1}^{\infty}$$

The inductive definition is analogous to the one at the end of Sect. 27.3. On any input $(a_1\, a_2\, \cdots a_n) \in S$ the n-th component of $\bullet R^{\#}$ thus produces an n-tuple of subsets of B:

$$(a_1\, a_2\, \cdots a_n)R^n = \Big\langle a_1 R_1,\ \ (a_1 a_2)R_2,\ \ldots,\ (a_1 \cdots a_{n-1})R_{n-1},\ \ (a_1 \cdots a_{n-1} a_n)R_n \Big\rangle \tag{27.45}$$

If each each function $S_n \xrightarrow{\bullet R^n} (\wp B)^n$ is viewed as a relation $S_n \xleftrightarrow{R^n} B^n$, then (27.45) says that they make the following tower commute

$$\begin{tikzcd}
S_1 & S_2 \arrow[l, "\overleftarrow{\pi}"'] & S_3 \arrow[l, "\overleftarrow{\pi}"'] & S_4 \arrow[l, "\overleftarrow{\pi}"'] & S_i \arrow[l, dotted] & \arrow[l, dotted] \\
& R_1 \arrow[ul] \arrow[dl] & R_2 \arrow[l] \arrow[ul] \arrow[dl] & R_3 \arrow[l] \arrow[ul] \arrow[dl] & R_4 \arrow[l] \arrow[ul] \arrow[dl] & R_i \arrow[l, dotted] \arrow[ul] \arrow[dl] & \arrow[l, dotted] \\
B & B^2 \arrow[l, "\overleftarrow{\pi}"] & B^3 \arrow[l, "\overleftarrow{\pi}"] & B^4 \arrow[l, "\overleftarrow{\pi}"] & B^i \arrow[l, dotted] & \arrow[l, dotted]
\end{tikzcd} \tag{27.46}$$

To preclude any nontrivial side-effects of processes with trivial outputs, we slice over the trivial type 1 again, and define the category of *safe synchronous relations extended in time* as

$$\mathsf{SProc} = \mathbf{S}_{+\wp}/1 \tag{27.47}$$

This is the original *interaction category*, introduced by Abramsky (1993), and further studied by Abramsky et al. (1995), Pavlovic and Abramsky (1997b). The descriptions were different, but it is easy to see that the objects coincide, since the morphisms $S \in \mathbf{S}_{+\wp}(A, 1)$ are the prefix-closed sets $S \subseteq A^+$. Reasoning like in Sect. 27.4.2.1, a morphism $S_{\trianglelefteq A} \xleftrightarrow{R} T_{\trianglelefteq B}$ in $\mathbf{S}_{+\wp}/1$ is now reduced to a ladder of spans

$$\begin{tikzcd}
S_1 & S_2 \arrow[l, "\overleftarrow{\pi}"'] & S_3 \arrow[l, "\overleftarrow{\pi}"'] & S_4 \arrow[l, "\overleftarrow{\pi}"'] & S_i \arrow[l, dotted] & \arrow[l, dotted] \\
& R_1 \arrow[ul, two heads] \arrow[dl, two heads] & R_2 \arrow[l, "\ulcorner w"', "\llcorner w"] \arrow[ul, two heads] \arrow[dl, two heads] & R_3 \arrow[l, "\ulcorner w"', "\llcorner w"] \arrow[ul, two heads] \arrow[dl, two heads] & R_4 \arrow[l, "\ulcorner w"', "\llcorner w"] \arrow[ul, two heads] \arrow[dl, two heads] & R_i \arrow[l, dotted, "\ulcorner w"', "\llcorner w"] \arrow[ul, two heads] \arrow[dl, two heads] & \arrow[l, dotted] \\
T_1 & T_2 \arrow[l, "\overleftarrow{\pi}"] & T_3 \arrow[l, "\overleftarrow{\pi}"] & T_4 \arrow[l, "\overleftarrow{\pi}"] & T_i \arrow[l, dotted] & \arrow[l, dotted]
\end{tikzcd} \tag{27.48}$$

Like in (27.14), we have relations that are total in both directions, which means that the projections $R \to S$ and $R \to T$ are surjective, in this case componentwise. Like in (27.34), the surjections are extended in time, in the sense that all rhombi in (27.48) are weak pullbacks. Putting it all together, this tower says that R satisfies

$$\begin{aligned}
\vec{s}\, R\, \vec{t} \iff\ & \forall a \in A\ \left(\vec{s}a \in S \Rightarrow \exists b \in B.\, \vec{t}b \in T\ \wedge\ \vec{s}a\, R\, \vec{t}b\right)\ \wedge \\
& \forall b \in B\ \left(\vec{t}b \in T \Rightarrow \exists a \in A.\, \vec{s}a \in S\ \wedge\ \vec{s}a\, R\, \vec{t}b\right)
\end{aligned} \tag{27.49}$$

This condition means that $S_{\underline{\in} A} \overset{R}{\longleftrightarrow} T_{\underline{\in} B}$ is a *strong* or *synchronous bisimulation* relation (Milner, 1989; Park, 1981), as required in the original definition of SProc in Abramsky (1993).

Bisimulations are intrinsic. The notion of bisimulation was introduced in process theory as an imposed equivalence of the processes that are intended to be semantically indistinguishable (Milner, 1982; Park, 1981). The logical reconstruction of synchronous bisimulation from process-types-as-propositions shows that the same notion also arises as a property of morphisms in a category. The coinductive reconstructions of the whole gamut of bisimulations comprise a well-studied field of research. The present reconstruction, combining the nondeterminism monad $\wp$, the history comonad $(-)^+$, and the slicing over 1, displays the synchronous bisimulations as a logical property of processes arising from nondeterministic choices extended in time, provided that that nontrivial side-effects only arise when there are nontrivial outputs.

27.5.2.2 Asynchronous Safe Relations

Including in the model the silent, unobservable actions leads to asynchronicity, and to the notion of *weak* or *observational* bisimulation (Hennessy and Milner, 1980; Milner, 1989). Proceeding like in Sect. 27.4.2.2, we consider the final coalgebras of the functors

$$\begin{aligned} P_{AB\perp} : \mathbf{S} &\longrightarrow \mathbf{S} \\ X &\mapsto \big(A \Rightarrow (\wp(B_\perp) \times X)\big)_\perp \end{aligned} \tag{27.50}$$

as the hom-sets of the category

$$\begin{aligned} |\mathbf{S}_{+\wp\perp}| &= |\mathbf{S}| \\ \mathbf{S}_{+\wp\perp}(A, B) &= \coprod_{S \in \curlyvee A^+} \mathbf{S}(S, \wp(B_\perp)) \end{aligned} \tag{27.51}$$

The morphism tower is like (27.46), but with each S_n, R_n and B^n replaced replaced with $S_{\leq n}$, $R_{\leq n}$ and $B^{\leq n}$, as in (27.37) and (27.38). The category of *safe asynchronous relations extanded in time* is now

$$\mathsf{ASProc} = \mathbf{S}_{+\wp\perp}/1$$

and the morphism tower is like (27.48), with the same modification of the subscripts and the superscripts. This modified tower characterizes the following logical property of the *asynchronous* relation R extended in time:

$$\begin{aligned}\vec{s}\,R\,\vec{t} \quad\Longleftrightarrow\quad & \forall a \in A\,\Big(\vec{s}a \in S \Rightarrow \exists b \in B\,\big(\vec{t}b \in T\ \wedge\ \vec{s}a\,R\,\vec{t}b\big)\ \vee \\ & \qquad\qquad\qquad\qquad \exists \vec{x} \in A^*\,\big(\vec{s}a\vec{x} \in S\ \wedge\ \vec{s}a\vec{x}\,R\,\vec{t}\,\big)\Big) \qquad \wedge \\ & \forall b \in B\,\Big(\vec{t}b \in T \Rightarrow \exists a \in A\,\big(\vec{s}a \in S\ \wedge\ \vec{s}a\,R\,\vec{t}b\big)\ \vee \\ & \qquad\qquad\qquad\qquad \exists \vec{y} \in B^*\,\big(\vec{t}b\vec{y} \in T\ \wedge\ \vec{s}\,R\,\vec{t}b\vec{y}\big)\Big) \end{aligned} \tag{27.52}$$

This characterizes the *weak* or *observationsl* bisimulations of Hennessy and Milner (1980), Milner (1989). The category ASProc is equivalent to the one introduced and studied under the same name in Abramsky (1993), Pavlovic (1996), Pavlovic and Abramsky (1997b).

27.5.3 *External Nondeterminism*

27.5.3.1 Synchronous Dynamic Relations

The state space of the final coalgebra of the functor Q_{AB} from (27.42) should again come with a tower like

$$\begin{array}{l} 1 \xleftarrow{\;!\;} \wp(A \times B) \xleftarrow{\wp(A\times B\times !)} \wp(A \times B \times \wp(A \times B)) \leftarrow \cdots \\ \cdots\ Q^n_{AB}(1) \xleftarrow{P^n_{\wp AB}(!)} Q^{n+1}_{AB}(1) \leftarrow \cdots\cdots\cdots [A, B]_{\wp} \end{array} \tag{27.53}$$

The trouble is that such a tower never stabilizes within a universe of sets, since there is no set X such that $X \cong \wp X$. If we take $A = B = 1$, the tower boils down to

$$1 \xleftarrow{\;\cup\;} \wp 1 \xleftarrow{\;\cup\;} \wp\wp 1 \xleftarrow{\;\cup\;} \cdots\ \wp^n 1 \xleftarrow{\;\cup\;} \wp^{n+1}(1) \leftarrow \cdots\cdots [1, 1]_{\mathcal{P}} = \mathfrak{H} \tag{27.54}$$

where the coinductive fixpoint $\mathfrak{H}$ is the class of *hypersets*, or *non-wellfounded sets* (Aczel, 1988). It is dual to von Neumann's class of well-founded sets (von Neumann, 1923; Zermelo, 1930), which arises as the inductive fixpoint $\mathfrak{V}$ along the tower

$$\emptyset \xhookrightarrow{\;\epsilon\;} \wp\emptyset = 1 \xhookrightarrow{\;\epsilon\;} \wp\wp 1 \xhookrightarrow{\;\epsilon\;} \wp^n 1 \xhookrightarrow{\;\epsilon\;} \wp^{n+1} 1 \xhookrightarrow{\;\epsilon\;} \cdots\ \mathfrak{V} \tag{27.55}$$

Von Neumann, of course, did not draw categorical diagrams like (27.55), but specified his construction using the transfinite induction

$$V_0 = \emptyset \qquad\qquad V_\beta = \bigcup_{\alpha<\beta} \wp(V_\alpha) \qquad\qquad \mathfrak{V} = \bigcup_{\alpha\in\mathsf{Ord}} V_\alpha \tag{27.56}$$

The class Ord of ordinals, over which the union in the third clause is indexed, is assumed to be given. The construction thus provides an *inner* model of set theory within a given universe of sets and classes (Aczel, 1988), or equivalently within a universe with an inaccessible cardinal playing the role of the class Ord (Barr, 1993).[6]. In any case, reach a fixpoint within a given universe, the constructor $\wp$ must be restricted to stay within a smaller universe. Early on, Gödel restricted it to the subsets definable in the language of set theory, and constructed the universe $\mathfrak{L}$ of constructible sets, proving the independence of the Continuum Hypothesis, and launching the whole industry of the independence proofs (Gödel, 1940). Inner models of set-theory have also been constructed over topological spaces, concrete or abstract (Joyal & Moerdijk, 1995).

The above constructions also restrict to finite sets. The set theorists often explicitly exclude $\aleph_0$ from the definition of inaccessible cardinals, but the inequalities $2^n < \aleph_0$ and $\cup n < \aleph_0$ hold for all for all $n < \aleph_0$, and that makes $\aleph_0$ inaccessible from the universe **fS** of finite sets. Formally, **fS** can be viewed as the subcategory of **S** spanned by $U \in \mathbf{S}$ such that $\#U < \aleph_0$, where $\#U$ denotes the cardinality of U. Since computation is mostly concerned with finite sets, **fS** is often by the computer scientists to be the universe of "small sets", and **S** is interpreted as the universe of "classes". The powerset construction $\wp : \mathbf{S} \longrightarrow \mathbf{S}$ where $\wp X = \{U \subset X\}$ is then replaced with $\mathcal{P} : \mathbf{S} \longrightarrow \mathbf{S}$ where

$$\mathcal{P}X = \wp_{<\omega}X \quad = \quad \{U \subset X \mid \#U < \aleph_0\} \tag{27.57}$$

which restricts to $\mathcal{P} : \mathbf{fS} \longrightarrow \mathbf{fS}$. The tower (27.54) with $\mathcal{P}$ replacing $\wp$ thus lies in **fS**, and reaches a fixpoint $\mathcal{H} \cong \mathcal{PH}$ in **S** after countably many steps. Since $\mathcal{P}$ does not preserve limits, the tower does not stabilize at its limit, but it turns out to stabilize at a retract of its limit (Adámek and Koubek, 1995; Barr, 1993; Lambek, 1968b; Pavlovic, 1998). The projections from the fixpoint down the tower are still jointly monic, and support inductive reasoning about the universe of finite hypersets $\mathcal{H} = [1, 1]_{\mathcal{P}}$, which arises in the finite version of (27.54), and about the finite AB-relations extended in time $[A, B]_{\mathcal{P}}$ which arises in the finite version of (27.53). Continuing with the workflow from the preceding sections, we use the process implications arising from these finite versions of (27.53) to define the universe of sets with synchronous dynamic relations:

$$\begin{aligned} |\mathbf{S}^{\mathcal{P}}| &= |\mathbf{S}| \\ \mathbf{S}^{\mathcal{P}}(A, B) &= [A, B]_{\mathcal{P}} \end{aligned} \tag{27.58}$$

Like before, we factor out any nontrivial side-effects of processes with trivial outputs by slicing over the trivial type 1 again and define the *category synchronous dynamic relations*

[6] A universe of sets and classes is a model of the *NBG* set theory, whereas the one with an inaccessible cardinal can be interpreted in terms of the *ZFC* axioms (Mendelson, 2015, Chap. 4).

$$\mathsf{DProc} = \mathbf{S}^{\mathcal{P}}/1 \tag{27.59}$$

But now something new happens, and a path beyond the workflow from the previous sections opens up. When nondeterminism is internalized, the powerset constructor $\mathcal{P}$ generates types with enough structure to play the role of the labels. More precisely, the states built along the towers (27.53) to be cumulatively stored and distinguished using the intrinsic structure, making the label sets $A, B \in \mathbf{S}$ dispensable. In the constructions so far, the labels were used to identify the same action when it occurs in different processes. Now action can be identified by its history, which the type constructor, that generates the action, stores in the constructed type.

27.5.3.2 Internalising the Labels

All process universes presented up to so far have been built starting from a given universe $\mathbf{S}$ of labels. The coinductive construction leading to DProc has a novel feature that it can be built starting from nothing: the role of the label sets $A \in \mathbf{S}$ can be played by structures arising from the construction itself. The role of the labels $a \in A$ is to identify the same action when it occurs in different observations, or safety specifications S or T. This is assured by modeling them as subsets $S, T \subseteq A^+$. The upshot is that there can be at most one label-preserving function $S \longrightarrow T$, namely the inclusion $S \hookrightarrow T$.

When all actions arise in a cumulative hierarchy, by iterating the constructor $\mathcal{P}$, be it inductively (27.55) or coinductively (27.54), they are always given as sets with the element relation ϵ, which records the elements of each set, their elements, and so on. The axiom of extensionality

$$a = b \iff (\forall x.\, x \epsilon a \iff x \epsilon b) \tag{27.60}$$

says that this ϵ-structure completely determines the identity of each set: two sets with the same elements are the same set. In the cumulative hierarchy, the elements of sets are sets too, so the same elements are also the sets with the same elements. If such hereditary ϵ-relations are unfolded into trees, the extensionality axiom means that these trees must be *irredundant*: they have no nontrivial automorphisms. In other words, they cannot contain isomorphic subtrees at the same level (Pavlovic, 1995). The ϵ-structures that arise from the cumulative processes in (27.55) and (27.54) are extensional, thus irredundant, because the powerset constructors impose $\{a, a, b, c, \ldots\} = \{a, b, c, \ldots\}$. The other way around, Mostowski's *Collapse Lemma* (Mostowski, 1949) says that every well-founded extensional relation corresponds to the ϵ-structure of a set somewhere in $\mathfrak{V}$. Aczel's crucial observation in (Aczel, 1988) is that the well-foundedness assumption can be dropped: any extensional relation, including non-wellfounded, can be reconstructed as the ϵ-relation of a hyperset, somewhere in $\mathfrak{H}$, or for finite sets somewhere in $\mathcal{H}$. The upshot is that any two hypersets $S, T \in \mathcal{H}$, there is at most one ϵ-preserving function $S \longrightarrow T$, or else nontrivial automorphisms arise. The role of the label sets can now be played by the ϵ -structures.

Lemma 27.5.1 *For every countable $A \in \mathbf{S}$, i.e. such that $\#A \leq \aleph_0$, there are dynamic relations $A \overset{m}{\to} 1$ and $1 \overset{e}{\to} A$ in $\mathbf{S}^{\mathcal{P}}$ which make A into a retract of* 1*, i.e. their composite in $\mathbf{S}^{\mathcal{P}}$ is*

$$\mathrm{id}_A = \left(A \overset{e}{\twoheadrightarrow} 1 \overset{m}{\rightarrowtail} A\right)$$

A **proof** is sketched in Appendix C. To a category theorist, Lemma 27.5.1 says that the subcategory $\mathbf{S}^{\mathcal{P}}_{\leq\aleph_0} \hookrightarrow \mathbf{S}^{\mathcal{P}}$ spanned by the countable sets is the idempotent completion within $\mathbf{S}^{\mathcal{P}}$ of the endomorphism monoid $\boldsymbol{H} = \mathbf{S}^{\mathcal{P}}(1, 1)$. The underlying set of this monoid is the set $\mathcal{H}$ of finite hypersets. The monoid operation is the dynamic synchronous relational composition, spelled out below. For the categories

$$\mathsf{dProc} \;=\; \boldsymbol{H}/1 \qquad\qquad \mathsf{DProc}_{\leq\aleph_0} \;=\; \mathbf{S}^{\mathcal{P}}_{\leq\aleph_0}/1 \tag{27.61}$$

we have the following corollary, proved in Appendix D.

Corollary 27.5.2 *The inclusion*

$$\mathsf{dProc} \hookrightarrow \mathsf{DProc}_{\leq\aleph_0} \tag{27.62}$$

is an equivalence of categories.

Remark. The equivalence in the preceding corollary means that the embedding is full and faithful, and essentially surjective, i.e. that every type in $\mathsf{DProc}_{\leq\aleph_0}$ is isomorphic to a type in the image of dProc. This notion of equivalence allows finding an adjoint functor in the opposite direction *provided* that the axiom of choice is given, in this case for classes. The equivalence in (27.62) therefore does not provide an effective global representation of $\mathsf{DProc}_{\leq\aleph_0}$ in dProc. Locally, however, any structure present in DProc can be found in dProc, as long as we do not need uncountable sets of labels. In the rest of the paper, we elide the labels, and work in dProc.

27.5.3.3 Synchronous Dynamic Relations as Hypersets

The objects of the category dProc boil down the elements of the universe of finite $\mathcal{H}$, that arises as the coinductive fixpoint of the tower like (27.54), but with $\wp$ restricted to $\mathcal{P} = \wp_{\leq\aleph_0}$. Since $\mathcal{H} \cong \mathcal{P}\mathcal{H}$, an element of $\mathcal{H}$ can also be viewed as its finite subset, which unfolds it into a tower

$$\begin{tikzcd}
S_1 \arrow[d, hook] & S_2 \arrow[l, "\ni"'] \arrow[d, hook] & S_3 \arrow[l, "\ni"'] \arrow[d, hook] & S_4 \arrow[l, "\ni"'] \arrow[d, hook] & S_n \arrow[l, dotted] \arrow[d, hook] & S \arrow[l, dotted] \arrow[d, hook] \\
\mathcal{P}1 & \mathcal{P}^2 1 \arrow[l, "\ni"'] & \mathcal{P}^3 1 \arrow[l, "\ni"'] & \mathcal{P}^4 1 \arrow[l, "\ni"'] & \mathcal{P}^n 1 \arrow[l, dotted] & \mathcal{H} \arrow[l, dotted]
\end{tikzcd} \tag{27.63}$$

where all S_n and $\mathcal{P}^n 1$ are in **fS**. This seems like the most convenient presentation of the objects of dProc. A tower corresponding to a morphism $R \in \mathsf{dProc}(\mathsf{S}, \mathsf{T})$ looks just like (27.48) in Sect. 27.5.2.1, except that the projections $\overleftarrow{\pi}$ are replaced by the set-theoretic operation $\cup$. The bisimulation condition (27.49) now becomes

$$s\,R\,t \iff \forall s' \epsilon\, s\ \exists t' \epsilon\, t.\ s'\,R\,t' \quad \wedge \quad \forall t' \epsilon\, t\ \exists s' \epsilon\, s.\ s'\,R\,t' \tag{27.64}$$

27.5.3.4 Asynchronous Dynamic Relations

So far, the asynchrony has been modeled using a silent action $\perp$, which enabled waiting. When the actions are modeled using the element relation ϵ, i.e. each state transition is a choice of an element, then waiting can be enabled by allowing sets to contain and choose themselves, i.e. by making the relation ϵ reflexive, satisfying $x \epsilon x$ for all x. The objects of the category aProc of *asynchronous dynamic relations* are now the reflexive finite hypersets, conveniently viewed as towers of finite subsets

$$\begin{tikzcd}
S_{\leq 1} \arrow[d, hook] & S_{\leq 2} \arrow[l, two heads, "\ni"'] \arrow[d, hook] & S_{\leq 3} \arrow[l, two heads, "\ni"'] \arrow[d, hook] & \cdots \arrow[l, two heads, dotted] & S_{\leq n} \arrow[l, two heads, dotted] \arrow[d, hook] & \cdots \arrow[l, two heads, dotted] & S \arrow[l, two heads, dotted] \arrow[d, hook] \\
\mathcal{P}^{\leq 1}1 & \mathcal{P}^{\leq 2}1 \arrow[l, two heads, "\ni"'] & \mathcal{P}^{\leq 3}1 \arrow[l, two heads, "\ni"'] & \cdots \arrow[l, two heads, dotted] & \mathcal{P}^{\leq n}1 \arrow[l, two heads, dotted] & \cdots \arrow[l, two heads, dotted] & \mathcal{H}^{\circlearrowleft} \arrow[l, two heads, dotted]
\end{tikzcd} \tag{27.65}$$

where $\mathcal{P}^{\leq n} X = \coprod_{i=0}^{n} \mathcal{P}^i X$, and $\mathcal{H}^{\circlearrowleft}$ is the universe of reflexive finite hypersets. A morphism $R \in \mathsf{aProc}(\mathsf{S}, \mathsf{T})$ is now a reflexive hyperset relation, satisfying the following property

$$\begin{aligned} s\,R\,t \iff\ & \forall s' \epsilon\, s\ \big(\exists t' \epsilon\, t.\ s'\,R\,t' \ \vee\ \exists s'' \epsilon\, s'.\ s''\,R\,t\big) \ \wedge \\ & \forall t' \epsilon\, t\ \big(\exists s' \epsilon\, s.\ s'\,R\,t' \ \vee\ \exists t'' \epsilon\, t'.\ s\,R\,t''\big) \end{aligned} \tag{27.66}$$

The computational origins of this simulation strategies were studied by van Glabbeek (1993), van Glabbeek and Weijland (1996). Like before, it also arises from the mathematical structure of final coalgebras, and can be logically reconstructed from the paradigm of process-types-as-propositions.

27.6 Integers, Interactions, and Real Numbers

27.6.1 *The Common Denominator of Integers and Interactions*

Counting generates the ordinals (von Neumann, 1923), but the integers arise from the duality of counting up and down. Geometric and algebraic transformations generate monoids, but capturing the symmetries requires groups. Interactions between the system and the environment generate process universes, some of which we studied; but the dual interactions between the environment and the system were not captured. The duality inherent in process interactions was noted, albeit in passing, very early on in process theory:

> The *whole* meaning of any computing agent [would be that it is] a transducer, whose input sequence consists of enquiries by, or responses from, its environment, and whose output sequence consists of enquiries of, or responses to, its environment (Milner, 1975, p. 160).

A similar vision of dual interactions between the system and the environment, as an ongoing session of a question-answer protocol, re-emerged in linear logic (Girard, 1989a). It was formalized categorically in Abramsky and Jagadeesan (1994), and retraced in Abramsky (1996). The mathematical underpinning turned out to be the Int-construction, generating free compact categories over traced monoidal categories (Joyal et al., 1996). The name does not refer to ***interactions*** but to ***integers***. Appying the Int-construction to the additive monoid $\mathbb{N}$ of natural numbers, viewed as a discrete monoidal category, gives rise to the additive group $\mathbb{Z}$ of integers, viewed as a discrete compact category. The the trace structure on the monoid $\mathbb{N}$ is the cancellation property:

$$m + k = n + k \implies m = n$$

The set of integers is defined as the quotient

$$\mathbb{Z} = \mathsf{Int}_{\mathbb{N}} \quad = \quad \mathbb{N}_{-} \times \mathbb{N}_{+} / \sim$$

where the equivalence relation $\sim$ is:

$$< m_{-}, m_{+} > \sim < n_{-}, n_{+} > \iff m_{-} + n_{+} = n_{-} + m_{+}$$

The two components of the product are annotated for convenience, e.g. as $\mathbb{N}_{-} = \{''-''\} \times \mathbb{N}$ and $\mathbb{N}_{+} = \{''+''\} \times \mathbb{N}$. The cancellation property guarantees that each $\sim$-equivalence class contains a unique canonical representative in the form $< n, 0 >$ or $< 0, n >$. The former can be written as $-n$, the latter as $+n$.

The structural common denominator of integers and interactions, which makes the Int-construction applicable to both, is the *trace* operation. It will also take us from relations extended in time to the reals. Towards that goal, we spell out how the trace operation arises in categories of relations. This will makes the Int-construction applicable to the interaction categories.

Since the categories of relations, described in Appendix 27.9, are self-dual, the coproducts $+$ from the universe of sets and functions $\mathbf{S}$ become biproducts $\oplus$ in the category $\mathbf{R}$ of sets and relations. As the coproduct lifts to the universes of functions extended in time, the biproducts lift to the universes SProc, ASProc, dProc and aProc of relations extended in time. By definition, the biproducts are both products and coproducts. Since the relation biproducts are induced by the function coproducts, their unit is the function coproduct unit 0. For every type X, the biproduct structure consists of

- a monoid $0 \xrightarrow{!} X \xleftarrow{[\mathrm{id},\mathrm{id}]} X \oplus X$, and
- a comonoid $0 \xleftarrow{!} X \xrightarrow{<\mathrm{id},\mathrm{id}>} X \oplus X$,

which are natural with respect to all morphisms in and out of X. The projections $X \stackrel{\pi}{\leftarrow} X \oplus Y \stackrel{\pi'}{\rightarrow} Y$ and the injections $X \stackrel{\iota}{\rightarrow} X \oplus Y \stackrel{\iota'}{\leftarrow} Y$ are derived from the comonoid counits and from the monoid units respectively. A propositions-as-types interpretation of biproducts is tenuous but a process category with the biproducts and the hom-sets $[A, B]$ supporting a coinductive rule

$$\frac{A \oplus X \stackrel{\xi}{\rightarrow} B \oplus X}{X \xrightarrow{[\![\xi]\!]} [A, B]}$$

comes with the trace structure Tr derived by

$$\frac{A \stackrel{\iota}{\rightarrow} A \oplus Y \qquad \overline{A \oplus Y \oplus [A \oplus Y, B \oplus Y] \stackrel{\upsilon}{\rightarrow} B \oplus Y \oplus [A \oplus Y, B \oplus Y]} \qquad B \oplus Y \stackrel{\pi}{\rightarrow} B}{\dfrac{A \oplus [A \oplus Y, B \oplus Y] \xrightarrow{\iota;\upsilon;\pi} B \oplus [A \oplus Y, B \oplus Y]}{[A \oplus Y, B \oplus Y] \xrightarrow{\mathrm{Tr}=[\![\iota;\upsilon;\pi]\!]} [A, B]}}$$

Each of the categories of relations, **R**, SProc, dProc, etc., is easily seen to give rise to the trace structure in this way. See Appendix 27.13 for more.

27.6.2 *Games as Labelled Polarized Relations Extended in Time*

The biproducts in ASProc are in the form

$$\begin{array}{ccccccccc}
(S \oplus T)_{\leq 1} & \longleftarrow & (S \oplus T)_{\leq 2} & \longleftarrow & (S \oplus T)_{\leq 3} & \leftarrow\cdots\cdots & (S \oplus T)_{\leq i} & \leftarrow\cdots\cdots \\
\downarrow & & \downarrow & & \downarrow & & \downarrow & \\
(A + B)^{\leq 1} & \xleftarrow{\pi} & (A + B)^{\leq 2} & \xleftarrow{\pi} & (A + B)^{\leq 3} & \leftarrow\cdots\cdots & (A + B)^{\leq i} & \leftarrow\cdots\cdots
\end{array} \tag{27.67}$$

where $(S \oplus T)_{\underline{\oplus} A+B}$ are all shuffles of $S_{\underline{\oplus} A}$ and $T_{\underline{\oplus} B}$.

$$(S \oplus T)_{\leq i} = \left\{\vec{x} \in (A + B)^{\leq i} \mid \vec{x} \restriction_A \in S \ \wedge \ \vec{x} \restriction_B \in T\right\}$$

The trace structure of categories of relations with respect to the biproducts as the monoidal structure was analyzed already in the final section of Joyal et al. (1996), and explained in more detail for the interaction categories in Abramsky (1996).

The analysis presented in that paper suggests that the *AJM-games* (Abramsky and Jagadeesan, 1992; Abramsky et al., 2000, 1994) should be construed in terms of the Int-construction. The AJM-games are, of course, one of the crowning achievements of the quest for fully abstract models of PCF, and a tool of many other semantical results. They appeared in many different semantical contexts (Abramsky, 1997; Abramsky et al., 2000; Vákár et al., 2018), with many refinements and different presentation details. A crude common denominator can be obtained by applying the Int-construction from Appendix 27.13 to the category ASProc, leading to

$$\begin{aligned} |\mathsf{Gam}| &= |\mathsf{ASProc}|_- \times |\mathsf{ASProc}|_+ \\ \mathsf{Gam}(\mathsf{S}, \mathsf{T}) &= \mathsf{ASProc}(\mathsf{S}_- \oplus \mathsf{T}_+, \mathsf{T}_- \oplus \mathsf{S}_+) \end{aligned}$$

Some of the crucial features of game semantics, such as the copycat strategy, and the various switching and starting conditions, arise in such reconstructions as abstract mathematical properties, like the notions of bisimulations arose before.

27.6.3 Polarized Dynamics

Since $\mathcal{P}(A + B) \cong \mathcal{P}A \times \mathcal{P}B$, applying the powerset constructor on polarized sets $X_- + X_+$ leads to the functor

$$\begin{aligned} Q \colon \mathbf{S} &\longrightarrow \mathbf{S} \\ X &\mapsto \mathcal{P}_- X \times \mathcal{P}_+ X \end{aligned}$$

where the subscripts are still just annotations, and we can take, e.g., $\mathcal{P}_- X = \{''-''\} \times \mathcal{P}X$ and $\mathcal{P}_+ X = \{''+''\} \times \mathcal{P}X$.

27.6.3.1 Synchronous Case

The universe of *signed* finite hypersets can be constructed just like the universe of hypersets in Sect. 27.5.3, but bifurcating at each step into positive and negative hypersubsets:

$$\begin{array}{l} 1 \xleftarrow{\ !\ } \mathcal{P}_- 1 \times \mathcal{P}_+ 1 \xleftarrow{\ Q!\ } \mathcal{P}_-\left(\mathcal{P}_- 1 \times \mathcal{P}_+ 1\right) \times \mathcal{P}_+\left(\mathcal{P}_- 1 \times \mathcal{P}_+ 1\right) \leftarrow \cdots \\ \cdots\ Q^n 1 \xleftarrow{\qquad Q^n! \qquad} \mathcal{P}_-\left(Q^n 1\right) \times \mathcal{P}_+\left(Q^n 1\right) \leftarrow \cdots\cdots \mathcal{H}_{\pm} \end{array} \tag{27.68}$$

The final coalgebra structure still maps each hyperset to its elements, but this time they can be positive or negative

$$\mathcal{H}_{\pm} \underset{\left\langle \ni_{-}, \ni_{+} \right\rangle}{\overset{\cup}{\leftrightarrows}} \mathcal{P}_{-}\mathcal{H}_{\pm} \times \mathcal{P}_{+}\mathcal{H}_{\pm} \tag{27.69}$$

Notation. Given $s \in \mathcal{H}_{\pm}$, we write $s^{-} = \ni_{-}(s)$ for the negative part and $s^{+} = \ni_{+}(s)$ for the positive part. We often tacitly identify $\mathcal{H}_{\pm}$ with $\mathcal{P}_{-}\mathcal{H}_{\pm} \times \mathcal{P}_{+}\mathcal{H}_{\pm}$, in which case $s \in \mathcal{H}_{\pm}$ becomes a pair $s =< s^{-}|s^{+} >$, where $s^{-} = \ni_{-}(s)$ and $s^{+} = \ni_{+}(s)$. We follow Conway (2001) and denote a generic element of s^{-} by s_{-}, and a generic element of s^{+} by s_{+}, and abbreviate $s_{-} \in \ni_{-}(s)$ and $s_{+} \in \ni_{+}(s)$ to $s_{-}, s_{+} \epsilon\, s$. Writing $s = \{s_{-} \mid s_{+}\}$ instead of $s =< s^{-}, s^{+} >$ is yet another well-established notational abuse, used to great effect by Conway (2001). E.g., instead of $\cup s = \left\langle \cup s^{-}, \cup s^{+} \right\rangle$, the polarized union operation, pointing left in (27.69), can be written in the form

$$\cup s = \{s_{--}, s_{-+} \mid s_{+-}, s_{++}\}$$

Other coinductive definitions become even simpler, e,g.

$$\ominus\, s \;=\; \{\ominus s_{+} \mid \ominus\, s_{-}\} \qquad\qquad s \oplus t = \{s_{-} \oplus t, s \oplus t_{-} \mid s_{+} \oplus t, s \oplus t_{+}\} \tag{27.70}$$

Synchronous hypergames. The objects of the category gam are the signed finite hypersets from the universe $\mathcal{H}_{\pm}$. The final coalgebra structure (27.69) separates their elements into a negative and a positive part. In the game semantics, this is interpreted as separating a game $s \in \mathcal{H}_{\pm}$ into a pair $s =< s^{-}, s^{+} >$, where $s^{-} = \{s_{-} \epsilon\, s\} \in \mathcal{P}_{-}(\mathcal{H}_{\pm})$ are the moves available to the player $-$, whereas $s^{+} = \{s_{+} \epsilon\, s\} \in \mathcal{P}_{+}(\mathcal{H}_{\pm})$ are the moves available to the player $+$. The projections $\mathcal{H}_{\pm} \xrightarrow{q_i} Q^n 1$ down the tower (27.68) represent each game $s \in \mathcal{H}_{\pm}$ as a stream $[s^1, s^2, s^3, \ldots, s^{n+1}, \ldots]$, where $s^{n+1} = q_{n+1}(s) \in Q^{n+1}1 = \mathcal{P}_{-}(Q^n 1) \times \mathcal{P}_{+}(Q^n 1)$, and thus $s^{n+1} =< s^{n+1}_{-}, s^{n+1}_{+} >$, where $s^{n+1}_{-}, s^{n+1}_{+} \subseteq Q^n 1$.

A morphism $R \in \mathsf{gam}(\mathsf{s}, \mathsf{t})$ should be a **synchronous hyperstrategy**. It is a *hyper*strategy because the players $-$ and $+$ play not one, but two games, s and t, distinguished by the dual goals that the two players have in each of them:

$$s\,R\,t \iff \forall s_{-} \epsilon\, s\; \exists t_{-} \epsilon\, t.\;\; s_{-}\,R\,t_{-} \;\;\wedge\;\; \forall t_{+} \epsilon\, t\; \exists s_{+} \epsilon\, s.\;\; s_{+}\,R\,t_{+} \tag{27.71}$$

The player $-$ is thus tasked with simulating every s-step by a t-step, whereas the player $+$ is tasked with simulating every t-step by an s-step. A hyperstrategy into a polarized version of a synchronous bisimulation (27.64). While a bisimulation relation between two processes provides a simulation relation of each of them in the other one, both ways, the polarization of a hyperstrategy separates the two simulation tasks, and each player is tasked with one.

27.6.3.2 Asynchronous Case

Using the functor $\overline{Q} : \mathbf{S} \longrightarrow \mathbf{S}$ where $\overline{Q}X = X + QX$, the tower in (27.68) becomes

$$1 \xleftarrow{\;!\;} Q^{\leq 1}(1) \xleftarrow{\overline{Q}!} Q^{\leq 2}(1) \leftarrow\cdots\cdots Q^{\leq n}(1) \xleftarrow{\overline{Q}^n!} Q^{\leq n+1}(1) \leftarrow\cdots\cdots \mathfrak{R} \tag{27.72}$$

where $Q^{\leq n}(1) = \coprod_{i=0}^{n} Q^i(1)$. The final coalgebra structure is thus

$$\mathfrak{R} \underset{\ni}{\overset{\cup}{\rightleftarrows}} \mathfrak{R} + \mathcal{P}_{-}\mathfrak{R} \times \mathcal{P}_{+}\mathfrak{R} \tag{27.73}$$

The coalgebra structure $\ni$ maps $s =< s^-, s^+ >$ to $s = \ni(s)$ if $s^- \in s^-$ and $s^+ \in s^+$. Otherwise it unfolds its elements into $s^- = \ni_-(s)$ and $s^+ = \ni_+(s)$ like before. A straightforward induction along the tower gives the following.

Lemma 27.6.1 *Every $s \in \mathfrak{R}$ is ϵ-transitive, in the sense that for all $s_-, s_+ \in s$ holds*

$$s_-^- \subseteq s^- \subseteq s_+^- \qquad\qquad s_+^+ \subseteq s^+ \subseteq s_-^+ \tag{27.74}$$

The elements of the universe $\mathfrak{R}$ of transitive finite signed hypersets can be thought of as **asynchronous hypergames**. They are the objects of the category $\mathcal{R}$. An **asynchronous hyperstrategy** $R \in \mathcal{R}(s,t)$ resembles a branching bisimulation from (27.66), except that the two simulation tasks are again separated, like in (27.71), and assigned to the two players:

$$\begin{aligned} s\,R\,t \iff\ & \forall s_- \epsilon\, s\ (\exists t_- \epsilon\, t.\ s_-\,R\,t_-\ \vee\ \exists s_{-+} \epsilon\, s_-.\ s_{-+} R\, t)\ \wedge \\ & \forall t_+ \epsilon\, t\ (\exists s_+ \epsilon\, s.\ s_+\,R\,t_+\ \vee\ \exists t_{+-} \epsilon\, t_+.\ s\,R\,t_{+-}) \end{aligned} \tag{27.75}$$

Lemma 27.6.1 makes the relations induced by the coalgebra structure on $\mathfrak{R}$ into hyperstrategies. Remember that $s_- \epsilon\, s$ abbreviates $s_- \in \ni(s) \in \mathcal{P}_-\mathfrak{R}$, whereas $s \ni s_+$ abbreviates $s_+ \in \ni(s) \in \mathcal{P}_+\mathfrak{R}$.

Lemma 27.6.2 (27.75) *holds when sRt is instantiated to $s_- \epsilon\, s$ and $s \ni s_+$, for any $s \in \mathfrak{R}$ and $s_-, s_+ \epsilon\, s$.*

Proof $s_-^- \subseteq s^-$ implies that for every s_{--} there is s'_- with $s_{--} \epsilon\, s'_-$. $s^+ \subseteq s_-^+$ implies that for every s_+ there is some s_{-+} with $s_{-+} \epsilon\, s_+$. Hence (27.75) for $s_- \epsilon\, s$. $s^- \subseteq s_+^-$ implies that for every s_- there is s_{+-} with $s - \ni s_{+-}$. $s_+^+ \subseteq s^+$ implies that for every s_{++} there is some s'_+ with $s'_+ \ni s_{++}$. Hence (27.75) for $s \ni s_+$. □

Remark. The property in (27.75) is not self-dual under the relational converse, but under the polarity change $\ominus$ in (27.70). In game semantics, the polarity change switches the roles of the player and the opponent. The game-theoretic concept of *equilibrium*, where both players play their best responses, reimposes the *bi*simulation requirement: that the *same* relation is a winning strategy (simulation) in *both* directions. The equilibrium strategies are thus fixed under two dualities: under the polarity change (switching the players $-$ and $+$), and under the relational converse (switching the component games s and t). The two dualities are generally not independent, as there are situations when they do not commute. However, for games where they do commute, they induce a *dagger-compact* structure, akin to the adjunctions over the complex linear operators, which is induced by two commuting dualities: the complex conjugation and the matrix transposition. This structure also arises in many other areas of abelian and nonabelian geometry. It was not used in the game semantics, but it emerged in the Abramsky-Coecke models of quantum protocols and has been explored in other areas of the semantics of computation (Abramsky and Coecke, 2004; Coecke et al., 2009; Pavlovic, 2011).

27.6.4 A Category of Real Numbers

In closing this section, we encounter a remarkable and somewhat disturbing fact: that the posetal collapse of the category $\mathcal{R}$ boils down to the ordered field $\mathbb{R}$ of the real numbers. It is disturbing because it shows that the described process logic and game semantic constructions impose on the processes no computability restrictions whatsoever since they include all real numbers. On one hand, this observation should not be surprising, since John Conway reconstructed numbers from games a long time ago (Conway, 2001), and game semantics was inspired by his ideas and informed by his constructions (Abramsky and Jagadeesan, 1992). On the other hand, it should be surprising, because game semantics has been developed as the semantics of *computational* processes, albeit as a quotient of an undecidable term calculus (Berry and Curien, 1982; Loader, 2001; Milner, 1977).

27.6.4.1 Coalgebra of Reals

We adapt the alternating dyadics from Pavlovic and Pratt (1999, Sect. 3.2)[7] to present the real numbers. Consider the alphabet $\Sigma = \{-, +\}$, and denote by $\Sigma^{\circledast}$ the set of finite and infinite strings over it. It comes with the coalgebra structure

$$\begin{tikzcd} \Sigma^{\circledast} \arrow[r, bend right, "\chi"'] \arrow[r, phantom, "\cong"] & 1 + \Sigma \times \Sigma^{\circledast} \arrow[l, bend right, "(::)"'] \end{tikzcd} \tag{27.76}$$

where χ maps the empty string $()$ into 1 and each nonempty strings into its head symbol and the tail string. Equivalently, this coalgebra can be written in the form

[7] See also Pavlovic and Pratt (2002), Pavlović and Escardó (1998) for a broader context.

$$\begin{tikzcd} \Sigma^{\circledast} \arrow[r, bend right, "\kappa"'] & 1+\Sigma^{\circledast}_{-}+\Sigma^{\circledast}_{+} \arrow[l, bend right, "{[o,h_-,h_+]}"'] \arrow[l, phantom, "\cong"] \end{tikzcd} \tag{27.77}$$

where the product $\Sigma \times \Sigma^{\circledast}$, which is $\{-,+\} \times \Sigma^{\circledast}$ is expanded into $\{-\} \times \Sigma^{\circledast} + \{+\} \times \Sigma^{\circledast}$, and the products with the singletons are abbreviated as subscripts. The structure map κ now maps the empty string into 1, and the strings in the form $\pm::\vec{x}$ as $\vec{x}$ into $\Sigma^{\circledast}_{\pm}$, whereas the components h_- and h_+ add $-$ and $+$ as the head, while o maps the singleton from 1 into the empty string.

Each Σ-string encodes a unique real number. The idea is that we count the first string of $-$s or $+$s in the unary, and after that proceed in the alternating dyadics, e.g.

$$\begin{aligned} ++---+-- &\mapsto +1+1-\tfrac{1}{2}-\tfrac{1}{4}-\tfrac{1}{8}+\tfrac{1}{16}-\tfrac{1}{32}-\tfrac{1}{64} \\ ----+-+\cdots &\mapsto -1-1-1-1+\tfrac{1}{2}-\tfrac{1}{4}+\tfrac{1}{8}\cdots \end{aligned}$$

Since the infinite strings of $-$s and of $+$s encode the two infinities, we will have a map into the extended reals $\overline{\mathbb{R}} = \mathbb{R} \cup \{\infty, -\infty\}$. The bijection $\Sigma^{\circledast} \cong \overline{\mathbb{R}}$ is described in Appendix 27.14. We henceforth identify the two, and use both names interchangeably, since $\Sigma^{\circledast}$ refers to the encoding, and $\overline{\mathbb{R}}$ says what is encoded.

Ordering. The usual ordering of the reals in $\overline{\mathbb{R}}$ corresponds to the lexicographic ordering of $\Sigma^{\circledast}$. When the finite strings are padded by 0s, the symbol ordering is $- < 0 < +$.

27.6.4.2 Numbers Extended in Time: Conway's Version of Dedekind Cuts

Theorem 27.6.3 *There are functors*

$$\begin{tikzcd} \overline{\mathbb{R}} \arrow[r, bend right, "\Gamma"'] & \mathcal{R} \arrow[l, bend right, "\Upsilon"'] \end{tikzcd} \tag{27.78}$$

which make the extended continuum $\overline{\mathbb{R}}$ into the posetal collapse of the category $\mathcal{R}$ of asynchronous hypergames. In particular,

- *for every real number $\varsigma \in \overline{\mathbb{R}}$ holds $\Upsilon\Gamma(\varsigma) = \varsigma$;*
- *for every asynchronous hypergame $s \in \mathcal{R}$ there are natural hyperstrategies*

$$s \xrightarrow{\eta} \Gamma\Upsilon(s) \qquad \textit{and} \qquad \Gamma\Upsilon(s) \xrightarrow{\varepsilon} s$$

Proof (sketch) The functor Γ can be obtained from the anamorphism $[\![\kappa]\!]$

$$
\begin{tikzcd}
\overline{\mathbb{R}} \arrow[r, "\kappa"] \arrow[d, "\llbracket\kappa\rrbracket"'] & \overline{\mathbb{R}} + \mathcal{P}_{-}\overline{\mathbb{R}} \times \mathcal{P}_{+}\overline{\mathbb{R}} \arrow[d, "\llbracket\kappa\rrbracket + \mathcal{P}_{-}\llbracket\kappa\rrbracket + \mathcal{P}_{+}\llbracket\kappa\rrbracket"] \\
\mathfrak{R} \arrow[r, "\ni"'] & \mathfrak{R} + \mathcal{P}_{-}\mathfrak{R} \times \mathcal{P}_{+}\mathfrak{R}
\end{tikzcd}
$$

where κ is derived from (27.77), by mapping the empty string to the empty string, the Σ-strings in the form $(-::\varsigma)$ to the pair $\left\langle\{\varsigma\}, \emptyset\right\rangle$, and the strings in the form $(+::\varsigma)$ to $\left\langle\emptyset, \{\varsigma\}\right\rangle$. Setting $\Gamma\varsigma = \llbracket\kappa\rrbracket\, \varsigma$, the functoriality of Γ boils down to the observation that the lexicographic order $\varsigma \le \vartheta$ on $\Sigma^{\circledast}$ lifts to a relation $s \le t$ on $s = \Gamma\varsigma$ and $t = \Gamma\vartheta$ which satisfies (27.75), i.e.

$$
\begin{aligned}
s \le t \iff & \forall s_{-} \in s\ (\exists t_{-} \in t.\ \ s_{-} \le t_{-}\ \vee\ \exists s_{-+} \in s_{-}.\ s_{-+} \le t)\ \ \wedge \\
& \forall t_{+} \in t\ \ (\exists s_{+} \in s.\ \ s_{+} \le t_{+}\ \vee\ \exists t_{+-} \in t_{+}.\ s \le t_{+-})
\end{aligned}
\tag{27.79}
$$

As long as ς and ϑ are unpadded by 0s, their lexicographic ordering leads to $s = \Gamma\varsigma$ and $t = \Gamma\vartheta$ satisfying the synchronous comparison clauses $s_{-} \le t_{-}$ and $s_{+} \le t_{+}$ of (27.79). If ϑ is padded by 0s, then (27.79) is satisfied because the lexicographic ordering induces $s_{-+} \le t$. If ς is padded by 0s, then it induces $s \le t_{+-}$. This completes the definition of Γ.

The functor Υ arises from Conway's *simplicity theorem* (Conway, 2001, Theorem 11). It picks the simplest representatives of the equivalence classes of the posetal collapse of $\mathcal{R}$, where the simplicity is measured in Conway (2001) by the "birthday ordinal", which for our finite hypersets, signed or not, boils down each element's position on its coinduction tower. The simplicity theorem plays a central role in all presentations of surreal numbers, and suitable versions have been proved in detail in Alling (1987) and Gonshor (1986). The arrow part of Υ collapses the $\mathcal{R}$-morphisms to the lexicographic order on $\Sigma^{\circledast}$. Conway shortcuts his proof of the simplicity theorem by imposing the posetal collapse directly signed hypersets by

$$
s \le t \iff \forall s_{-} \in s\ \forall t_{+} \in t.\ \ t \not\le s_{-}\ \ \wedge\ \ t_{+} \not\le s \tag{27.80}
$$

Instantiating this definition to $t \le s_{-}$ and to $t_{+} \le s$ (27.80) gives

$$
\begin{aligned}
t \not\le s_{-} &\iff \exists t_{-} \in t.\ s_{-} \le t_{-}\ \ \vee\ \ \exists s_{-+} \in s_{-}.\ s_{-+} \le t \\
t_{+} \not\le s &\iff \exists t_{+-} \in t_{+}.\ s \le t_{+-}\ \ \vee\ \ \exists s_{+} \in s.\ s_{+} \le t_{+}
\end{aligned}
$$

and shows that (27.80) implies (27.79). The converse, spelled out along the lines of the proofs of the simplicity theorem that can be found in Alling (1987) and Gonshor (1986), involves extensive but routinely case reasoning. The equivalence classes of the posetal quotient of $\mathcal{R}$ are thus ordered by (27.80), which on $\Sigma^{\circledast}$ boils down to the lexicographic order. □

Remarks. Conway's proof of the simplicity theorem demonstrates coinduction in action, not only at the formal level in (27.80), but also at the meta-level. In order to define the $\overline{\mathbb{R}}$-ordering of the minimal representatives of the equivalence classes of his games, reduced to numbers, he imposes the sought ordering as a preorder on arbitrary representatives and then uses that preorder to prove the existence of the minimal representatives. Lemma 27.6.2 also shows how the simplicity follows from the coinductive construction, as it implies $\Upsilon(s_-) \leq \Upsilon(s) \leq \Upsilon(s_+)$, and steers the coinductive descent towards the simplest representative.

27.6.4.3 Real Numbers as Processes

Theorem 27.6.3 says that the real numbers can be viewed as processes; and the other way around, that the asynchronous, polarized, reflexive processes boil down to real numbers. The heart of the theorem is in the "boil down" part of the second statement. Its precise meaning is that the simulations between the asynchronous, polarized, reflexive processes implement (and are thus consistent with) the real number ordering. If these processes are thought of as the processes of observing, then the reals are the outcomes of the measurements. On the other hand, computing with the reals involves some embedding into a universe where each number is the outcome of many processes. This is a consequence of the observation, going back to Brouwer Brouwer (1921), that the irredundant representations of the reals, where each real number corresponds to a unique stream of digits, there are always basic arithmetical operations, and easily defined inputs, where no finite prefix suffices to determine a finite prefix of the output. Such operations are obviously not computable.

Dropping the infinite strings $-\infty = (- - - \cdots)$ and $\infty = (+ + + \cdots)$ on the left-hand side of the retraction $\overline{\mathbb{R}} \leftrightarrows \mathcal{R}$ in (27.78), and the signed hypersets bisimilar to $-\infty = \{-\infty|\}$ and $\infty = \{|\infty\}$ on the right-hand side, we get the retraction $\mathbb{R} \leftrightarrows \underline{\mathcal{R}}$. It lifts to $\mathbb{R}^n \leftrightarrows \underline{\mathcal{R}}^n$, i.e. it makes the real vector spaces into retracts of the discrete functor categories. A real matrix $L \in \mathbb{R}^{p\times q}$ becomes an $\underline{\mathcal{R}}$-profunctor $\Lambda = \left(p \xleftrightarrow{\Gamma L} q\right)$, and the linear operators $\mathbb{R}^p \xrightarrow{L} \mathbb{R}^q$ and $\mathbb{R}^q \xrightarrow{L^\ddagger} \mathbb{R}^p$ become the $\underline{\mathcal{R}}$-extensions of $\Lambda = \Gamma L$ along the Yoneda embeddings, in the enriched-category sense.[8]

$$\begin{tikzcd}
p \arrow[r, "\nabla"] \arrow[d, leftrightarrow, "\Lambda"] & \underline{\mathcal{R}}^p \arrow[d, bend right, "\Lambda^*"'] \arrow[d, phantom, "\dashv"] \\
q \arrow[r, "\Delta"'] & \underline{\mathcal{R}}^q \arrow[u, bend right, "\Lambda_*"']
\end{tikzcd}$$

The left Kan extension Λ^* maps the functor $\alpha \in \underline{\mathcal{R}}^p$ into the coend, which is the colimit along α of its tensors with the left transpose of Λ. The right Kan extension Λ_* maps the functor $\beta \in \underline{\mathcal{R}}^q$ into the end, which is the limit along β of its cotensors with the right transpose of Λ. But since α and β are discrete, the colimits boil down to coproducts, and the limits boil down to products. And since $\underline{\mathcal{R}}$ is self-dual, the

[8] The reader unfamiliar with what any of this means is welcome to skip the next paragraph.

products and the coproducts coincide as the biproducts, which we write $\oplus$; and the tensors and the cotensors also coincide as $\otimes$. The Kan extensions thus become

$$\Lambda^*(\alpha) = \left(\bigoplus_{i=1}^{p} \alpha_i \otimes \Lambda_{ij}\right)_{j=1}^{q} \qquad \Lambda_*(\beta) = \left(\bigoplus_{j=1}^{q} \Lambda_{ij} \otimes \beta_j\right)_{i=1}^{p} \tag{27.81}$$

where

$$s \oplus t = \left\{ s_- \oplus t,\ s \oplus t_- \ \middle|\ s_+ \oplus t,\ s \oplus t_+ \right\}$$
$$\begin{aligned} s \otimes t = \Big\{ & (s_- \otimes t) \oplus (s \otimes t_+) \ominus (s_- \otimes t_+),\ (s_+ \otimes t) \oplus (s \otimes t_-) \ominus (s_+ \otimes t_-) \ \Big| \\ & (s_- \otimes t) \oplus (s \otimes t_-) \ominus (s_- \otimes t_-),\ (s_+ \otimes t) \oplus (s \otimes t_+) \ominus (s_+ \otimes t_+) \Big\} \end{aligned}$$

correspond respectively to Conway's addition and multiplication operations Conway (2001). Formally, this correspondence means that the retraction Υ satisfies

$$\Upsilon(s \oplus t) \ = \ \Upsilon s + \Upsilon t \qquad\qquad \Upsilon(s \otimes t) \ = \ \Upsilon s \cdot \Upsilon t$$

The usual matrix operations are thus "rediscovered" as the Υ-image of the Kan extensions in (27.81) of $\underline{\mathcal{R}}$-profunctors (corresponding to the real matrices) along the Yoneda embeddings of the bases into their $\underline{\mathcal{R}}$-completions (corresponding to the real vector spaces).

27.6.5 Where Is Computation?

As exciting as it is to see the real numbers arising from the categorical structure of processes, it also suggests that we lost the computation from sight somewhere along the way, while retracing the paths of the categorical semantics of computation. The process universe $\underline{\mathcal{R}}$ contains a representative $\Gamma\varsigma$ of every real number ς from the field $\mathbb{R}$. Whatever can be computed on such process representatives of the reals in $\underline{\mathcal{R}}$ can be projected back into $\mathbb{R}$ along Υ. Any real number can ς can be obtained in that way, since $\Upsilon\Gamma\varsigma = \varsigma$. But most real numbers are not computable. Arbitrarily long prefixes of uncomputable reals can be defined by enumerating all computations and avoiding all computable reals by a diagonal argument. This idea has been refined in many directions, showing that almost all real numbers are uncomputable, whichever way we quantify them (Chaitin, 1969; Gacs, 1986; Martin-Löf, 1966). And they all live in $\underline{\mathcal{R}}$. Everything that any oracle can tell any computer is already there. Somewhere on the path from propositions-as-types, through process-propositions-as-types-extended-in-time, to dynamic interactions, the idea of process-computability-as-programmability got lost, and we got all processes.

In the final section, we retrace the path back to one of the original questions of categorical semantics: *How can* ***intensional*** *computation be characterized semantically?*

27.7 Categorical Semantics as a Programming Language

27.7.1 Computability-as-programmability

A process is computable if it is programmable.[9] In a universe of processes, types are used to specify requirements and to impose constraints. In a universe of *computable* processes, there is also a type $\mathbb{P}$ of *programs*. Since any Turing-complete language can encode its own interpreter, any model of a Turing-complete language must contain[10] the type $\mathbb{P}$ of programs in that language.

A model of computable processes is *extensional* if it only describes the extensions of computations, i.e. their input-output functions, and does not say anything about the process of computation. Each computable function is thus assigned a unique "program". Type-theoretically, this unique "program" is captured by the (cartesian) abstraction operation, which fold a function $f_x(a) : A \times X \longrightarrow B$ with parameters from X to the X-indexed family of abstract functions $\lambda a.f_x(a) : X \longrightarrow (A \Rightarrow B)$. The application operation applies an abstraction to its inputs and recovers the corresponding function. The bijection between the abstractions and their applications was displayed in (27.4) and formalized in Definition 27.1.2 using the structure of *cartesian closed* categories. If "programs" do not specify some input-output mappings, but also how they change during computation, then the X-indexing becomes a state dependency, and the computations are presented as state machines $\xi : A \times X \longrightarrow B \times X$, producing the outputs by $\xi^{\bullet} : A \times X \longrightarrow B$ and updating the states by $\xi^{\circ} : A \times X \longrightarrow X$. The bijection between the parametrized functions and their abstractions (27.4) becomes a mapping (27.16) of machines $\xi : A \times X \longrightarrow B \times X$ to the anamorphisms $[\![\xi]\!] : X \longrightarrow [A, B]$ assigning to each state in X a dynamic function as the induced computational behavior. This *machine abstraction* was formalized in Definition 27.2.1 using the structure of *process closed* categories. The machine abstraction is not injective because many different machines realize the same behaviors; and it is not surjective because some dynamic functions are not implementable by machines. A categorical structure capturing how actual computable functions are specified by actual programs (without the quotation marks) is formalized in Definition 27.7.1. It characterizes computable functions using the language of Definitions 27.1.2 and

[9] Network processes are sometimes also called computations, although they are not globally controllable, and thus not programmable. They can be steered by interacting programs and protocols, but that is a different story. The notion of computability was originally defined as computability by computers, and the term is still used in that sense.

[10] The tacit assumption is that a model of a programming language contains all types recognizable in that language.

27.2.1, but not in terms of an abstraction operation, since program abstraction is not an operation.

The conceptual distinction between the static view of the function abstraction in (27.4), and the dynamic view of the process abstraction in (27.16) is echoed to some extent by the technical distinction between the *denotational* and the *operational* semantics of computation (Abramsky, 1997, 2014; Bruni and Montanari, 2017). Overarching all such distinction is the logical distinction between the *extensional* and the *intensional* models of meaning, going back to Frege, Carnap, Church and Martin-Löf (Fitting, 2020). All models of computation that capture abstraction *as an operation* fall squarely on the extensional side. The intuitive reason is that abstraction as an operation readily produces a "program" to each computation; but programming is not such an easy operation. It is a process that involves programmers and evolves other processes.

In contrast with the denotational models of the λ-abstraction of functions, and with the operational models of the $[\![-]\!]$-abstraction of processes, the intensional models of computations are based on the operations for evaluating programs and executing computations. There are many programs for each computation, but there is no operation that transforms computations into programs.

27.7.2 Categorical Semantics of Intensional Computation

The logical schema of intensional computation is dual to (27.16):

$$\frac{X \overset{p}{\mapsto} \mathbb{P}}{A \wedge X \overset{\{\!|p|\!\}}{\longmapsto} \Diamond(B \wedge X)}\ \{\!|-|\!\}\Diamond \qquad\qquad \begin{array}{c} \mathbf{S}\,(X, \mathbb{P}) \\ \Big\downarrow{\scriptstyle \{\!|-|\!\}} \\ \mathbf{S}_M(A \times X, B \times X) \end{array} \tag{27.82}$$

The idea of computability-as-programmability is expressed by the requirement that the maps $\{\!|-|\!\}$ are surjective: for any computation $A \times X \overset{g}{\to} M(B \times X)$ there is a program ρ such that $\{\!|\rho|\!\} = g$. Computations are presented as state machines to help capturing the dynamics of computation. Proposition 27.7.2 shows that this view of computation is equivalent to the standard view in terms of acceptable enumerations.

The naturality of the program executions $\{\!|-|\!\}$ in (27.82) can be described, *mutatis mutandis*, in a similar way like the naturality of $[\![-]\!]$ in (27.16). An X-indexed family of functions $\{\!|-|\!\}_X^{AB} : \mathbf{S}(X, \mathbb{P}) \longrightarrow \mathbf{S}_M(A \times X, B \times X)$ constitutes a natural transformation $\{\!|-|\!\}^{AB} : \nabla_{\mathbb{P}} \longrightarrow \Theta_{AB}$ between the functors

$$\begin{aligned} \nabla_{\mathbb{P}} : \mathbf{S}^o &\longrightarrow \mathbf{R} \\ X &\mapsto \mathbf{S}(X, \mathbb{P}) \end{aligned} \qquad\qquad \begin{aligned} \Theta_{AB} : \mathbf{S}^o &\longrightarrow \mathbf{R} \\ X &\mapsto \mathbf{S}_M(A \times X, B \times X) \end{aligned} \tag{27.83}$$

See (27.19) and (27.21) in Sect. 27.2.2.2 for the arrow parts of these functors. The naturality requirement is dual to (27.17). It implies that the diagram here on the left commutes for every $p \in \mathbf{S}(X, \mathbb{P})$.

$$\begin{tikzcd}
\mathbf{S}\,(\mathbb{P},\mathbb{P}) \arrow[rr, "(-\circ p)"] \arrow[d, "\{\!|-|\!\}^{AB}_{\mathbb{P}}"] && \mathbf{S}\,(X,\mathbb{P}) \arrow[d, "\{\!|-|\!\}^{AB}_{X}"] \\
\mathbf{S}_M\,(A\times\mathbb{P}, B\times\mathbb{P}) \arrow[rr, leftrightarrow, "(p)"] && \mathbf{S}_M\,(A\times X, B\times X)
\end{tikzcd}
\qquad
\begin{tikzcd}
A\times X \arrow[r, "\{\!|p|\!\}"] \arrow[d, "A\times p"] & M(B\times X) \arrow[d, "M(B\times p)"] \\
A\times\mathbb{P} \arrow[r, "\{\!|\mathrm{id}|\!\}"'] & M(B\times\mathbb{P})
\end{tikzcd}
\tag{27.84}$$

The diagram on the right arises by chasing id $\in \mathbf{S}(\mathbb{P}, \mathbb{P})$ around the diagram on the left. The left-hand diagram says that $\{\!|\mathrm{id}|\!\}^{AB}_{\mathbb{P}}$ and $\{\!|p|\!\}^{AB}_{X}$ are related under $\Theta_{AB}\,p$, which by the definition in (27.19) means that the right-hand square commutes. Since the naturality implies that

$$\{\!|pf|\!\}_Y \;=\; \{\!|p|\!\}_X\,(A\times f) \;=\; \{\!|\mathrm{id}|\!\}_{\mathbb{P}}\,(A\times pf)$$

holds for all $f \in \mathbf{S}(Y, X)$ and $p \in \mathbf{S}(X, \mathbb{P})$, droping the subcripts X from $\{\!|-|\!\}_X$ seldom causes confusion. The other way around, by the surjectivity of $\{\!|-|\!\}$, for every computation $g \in \mathbf{S}_M(A\times X, B\times X)$ there is an X-indexed program $\rho \in \mathbf{S}(X, \mathbb{P})$ such that $\{\!|\rho|\!\}^{AB}_{X} = g$, making the right-hand square in (27.84) commute. Since this is true for all A and B, the claim is thus that $\mathbb{P}$ is the state space of a weakly[11] final AB-machine $\{\!|\mathrm{id}|\!\}^{AB}_{\mathbb{P}} \in \mathbf{S}_M(A\times\mathbb{P}, B\times\mathbb{P})$ — for *all types* A and B in $\mathbf{S}$. The categories of computable-as-programmable functions, induced by (27.84), are thus process-closed in a suitable intensional sense that is both weaker and stronger than the extensional process-closed structure (27.16). It is weaker in the sense that the abstractions are not unique, but it is stronger in the sense that all abstractions, over all types, are of the same type $\mathbb{P}$. They are the programs. Hence the intensional cousin of the cartesian-closed and the process-closed categories defined in Definitions 27.1.2 and 27.2.1:

Definition 27.7.1 A *categorical computer* is a cartesian category $\mathbf{S}$ with a commutative monad $M : \mathbf{S} \longrightarrow \mathbf{S}$, a fixed type $\mathbb{P}$ of *programs*, and for any pair of types A, B an X-natural family of surjections, called *program executions*:

$$\mathbf{S}(X,\mathbb{P}) \xrightarrow{\{\!|-|\!\}^{AB}_{X}} \mathbf{S}_M(A\times X, B\times X) \tag{27.85}$$

The naturality of the program executions $\{\!|-|\!\}^{AB}$ is with respect to the functors $\nabla_{\mathbb{P}}, \Theta_{AB} : \mathbf{S}^o \longrightarrow \mathbf{R}$ from (27.83).

Proposition 27.7.2 *Let* $\mathbf{S}$ *be a cartesian category,* $\mathbb{P} \in \mathbf{S}$ *a fixed type, and* $M : \mathbf{S} \longrightarrow \mathbf{S}$ *a commutative monad. Specifying the the program executions* $\{\!|-|\!\}$ *in* (27.85)*, and*

[11] The word “weakly” refers to the fact that the programs ρ_g are not unique: each machine g can be represented by many of them; in fact infinitely many.

establishing $\mathbf{S}$ *as a categorical computer, is equivalent to specifying the following data for all types* A, B, X*:*

(a) a universal evaluator *(or* interpreter*)* $\varphi^{AB} \in \mathbf{S}_M(A \times \mathbb{P}, B)$ *and*
(b) a partial evaluator *(or* specializer*)* $\sigma^X \in \mathbf{S}(X \times \mathbb{P}, \mathbb{P})$

such that for any $f \in \mathbf{S}_M(A, B)$ *there is* $p \in \mathbf{S}(1, \mathbb{P})$ *with*

$$
\begin{aligned}
f &= \varphi^{AB} \circ \left(A \times p\right) \\
\varphi^{(AX)B} &= \varphi^{AB} \circ \left(A \times \sigma^X\right)
\end{aligned}
\qquad
\begin{tikzcd}
A \arrow[rr, "f"] \arrow[d, "A \times p"'] & & B \\
A \times \mathbb{P} \arrow[urr, "\varphi^{AB}"] & & A \times X \times \mathbb{P} \arrow[ll, "A \times \sigma^X"] \arrow[u, "\varphi^{(AX)B}"']
\end{tikzcd}
\tag{27.86}
$$

Proof (sketch) Given a categorical computer, the interpreters are $\varphi^{AB} = \pi_B \circ \{\!|\mathrm{id}|\!\}^{AB}_{\mathbb{P}}$ and the specializers σ^X are chosen using the surjectivity of $\{\!|-|\!\}^{AB}_{(X\mathbb{P})}$. Showing that the same σ^X can be chosen for all A and B is the only part which requires work.[12] Towards he converse, setting $\{\!|p|\!\}^{AB}_X = \varphi^{A(BX)} \circ (A \times p)$ defines a natural transformation. To show that its components are surjective, for an arbitrary computation $A \times X \xrightarrow{g} M(B \times X)$, set $\rho(x) = \sigma^{X\mathbb{P}}(x, r, r)$ using in the following diagram.

$$
\begin{tikzcd}
A \times X \arrow[rr, "g"] \arrow[d, "A \times X \times r"] \arrow[ddd, dashed, bend right=60, "A \times \rho"'] & & M(B \times X) \arrow[d, "M(B \times X \times r)"'] \arrow[ddd, dashed, bend left=60, "B \times \rho"] \\
A \times X \times \mathbb{P} \arrow[r, "g \times \eta"] \arrow[d, dashed, "A \times X \times \mathbb{P} \times r"] & M(B \times X) \times M\mathbb{P} \arrow[r, "\phi"] & M(B \times X \times \mathbb{P}) \arrow[d, "M(B \times X \times \Delta)"'] \\
A \times X \times \mathbb{P} \times \mathbb{P} \arrow[d, "A \times \sigma^{X\mathbb{P}}"] \arrow[ddrr, phantom] \arrow[drr, "\varphi^{(AX\mathbb{P})(B\mathbb{P})}"] & & M(B \times X \times \mathbb{P} \times \mathbb{P}) \arrow[d, "M(B \times \sigma^{X\mathbb{P}})"'] \\
A \times \mathbb{P} \arrow[rr, "\varphi^{A(B\mathbb{P})}", "= \{\!|\mathrm{id}|\!\}^{AB}_{\mathbb{P}}"'] & & M(B \times \mathbb{P})
\end{tikzcd}
\tag{27.87}
$$

The program r is defined by the commutative trapezoid in the middle. It encodes the computation where the state output $A \times X \xrightarrow{g^\circ} MX$ is fed into the function $X \times \mathbb{P} \xrightarrow{X \times \Delta} X \times \mathbb{P} \times \mathbb{P} \xrightarrow{\sigma^{XP}} \mathbb{P}$ where $\sigma^{X\mathbb{P}}$ partially evaluates any program on itself. This computation is the composite of the arrows going from $A \times X \times \mathbb{P}$ right along the top and down along the right side of the trapezoid. Some programs r that make the trapezoid commute when substituted as the left dashed side exist by Proposition 27.7.2a. The top rectangle is obtained by feeding some such r as the input to

[12] Most computability theory goes through with non-uniform specializers, which may vary with the context A, B.

the partial evaluator, to evaluate it on itself. The triangle at the bottom commutes by Proposition 27.7.2b. The commutativity of the whole diagram gives $\{\!|\rho|\!\}_X^{AB} = g$. □

Historic background. Proposition 27.7.2 says that the structure of categorical computer is a categorical version of the standard concept of *acceptable enumeration* Rogers (1987). In the standard notation, the enumeration would be a sequence $\left(\varphi_x^n\right)_{x\in\mathbb{P}}^{n\in\mathbb{N}}$, where x is the program index, and n is the arity of the computable function φ_x. While the computable functions are usually modeled over natural numbers, and the arity n means that the function takes the inputs of type $\mathbb{N}^n$, and always produces a single output of type $\mathbb{N}$, the categorical treatment is over abstract types, so we write φ^{AB} to specify the input type A and the output type B.

Programming background. The construction in the proof of Proposition 27.7.2 is easily seen to be a version of Kleene's construction of the fixpoint in his Second Recursion Theorem (Rogers, 1987, Chap. 11). The partial evaluator evaluating all programs on themselves plays the central role. This capability of self-evaluation lies at the heart of many computational constructions (Moschovakis, 2010). While the diagram chase above elides many equations, the string diagrammatic versions do not just abridge the constructions but display the geometric patterns behind many of them. They support a diagrammatic programming language with convenient implementations of computable logic and arithmetic, program schemas, abstract metaprogramming concepts like compilation, supercompilation, synthesis, and to derive static, dynamic, and algorithmic complexity measures (Pavlovic, 2023; Pavlovic and Yahia, 2018).

The λ-calculus and the underlying type theories have been used as abstract programming languages in the semantics of computation from the outset (Scott, 1993), and remained at the heart of the semantical investigations (Abramsky et al., 2000; Hyland and Luke Ong, 2000). Programming in abstract programming languages has also been pursued since early on (Plotkin, 1977). It led to functional programming, which now permeates programming practices beyond the realm of. However, the mere presence of the abstraction operations makes the underlying type systems essentially extensional. Dropping the extensional λ-conversions allows that multiple programs may correspond to a single computation, but still provides a canonical choice among them, maintains a canonical extensional core of the type system (Hayashi, 1985). This has been the main obstacle to studying genuinely intensional algorithmic phenomena, such as complexity, within the semantics of computation.

27.7.3 Computability as an Intrinsic Property

A poset may be a monoid in many different ways: e.g., the reals are a monoid for addition, for multiplication, and for many other operations. But a poset may be a lattice (an idempotent monoid) in at most one way: the joins are the least upper bounds, the meets are the greatest lower bounds, and if they exist, they are uniquely determined by the order. A category can be monoidal in many different ways, but it can be cartesian in at most one way because the cartesian products are uniquely determined. The lattice structure of a poset and the cartesian structure of a category are unique, and they are therefore the *properties* of their carriers. When the meets in a poset have the right adjoints, the implications that arise are also unique, and the structure of a Heyting algebra in is also a property. For the same reason, the cartesian-closed structure from Definition 27.1.2 is a property of a category, as is the process-closed structure from Definition 27.2.1.

The structure of a categorical computer from Definition 27.7.1 is also essentially unique and thus a property of the category that carries it. Proving this requires a little more work than the simple arguments above, but not much more. It boils down to a categorical encoding of the theorem that all parametrized interpreters isomorphically interpret one another (Rogers, 1987). A categorical computer thus displays computability as a categorical property: that all of its morphisms are programmable functions.

On the other hand, it was explained in Abramsky (2014, Sect. 1.2.3) that the notion of computability, *as defined in the standard Church-Turing approach*, is *extrinsic*, in the sense that a particular computable function is recognized as such only by referring to a particular external model of computation, say a Turing machine or a definitional schema. The invoked model then describes a particular process of computing the function, which is not recorded or recognizable on the function itself. It was thus argued in Abramsky (2014) that the standard definitions do not specify computability as an intrinsic structure, even less a property of a function. In contrast, (27.82) expresses the idea of *computability-as-programmability* as a logical structure; and by the virtue of uniqueness of that structure, as a logical property. Whatever programming language $\mathbb{P}$ might be used to encode programs, they are always assigned semantics along some program executions $\mathbf{S}(X, \mathbb{P}) \twoheadrightarrow \mathbf{S}_M(A \times X, B \times X)$, or along some equivalent mappings. The Rogers' isomorphism theorem says that all programming languages are isomorphic along semantics-preserving computable functions. Whichever Church-Turing model of computation might be used to define computability, the underlying execution model will map its process descriptions to the corresponding computational processes, and this mapping will make it into a categorical computer. This structure provides a "*canonical form witnessing computability*", sought in Abramsky (2014, Sect. 1.2.3).

Many languages of logic claim universality and establish their universality on their own terms. The set theory proves that it is the foundation of all mathematics, first-order logic is the language of predicates, category theory is the language of structures. The statement that logic is tasked with discovering the universal laws of logic is a tautology, in a logic of logic. A universal law should not be misunderstood as the last word about anything, but as the first word about something else. The idea that *computability-as-programmability*is a model-invariant, syntax-independent, device-free concept, and a property intrinsic to all computable objects and processes, is broader than any particular structure, categorical or otherwise, in which it may be expressed. The idea of computability-as-programmability lurks behind Kolmogorov's invariance theorem (Li and Vitányi, 1997, Sect. 2.1). While recognizing a particular function as computable depends on encodings in a particular model, the invariance theorem is built upon the fact that the encodings and their transformations are programmable, and that the programs are of constant lengths. Kolmogorov's invariance theorem can be construed as a quantitative counterpart of Rogers' isomorphism theorem (Calude, 1988, Theorem 2.4.14). Both theorems characterize computability as an intrinsic property. Computability-as-programmability is not just testable by any of the equivalent models of computation, as claimed by the Church-Turing thesis, but it is also *quantifiable*, in Kolmogorov's formulation by the length of programs. Kolmogorov's *algorithmic complexity* is thus the quantitative view of the intrinsic property of computability-as-programmability. By displaying programmability as a structure, categorical semantics provides the qualitative view of this property.

It should be noted that the qualitative and the quantitative views of computability as an intrinsic property of processes come about in disguise in many arenas of science. Although the search for a program that makes a process computable is generally not a computable process, its average algorithmic complexity is an intrinsic quantity again: the Shannon entropy (Muchnik & Vereshchagin, 2006; Zurek, 1989). Information theory as the theory of information processing has been viewed as a theory of computation in microsystems, averaged out in thermodynamics. Domain theory has been viewed as a theory of computability-as-approximation in suitable topologies (Abramsky, 1991; Smyth, 1993, Sect. 5.1). A natural task for categorical semantics is to bring such conceptual threads together. That is the message that I got from Samson Abramsky's categories that no one had seen before.

27.8 Summary

In the propositions-as-types view, the extensional operations of abstraction and application, *viz* the structure of cartesian closed categories, correspond to the introduction and the elimination of the propositional implication:

$$\frac{(A \wedge X) \vdash B}{X \vdash (A \supset B)} \supset \qquad\qquad \begin{tikzcd} \mathbf{S}(A \times X, B) \arrow[d, bend right, "(A\Rightarrow -)\circ\eta_X"'] \\ \mathbf{S}\big(X, (A \Rightarrow B)\big) \arrow[u, bend right, "\varepsilon_X \circ (A\times -)"'] \end{tikzcd}$$

In process logics, the process implication introduction rule corresponds to the coinductive interpretation of arbitrary states as process behaviors, captured in the final machine:

$$\frac{A \wedge X \xmapsto{\varphi} \Diamond(B \wedge X)}{X \xmapsto{[\![\varphi]\!]} [A, B]_{\Diamond}}\;[\![-]\!] \qquad\qquad \begin{array}{c} \mathbf{S}_M(A \times X, B \times X) \\ \Big\downarrow{\scriptstyle [\![-]\!]_X} \\ \mathbf{S}\,(X, [A, B]_M) \end{array}$$

In terms of dynamic types, computation corresponds to program execution. In terms of process propositions, computability-as-programmability is thus an elimination rule, mapping programs, as intensional proofs of the universal proposition, the programming language, into computations as their extensions:

$$\frac{X \xmapsto{p} \mathbb{P}}{A \wedge X \xmapsto{\{p\}} \Diamond(B \wedge X)}\;\{-\} \qquad\qquad \begin{array}{c} \mathbf{S}\,(X, \mathbb{P}) \\ \Big\downarrow{\scriptstyle \{-\}} \\ \mathbf{S}_M(A \times X, B \times X) \end{array}$$

Categorical semantics provides convenient and sometimes effective tools for reasoning about types and processes. Samson Abramsky led many of us through its vast landscape. I followed him to the best of my ability. The present paper is an attempt at a travel report. But the territory is largely uncharted, and there were times when I lost sight of Samson, probably somewhere far ahead. It is thus likely that the travel report is not just about what I learned from Samson, but also about what I misunderstood by getting lost, and maybe most of all about what I did not learn at all. Categorical semantics of computational processes is a computational process itself, and it is the nature of such processes that they may terminate, or not.

Appendices

27.9 Category R of Sets and Relations

Relations $A \longleftarrow R \longrightarrow B$ arise in two ways:

(a) as subsets $R \overset{r}{\rightarrowtail} A \times B$, so that

$$aRb \iff \exists x \in X.\, a = r_A(x) \wedge r_B(x) = b$$

(b) as a nondeterministic functions $A \overset{\varrho}{\to} \wp B$ and $B \overset{\varrho^o}{\to} \wp A$, so that

$$aRb \quad\iff\quad \varrho(a) \ni b \quad\iff\quad a \in \varrho^o(b)$$

where $\wp : \mathbf{S} \longrightarrow \mathbf{S}$ is the powerset monad.

The equivalence between the two views lies at the heart of the elementary structure of topos (Barr and Wells, 2005; Freyd and Scedrov, 1990; Lambek and Scott, 1986), which can be defined in terms of the correspondece between the subsets $R \rightarrowtail A \times B$ and the elements $\chi_R \in \wp(A \times B)$, and the natural bijections

$$\mathbf{S}(X \times A, \wp B) \;\cong\; \mathbf{S}(X, \wp(A \times B)) \;\cong\; \mathbf{S}(X \times B, \wp A) \tag{27.88}$$

A relational calculus can, however, be developed entirely in terms of subobjects $R \rightarrowtail A \times B$, in type universes without the powerset monad. Process relations are presented from this angle. The universe $\mathbf{S}$ only needs to be *regular* (Barr et al., 1971; Pavlovic, 1995). In addition to the cartesian structure, it is thus also assumed to have the equalizers (i.e., the subsets characterized by equations), which induce the pullback squares. The final assumption, crucial for the relational calculus, is that every function $f : A \longrightarrow B$ has an epi-mono (surjective-injective) factorization: it can be decomposed in the form $f = \left(A \overset{e_f}{\twoheadrightarrow} A' \overset{m_f}{\rightarrowtail} B\right)$, where $e_f \in \mathcal{E}$ and $m_f \in \mathcal{M}$. The family $\mathcal{E}$ can be thought of as the quotient maps (coequalizers), whereas $\mathcal{M}$ are all monics. The family $\mathcal{E}$ is required to be stable under the pullbacks. The category of relations in $\mathbf{S}$ is then defined to be

$$\begin{aligned} |\mathbf{R}| &= |\mathbf{S}| \\ \mathbf{R}(A, B) &= \mathcal{M}_{\cong}\big/(A \times B) \end{aligned} \tag{27.89}$$

where $\mathcal{M}_{\cong}$ is the set of the equivalence classes modulo the relation

$$m \cong m' \iff \begin{tikzcd} R \arrow[rr, bend left, tail, "\cong"] \arrow[dr, tail, "m"'] & & R' \arrow[ll, bend left, tail] \arrow[dl, tail, "m'"] \\ & X & \end{tikzcd}$$

Without this quotienting, $\mathbf{R}(A, B)$ would in general be a proper class. The composition of relations $A \overset{R}{\longleftrightarrow} B$ and $B \overset{S}{\longleftrightarrow} C$, viewed as the $\mathcal{M}$-monics $R \rightarrowtail A \times B$ and $S \rightarrowtail B \times C$, is defined using the pullback $R \underset{B}{\times} S$ and the factorization in the following diagram.

$$\begin{tikzcd} & & R \underset{B}{\times} S \arrow[dl] \arrow[d, two heads] \arrow[dr] & & \\ & R \arrow[dl] \arrow[dr] & (R;S) \arrow[dll, bend right, tail] \arrow[drr, bend left, tail] & S \arrow[dl] \arrow[dr] & \\ A & & B & & C \end{tikzcd}$$

The identity $A \longleftrightarrow A$ in $\mathbf{R_S}$ is the diagonal $A \longrightarrow A \times A$ in $\mathbf{S}$. More general categories of relations can be defined in more general situations using technically different but conceptually similar constructions (Pavlovic, 1995, 1996). If $\mathbf{S}$ has the coproducts $+$, they become biproducts in $\mathbf{R}$. The products $\times$ from $\mathbf{S}$ induce a canonical monoidal structure in $\mathbf{R}$, with the compact structure $\eta : 1 \leftrightarrow A \leftrightarrow A \times A$ and $\varepsilon : A \times A \leftrightarrow A \leftrightarrow 1$ on every A (Kelly and Laplaza, 1980).

27.10 Proof of Proposition 27.2.3.3

(a) Suppose that $\mathbf{S}$ is a cartesian closed category with the static implication $(A \Rightarrow B)$, and with the process of A-histories $A \xrightarrow{(-)} A^+ \xleftarrow{(::)} A \times A^+$ for every A. Then $[A, B] = \left(A^+ \Rightarrow B\right)$ is the state space of the final AB-machine with the structure map

$$A \times \left(A^+ \Rightarrow B\right) \xrightarrow{\upsilon\, =\, <\upsilon^\bullet, \upsilon^\circ>} B \times \left(A^+ \Rightarrow B\right) \tag{27.90}$$

where the components are derived by evaluating along the components of the A-history process

$$\frac{A \times \left(A^+ \Rightarrow B\right) \xrightarrow{\left(A \times (-) \Rightarrow B\right)} A \times (A \Rightarrow B) \xrightarrow{\varepsilon} B}{A \times \left(A^+ \Rightarrow B\right) \xrightarrow{\upsilon^\bullet} B}$$

$$\frac{A^+ \times A \times \left(A^+ \Rightarrow B\right) \cong A \times A^+ \times \left(A^+ \Rightarrow B\right) \xrightarrow{(::) \times \left(A^+ \Rightarrow B\right)} A^+ \times \left(A^+ \Rightarrow B\right) \xrightarrow{\varepsilon} B}{A \times \left(A^+ \Rightarrow B\right) \xrightarrow{\upsilon^\circ} \left(A^+ \Rightarrow B\right)}$$

To show that (27.90) is a final machine, first note that every AB-machine $A \times X \xrightarrow{\xi\, =\, <\xi^\bullet, \xi^\circ>} B \times X$ induces an A-history process

$$A \xrightarrow{\kappa(-)} (X \Rightarrow B) \xleftarrow{\kappa(::)} A \times (X \Rightarrow B) \tag{27.91}$$

with the components

$$\frac{A \times X \xrightarrow{\xi^\bullet} B}{A \xrightarrow{\kappa(-)} (X \Rightarrow B)} \qquad \frac{X \times A \times (X \Rightarrow B) \cong A \times X \times (X \Rightarrow B) \xrightarrow{\xi^\circ \times (X \Rightarrow B)} X \times (X \Rightarrow B) \xrightarrow{\varepsilon} B}{A \times (X \Rightarrow B) \xrightarrow{\kappa(::)} (X \Rightarrow B)}$$

By Sect. 27.2.3.1, the A-history process κ induces the catamorphism (i.e. fold, banana-function) $(\!|\kappa|\!)$

$$
\begin{tikzcd}
 & A^{+} \arrow[d, dashed, "(\!|\kappa|\!)"] & A \times A^{+} \arrow[l, "(::)"'] \arrow[d, dashed, "A\times(\!|\kappa|\!)"] \\
A \arrow[ur, "(-)"] \arrow[dr, "\kappa(-)"'] & & \\
 & (X \Rightarrow B) & A \times (X \Rightarrow B) \arrow[l, dashed, "\kappa(::)"']
\end{tikzcd}
\tag{27.92}
$$

On the other hand, the transposition

$$
\frac{A^{+} \xrightarrow{(\!|\kappa|\!)} (X \Rightarrow B)}{X \xrightarrow{[\![\xi]\!]} \left(A^{+} \Rightarrow B\right)}
$$

induces the anamorphism (unfold, lens-function) $[\![\xi]\!]$

$$
\begin{tikzcd}
X \times A \arrow[r, "\xi"] \arrow[d, dashed, "{[\![\xi]\!]\times A}"'] & X \times B \arrow[d, dashed, "{[\![\xi]\!]\times B}"] \\
\left(A^{+} \Rightarrow B\right) \times A \arrow[r, "\upsilon"'] & \left(A^{+} \Rightarrow B\right) \times B
\end{tikzcd}
\tag{27.93}
$$

which shows that υ makes $[A, B]$ into the process implication as in Sect. 27.2.2.2. The diagram chase showing that the catamorphism (27.92) commutes if and only if (27.93) commutes is an instructive exercise.

(b) The assumption is that **S** has final AB-machines

$$
A \times [A, B] \xrightarrow{\upsilon \,=\, <\upsilon^{\bullet}, \upsilon^{\circ}>} B \times [A, B]
$$

Replacing the second component by the projection gives the machine which induces the anamorphism $[\![\upsilon^{\bullet}, \pi_1]\!]$, which makes the outer square in the following diagram commute.

$$
\begin{tikzcd}
A \times [A, B] \arrow[rr, "<\upsilon^{\bullet}{,}\pi_1>"] \arrow[d, dashed, "{A\times[\![\upsilon^{\bullet},\pi_1]\!]}"'] \arrow[dr, two heads, "A\times q"] & & B \times [A, B] \arrow[d, dashed, "{B\times[\![\upsilon^{\bullet},\pi_1]\!]}"] \arrow[dl, two heads, "B\times q"'] \\
 & A \times (A \Rightarrow B) \arrow[r, dashed, "<\varepsilon{,}\pi_1>"] \arrow[dl, tail, "A\times m"] & B \times (A \Rightarrow B) \arrow[dr, tail, "B\times m"'] \\
A \times [A, B] \arrow[rr, "<\upsilon^{\bullet}{,}\upsilon^{\circ}>"'] & & B \times [A, B]
\end{tikzcd}
\tag{27.94}
$$

Since $[\![\upsilon^{\bullet}, \pi_1]\!]$ is also endomorphism on the AB-machine $A \times [A, B] \xrightarrow{<\upsilon^{\bullet}, \pi_1>} B \times [A, B]$, the uniqueness of $[\![\upsilon^{\bullet}, \pi_1]\!]$ as an AB-machine homomorphism from $< \upsilon^{\bullet}, \pi_1 >$ to $< \upsilon^{\bullet}, \upsilon^{\circ} >$ implies that it is an idempotent:

$$[\![\upsilon^\bullet, \pi_1]\!] \circ [\![\upsilon^\bullet, \pi_1]\!] = [\![\upsilon^\bullet, \pi_1]\!]$$

Here we use the assumption that the idempotents split in $\mathbf{S}$, and define $(A \Rightarrow B)$ as the splitting

$$[\![\upsilon^\bullet, \pi_1]\!] = \left([A, B] \stackrel{q}{\twoheadrightarrow} (A \Rightarrow B) \stackrel{m}{\rightarrowtail} [A, B]\right)$$

also displayed in (27.94). The component $A \times (A \Rightarrow B) \stackrel{\varepsilon}{\longrightarrow} B$ of the factoring defined there is the counit of the adjunction $A \times (-) \dashv (A \Rightarrow -)$, defined

$$\begin{aligned} \mathbf{S}(X \times A, B) &\stackrel{\lambda}{\longrightarrow} \mathbf{S}(X, (A \Rightarrow B)) \\ f &\longmapsto \lambda f = q \circ [\![f, \pi_1]\!] \end{aligned}$$

To show that $\varepsilon \circ (\lambda A \times f) = f$, chase the following diagram:

$$\begin{tikzcd}
A \times X \arrow[rrr, "<f{,}\pi_1>"] \arrow[dr, "A\times[\![f{,}\pi_1]\!]"] \arrow[dd, "A\times\lambda f"'] & & & B \times X \arrow[dl, "B\times[\![f{,}\pi_1]\!]"'] \arrow[dd, "B\times\lambda f"] \\
& A \times [A, B] \arrow[r, "<\upsilon^\bullet{,}\pi_1>"] \arrow[dl, two heads, "A\times q"] & B \times [A, B] \arrow[dr, two heads, "B\times q"'] & \\
A \times (A \Rightarrow B) \arrow[rrr, "<\varepsilon{,}\pi_1>"'] & & & B \times (A \Rightarrow B)
\end{tikzcd}$$

□

27.11 Proof Sketch for Lemma 27.5.1

Since $\#A \leq \aleph_0$, there is an ordinal number $\kappa \leq \omega$ large enough to support an retraction $\mathcal{P}(A \times A) \rightarrowtail \mathcal{P}^\kappa(A) \twoheadrightarrow \mathcal{P}(A \times A)$, and thus also $Q_{AA}1 \rightarrowtail Q^\kappa_{A1}1 \twoheadrightarrow Q_{AA}1$ for the functor Q defined in (27.42). Hence the tower of retractions:

$$\begin{tikzcd}
1 \arrow[d, equal] & Q_{AA}1 \arrow[l, "!"'] \arrow[d, bend left, tail] & Q^2_{AA}1 \arrow[l, "Q_{AA}!"'] \arrow[d, bend left, tail] & Q^n_{AA}1 \arrow[l, dotted] \arrow[d, bend left, tail] & {[A, A]_{\mathcal{P}}} \arrow[l, dotted] \arrow[d, bend left, dashed, tail, "m_0"] \\
1 & Q^\kappa_{A1}1 \arrow[l, "!"'] \arrow[u, bend left, two heads] & Q^{\kappa 2}_{A1}1 \arrow[l, "Q^\kappa_{A1}!"'] \arrow[u, bend left, two heads] & Q^{\kappa n}_{A1}1 \arrow[l, dotted] \arrow[u, bend left, two heads] & {[A, 1]_{\mathcal{P}}} \arrow[l, dotted] \arrow[u, bend left, dashed, two heads, "e_0"]
\end{tikzcd}$$

The symmetry $A \times 1 \cong 1 \times A$ lifts to a smilar retraction

$$[A, A]_{\mathcal{P}} \stackrel{m_1}{\rightarrowtail} [1, A]_{\mathcal{P}} \stackrel{e_1}{\twoheadrightarrow} [A, A]_{\mathcal{P}}$$

With these retractions, the proof boils down to showing the commutativity of the following diagram

$$
\begin{tikzcd}
1 \arrow[rr, "\llbracket \mathrm{id} \rrbracket"] \arrow[dd, "\llbracket \mathrm{id} \rrbracket"'] & & {[A, A]_{\mathcal{P}}} \arrow[dl, tail, "<m_0, m_1>"'] \arrow[dd, "<\mathrm{id}, \mathrm{id}>"] \\
& {[A, 1]_{\mathcal{P}} \times [1, A]_{\mathcal{P}}} \arrow[dl, "\llbracket -;- \rrbracket"'] \arrow[dr, two heads, "<e_0, e_1>"] & \\
{[A, A]_{\mathcal{P}}} & & {[A, A]_{\mathcal{P}} \times [A, A]_{\mathcal{P}}} \arrow[ll, "\llbracket -;- \rrbracket"]
\end{tikzcd}
$$

where $\llbracket -; - \rrbracket$ are the enriched compositions, constructed like in (27.29) (or see Krstić et al. (2001) for more details), whereas $\llbracket \mathrm{id} \rrbracket$ is the enriched identity, constructed as the anamorphism (final coalgebra homomorphism) from the identity machine $A \times 1 \xrightarrow{\eta} \mathcal{P}(A \times 1)$, where η is the unit of the monad $\mathcal{P}$. This diagram says that $j = m_0 \llbracket \mathrm{id} \rrbracket \in \mathbf{S}^{\mathcal{P}}(A, 1)$ and $r = m_1 \llbracket \mathrm{id} \rrbracket \in \mathbf{S}^{\mathcal{P}}(1, A)$ display A as a retract of 1 in $\mathbf{S}^{\mathcal{P}}$, i.e. that they compose to

$$
\mathrm{id}_A = \left(A \xrightarrow{j = m_0 \llbracket \mathrm{id} \rrbracket} 1 \xrightarrow{r = m_1 \llbracket \mathrm{id} \rrbracket} A \right) \tag{27.95}
$$

□

27.12 Proof of Corollary 27.5.2

Since the embedding (27.62) is full and faithful by definition, we only need to prove that it is essentially surjective: for an arbitrary object $S \in \mathsf{DProc}_{\leq \aleph_0}$ we must find $S' \in \mathsf{dProc}$ such that $S \cong S'$ in $\mathsf{DProc}_{\leq \aleph_0}$. An object of $\mathsf{DProc}_{\leq \aleph_0}$ is a dynamic relation $A \overset{S}{\longleftrightarrow} 1$ in $\mathbf{S}^{\mathcal{P}}$, where $\#A \leq \aleph_0$. An object of dProc is a hyperset S', viewed as a dynamic relation $1 \overset{S'}{\longleftrightarrow} 1$ in $\mathbf{S}^{\mathcal{P}}$. By Lemma 27.5.1, there are the relations $j \in \mathbf{S}^{\mathcal{P}}(A, 1)$ and $r \in \mathbf{S}^{\mathcal{P}}(1, A)$ such that $(j; r) = \mathrm{id}_A$. Setting

$$
S' = \left(1 \overset{r}{\longleftrightarrow} A \overset{S}{\longleftrightarrow} 1 \right)
$$

assures that the inner triangle in the following diagram commutes.

$$
\begin{tikzcd}
1 \arrow[rr, bend right, "r"'] \arrow[dr, "S'"'] & & A \arrow[ll, bend right, "j"'] \arrow[dl, "S"] \\
& 1 &
\end{tikzcd}
$$

The outer triangle commutes because $S' \circ j = S \circ r \circ j = S$ by (27.95). So we have the morphisms $r \in \mathsf{DProc}_{\leq \aleph_0}(\mathsf{S}', \mathsf{S})$ and $j \in \mathsf{DProc}_{\leq \aleph_0}(\mathsf{S}, \mathsf{S}')$. They form an isomorphism because $r \circ j = \mathrm{id}_S$ by (27.95) again, and $j \circ r \in \mathsf{DProc}_{\leq \aleph_0}(\mathsf{S}', \mathsf{S}')$ must be an identity because S' is a subobject of the terminal object in $\mathsf{DProc}_{\leq \aleph_0}$. □

27.13 Traces and the Int-construction

The *trace* operation on a symmetric (or braided) monoidal category $(C, \otimes, I)$ is typed by the rule

$$\frac{A \otimes Y \xrightarrow{f} B \otimes Y}{A \xrightarrow{\mathrm{Tr}_Y(f)} B}$$

The equations for this operation, with some examples and explanations can be found in Abramsky (2005), Joyal et al. (1996), Pavlovic (2012). The free compact category over any traced monoidal C

$$
\begin{aligned}
|\mathsf{Int}_C| &= |C|_- \times |C|_+ \\
\mathsf{Int}_C(\mathsf{A}, \mathsf{B}) &= C\left(A_- \otimes B_+, B_- \otimes A_+\right)
\end{aligned}
\tag{27.96}
$$

where $X_- = \{-\} \times X$ and $X_+ = \{+\} \times X$. The composition of $\mathsf{Int}_C(\mathsf{A}, \mathsf{B}) \times \mathsf{Int}_C(\mathsf{B}, \mathsf{C}) \xrightarrow{\bullet} C(\mathsf{A}, \mathsf{C})$ is defined by

$$\frac{\dfrac{A_- \otimes B_+ \xrightarrow{f} B_- \otimes A_+ \qquad\qquad B_- \otimes C_+ \xrightarrow{g} C_- \otimes B_+}{A_- \otimes C_+ \otimes B_- \otimes B_+ \overset{\sigma}{\cong} A_- \otimes B_+ \otimes B_- \otimes C_+ \xrightarrow{f \otimes g} B_- \otimes A_+ \otimes C_- \otimes B_+ \overset{\sigma}{\cong} C_- \otimes A_+ \otimes B_- \otimes B_+}}{g \bullet f \;=\; \left(A_- \otimes C_+ \xrightarrow{\mathrm{Tr}_{B_- \otimes B_+}(\sigma \circ (g \otimes f) \circ \sigma)} C_- \otimes A_+\right)}$$

27.14 The Extended Reals as Alternating Dyadics

Recall from Sect. 27.6.4.1 that $\overline{\mathbb{R}} = \mathbb{R} \cup \{\infty, -\infty\}$ is the extended real continuum, and that $\Sigma^{\circledast} = \coprod_{i=0}^{\omega+1} \Sigma^i$ is the set of finite or infinite (countable) strings of symbols from $\Sigma = \{-, +\}$, which are treated in (27.97) as $\{-1, 1\}$.

Define the value of the function $\Phi : \Sigma^{\circledast} \longrightarrow \overline{\mathbb{R}}$ on an arbitrary string $\varsigma = (\varsigma_0\, \varsigma_1\, \varsigma_2, \ldots)$ to be

$$\Phi(\varsigma) = z \cdot \varsigma_0 + \sum_{i=z+1}^{\infty} \frac{\varsigma_i}{2^{i-z}} \tag{27.97}$$

where $z = \mu n.\ \varsigma_n \neq \varsigma_{n+1}$ is the length of the initial segment before the sign flips. If ς is the infinite string of either one sign or the other, then z is infinite, and the value of $\Phi(\varsigma)$ is either ∞ or $-\infty$. Leaving the two infinities aside, Φ establishes a bijection between the remaining Σ-stings, where the sign eventually flips, and the finite real numbers from $\mathbb{R}$. For an arbitrary $x \in \mathbb{R}$, the string $\upsilon \in \Sigma^{\circledast}$ such that $x = \Phi(\upsilon)$ can be constructed as follows:

- Decompose the real line as the disjoint union of the closed-open and open-closed intervals

  $$\mathbb{R} = \coprod_{n=1}^{\infty} [-n, -n+1) + \{0\} + \coprod_{n=1}^{\infty} (n-1, n]$$

  leaving the 0 on its own. Then there are 3 cases:
- (0) If $x = 0$ then υ is the empty string ().
- $(-)$ If $x \in [-n_0, -n_0 + 1)$, then υ begins with $\underbrace{- - \cdots -}_{n_0}$.
- $(+)$ If $x \in [n_0 - 1, n_0)$, then υ begins with $\underbrace{+ + \cdots +}_{n_0}$.
- In case (-), find
  - the smallest n_1 such that $x \leq -n_0 + \sum_{i=1}^{n_1} \frac{1}{2^i}$ and append $\underbrace{+ \cdots +}_{n_1}$ to υ;
  - the smallest n_2 such that $x \geq -n_0 + \sum_{i=1}^{n_1} \frac{1}{2^i} - \sum_{i=1}^{n_2} \frac{1}{2^{n_1+i}}$ and append $\underbrace{- \cdots -}_{n_2}$ to υ;
  - the smallest n_3 such that $x \leq \cdots$, etc.
- In case (+), find
  - the smallest n_1 such that $x \geq n_0 - \sum_{i=1}^{n_1} \frac{1}{2^i}$ and append $\underbrace{- \cdots -}_{n_1}$ to υ;
  - the smallest n_2 such that $x \leq \cdots$, etc.

- If you ever reach a sum equal to x, then halt and leave υ finite. Otherwise υ is infinite.

In any case, it is easy to see that $\Phi(\upsilon) = x$ and that $\Phi(\upsilon) = \Phi(\zeta)$ implies $\upsilon = \zeta$. So Φ is an injection. And we have just shown that it is a surjection by constructing for an arbitrary $x \in \overline{\mathbb{R}}$ a $\upsilon \in \Sigma^{\circledast}$ such that $x = \Phi(\upsilon)$. The function Φ defined by (27.97) is thus the claimed bijection.

References

Abramsky, S. (1987). Observation equivalence as a testing equivalence. *Theoretical Computer Science, 53*(2–3), 225–241.

Abramsky, S. (1991). Domain theory in logical form. *Annals of Pure and Applied Logic, 51*(1–2), 1–77.

Abramsky, S. (1993). Interaction categories. In G. L. Burn, S. J. Gay & M. Ryan (Eds.), *Theory and Formal Methods 1993*, Workshops in Computing (pp. 57–69). Springer.

Abramsky, S. (1996). Retracing some paths in process algebra. In U. Montanari & V. Sassone (Eds.), *CONCUR*, volume 1119 of Lecture Notes in Computer Science (pp. 1–17). Springer.

Abramsky, S. (1997). Semantics of interaction: an introduction to game semantics. In *Semantics and logics of computation*, volume 14 of Publications of the Newton Institute (pp. 1–31). Cambridge University Press.

Abramsky, S. (2005). Abstract scalars, loops, and free traced and strongly compact closed categories. In J. L. Luiz Fiadeiro et al. (Eds.), *Proceedings of the First International Conference on Algebra and Coalgebra in Computer Science (CALCO)*, volume 3629 of Lecture Notes in Computer Science (pp. 1–29). Springer.

Abramsky, S. (2010). Coalgebras, chu spaces, and representations of physical systems. In *Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science, LICS 2010*, 11–14 July 2010, Edinburgh, UK (pp. 411–420). IEEE Computer Society.

Abramsky, S. (2012). Big toy models: Representing physical systems as Chu spaces. *Synthese, 186*(3), 697–718.

Abramsky, S. (2014). Intensionality, definability and computation. In A. Baltag & S. Smets (Eds.), *Johan van Benthem on Logic and Information Dynamics* (pp. 121–142). Springer.

Abramsky, S., & Coecke, B. (2004). A categorical semantics of quantum protocols. In *Proceedings of 19th IEEE Symposium on Logic in Computer Science (LICS 2004).*

Abramsky, S., Gay, S. J., & Nagarajan, R. (1995). Specification structures and propositions-as-types for concurrency. In F. Moller & G. M. Birtwistle (Eds.), *Logics for Concurrency (Proceedings of 8th Banff Higher Order Workshop)*, volume 1043 of Lecture Notes in Computer Science (pp. 5–40). Springer.

Abramsky, S., & Jagadeesan, R. (1992). Games and full completeness for multiplicative linear logic (extended abstract). In R. K. Shyamasundar (Eds.), *Foundations of Software Technology and Theoretical Computer Science, 12th Conference, New Delhi, India*, December 18–20, 1992, volume 652 of *Lecture Notes in Computer Science* (pp. 291–301). Springer.

Abramsky, S., & Jagadeesan, R. (1994). New Foundations for the Geometry of Interaction. *Information and Computation, 111*(1), 53–119.

Abramsky, S., Jagadeesan, R., & Malacaria, P. (2000). Full abstraction for PCF. *Information and Computation,163*(2), 409–470.

Abramsky, S., Malacaria, P., & Jagadeesan, R. (1994). Full abstraction for PCF. In M. Hagiya & J. C. Mitchell (Eds.), *TACS*, volume 789 of Lecture Notes in Computer Science (pp. 1–15). Springer.

Aczel, P. (1988). Non-well-founded Sets. In *Center for the Study of Language and Information Publication Lecture Notes*. Cambridge University Press.
Adámek, J., & Koubek, V. (1995). On the greatest fixed point of a set functor. *Theoretical Computer Science, 150*(1), 57–75.
Alling, N. L. (1987). *Foundations of Analysis over Surreal Number Fields*, volume 141 of Notas de Matemática, North-Holland.
Alpern, B., & Schneider, F. B. (1987). Recognizing safety and liveness. *Distributed Computing, 2*(3), 117–126.
Ash, R. B. (1990). *Information Theory*. Dover Publications.
Barr, M. (1993). Terminal coalgebras in well-founded set theory. *Theoretical Computer Science, 114*(2), 299–315.
Barr, M., Grillet, P. A., & van Osdol, D. H. (1971). *Exact Categories and Categories of Sheaves*, volume 236 of Lecture Notes in Mathematics. Springer.
Barr, M., & Wells, C. (2005). *Toposes, Triples, and Theories*. Number 278 in Grundlehren der mathematischen Wissenschaften. Springer, 1985. Republished in: Reprints in Theory and Applications of Categories, No. 12 (pp. 1–287).
Berry, G., & Curien, P.-L. (1982). Sequential algorithms on concrete data structures. *Theoretical Computer Science, 20*(3), 265–321.
S. D. Brookes, Geva, S. (1992). Computational commands and intensional semantics. In M. P. Fourman et al. (Eds.), *Applications of Categories in Computer Science*, volume 177 of London Mathematical Society Lecture Note Series (pp. 1–44). Cambridge University Press.
Brouwer, L. E. J. (1921). Besitzt jede reelle Zahl eine Dezimalbruchentwicklung? *Mathematische Annalen, 83*(3), 201–210.
Bruni, R., & Montanari, U. (2017). *Models of Computation*. Springer International: Texts in Theoretical Computer Science. An EATCS Series.
Calude, C. (1988). *Theories of Computational Complexity*, volume 35 of Annals of Discrete Mathematics. North-Holland.
Chaitin, G. (1969). On the length of programs for computing finite binary sequences: statistical considerations. *J. ACM, 16,* 145–159.
Church, A. (1940). A formulation of the simple theory of types. *The Journal of Symbolic Logic, 5*(2), 56–68.
Cockett, J. R. B., & Spooner, D.A. (1995). Categories for synchrony and asynchrony. *Electronic Notes in Theoretical Computer Science, 1*, 66 – 90. MFPS XI, Mathematical Foundations of Programming Semantics, Eleventh Annual Conference.
Coecke, B., Paquette, É. O., & Pavlovic, D. (2009). Classical and quantum structuralism. In S. Gay & I. Mackie (Eds.), *Semantical Techniques in Quantum Computation* (pp. 29–69). Cambridge University Press.
Coecke, B., & Pavlovic, D. (2007). Quantum measurements without sums. In G. Chen, L. Kauffman & S. Lamonaco (Eds.), *Mathematics of Quantum Computing and Technology*, Taylor and Francis (36pp.). arxiv.org/quant-ph/0608035.
Conway, J. H. (2001). *On numbers and games* (2nd ed.). A K Peters.
Coquand, T. (1993). Infinite objects in type theory. In *International Workshop on Types for Proofs and Programs*, volume 806 of Lecture Notes in Computer Science (pp. 62–78). Springer.
Fisher, R. A. (1973). *Statistical methods and scientific inference*. Hafner Press.
Fitting, M. (2020). Intensional logic. In E. N. Zalta (Eds.), *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, spring 2020 edition.
Freyd, P., & Scedrov, A. (1990). *Categories, Allegories*. Number 39 in Mathematical Library, North-Holland.
Gabbay, D. M. (1996). Labelled deductive systems. *Labelled Deductive Systems*. Clarendon Press.
Gacs, P. (1986). Every sequence is reducible to a random one. *Information and Control, 70*(2), 186–192.

Girard, J.-Y. (1989). Towards a geometry of interaction. In J. W. Gray & A. Scedrov (Eds.), *Categories in computer science and logic*, volume 92 of Contemporary Mathematics (pp. 69–108). American Mathematical Society.

Girard, J.-Y., Lafont, Y., & Taylor, P. (1989). *Proofs and Types*. Number 7 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press.

Gödel, K. (1940). *The Consistency of the Axiom of Choice and of the Generalized Continuum-hypothesis with the Axioms of Set Theory*. Number 3 in Annals of Mathematics Studies. Princeton University Press.

Gonshor, H. (1986). *An Introduction to the Theory of Surreal Numbers*, volume 110 of London Mathematical Society Lecture Note Series. Cambridge University Press.

Gunter, C. (1992). *Semantics of programming languages: Structures and techniques*. Foundations of Computing. MIT Press.

Hayashi, S. (1985). Adjunction of semifunctors: Categorical structures in nonextensional lambda calculus. *Theoretical Computer Science, 41,* 95–104.

Hennessy, M., & Milner, R. (1980). On observing nondeterminism and concurrency. In J. W. de Bakker & J. van Leeuwen (Eds.), *Proceedings of the Automata, Languages and Programming, 7th Colloquium, Noordweijkerhout, The Netherlands*, July 14–18, volume 85 of *Lecture Notes in Computer Science* (pp. 299–309). Springer.

Hyland, J. M. E., & Luke, C.-H. (2000). Ong. On full abstraction for PCF: I, II, and III. *Information Computation, 163*(2), 285–408.

Joyal, A., & Moerdijk, I. *Algebraic Set Theory*. Number 220 in London Mathematical Society Lecture Notes. Cambridge University Press.

Joyal, A., Street, R. & Verity, D. (1996). Traced monoidal categories. *Mathematical Proceedings of the Cambridge Philosophical Society, 119*(3), 447–468.

Kelly, G. M., & Laplaza, M. L. (1980). Coherence for compact closed categories. *Journal of Pure and Applied Algebra, 19*, 193–213.

Kleene, S. C. (1935). A theory of positive integers in formal logic. *American Journal of Mathematics, 57,* 153–173.

Kock, A. (1971). Closed categories generated by commutative monads. *Journal of the Australian Mathematical Society, 12*(4), 405–424.

Kock, A. (2012). Commutative monads as a theory of distributions. *Theory and Applications of Categories, 26*(4), 97–131.

Kolmogoroff, A. (1932). Zur Deutung der intuitionistischen Logik. *Mathematische Zeitschrift, 35*(1), 58–65.

Kreisel, G. (1968). A survey of proof theory. *Journal of Symbolic Logic, 33,* 321–388.

Krstić, S., Launchbury, J., & Pavlović, D. (2001). Categories of processes enriched in final coalgebras. In F. Honsell (Ed.), *Proceedings of FoSSaCS 2001*, volume 2030 of Lecture Notes in Computer Science (pp. 303–317). Springer.

Lambek, J. (1968). Deductive Systems and Categories I: Syntactic Calculus and Residuated Categories. *Mathematical Systems Theory, 2,* 287–318.

Lambek, J. (1968). A fixpoint theorem for complete categories. *Mathematische Zeitschrift, 103,* 151–161.

Lambek, J. (1969). Deductive Systems and Categories II: Standard Constructions and Closed Categories. In P. Hilton (Ed.), *Category Theory, Homology Theory and their Applications*, number 86 in Lecture Notes in Mathematics (pp. 76–122). Springer.

Lambek, J. (1972). Deductive systems and categories III: Cartesian closed categories, intuitionist propositional calculus, and combinatory logic. In F. W. Lawvere (Ed.), *Toposes, Algebraic Geometry, and Logic*, number 274 in Lecture Notes in Mathematics (pp. 57–82). Springer.

Lambek, J. (1980). From types to sets. *Advances in Mathematics, 36,* 113–164.

Lambek, J., & Scott, P. (1986). *Introduction to Higher Order Categorical Logic*. Number 7 in Cambridge Studies in Advanced Mathematics. Cambridge University Press.

Lawvere, F. W. (1964). An elementary theory of the category of sets. In *Proceedings of the National Academy of Sciences of the United States of America* (Vol. 52, pp. 1506–1511). Reprinted in *Theory and Applications of Categories* 11(2005) 1–35.

Lawvere, F. W. (1969). Adjointness in foundations. *Dialectica, 23*, 281–296. Reprint in Theory and Applications of Categories. *No.,16*(2006), 1–16.

Li, M., & Vitányi, P. M. B. (1997). *An introduction to Kolmogorov complexity and its applications* (2nd ed.). Graduate texts in computer science. Springer.

Loader, R. (2001). Finitary PCF is not decidable. *Theoretical Computer Science, 266*(1), 341–364.

Manes, E. (1976). *Algebraic Theories*. Number 26 in Graduate Texts in Mathematics. Springer.

Martin-Löf, P. (1966). The definition of random sequences. *Information and Control, 9*(6), 602–619.

Martin-Löf, P. (1975). An intuitionistic theory of types: Predicative part. In H. Rose & J. Sheperdson (Eds.), *Logic Colloquium '73*, number 80 in Studies in Logic and the Foundations of Mathematics (pp. 73–118). North-Holland.

Martin-Löf, P. (1984). *Intuitionistic type theory*. Naples: Bibliopolis.

Mendelson, E. (2015). *Introduction to Mathematical Logic* (6th ed.). Discrete Mathematics and Its Applications, Taylor & Francis.

Milner, R. (1975). Processes: a mathematical model of computing agents. In *Proceedings of the Logic Colloquium. Bristol, July 1973*, volume 80 of Studies in Logic and the Foundations of Mathematics (pp. 157–173). Elsevier.

Milner, R. (1977). Fully abstract models of typed λ-calculi. *Theoretical Computer Science, 4*(1), 1–22.

Milner, R. (1982). *A Calculus of Communicating Systems*. Lecture Notes in Computer Science (Vol. 92). Berlin, Heidelberg: Springer.

Milner, R. (1989). *Communication and concurrency*. Series in Computer Science (Vol. 84). New York: Prentice Hall.

Mitzenmacher, M. (2009). A survey of results for deletion channels and related synchronization channels. *Probability Surveys, 6,* 1–33.

Moggi, E. (1991). Notions of computation and monads. *Information and Computation, 93*(1), 55–92. Selections from 1989 IEEE Symposium on Logic in Computer Science.

Moschovakis, Y. N. (2010). Kleene's amazing Second Recursion Theorem. *Bulletin of Symbolic Logic, 16*(2), 189–239.

Mostowski, A. (1949). An undecidable arithmetical statement. *Fundamenta Mathematicæ, 36,* 143–164.

Muchnik, A., & Vereshchagin, N. (2006) Shannon Entropy versus Kolmogorov Complexity. In G. Dima et al. (Ed.), *Computer Science—Theory and Applications* (pp. 281–291). Berlin, Heidelberg: Springer.

Paré, R. (1971). On absolute colimits. *Journal of Algebra, 19,* 80–95.

Park, D. (1981). Concurrency and automata on infinite sequences. In P. Deussen (Ed.), *Theoretical Computer Science*, number 104 in Lecture Notes in Computer Science (pp. 167–183). Springer.

Pavlovic, D. (1995). Convenient categories of processes and simulations I: modulo strong bisimilarity. In D. Pitt et al. (Eds.), *Category Theory and Computer Science '95*, volume 953 of Lecture Notes in Computer Science (pp. 3–24). Springer.

Pavlovic, D. (1995). Maps I: relative to a factorisation system. *Journal of Pure and Applied Algebra, 99,* 9–34.

Pavlovic, D. (1996). Convenient categories of processes and simulations II: modulo weak and branching bisimilarities. In A. Edalat et al. (Eds.), *Theory and Formal Methods of Computing* (Vol. 96, pp. 156–167). World Scientific.

Pavlovic, D. (1996). Maps II: Chasing diagrams in categorical proof theory. *Journal of the IGPL, 4*(2), 1–36.

Pavlovic, D. (1997). Categorical logic of names and abstraction in action calculus. *Mathematical Structures in Computer Science, 7,* 619–637.

Pavlovic, D. (1998). Guarded induction on final coalgebras. *Electronic Notes in Theoretical Computer Science, 11,* 143–160.

Pavlovic, D. (2011). Relating toy models of quantum computation: comprehension, complementarity and dagger autonomous categories. *Electronic Notes in Theoretical Computer Science, 270*(2), 121–139. arxiv.org:1006.1011.

Pavlovic, D. (2012). Geometry of abstraction in quantum computation. In *Proceedings of Symposia in Applied Mathematics* (Vol. 71, pp. 233–267). arxiv.org:1006.1010.

Pavlovic, D. (2012). Tracing Man-in-the-Middle in monoidal categories. In D. Pattinson & L. Schroeder (Eds.), *Proceedings of CMCS 2012*, volume 7399 of *Lecture Notes in Computer Science* (pp. 191–217). Springer. arXiv:1203.6324.

Pavlovic, D. (2020). Logic of fusion. In C. Talcott et al. (Ed.s), *Proceedings of the Symposium in Honor of Andre Scedrov*, volume 12300 of *Lecture Notes in Computer Science* (pp. 32). Springer.

Pavlovic, D. (2023).

Pavlovic, D., & Abramsky, S. (1997). Specifying interaction categories. In E. Moggi & G. Rosolini (Eds.), *Category Theory and Computer Science '97*, volume 1290 of Lecture Notes in Computer Science (pp. 147–158). Springer.

Pavlovic, D., & Hughes, D. J. D. (2020). The nucleus of an adjunction and the Street monad on monads. arxiv:abs/2004.07353:87. to appear.

Pavlovic, D., & Pratt, V. (1999). On coalgebra of real numbers. *Electronic Notes in Theoretical Computer Science, 19,* 133–148.

Pavlovic, D., & Pratt, V. (2002). The continuum as a final coalgebra. *Theoretical Computer Science, 280*(1–2), 105–122.

Pavlovic, D., & Yahia, M. (2018). Monoidal computer III: A coalgebraic view of computability and complexity. In C. Cîrstea (Ed.), *Coalgebraic Methods in Computer Science (CMCS) 2018—Selected Papers*, volume 11202 of Lecture Notes in Computer Science (pp. 167–189). Springer. arxiv:1704.04882.

Pavlović, D., & Escardó, M. (1998). Calculus in coinductive form. In V. Pratt (Ed.), *Proceedings. Thirteenth Annual IEEE Symposium on Logic in Computer Science* (pp. 408–417). IEEE Computer Society.

Plotkin, G. (1977). LCF considered as a programming language. *Theoretical Computer Science, 5,* 223–255.

Reynolds, J. (1998). *Theories of Programming Languages*. Cambridge University Press.

Rogers, H. Jr. (1987). *Theory of recursive functions and effective computability*. Cambridge, MA, USA: MIT Press.

Russell, B. (1908). Mathematical logic based on the theory of types. *American Journal of Mathematics, 30*, 222–262. Reprinted in van Heijenoort (1967), pages 150–182.

Scott, D. S. (1993). A type-theoretical alternative to ISWIM, CUCH, OWHY. *Theoretical Computer Science, 121*(1–2), 411–440. Technical report written in 1969.

Seldin, J. P., & Hindley, J. R. (Eds.), *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*. London: Academic Press.

Smyth, M. B. (1993). Topology. In S. Abramsky & D. Gabbay (Eds.), *Handbook of Logic in Computer Science* (Vol. 1). Background: Mathematical Structures (pp. 641–761). USA: Oxford University Press, Inc.

Uustalu, T., & Vene, V. (2008). Comonadic notions of computation. *Electronic Notes in Theoretical Computer Science, 203*(5), 263–284.

Vákár, M., Jagadeesan, R., & Abramsky, S. (2018). Game semantics for dependent types. *Information and Computation, 261*(Part), 401–431.

van Glabbeek, R. J. (1993). The Linear Time—Branching Time Spectrum II. The semantics of sequential processes with silent moves. In E. Best (Ed.), *Proceedings of CONCUR '93*, volume 715 of Lecture Notes in Computer Science, (pp. 66–81). Springer.

van Glabbeek, R. J., & Weijland, W. P. (1996). Branching time and abstraction in bisimulation semantics. *Journal of the ACM, 43*(3), 555–600.

van Heijenoort, J. (Ed.). (1967). *From Frege to Gödel: a Source Book in Mathematical Logic* (pp. 1879–1931). Harvard University Press. Reprinted 1971, 1976.

von Neumann, J. (1923). Zur Einführung der transfiniten Zahlen. *Acta litt. Acad. Sc. Szeged, X*(1), 199–208. English translation, "On the introduction of transfinite numbers" in van Heijenoort (1967), pages 393–413.

Wadler, P. (2015). Propositions as types. *Communications of the ACM, 58*(12), 75–84 November.

Zermelo, E. (1930). Über Grenzzahlen Und Mengenbereiche: Neue Untersuchungen Über Die Grundlagen der Mengenlehre. *Fundamenta Mathematicæ, 16,* 29–47.

Zurek, W. H. (1989). Algorithmic randomness and physical entropy. *Physical Review A, 40*(8), 4731.

Part VI
Probabilistic Computation

Chapter 28
(Towards a) Statistical Probabilistic Lazy Lambda Calculus

Radha Jagadeesan

Abstract We study the desiderata on a model for statistical probabilistic programming languages. We argue that they can be met by a combination of traditional tools, namely open bisimulation and probabilistic simulation.

Keywords Lambda calculus · Probability · Open bisimulation · Semantics

28.1 Introduction

The thesis of this article is that open bisimulation is a useful methodology to describe the semantics of statistical probabilistic languages (henceforth StatProb).

van de Meent et al. (2018) provides a textbook introduction to these languages. To a base programming language, e.g., with state, higher order functions and recursion, add two key features to get a statistical probabilistic language.

Forward simulation: The ability to sample/draw values at random from (perhaps continuous) distributions, to describe forward stochastic simulation.

Inference: The ability to condition the posterior distributions of values of variables via observations, and compute them via inference.

van de Meent et al. (2018) describes StatProb-languages as "performing Bayesian inference using the tools of computer science: programming language for model denotation and statistical inference algorithms for computing the conditional distribution of program inputs that could have given rise to the observed program output.". Vákár et al. (2019) illustrates these key features with the following example:

R. Jagadeesan (✉)
School of CTI, DePaul University, Chicago, IL 60604, USA
e-mail: rjagadee@depaul.edu

A. Palmigiano and M. Sadrzadeh (eds.), *Samson Abramsky on Logic and Structure in Computer Science and Beyond*, Outstanding Contributions to Logic 25,
https://doi.org/10.1007/978-3-031-24117-8_28

```
let a = normal(0,2) in
  score(normal(1.1 | a*1, 0.25))
  score(normal(1.9 | a*2, 0.25))
  score(normal(2.7 | a*3, 0.25))
in a
```

The program postulates a linear function $a \times x$ with slope a. Three noisy measurements 1.1, 1.9, 2.7 at $x = 1, 2, 3$ respectively are postulated to have noise modeled as a normal distribution with standard deviation 0.25. The goal is to find a posterior distribution on the slope a. Operationally, a sampler draw from the prior, which is a normal with mean 0 and standard deviation 2. The posterior distribution is constrained using the resulting samples of the three data points. While the forward simulation yields the un-normalised posterior distribution, an inference algorithm is used to approximate its normalization and compute the posterior distribution on a. Arguably, the evaluators that implement such conditioning are the proximate cause for the recent explosion in interest in this area.

In the rest of this introduction, we provide an overview of open bisimulation and the rationale for its use in the current setting.

28.1.1 Open Bisimulation

Consider the study of applicative bisimulation in the pure untyped lazy lambda calculus by Abramsky (1990), Abramsky and Ong (1993) and Ong (1988). In particular, recall Gordon (1995)'s LTS approach to applicative bisimulation.

Internal: A non-value term M has a τ transition to M' if M reduces in one step to M'.

Applicative test: $U = \lambda x.M$ has a transition labeled U' to the application UU'.

Two terms are bisimilar if the associated transition systems are bisimilar, i.e., if their convergence properties agree and each applicative test yields bisimilar terms. In this article, we build on Lassen (1999)'s presentation of Sangiorgi (1994)'s treatment. We follow the "open bisimulation" presentation of Lassen's later work (Lassen, 2005; Lassen & Levy, 2008; Støvring & Lassen, 2009) in a form that validates η-expansion. In contrast to applicative bisimulation, the applicative tests are restricted to (perhaps new) variable names ϕ:

Applicative test: $U = \lambda x.M$ has a transition labeled $\mathsf{app}\ \phi$ to the application $U\phi$.

"Open" application results in a new category of normal forms of the form $\phi\overline{M}$, for a vector of terms $\overline{M}$, forcing the consideration of "external call" tests:

External call: A value $\phi\overline{M}$ has a transition labeled $\mathsf{fcall}\ (\phi, (i, n))$ to $\overline{M}(i)$ (the i'th term in $\overline{M}$), where $|\overline{M}| = n$.

Lassen (1999) shows that Levy–Longo tree equivalence is the biggest bisimulation for such a transition system. In subsequent papers beginning with Lassen (2005), Lassen and his coauthors studies a full suite of programming language features, including control (Lassen, 2006), and state (Støvring & Lassen, 2009). Jagadeesan et al. (2009) adapt this approach to aspect-oriented programming languages. The open-bisimulation approach has also been used to study issues related to types and parametricity, e.g., Lassen and Levy (2008) provide a coinduction principle to simplify proofs about existential types and Jaber and Tzevelekos (2018) show that Strachey parametricity implies Reynolds parametricity.

Why does this approach work so well? Levy and Staton (2014) provides a hint by demonstrating a tight correspondence between open bisimulation and game semantics, Abramsky et al. (2000), Hyland and Ong (2000), and Nickau (1994), albeit for a small calculus with a limited type theory. This has led Paul Levy to coin the evocative (but arguably unnattractive?!) term "Operational Game Semantics" to describe the approaches based on open bisimulation. In this article, we do not further the study needed to resolve the suspicion of a more pervasive and general connection between game semantics and open bisimulation. Rather, what is crucial is the commonality between the semantics of programs in these two approaches, namely:

The semantics of a program is a set of traces.

Remarkably, there is no explicit mention of any higher order structure, such as abstraction and higher order functionals.

On the games side, this is particularly true of the Abramsky et al. (2000) style of games; the Hyland and Ong (2000) games and Nickau (1994) games also appeal to a "pointer" structure that is not intrinsic to a linear trace. The AJM games inherit this feature from the Geometry of Interaction of Girard (1989). In this style of model, a single top level, first order, fixpoint iterator serves as the control engine driving the execution. The complications and subtlety of higher order control flow is handled purely by dataflow over structured tokens, an idea made precise in Abramsky and Jagadeesan (1994).

On the open bisimulation side, the transition system identifies the (set of) paths through the normal form of the program as the invariant of the bisimulation class of the program; a familiar perspective of GOI models as seen in Malacaria and Regnier (1991). The only residue of higher order computation in this first-order world view is perhaps the partition of the labels of the transition system into moves associated with the program and the environment.

This seemingly abstruse technical point acquires particular relevance when considering the semantics of statistical programming languages.

28.1.2 Semantics of Statistical Probabilistic Languages

There are two serious impediments to a systematically structured study of statistical probabilistic languages.

First, the semantics of a simple (pseudo)-random value generators is already troublesome. Standard probability theory does not interact well with higher order functions, since the category of measurable spaces is not cartesian closed. From the domain theoretic perspective, Jones and Plotkin (1989) provide a construction of a probablistic powerdomain, and show that the probabilistic powerdomain of a continuous domain is again continuous. The issues with reconciling the probabilistic powerdomain and function spaces are well known, and remain unresolved to date; see Jung and Tix (1998). This leads Heunen et al. (2017) to identify the mismatch with probabilistic programming languages where "programs may use both higher-order functions and continuous distributions, or even define a probability distribution on functions."

Second, the accurate modeling of conditional probabilities is subtle in the presence of possibly infinite computations. This computation requires normalization that is not monotone w.r.t the order on (sub)-probability distributions; e.g., consider the following weighted sums on distinct constants tru, fls:

$$((0.2, \mathsf{tru}), (0.2, \mathsf{fls})) \leq ((0.2, \mathsf{tru}), (0.3, \mathsf{fls}))$$

whereas after normalization, we get:

$$((0.5, \mathsf{tru}), (0.5, \mathsf{fls})) \not\leq ((0.4, \mathsf{tru}), (0.6, \mathsf{fls}))$$

Thus, the usual view of infinite computations as a supremum of an increasing chain of finite approximants needs to be revisited; we are unable to rely purely on the order structure to compute the probability. This issue may not be fatal in the setting of purely countable probabilities; however, the setting of continuous probabilities perforce requires a notion of limits that is different from the one provided by the order theory of simulation.

Our approach. We address these issues as follows. Our calculus of choice is an untyped lambda calculus with binary choice Λ, with a lazy reduction scheme to weak head normal forms.[1]

We define open probabilistic simulation, $\lesssim$, by a direct combination of two traditional ingredients.

- Open (bi)simulation for lazy lambda calculi, adapted to account for discrete distributions.
  Probability impacts the notion of convergence. Let $\mathtt{Y}$ be a fixpoint combinator, Id be $\lambda x\,.\,x$, and let $M = \lambda x\,.\,\{(0.5, \mathsf{Id}), (0.5, x)\}$. Then, $\mathtt{Y}M$ converges with probability 1 to Id, but the recognition of this fact *requires* the complete infinite unfolding of the computation.
- Probabilistic (weak) simulation for labelled transition systems (Segala & Lynch, 1995).
  This definition incorporates a "splitting lemma" characteristic of probabilities in various treatments in the literature, recognized by Deng et al. (2007) to be the

[1] The choice of an untyped calculus is merely to reduce the syntactic overhead in the treatment.

Kleisli construction on the probabilistic power domain. Probabilistic determinacy is crucial, and ensures that the closure ordinal of the simulation functional is ω.

We demonstrate that simulation is a congruence. In the context of the simple discrete probability embodied in this calculus, this development already shows how the first-order treatment of open bisimulation allows us to sidestep the troublesome interaction of probability and function spaces.

Next, we develop an order theory of simulation. In effect, this study is an examination of $D = \mathscr{P}_{\text{prob}}(D \Rightarrow\ D)$, except for the unfortunate accident that we do not know how to construct such a D! We adapt the finite Levy–Longo trees of Ong (1988) to account for discrete probabilistic sums, and show that it constitutes a basis for $(\Lambda, \lesssim)$. This structure suffices to define "the" Lawson topology (Gierz et al., 1980) on $(\Lambda, \lesssim)$, and define a metric completion that yields a compact Polish space $\overline{\boldsymbol{\Lambda}}$. This has two consequences:

- Since an omega $\lesssim$-chain is a convergent sequence in $\overline{\boldsymbol{\Lambda}}$, this completion makes $\overline{\boldsymbol{\Lambda}}$ into a continuous dcpo. Since $(\Lambda, \lesssim)$ already contains finite valuations, we deduce that $\overline{\boldsymbol{\Lambda}}$ is rich enough to contain continuous probability distributions, along with an approximation theory by finite valuations.
- van Breugel et al. (2003) show that the Lawson topology on the probabilistic powerdomain of a continuous dcpo D coincides with the weak topology on the probabilistic measures of the Lawson topology of D. Convergence in $\overline{\boldsymbol{\Lambda}}$ thus provides a meaningful definition of the convergence of probability measures. In the specific example of normalization, this suggests a compelling operational intuition to describe the results of normalizing the computation represented by a term A. Consider the net of normalized finite evolutions of A; the result of normalization exists if the net is convergent.

Other approaches. Rather than consider a separate probabilistic powerdomain construction, there is a line of work that considers types that are already closed under probability. In this spirit, Danos and Harmer (2002) describe a semantics for a probabilistic extension of Idealized Algol; the key enhancement to traditional game semantics is to move from deterministic strategies to probabilistically deterministic strategies. Danos and Ehrhard (2011) describe probabilistic coherence spaces where types are interpreted as convex sets and programs as power series. Ehrhard et al. (2014) establish a remarkable equational (but not inequational!) full abstraction result for this semantics. Goubault-Larrecq (2019) motivates why such a result holds by showing how to define a "poor mans parallel or" in a probabilistic language.

Sangiorgi and Vignudelli (2019) use environmental bisimulation in a thorough treatment of probabilistic languages. Environmental bisimulation tracks an observer's knowledge about values computed during the bisimulation game with aim of simplifying the delicate proofs of congruence in applicative bisimulation. We gratefully acknowledge their impact on the formalization of the operational aspects of the calculus.

Both the above lines of research resolve the tension between the probabilistic powerdomain and function spaces. However, they do not account for the normalization of probability distributions. It is plausible that our techniques that study the order theory of simulation can be adapted to these settings.

A second line of work is on Quasi Borel Spaces (Vákár et al., 2019; Heunen et al., 2017). In our reading of this approach, it diagnoses the problems of modeling statistical probabilistic languages as arising from the interplay of approximating data and approximating probabilities. So, the radical design idea is just to separate them altogether. Heunen et al. (2017) shows that this category is cartesian closed, and Vákár et al. (2019) shows how to interpret a rich type theory by exploiting an elegant integration with axiomatic domain theory. This approach addresses the full expressiveness of statistical probabilistic languages. However, since we are forced out of the realm of standard measure theory, we are perforce forced to "reinvent the wheel" for probabilistic reasoning. The full integration with classical reasoning techniques of programming languages remains to be resolved in future work.

Our suggested approach for StatProb-languages inherits powerful coinductive reasoning from open bisimulation, approximation techniques from the underlying domain theory and its intimately related metric space, and stays within the confines of classical measure theory. However, as it stands, this paper is merely a promise. Its full realization via a thorough study of a full language with continuous probability distributions in this setting is left to future work.

Samson's impact on this article. This article falls squarely into the areas pioneered by Samson, starting with the origins of open bisimulation from his study of the lazy lambda calculus and game semantics. We conclude this introduction by tracing the (personal) impact of Samson's work even in the matter of probabilistic methods. My interest in probabilistic computation is inspired by my thesis advisor and friend, Prakash Panangaden, an energetic and enthusiastic advocate (Panangaden, 2009) of probabilistic methods in semantics. In analogy to Abramsky (1991a)'s analysis of strong bisimulation, Desharnais et al. (2003) explores a domain equation for strong probabilistic bisimulation. Desharnais et al. (2003)'s study of simulation and this article's analysis of the simulation relation in this article draw heavily on the intuitions inspired by Abramsky (1991b).

28.2 Λ: Probabilistic Lazy Lambda Calculus

We study a lazy lambda calculus with countable formal sums. Despite the title of this section, at this stage, we do not associate formal probabilities with the terms; instead, we view the formal sums of terms merely as weighted terms.

Our technical definitions follow the general style of Sangiorgi and Vignudelli (2019), adapted to our setting of open bisimulation.

28.2.1 Syntax

Λ. SYNTAX.

$\mathtt{p}, \mathtt{q}$	Probabilities in $[0\ldots 1]$	
$\mathsf{A}, \mathsf{B}, \mathsf{C}$ ::=	Distributions of terms	
$\{(\mathtt{p}, M)\}$	$\mathtt{p}$	$x fn(M)$
$\mathsf{A} \uplus \mathsf{B}$	$m(\mathsf{A}) + m(\mathsf{B})$	$fn(\mathsf{A}) \cup fn(\mathsf{B})$
$x, y, z, \phi, \psi, \theta$	Variable Names	
M, N, L ::=	Terms	
x		$\{x\}$
$\lambda x\,.\,\mathsf{A}$		$fn(\mathsf{A}) \setminus \{x\}$
AB		$fn(\mathsf{A}) \cup fn(\mathsf{B})$
$\overline{\mathsf{A}}, \overline{\mathsf{B}}$ ::=	Vector of finite distributions of terms	
S, T ::=	Sets of terms	

$|\overline{\mathsf{A}}|$ is the length of the vector. We use $\overline{\mathsf{A}}(i)$ for the i'th term of $\overline{\mathsf{A}}$ when $1 \leq i \leq |\overline{\mathsf{A}}|$.

$fn(\mathsf{A})$ is the set of free names of A. As usual, we consider terms upto renaming of bound variables.

$m_{\mathsf{A}}(\mathsf{B}) = \mathtt{p}$ if $(\mathtt{p}, \mathsf{B}) \in \mathsf{A}$, and 0 otherwise. $m(\mathsf{A})$ is the full weight of A. We will only consider A such that $m(\mathsf{A}) \leq 1$.

Let $\mathtt{p} \times \mathsf{B} = \{(\mathtt{p} \times \mathtt{q}_i, M_i) \mid (\mathtt{q}_i, M) \in \mathsf{B}\}$ If $\sum_i \mathtt{p}_i \leq 1$, then the *sub-convex combination* $\{\mathtt{p}_i \times \mathsf{A}_i\}$ is a valid multiset of terms. An important special case is written $\mathsf{A} \uplus \mathsf{B}$, that stands for for the weighted term such that $m_{\mathsf{A}\uplus\mathsf{B}}(\mathsf{C}) = m_{\mathsf{A}}(\mathsf{C}) + m_{\mathsf{B}}(\mathsf{C})$, if it exists. Traditional probabilistic choice, written $(0.5, M) \uplus (0.5, N)$, thus stands for $\{(0.5, M), (0.5, N)\}$.

The above grammar only generates finite distributions. The forthcoming reduction semantics can generate countable distributions; we will also use A, B for the countable distributions generated by the reduction semantics.

Definition 1 (*Ordering distributions*) $\mathsf{A} \leq \mathsf{B} \ := \ (\exists \mathsf{C})\ [\mathsf{B} = \mathsf{A} \uplus \mathsf{C}]$.

Thus, if $\mathsf{A} \leq \mathsf{B}$, then $(\forall(\mathtt{p}, M) \in \mathsf{A})\ (\exists(\mathtt{q}, M) \in \mathsf{B})\ \mathtt{p} \leq \mathtt{q}$. Under this order, distributions with countable support form a bounded complete, continuous dcpo. Bounds do not exist only when total weight adds up to more than 1. The least upper bounds of bounded sets, written $\bigsqcup \mathsf{A}_i$, given by $\{(\sup \mathtt{p}_i, M) \mid (\mathtt{p}_i, M) \in \mathsf{A}_i\}$ and a countable basis given by distributions with finite support and rational weights, with the "way-below" relation given by:

$$\mathsf{A} \ll \mathsf{B} \ := \ |\mathsf{A}| \text{ finite },\ (\forall(\mathtt{p}, M) \in \mathsf{A})\ (\exists(\mathtt{q}, M) \in \mathsf{B})\ \mathtt{p} < \mathtt{q}$$

We will often avoid some explicit coercions between terms and their associated point measures, e.g., we will often write $\{(1, M)\}$ simply as M.

28.2.2 Reduction Semantics

We consider a lazy reduction strategy to weak head normal forms. For pure lambda terms, this reduction strategy corresponds to Levy–Longo trees.

The *evaluation contexts* are defined as follows.

$$\mathscr{E}, \mathscr{F} ::= [-] \mid \mathscr{E}\mathsf{A}$$

The small step evaluation relation $\mathsf{A} \rightsquigarrow \mathsf{B}$ below tracks the full ensemble of possible results. The evaluation relation ensures that application is left linear. Since the evaluation is call-by-name, any choices in the argument are resolved separately and independently at the point of use of the argument. We write $\mathsf{A}[x := \mathsf{B}]$ for the usual capture free substitution lifted to sets.

EVALUATION $(\mathsf{A} \rightsquigarrow \mathsf{B})$. $\rightsquigarrow^\star$: TRANSITIVE CLOSURE OF $\rightsquigarrow$.

$$\lambda x\,.\,\mathsf{AB} \rightsquigarrow \mathsf{A}[x := \mathsf{B}]$$

$$\frac{\mathsf{A} \rightsquigarrow \mathsf{A}'}{\mathscr{E}[\mathsf{A}] \rightsquigarrow \mathscr{E}[\mathsf{A}']}$$

$$\mathsf{AB} \rightsquigarrow \biguplus_{(\mathrm{p}, M) \in \mathsf{A}} \{(\mathrm{p}, M\mathsf{B})\}$$

$$\frac{M \rightsquigarrow \mathsf{B}}{(\mathrm{p}, M) \uplus \mathsf{A} \rightsquigarrow (\mathrm{p} \times \mathsf{B}) \uplus \mathsf{A}}$$

We summarize some properties of $\rightsquigarrow$.

- $\rightsquigarrow$ is one-step Church Rosser since there are no critical pairs.
- If $\mathsf{A} \leq \mathsf{B}$ and $\mathsf{A} \rightsquigarrow \mathsf{A}'$, then there exists B' such that $\mathsf{B} \rightsquigarrow \mathsf{B}'$ and $\mathsf{A}' \leq \mathsf{B}'$
- Let A_i be a directed set. Let B_i be a directed set such that $\mathsf{A}_i \rightsquigarrow \mathsf{B}_i$. Then, $\bigsqcup \mathsf{A}_i \rightsquigarrow \bigsqcup \mathsf{B}_i$.
- $\rightsquigarrow$ is linear w.r.t $\uplus$. If $\mathsf{A} = \biguplus_i \mathsf{A}_i$ and $(\forall i)\mathsf{A}_i \rightsquigarrow^\star \mathsf{B}_i$, then $\mathsf{A} \rightsquigarrow^\star \biguplus_i \mathsf{B}_i$.

$\rightsquigarrow^\star$ inherits the linearity and monotonicity properties from $\rightsquigarrow$.

Definition 2 (*Weak head normal forms*) The weak head normal forms are of the form $(\mathrm{p}, \lambda x\,.\,\mathsf{A})$ and $(\mathrm{p}, x\overline{\mathsf{A}})$.

We use $vals(\mathsf{A})$ for the sub distribution of A with support only on the (weak head) normal forms in A.

$$vals(\mathsf{A}) = \{(\mathrm{p}, M) \in \mathsf{A} \mid M \text{ in } whnf\}$$

$vals$ is monotone: if $\mathsf{A} \rightsquigarrow \mathsf{B}$, then $vals(\mathsf{A}) \leq vals(\mathsf{B})$. The result of big-step evaluation is a distribution on normal forms.

Definition 3 (*Big step evaluation*) $whnf(\mathsf{A}) = \bigsqcup \{vals(\mathsf{C}) \mid \mathsf{A} \rightsquigarrow^\star \mathsf{C}\}$.

This evaluation might be necessarily infinite, as shown by the following example, reproduced from the introduction.

Example 4 Let $\mathtt{Y}$ be a fixpoint combinator such that $\mathtt{YM} \rightsquigarrow^\star \mathtt{M(YM)}$, Id be $\lambda x\,.\,x$, and let $M = \lambda x\,.\{(0.5, \mathsf{Id}), (0.5, x)\}$. Then, $\mathtt{YM} \Downarrow \mathsf{Id}$ but for any finite evolution $\mathtt{YM} \rightsquigarrow^\star \mathsf{B}$, $vals(\mathsf{B})$ necessarily assigns a probability strictly less than 1 to Id.

Following Sangiorgi and Vignudelli (2019), we generally use big-step evaluation for definitions and small-step evaluation for proofs.

We summarize the linearity and monotonicity properties of big-step evaluation.

- If $\mathsf{A} \leq \mathsf{B}$, then $whnf(\mathsf{A}) \leq whnf(\mathsf{B})$.
- Let A_i be a directed set. Then, $whnf(\bigsqcup \mathsf{A}_i) = \bigsqcup whnf(\mathsf{A}_i)$.
- If $\mathsf{A} = \biguplus_i \mathsf{A}_i$, then $whnf(\mathsf{A}) = \biguplus_i whnf(\mathsf{A}_i)$.

28.3 Open Simulation

28.3.1 LTS Basis for Open Simulation

We present an LTS on terms as a prelude to defining a notion of simulation on terms, building on Lassen (1999)'s presentation of Sangiorgi (1994)'s treatment. In particular, we follow the "open bisimulation" presentation of Lassen's later work (Lassen, 2005; Lassen & Levy, 2008; Støvring & Lassen, 2009) in a form that validates η-expansion (Jagadeesan et al., 2009).

LTS LABELS

$\varkappa ::= \tau \mid \kappa$	All Labels	
$\tau ::=$	Silent Label	
$\kappa ::=$	Visible Labels	
	$\mathsf{fcall}\ (\phi, (i, n))$	Term uses arg i out of n of context function ϕ
	$\mathsf{app}\ \phi$	Context calls term with argument ϕ ($dn = \{\phi\}$)
	$\lambda\lightning$	Convergence

The choice of labels is determined by the possibilities available to the context to interact with the term. In Gordon (1995)'s terminology, the visible label $\mathsf{fcall}(,)$ has an *active* component (representing actions initiated by the term), whereas $\mathsf{app}\ \phi$ is completely *passive* (representing actions initiated by the environment). $\lambda\lightning$ is also passive.

The silent label τ stands for internal computation of the term.

LTS ON TERMS $\quad M \xrightarrow{\varkappa} \mathsf{B}$

(SILENT)

$$\frac{\mathsf{A} \rightsquigarrow \mathsf{B}}{\mathsf{A} \xrightarrow{\tau} \mathsf{B}}$$

(RETURN)

$$\frac{}{\lambda x\,.\,M \xrightarrow{\mathsf{ret}\ \phi} (\lambda x\,.\,M)\phi}$$

(CONVERGE)

$$\frac{}{\lambda x\,.\,M \xrightarrow{\lambda\lightning} \mathsf{Id}}$$

(CALL)

$$\frac{0 \leq i \leq |\overline{\mathsf{A}}|}{\phi\overline{\mathsf{A}} \xrightarrow{\mathsf{fcall}\ (\phi,(i,|\overline{\mathsf{A}}|))} \overline{\mathsf{A}}(i)}$$

(LIFT-κ)

$$\frac{M \xrightarrow{\kappa} N}{(\mathtt{p}, M) \xrightarrow{\kappa} (\mathtt{p}, N)}$$

The LTS affords priority to the internal reductions of the term, since the visible transitions are only applicable to weak-head normal forms.

$\mathsf{app}\ \phi$ performs applicative tests. Rather than providing a term as an argument for the applicative test, this rule provides a (possibly fresh) symbolic argument ϕ. $\lambda\lightning$ facilitates the measurement of the weight of convergence to an abstraction.

$\mathsf{fcall}\ (\phi, (i, n))$ records the call to an external and unknown parameter ϕ. It chooses an argument, the one at the i'th position, to inspect further. The addition of parameter n provides a way to distinguish argument vectors of different lengths. We adopt the convention that $\overline{\mathsf{A}}(0)$ is Id. Mirroring $\lambda\lightning$, the label $\mathsf{fcall}\ (\phi, (0, n))$ is a way to measure the weight of convergence to an head application of ϕ to n arguments.

Notice that the LTS is being defined on terms rather than formal sums of terms. This is because of a vexing difference between the usual weak-head normal forms $\lambda x\,.\,\mathsf{A}$ and the ones introduced by open application $\phi\overline{\mathsf{A}}$, that prevents us from following the correct approach of directly building a labeled Markov chain by lifting these LTS transitions.

Example 5 (*Linearity and Weak head normal forms*) Application is linear in the head variable. This is reflected in a candidate LTS extension to formal sums of abstractions as follows:

$$\frac{M_i = \lambda x\,.\,\mathsf{A}_i,\ M_i \xrightarrow{\kappa} \mathsf{B}_i}{\mathsf{A} \uplus \mathsf{B} \xrightarrow{\kappa} \uplus_i \mathsf{B}_i}$$

This rule permits us to show that abstraction is linear w.r.t formal sums. i.e..

$$\lambda x\,.\,\overline{\mathsf{A}} \uplus \overline{\mathsf{B}} \sim \lambda x\,.\,\overline{\mathsf{A}} \uplus \lambda x\,.\,\overline{\mathsf{B}}$$

where the matching of transitions of the left by the right makes essential use of the lifting of transitions on terms to transitions on formal sums of terms.

Alas, this lifting is not sound for formal sums of the form $\phi\overline{\mathsf{A}} \uplus \phi\overline{\mathsf{B}}$. To see this, consider $(x\mathsf{tru}\ \mathsf{fls}) \uplus (x\mathsf{fls}\ \mathsf{tru})$ and $(x\mathsf{fls}\ \mathsf{fls}) \uplus (x\mathsf{tru}\ \mathsf{tru})$, where $\mathsf{tru}, \mathsf{fls}$ are the Church booleans. Setting x to be the Church terms for exclusive-OR differentiates them. However, both formal sums have transitions with labels $\mathsf{fcall}\ (x, (i, |2|))$ to $\mathsf{tru} \uplus \mathsf{fls}$ for $i \in \{1, 2\}$, if we permitted unrestricted linearity of transition system.

28.3.2 Probabilistic Lifting

In order to establish the proper foundations for our forthcoming definition of simulation, we take a short detour into lifting of relations to probability distributions. We try to make the material in this subsection self-contained.

Let $\mathscr{R}$ be a binary relation on a countable X. We use P, Q for discrete sub probability distributions, and $|\mathsf{P}|, |\mathsf{Q}|$ for their carrier sets. We write $\sum_i \mathtt{p}_i \times \mathsf{a}_i$ for the sub-convex combination of point measures at a_i. Lift binary relations on X are to probability distributions as follows.

Definition 6 (*Probabilistic lift*) Let $\mathsf{P} = \sum_i \mathtt{p}_i \times \mathsf{a}_i, \mathsf{Q} = \sum_j \mathtt{q}_j \times \mathsf{b}_j$. Then: $\mathsf{P}\ \overline{\mathscr{R}}\ \mathsf{Q}$ if there is a matching $\mathsf{m}_{i,j}$ such that:

$$\mathsf{m}_{i,j} \neq 0 \Rightarrow \mathsf{a}_i\ \mathscr{R}\ \mathsf{b}_j$$
$$\sum_i \mathsf{m}_{i,j} = \mathtt{p}_i$$
$$\sum_j \mathsf{m}_{i,j} \leq \mathtt{q}_j$$

In contrast to the definitions of Segala and Lynch (1995); Deng et al. (2007), the inequality in the last line above permits us to address subprobabilities.

The following lemma shows that $\overline{\mathscr{R}}$ can be understood as a Kleisli construction on the probabilistic power domain of Jones and Plotkin (1989). This"splitting lemma" for (countable) probability distributions is a small generalization from the finite in Chap. 4 of Jones and Plotkin (1989).

Lemma 7 $\mathsf{P}\ \overline{\mathscr{R}}\ \mathsf{Q}$ *iff for all* $\mathsf{S} \subseteq |\mathsf{P}|$ *such that* $(\mathsf{S}\ ;\ \mathscr{R}) \bigcap |\mathsf{P}| = \mathsf{S}$*, it is the case that:*

$$\sum_{\mathsf{a}_k \in \mathsf{S}} \mathtt{p}_k \leq \sum_{\mathsf{b}_j \in |\mathsf{Q}|} \mathtt{q}_j \mid \mathsf{b}_j \in \mathsf{S};\ \mathscr{R}$$

Proof We follow Chap. 4 of Jones and Plotkin (1989) essentially verbatim, using the countable version of MaxFlow-MinCut from Aharoni et al. (2011); in any (possibly infinite) network there exists an orthogonal pair of a flow and a cut.[2] We use this result for very simple countably infinite directed bipartite graph with *no* infinite paths.

The vertices of the graph are given by $\{\mathsf{src}, \mathsf{sink}\} \cup |\mathsf{P}| \cup |\mathsf{Q}|$. src is connected to a_i with capacity p_i. b_i is connected to sink with capacity q_i. a_i is connected to b_j with a very high capacity, say 100.

Consider (F,C), the orthogonal pair of a flow and a cut in this graph.

If all the edges from the src are in C, F validates the criterion, yielding the result for the forward direction.

If not. Since the capacity of edges from $|\mathsf{P}|$ into $|\mathsf{Q}|$ is large, they cannot be exhausted. So, C consists only of edges from the src or into the sink. The set of vertices connected to src in C provides evidence of violation of the criterion.

In the forthcoming development, we use the matching definition in the precongruence proof and the characterization of Lemma 7 almost everywhere else.

Since $\mathscr{R} \ ; \ \bigcap_i \mathscr{R}_i = \bigcap_i \mathscr{R} \ ; \ \mathscr{R}_i$, we deduce:

Corollary 8

$$\overline{\bigcap_i \mathscr{R}_i} = \bigcap_i \overline{\mathscr{R}}$$

Let $\sum_i \mathrm{p}'_i \times \mathsf{a}_i \le \sum_i \mathrm{p}_i \times \mathsf{a}_i$ if $\mathrm{p}'_i < \mathrm{p}_i$ for all i. It is immediate that:

Corollary 9

$$\mathsf{P}\, \overline{\mathscr{R}}\, \mathsf{Q} \iff (\forall \mathsf{P}' \ll \mathsf{P})\mathsf{P}'\, \overline{\mathscr{R}}\, \mathsf{Q}$$

28.3.3 Simulation via Coinduction

$\Lambda_{\mathsf{n}} \subseteq \Lambda$ is the subset of weak-head normal forms. Λ_{nabs} restricts to formal sums of only abstractions. Λ_{nopa} restricts to point formal sums of open applications.

Definition 10 (Λ_{n})

$$\begin{aligned}
\mathsf{A} \in \Lambda_{\mathsf{n}} &:= [\mathsf{A} \leadsto^{\star} N \Rightarrow \mathsf{A} = N] \\
\mathsf{A} \in \Lambda_{\mathsf{nabs}} &:= [\forall (\mathrm{p}, M) \in \mathsf{A}, M \text{ is an abstraction}] \\
\mathsf{A} \in \Lambda_{\mathsf{nopa}} &:= [(\exists \mathrm{p}, \overline{\mathsf{A}}, x)\ \mathsf{A} = (\mathrm{p}, x\ \overline{\mathsf{A}})] \\
\mathsf{rel} &:= (\Lambda_{\mathsf{nabs}} \times \Lambda_{\mathsf{nabs}}) \bigcup (\Lambda_{\mathsf{nopa}} \times \Lambda_{\mathsf{nopa}})
\end{aligned}$$

The very type of rel addresses the linearity issue of Example 5.

Definition 11 (*Weak max transitions:* $\xrightarrow[max]{\varkappa}\!\!\!\!\rightarrow \ :\subseteq \Lambda_{\mathsf{n}} \times \Lambda_{\mathsf{n}}$)

[2] A flow and a cut are orthogonal if the flow exhausts the capacity of the edges that go forward in the cut, even as it assigns 0 flow to the edges that flow back in the cut.

$\mathsf{A} \xrightarrow[max]{\tau} whnf(\mathsf{A})$.

$\mathsf{A} \xrightarrow[max]{\kappa} \mathsf{C}$, if $\mathsf{A} \xrightarrow{\kappa} \mathsf{B} \xrightarrow[max]{\tau} \mathsf{C}$

The unique target of $\xrightarrow[max]{\varkappa}$ exemplifies the probabilistic determinacy of our language.

We consider binary relations of terms that have a kernel in rel.

Definition 12 REL is the set of relations $\mathscr{R}$ over Λ that are induced by a kernel in rel as follows:

$$\mathscr{R} = \{(\mathsf{A}, \mathsf{B}) \mid (whnf(\mathsf{A}), whnf(\mathsf{B})) \in \overline{\mathscr{R} \cap \mathsf{rel}}\}$$

REL is a complete lattice under subset with maximum element induced by rel and arbitrary least upper bounds induced by the union of their kernels. Thus, our definition of simulation fits in the framework described by Pous (2016) for simulation relations and upto-reasoning.

Definition 13 (*Simulation Functional*) Define a monotone operator *sim* on REL as follows. Let $\mathscr{R} \in \mathsf{REL}$. Let $(\mathsf{A}, \mathsf{B}) \in \mathsf{rel}$.

$\mathsf{A}\, sim\, (\mathscr{R})\, \mathsf{B}$ if whenever $\mathsf{A} \xrightarrow[max]{\varkappa} \mathsf{A}'$ and $\mathsf{B} \xrightarrow[max]{\varkappa} \mathsf{B}'$, it is the case that $\mathsf{A}'\, \overline{\mathscr{R}}\, \mathsf{B}'$.

In the above definition, we only define $sim(\mathscr{R}) \subseteq \mathsf{rel}$, since the extension to REL is unique. There is a maximum simulation, that we write as $\lesssim$. Let $\sim = \lesssim \cap \lesssim^{-1}$. $\lesssim$ is extended to $\Lambda \times \Lambda$ by extension by closing and reduction.

We list out some useful properties of *sim*. As a consequence of the inequality in the definition of $\overline{\mathscr{R}}$, we deduce that:

$$\leq\, ;\, sim(\mathscr{R})\, ;\, \leq = sim(\mathscr{R})$$

and

$$\overline{sim(\mathscr{R})} = sim(\mathscr{R})$$

$sim(\mathscr{R})$ is transitive if $\mathscr{R}$ is.

From Lemma 9, $sim(\mathscr{R})$ is admissible, in the sense of Abramsky and Jung (1995), for $\leq$-chains, i.e.: if $\{\mathsf{A}_i\}$ is a $\leq$-chain, and $(\forall i)\ \mathsf{A}_i\ sim(\mathscr{R})\ \mathsf{B}$, then $\bigsqcup \{\mathsf{A}_i\} sim(\mathscr{R}) \mathsf{B}$. This provides a finitary operational perspective that is crucial to the forthcoming precongruence proof. A motivating example is to show that $\mathsf{Id} \lesssim$ `YM` from Example 4; notably, none of the finite unwindings of `YM` suffice to reach the limit Id. Since every finite approximant to A can be achieved in a finite computation, admissibility ensures that in order for A to be simulated by B, it suffices for the finite computations from A to be simulated by (finite computations of) B.

Finally, as a consequence of probabilistic determinacy and Lemma 8, we deduce that the closure ordinal of $\lesssim$ is ω; i.e., let

$$\begin{aligned} \lesssim_0 &= \Lambda \times \Lambda \\ \lesssim_{k+1} &= sim(\ \lesssim_k\ ) \end{aligned}$$

Then, $\lesssim = \bigcap_k \lesssim_k$.

These properties translate to $\lesssim$ as follows.

- $\lesssim$ is a preorder with least element Ω, where Ω is any term that does not converge to a weak head normal form, e.g., $(\lambda x\,.\,xx)(\lambda x\,.\,xx)$.
- $() \sim \Omega$. In probabilistic programming languages, divergence causes loss of probability.
- $\leq \mathbin{;} \lesssim \mathbin{;} \leq \,=\, \lesssim$
- $\mathsf{A} \sim \ whnf(\mathsf{A})$; Sangiorgi and Vignudelli (2019) call this principle "simulation up-to lifting".
- Simulation is closed under under sub-convex combinations, i.e., if $\mathsf{A}_i \lesssim \mathsf{B}_i$ forall i, then $\biguplus_i \mathrm{p}_i \times \mathsf{A}_i \lesssim \biguplus_i \mathrm{p}_i \times \mathsf{B}_i$; thus, $\lesssim$ is a precongruence for choice.

28.3.4 Precongruence Proofs

We will work with substitutions generated by the following grammar.

SUBSTITUTIONS

σ, σ' ::=	Substitutions
$[x := \mathsf{A}]$	
$\sigma\sigma'$	

We will use $dom(\sigma)$ for the domain of a substitution σ. We write $\sigma(x) \uparrow$ if $x \notin dom(\sigma)$, and $\sigma(x) \downarrow$ otherwise. We will only use substitutions that satisfy the following restriction:

$$(\forall x \in dom(\sigma))\ x \notin \bigcup_{y \in dom(\sigma)} fn(\sigma(y))$$

The substitutive version of simulation is defined as follows.

Definition 14 $\sigma \lesssim \tau$ if $dom(\sigma) = dom(\tau)$ and for all $x \in dom(\sigma)$ $[\sigma(x) \lesssim\ \tau(x)]$.

$\trianglelefteq$ is defined as the smallest relation that satisfies for all M, M', σ, τ,

$$\mathsf{A} \lesssim \mathsf{A}', \sigma \lesssim \tau \implies \mathsf{A}\sigma \trianglelefteq \mathsf{A}'\tau$$

In particular, $\lesssim \subseteq \trianglelefteq$.

In the Appendix 1, we prove that $\trianglelefteq$ is a postfixed point of *sim*.

Lemma 15 $\trianglelefteq\ \subseteq sim(\ \trianglelefteq\ )$.

Theorem 16 *(Simulation is a precongruence)* $\lesssim$ *is a precongruence for all program combinators.*

Proof The proof follows from definitions for abstraction and $\uplus$. For application, we are given $\mathsf{A}_i, \mathsf{B}_i$ for $i = 1, 2$ with $\mathsf{A}_1 \lesssim \mathsf{A}_2$ and $\mathsf{B}_1 \lesssim \tau_2$. Use above Lemma 15 with the term xy and substitutions σ_i, for $i = 1, 2$ such that $\sigma_i(x) = \mathsf{A}_i, \sigma_i(y) = \mathsf{B}_i$.

28.4 Building a Polish Space

We adapt the definition of "compact" trees of Ong (1992) to a setting with weighted terms, by incorporating distributions with finite support.

Λ_{b}

$\mathtt{p}$	Rational Probabilities in $[0 \ldots 1]$
C, D ::=	Finite Distributions of terms
$\{(\mathtt{p}, L)\}$	
$\mathsf{C} \uplus \mathsf{D}$	
M ::=	Terms
Ω	
$\lambda x\,.\,M$	
$y\overline{\mathsf{C}}$	
$\overline{\mathsf{C}}$ ::=	Finite vector of finite distributions of terms

When restricted to Longo trees, the definition coincides with Ong (1992), also Chap. 2 of Ong (1988). In this section, we will restrict the use of $\mathsf{C}, \mathsf{D}, \ldots$ for elements of Λ_{b}, whereas $\mathsf{A}, \mathsf{B}, \ldots$ will be used for general terms of Λ.

Definition 17 (*Approximants*) Let $\mathsf{A} \in \Lambda$. For each $n \geq 0$, define $(\mathsf{A})^n \subseteq \Lambda_{\mathsf{b}}$, inductively as follows.

$$
\begin{array}{lll}
\Omega & \in & (\mathsf{A})^0 \\
(\mathsf{A})^k & \subseteq & (\mathsf{A})^{k+1} \\
\mathsf{C} & \in & (\mathsf{A})^k, \ \text{ if } \mathsf{C} \in \overline{(\mathsf{A})^k} \\
\mathsf{C} & \in & (\mathsf{A})^k, \ \text{ if } \ \mathsf{B} \ll \mathit{whnf}(\mathsf{A}), \ \mathsf{C} \in (\mathsf{B})^k \\
\overline{\mathsf{C}} & \in & (\overline{\mathsf{A}})^k, \ \text{ if } |\overline{\mathsf{C}}| = |\overline{\mathsf{A}}|, \ (\forall 1 \leq i \leq |\overline{\mathsf{A}}|)\ \overline{\mathsf{C}}(i) \in \overline{\mathsf{A}}(i) \\
(\mathtt{p}, \lambda x\,.\,\mathsf{C}) & \in & ((\mathtt{q}, \lambda x\,.\,\mathsf{A}))^{k+1}, \quad \text{if } \mathtt{p} < \mathtt{q}, \mathtt{p} \times \mathsf{C} \in (\mathtt{q} \times \mathsf{A})^k \\
(\mathtt{p}, x\overline{\mathsf{C}}) & \in & ((\mathtt{q}, x\overline{\mathsf{A}}))^{k+1}, \quad \text{if } \mathtt{p} < \mathtt{q}, |\overline{\mathsf{A}}| = |\overline{\mathsf{C}}|, \ (\forall 1 \leq i \leq |\overline{\mathsf{A}}|)\ \mathtt{p} \times \overline{\mathsf{C}}(i) \in (\mathtt{q} \times \overline{\mathsf{A}}(i))^k
\end{array}
$$

Definition 18 $\mathsf{C} \lll \mathsf{A}$ if $(\exists k)\ \mathsf{C} \in (\mathsf{A})^k$

We identify the key properties of $\lll$ following Proposition 2.3 of Lawson (1998).

Lemma 19 *Let* $\mathsf{C}, \mathsf{D} \in \Lambda_{\mathsf{b}}$.

$$\mathsf{C} \lesssim \mathsf{A} \impliedby \mathsf{C} \lll \mathsf{A} \tag{28.1}$$

$$\mathsf{C} \lll \mathsf{B} \impliedby \mathsf{C} \lesssim \mathsf{D} \lll \mathsf{A} \lesssim \mathsf{B} \tag{28.2}$$

$$\mathsf{C} \lll \mathsf{A} \implies (\exists \mathsf{D})\mathsf{C} \lll \mathsf{D} \lll \mathsf{A} \tag{28.3}$$

Proof All proofs proceed by routine induction on k such that $\mathsf{C} \in (\mathsf{A})^k$.

$\lll$ determines the simulation order.

Lemma 20

$$\mathsf{A} \lesssim \mathsf{B} \iff (\forall \mathsf{C})\ [\mathsf{C} \lll \mathsf{A} \implies \mathsf{C} \lesssim \mathsf{B}]$$

Proof The forward direction follows from Eq. 28.1 of Lemma 19 and transitivity of $\lesssim$.

For the converse, we prove by induction on k that:

$$(\forall k)\ [\mathsf{A} \lesssim_k \mathsf{B}] \impliedby (\forall \mathsf{C} \in (\mathsf{A})^k)\ [\mathsf{C} \lesssim \mathsf{B}]$$

The base case is immediate.

Consider the inductive case at $k+1$. Since $\overline{\lesssim} = \lesssim$, and $\overline{\lll} = \lll$, it suffices to prove for $\mathsf{A}, \mathsf{B}, \mathsf{C}$ such that $((\mathsf{C}, \mathsf{A}), (\mathsf{C}, \mathsf{B})) \subseteq \mathsf{rel}$. Simplifying a bit further, using linearity of abstraction w.r.t formal sums, we deduce that it suffices to consider just the following two cases.

- $M = (\mathtt{p}, x\ \overline{\mathsf{A}})$, $N = (\mathtt{q}, x\ \overline{\mathsf{B}})$, where $|\overline{\mathsf{A}}| = |\overline{\mathsf{B}}|$. We are aiming to prove $M \lesssim_{k+1} N$.

  Consider $L = (\mathtt{p}', x\ \overline{\Omega})$, for any $\mathtt{p}' < \mathtt{p}$. $L \in (\mathsf{A})^1 \subseteq (\mathsf{B})^1$. So, we deduce that $\mathtt{p} \le \mathtt{q}$.
  Let $1 \le i \le |\overline{\mathsf{A}}|$. We need to show that $\overline{\mathsf{A}}(i) \sim_k \overline{\mathsf{B}}(i)$. By induction hypothesis, it suffices to show that $(\overline{\mathsf{A}}(i))^k \subseteq (\overline{\mathsf{B}}(i))^k$. Let $\mathsf{D} \in (\overline{\mathsf{A}}(i))^k$. Then, $L = (\mathtt{p}', x\overline{\mathsf{C}}) \in (\overline{\mathsf{B}}(i))^{k+1}$, where $\mathtt{p}' < \mathtt{p}$ and $\overline{\mathsf{C}}(j) = \mathsf{D}$, $i = j$ and Ω otherwise. By induction hypothesis, $L \in (\overline{\mathsf{B}})^{k+1}$. Thus, we deduce that $\mathsf{D} \in (\overline{\mathsf{B}}(i))^k$.
- $M = (\mathtt{p}, \lambda x\,.\,\mathsf{A})$, $N = (\mathtt{q}, \lambda x\,.\,\mathsf{B})$. We are aiming to prove $M \lesssim_{k+1} N$.

  Consider $L = (\mathtt{p}', \lambda x\,.\,\Omega)$, for any $\mathtt{p}' < \mathtt{p}$. $L \in (\mathsf{A})^1 \subseteq (N)^1$. Thus, $\mathtt{p} \le \mathtt{q}$.

  We need to show that $\mathsf{A} \sim_k \mathsf{B}$. By induction hypothesis, it suffices to show that $(\mathsf{A})^k \subseteq (\mathsf{B})^k$. Let $\mathsf{D} \in (\mathsf{A})^k$. Then $(\mathtt{p}', \lambda x\,.\,\mathsf{D}) \in (\mathsf{A})^{k+1} \subseteq (\mathsf{B})^{k+1}$. Thus, we deduce that $\mathsf{D} \in (\mathsf{B})^k$.

Corollary 21 $\mathsf{A} \lesssim \mathsf{B} \iff (\forall \mathsf{C})\ [\mathsf{C} \lll \mathsf{A} \implies \mathsf{C} \lll \mathsf{B}]$

Proof The forward direction comes from Eq. 28.2. The backward direction follows from Eq. 28.1 and Lemma 20.

Definition 22 (*Lawson topology*) $\boldsymbol{\Lambda} = \Lambda/\sim$. Let

$$\mathsf{C} \Uparrow = \{\mathsf{A} \mid \mathsf{C} \lll \mathsf{A}\}$$
$$\mathsf{C} \uparrow = \{\mathsf{A} \mid \mathsf{C} \lesssim \mathsf{A}\}$$

The L-topology on $\boldsymbol{\Lambda}$ has subbasic open sets $\{\mathsf{C} \Uparrow, \boldsymbol{\Lambda} \setminus \mathsf{C} \uparrow \mid \mathsf{C} \in \Lambda_{\mathsf{b}}\}$.

$\mathsf{C} \Uparrow$ is up-closed w.r.t $\lesssim$, and $\mathsf{A} \downarrow$ is closed in the above topology.

Lemma 23 – $\mathsf{C} \Uparrow \,;\, \lesssim = \mathsf{C} \Uparrow$.
– $\mathsf{A} \downarrow = \{\mathsf{B} \mid \mathsf{B} \lesssim \mathsf{A}\} = \bigcap \{(\boldsymbol{\Lambda} \setminus \mathsf{C} \Uparrow) \mid \Lambda_{\mathsf{b}} \ni \mathsf{C} \not\lll \mathsf{A}\}$ *is closed.*

Proof The first item follows from Eq. 28.2.

Let $\mathsf{C} \not\lll \mathsf{A}$. $\boldsymbol{\Lambda} \setminus \mathsf{C} \Uparrow$ contains A. Let $\mathsf{B} \lesssim \mathsf{A}$. If $\mathsf{C} \lll \mathsf{B}$, then by Eq. 28.2, $\mathsf{C} \lll \mathsf{A}$, a contradiction. Thus, $\lesssim \,;\, (\boldsymbol{\Lambda} \setminus \mathsf{C} \Uparrow) = (\boldsymbol{\Lambda} \setminus \mathsf{C} \Uparrow)$, and we deduce that $\mathsf{A} \downarrow \subseteq \bigcap \{(\boldsymbol{\Lambda} \setminus \mathsf{C} \Uparrow) \mid \Lambda_{\mathsf{b}} \ni \mathsf{C} \not\lll \mathsf{A}\}$. By Corollary 21, for any $\mathsf{B} \not\lesssim \mathsf{A}$, there is a $\mathsf{D} \in \Lambda_{\mathsf{b}}$ such that $\mathsf{D} \lll \mathsf{B}$, $\mathsf{D} \not\lll \mathsf{A}$. Thus, $\mathsf{A} \downarrow \supseteq \bigcap \{(\boldsymbol{\Lambda} \setminus \mathsf{C} \Uparrow) \mid \Lambda_{\mathsf{b}} \ni \mathsf{C} \not\lll \mathsf{A}\}$.

We are now able to mimic the statement and proof of proposition 4.3 in Lawson (1998).

Lemma 24 $(\boldsymbol{\Lambda}, \mathsf{L} - topology)$ *is Hausdorff, regular, and second countable.*

Proof Λ_{b} is countable, so second countability follows.

To prove Hausdorff, let $\mathsf{A} \not\lesssim \mathsf{B}$. By Lemma 20, there exists C such that $\mathsf{C} \lll \mathsf{A}$ and $\mathsf{C} \not\lesssim \mathsf{B}$. The required (subbasic) open sets are $\mathsf{C} \Uparrow$, $\boldsymbol{\Lambda} \setminus \mathsf{C} \uparrow$.

For regularity, it suffices to show that each subbasic open set containing a point contains a closed neighborhood of the point.

– Let $\mathsf{A} \in \mathsf{C} \Uparrow$. Then, there exists $\mathsf{C} \lll \mathsf{D} \lll \mathsf{A}$. The required closed set is $\mathsf{D} \uparrow$.
– Let $\mathsf{A} \in \boldsymbol{\Lambda} \setminus \mathsf{C} \uparrow$. The required closed set is $\mathsf{C} \downarrow$.

As a corollary, $(\boldsymbol{\Lambda}, \mathsf{L} - topology)$ is metrizable.

Definition 25 Let $\overline{\boldsymbol{\Lambda}}$ be the completion of the metric space $(\boldsymbol{\Lambda}, \mathsf{L} - topology)$.

$\overline{\boldsymbol{\Lambda}}$ is a bounded Polish space; hence, compact. Thus, we are finally ready to formally view the weighted terms $\mathsf{A}, \mathsf{B}, \ldots$ as (countable) sub-convex sums of unit masses. Furthermore, it also serves as an appropriate universe to interpret a language with fully general continuous distributions.

28.5 Conclusions and Future Work

This article establishes the basic ingredients of a model for statistical probabilistic programming languages. In contrast to the current research, we develop the foundations using a combination of traditional tools, namely open bisimulation and probabilistic simulation.

These ideas remain but a promise, until used in a thorough investigation of the semantics of a programming language in this paradigm. Not least because the "competing" methods of Quasi Borel Spaces (Vákár et al., 2019; Heunen et al., 2017) provide precisely such a thorough investigation that includes evaluation strategies (call by value vs call by name), recursive types and continuous distributions. We take hope from the extant research that demonstrates that open bisimulation is amenable to all these features and also provides excellent accounts of state and parametricity.

The coinductive and metric foundations of our approach provide the opportunity to explore principled mechanisms for approximate reasoning. The opportunities for robust and approximate reasoning abounds in StatProb languages, driven by the desire to accommodate symbolic reasoning engines. For example:

– Approximating continuous distributions by other continuous distributions or finite distributions to simplify symbolic reasoning
– Approximating infinite computations by (large enough) finite unwindings

This motivates the desire to investigate coinductive principles for approximate reasoning, in the spirit of Desharnais et al. (2004)'s explorations into metric bisimulations for (concurrent) labelled Markov chains.

Acknowledgements I wish to thank Samson Abramsky for two decades of mentoring and friendship. This material is based upon work supported by the National Science Foundation under Grant No. 1617175. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

Appendix 1: Proof of Substitution Lemma

We use the following notation.

– $(\overline{\mathrm{p}}, \lambda\overline{x}\,.\,\overline{\mathsf{B}})$ for the weighted term $\{\overline{\mathrm{p}}(i) \times \lambda x\,.\,\overline{\mathsf{B}}(i)\}$. Thus, $(\overline{\mathrm{p}}, \lambda\overline{x}\,.\,\overline{\mathsf{A}})$ has a $\lambda\lightning$ transition to $(\sum_i \overline{\mathrm{p}}(i), \lambda x\,.\,x)$ and $\mathsf{ret}\ \phi$ transition to $\{\overline{\mathrm{p}}(i) \times \overline{\mathsf{B}}(i)[x := \phi]\}$ that we write as $(\overline{\mathrm{p}}, \overline{\mathsf{B}}[x := \phi])$.
– $\overline{\mathsf{A}} \lesssim \overline{\mathsf{B}}$, if $|\overline{\mathsf{A}}| = |\overline{\mathsf{B}}|$, and $(\forall 1 \leq i \leq |\overline{\mathsf{A}}|)\ \overline{\mathsf{A}}(i) \lesssim \overline{\mathsf{C}}(i)$

Lemma 26 *Let* $\mathsf{A}_1\sigma_1 \trianglelefteq \mathsf{A}_2\sigma_2$*, and whnf* $(\mathsf{A}_1\sigma_1) = \mathsf{A}_1\sigma_1$*. Then,* $\mathsf{A}_1\sigma_1$ *sim*$(\trianglelefteq)$ $\mathsf{A}_2\sigma_2$

Proof Since $\overline{sim(\trianglelefteq)} = sim(\trianglelefteq)$, it suffices to consider the following three cases for the shape of $\mathsf{A}_1\sigma_1$.

1. $\mathsf{A}_1 = (\overline{\mathrm{p}}, \lambda\overline{x}\,.\,\overline{\mathsf{B}})$
   By alpha renaming, we can assume that x is chosen such that $\sigma_i(x)\uparrow$, so that $\sigma_2[x := \phi]$ and $\sigma_1[x := \phi]$ are valid substitutions. Thus, $\mathsf{A}_1\sigma_1 = (\overline{\mathrm{p}}, \lambda\overline{x}\,.\,\overline{\mathsf{B}}\sigma_1)$. The two immediate transitions are:

   – Labeled $\lambda\lightning$ to $(\sum_i \overline{\mathrm{p}}(i), \lambda x\,.\,x)$.

– Labeled $\mathsf{ret}\ \phi$ to $(\overline{\mathrm{q}}, \overline{\mathsf{B}}\sigma_1)[x := \phi]$

By $\mathsf{A}_1 \lesssim \mathsf{A}_2$, $(\overline{\mathrm{q}}, \lambda\overline{x}\,.\overline{\mathsf{B}}') \leq \mathit{whnf}(\mathsf{A}_2)$, and $\overline{\mathsf{B}} \lesssim \overline{\mathsf{B}}'$, where $\sum_i \sum_i \overline{\mathrm{p}}(i) \leq \sum_i \overline{\mathrm{q}}(i)$. $(\overline{\mathrm{q}}, \lambda\overline{x}\,.\overline{\mathsf{B}}')$ has two immediate transitions as follows:

– Labeled $\lambda\lightning$ to $(\sum_i \overline{\mathrm{q}}(i), \lambda x\,.x)$.
– Labeled $\mathsf{ret}\ \phi$ to $(\overline{\mathrm{q}}, \overline{\mathsf{B}}'\sigma_2)[x := \phi]$

Result follows since $\mathit{sim}(\trianglelefteq)\,;\ \ \leq\, = \mathit{sim}(\trianglelefteq)$.

2. $\mathsf{A}_1 = (\mathrm{p}, \phi\overline{\mathsf{A}})$. Let $n = |\overline{\mathsf{A}}|$. $\sigma_1(\phi)\uparrow$.
$\mathsf{A}_1\sigma_1$ has the following transitions.

– $\mathsf{fcall}\ (\phi, (i, n))$ transition to $(\mathrm{p}, \overline{\mathsf{A}}(i)\sigma_1)$, for $1 \leq i \leq |\overline{\mathsf{A}}|$;
– $\mathsf{fcall}\ (\phi, (0, n))$ transition to $(\mathrm{p}, \mathsf{Id})$

Since $\mathsf{A}_1 \lesssim \mathsf{A}_2$, $(\mathrm{p}, \phi\overline{\mathsf{B}}) \leq \mathit{whnf}(\mathsf{A}_2)$, where $|\overline{\mathsf{B}}| = |\overline{\mathsf{A}}| = n$ and $\overline{\mathsf{A}} \lesssim \overline{\mathsf{B}}$. So, $\mathsf{A}_2\sigma_1$ has the following matching transitions.

– $\mathsf{fcall}\ (\phi, (i, n))$ transition to $(\mathrm{p}, \overline{\mathsf{B}}(i)\sigma_2)$, for $1 \leq i \leq |\overline{\mathsf{B}}|$;
– $\mathsf{fcall}\ (\phi, (0, n))$ transition to $(\mathrm{p}, \mathsf{Id})$

Result follows since $(\mathrm{p}, \phi\overline{\mathsf{B}})\sigma_2 \leq \mathsf{A}_2\ \sigma_2$.

3. $\sigma_1(\phi) = (\mathrm{q}, \psi\overline{\mathsf{B}})$. So, $\mathsf{A}_1\ \sigma_1 = (\mathrm{p}, (\mathrm{q}, \psi\overline{\mathsf{B}}))\overline{\mathsf{A}}\sigma_1$, and $\sigma_1(\psi)\uparrow$.
Since $\sigma_1(\phi) \lesssim \sigma_2(\phi)$, $(\exists\overline{\mathsf{B}}')\ \overline{\mathsf{B}} \lesssim \overline{\mathsf{B}}'$ and $(\mathrm{q}, \psi\overline{\mathsf{B}}') \leq \mathit{whnf}(\sigma_2(\phi))$.
Since $\mathsf{A}_1 \lesssim \mathsf{A}_2$, $(\mathrm{p}, \phi\overline{\mathsf{C}}) \in \mathit{whnf}(\mathsf{A}_2)$, where $\overline{\mathsf{A}} \lesssim \overline{\mathsf{C}}$.
Transitivity of $\mathit{sim}(\trianglelefteq)$ (inherited from $\trianglelefteq$) on the following subgoals yields the required result.

(a) $(\mathrm{p}, (\mathrm{q}, \psi\overline{\mathsf{B}}))\overline{\mathsf{A}}\sigma_1\ \ \mathit{sim}(\trianglelefteq)\ \ (\mathrm{p}, (\mathrm{q}, \psi\overline{\mathsf{B}}))\overline{\mathsf{A}}\sigma_2$.
From case(2) since $\sigma_1 \lesssim \sigma_2$, $\sigma_1(\psi)\uparrow$.
(b) $(\mathrm{p}, (\mathrm{q}, \psi\overline{\mathsf{B}}))\overline{\mathsf{A}}\sigma_2\ \ \mathit{sim}(\trianglelefteq)\ \ (\mathrm{p}, (\mathrm{q}, \psi\overline{\mathsf{B}}'))\overline{\mathsf{C}}\sigma_2$
From case (2), since $\sigma_1(\psi)\uparrow$, $\overline{\mathsf{B}} \lesssim \overline{\mathsf{B}}'$ and $\overline{\mathsf{A}} \lesssim \overline{\mathsf{C}}$.
(c) $(\mathrm{p}, (\mathrm{q}, \psi\overline{\mathsf{B}}'))\overline{\mathsf{C}}\sigma_2 \ \leq\ \mathsf{A}_2\sigma_2$

Proof of Lemma 15.

Proof Let $\mathsf{A}_1\sigma_1 \trianglelefteq \mathsf{A}_2\sigma_2$. So, $\mathsf{A}_1 \lesssim M_2$, $\sigma_1 \lesssim \sigma 2$.

We need to show that $\mathit{whnf}(\mathsf{A}_1\sigma_1)\ \mathit{sim}(\trianglelefteq)\ \ \mathsf{A}_2\sigma_2$. Since $\mathit{sim}(\trianglelefteq)$ is admissible for $\leq$-chains, it suffices to show this for $\mathit{vals}(\mathsf{E}_1)$ where $\mathsf{A}_1\ \sigma_1 \rightsquigarrow^\star \mathsf{E}_1$. The proof proceeds by induction on the length of $\mathsf{A}_1 \xrightarrow{\kappa}\!\!\!\!\rightarrow \mathsf{E}_1$.

The base case is proved in Lemma 26.

If $\mathsf{A}_1 \rightsquigarrow \mathsf{A}_1'$, then $\mathsf{A}_1\sigma_1 \rightsquigarrow \mathsf{A}_1'\sigma_1$. In this case, result follows from induction hypothesis on $\mathsf{A}_1'\sigma_1, \mathsf{A}_2\sigma_2$, since $\mathsf{A}_1' \sim \mathsf{A}_1 \lesssim \mathsf{A}_2$, so $\mathsf{A}_1'\sigma_1 \trianglelefteq \mathsf{A}_2\ \sigma_2$.

So, we can assume that $\mathsf{A}_1\sigma_1 \xrightarrow{\tau}$ is not induced by a reduction $\mathsf{A}_1 \rightsquigarrow$.

In this case, $\mathsf{A}_1 = (\mathrm{p}, \phi\overline{\mathsf{A}}_1)$. $\phi \in \mathit{dom}(\sigma_1)$. As above we can assume that $\mathsf{A}_1\sigma_1 \xrightarrow{\tau}$ is not induced by a reduction $\sigma_1(\phi) \rightsquigarrow$. So, $\sigma_1(\phi) = (\overline{\mathrm{q}}, \lambda\overline{x}\,.\overline{\mathsf{A}}_1')$. Wlog, we assume that $\sigma_1(x)\uparrow$. Then, $\mathsf{A}_1\sigma_1 \rightsquigarrow \mathrm{p} \times (\overline{\mathrm{q}}, \overline{\mathsf{A}}_1')\overline{x}\sigma_1'$, where

$$\sigma_1'(y) = \begin{cases} \sigma_1(y), & y \neq x, \sigma_1(y) \downarrow \\ \overline{\mathsf{A}}_1(i), & y = \overline{x}(i), i \geq 1 \end{cases}$$

and $\overline{x}$ is a vector of variables of same length as $|\overline{\mathsf{A}}_1|$ such that $x = \overline{x}(1)$.

Since $\mathsf{A}_1 \lesssim \mathsf{A}_2$, there exists $(\mathtt{p}, \phi\overline{\mathsf{A}}_2) \leq \mathit{whnf}(\mathsf{A}_2)$ such that $\overline{\mathsf{A}}_1 \lesssim \overline{\mathsf{A}}_2$. Since $\sigma_1(\phi) \lesssim \sigma_2(\phi)$, there exists $(\overline{\mathtt{q}}, \lambda\overline{x}\,.\overline{\mathsf{A}}_2') \leq \mathit{whnf}(\sigma_2(\phi))$ such that $(\overline{\mathtt{q}}, \overline{\mathsf{A}}_1') \lesssim (\overline{\mathtt{q}}, \overline{\mathsf{A}}_2')$. Consider $\mathtt{p} \times (\overline{\mathtt{q}}, \overline{\mathsf{A}}_2')\overline{x}\sigma_2'$, where

$$\sigma_2'(y) = \begin{cases} \sigma_2(y), & y \neq x, \sigma_1(y) \downarrow \\ \overline{\mathsf{A}}_2(i), & y = \overline{x}(i), i \geq 1 \end{cases}$$

and $\overline{x}$ is a vector of variables of same length as $|\overline{\mathsf{A}}| = |\overline{\mathsf{C}}|$ such that $x = \overline{x}(1)$. Result follows from induction hypothesis on $\overline{\mathsf{A}}'\sigma_1', \overline{\mathsf{C}}'\sigma_2'$ and $\leq$ monotonicity of $\mathit{sim}(\trianglelefteq)$.

References

Abramsky, S. (1990). *The lazy lambda calculus* (pp. 65–116). Inc, USA: Addison-Wesley Longman Publishing Co.

Abramsky, S. (1991a). A domain equation for bisimulation. *Information and Computation, 92*(2), 161–218. https://doi.org/10.1006/inco.1991.9999

Abramsky, S. (1991b). Domain theory in logical form*. *Annals of Pure and Applied Logic 51*(1), 1–77. https://doi.org/10.1016/0168-0072(91)90065-T

Abramsky, S., & Jagadeesan, R. (1994). New foundations for the geometry of interaction. *Information and Computation, 111*(1), 53–119. https://doi.org/10.1006/inco.1994.1041

Abramsky, S., Jagadeesan, R., & Malacaria, P. (2000). Full abstraction for PCF. *Information and Computation, 163*(2), 409–470.

Abramsky, S., & Jung, A. (1995). *Domain theory* (pp. 1–168). USA: Oxford University Press Inc.

Abramsky, S., & Ong, C. H. L. (1993). Full abstraction in the lazy lambda calculus. *Information and Computation, 105*(2), 159–267. https://doi.org/10.1006/inco.1993.1044

Aharoni, R., Berger, E., Georgakopoulos, A., Perlstein, A., & Sprüssel, P. (2011). The max-flow min-cut theorem for countable networks. *Journal of Combinatorial Theory, Series B, 101*(1), 1–17. https://doi.org/10.1016/j.jctb.2010.08.002

Danos, V., & Ehrhard, T. (2011). Probabilistic coherence spaces as a model of higher-order probabilistic computation. *Information and Computation, 209*(6), 966–991. https://doi.org/10.1016/j.ic.2011.02.001

Danos, V., & Harmer, R. S. (2002). Probabilistic game semantics. *ACM Transactions on Computational Logic, 3*(3), 359–382. https://doi.org/10.1145/.507385

Deng, Y., van Glabbeek, R., Hennessy, M., Morgan, C., & Zhang, C. (2007). Characterising testing preorders for finite probabilistic processes. In *Proceedings of the 22nd Annual IEEE Symposium on Logic in Computer Science (LICS '07)* (pp. 313–325). USA: IEEE Computer Society. https://doi.org/10.1109/LICS.2007.15

Desharnais, J., Gupta, V., Jagadeesan, R., & Panangaden, P. (2003). Approximating labelled Markov processes. *Information and Computation, 184*(1), 160–200. https://doi.org/10.1016/S0890-5401(03)00051-8

Desharnais, J., Gupta, V., Jagadeesan, R., & Panangaden, P. (2004). Metrics for labelled Markov processes. *Theoretical Computer Science, 318*(3), 323–354. https://doi.org/10.1016/j.tcs.2003.09.013

Ehrhard, T., Tasson, C., & Pagani, M. (2014). Probabilistic coherence spaces are fully abstract for probabilistic PCF. In *Proceedings of the 41st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (San Diego, California, USA) (POPL '14)* (pp. 309–320). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/2535838.2535865

Gierz, G., Hofmann, K. H., Keimel, K., Lawson, J. D., Mislove, M., & Scott, D. S. (1980). *A compendium of continuous lattices*. Berlin: Springer.

Girard, J.-Y. (1989). Geometry of interaction 1: Interpretation of system F. In R. Ferro, C. Bonotto, S. Valentini, & A. Zanardo (Eds.), *Logic colloquium '88*. Studies in logic and the foundations of mathematics (vol. 127, pp. 221–260). Elsevier. https://doi.org/10.1016/S0049-237X(08)70271-4

Gordon, A. D. (1995). Bisimilarity as a theory of functional programming. *Electronic Notes in Theoretical Computer Science, 1*.

Goubault-Larrecq, J. (2019). *Fooling the parallel or tester with probability 8/27* (pp. 313–328). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-31175-9_18

Heunen, C., Kammar, O., Staton, S., & Yang, H. (2017). A convenient category for higher-order probability theory. In *Proceedings of the 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (Reykjavík, Iceland) (LICS '17)*, Article 77, 12 p. IEEE Press.

Hyland, J. M. E., & Ong, C.-H. L. (2000). On full abstraction for PCF: I, II, and III. *Information and Computation, 163*(2), 285–408.

Jaber, G., & Tzevelekos, N. (2018). A trace semantics for system F parametric polymorphism. In *Foundations of Software Science and Computation Structures - 21st International Conference, FOSSACS 2018, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2018, Thessaloniki, Greece, April 14–20, 2018, Proceedings* (pp. 20–38). https://doi.org/10.1007/978-3-319-89366-2_2

Jagadeesan, R., Pitcher, C., & Riely, J. (2009). Open bisimulation for aspects. *LNCS Transactions on Aspect Oriented Software Development, 5*, 72–132. https://doi.org/10.1007/978-3-642-02059-9_3

Jones, C., & Plotkin, G. (1989). A probabilistic powerdomain of evaluations. In *Proceedings of the Fourth Annual Symposium on Logic in Computer Science (Pacific Grove, California, USA)* (pp. 186–195). IEEE Press.

Jung, A., & Tix, R. (1998). The troublesome probabilistic powerdomain. *Electronic Notes in Theoretical Computer Science, 13*, 70–91. https://doi.org/10.1016/S1571-0661(05)80216-6. Comprox III, Third Workshop on Computation and Approximation.

Lassen, S. B. (1999). Böhm trees and bisimulation up to context: Bisimulation in untyped lambda calculus. *Electronic Notes in Theoretical Computer Science, 20*, 346–374. https://doi.org/10.1016/S1571-0661(04)80083-5. MFPS XV, Mathematical Foundations of Progamming Semantics, Fifteenth Conference.

Lassen, S. B. (2005). Eager normal form bisimulation. In *20th IEEE Symposium on Logic in Computer Science (LICS 2005), 26–29 June 2005, Chicago, IL, USA, Proceedings* (pp. 345–354). IEEE Computer Society. https://doi.org/10.1109/LICS.2005.15

Lassen, S. B. (2006). Head normal form bisimulation for pairs and the lambda mu-calculus. In *21th IEEE Symposium on Logic in Computer Science (LICS 2006), 12–15 August 2006, Seattle, WA, USA, Proceedings* (pp. 297–306). https://doi.org/10.1109/LICS.2006.29

Lassen, S. B., & Levy, P. B. (2008). Typed normal form bisimulation for parametric polymorphism. In *Proceedings of the Twenty-Third Annual IEEE Symposium on Logic in Computer Science, LICS 2008, 24–27 June 2008, Pittsburgh, PA, USA* (pp. 341–352). https://doi.org/10.1109/LICS.2008.26

Lawson, J. (1998). Computation on metric spaces via domain theory. *Topology and its Applications, 85*(1), 247–263. https://doi.org/10.1016/S0166-8641(97)00152-1. Invited Papers of the 8th Prague Topological Symposium on General Topology and its Relations to Modern Analysis and Algebra

Levy, P. B., & Staton, S. (2014). Transition systems over games. In *Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS '14, Vienna, Austria, July 14–18, 2014* (pp. 64:1–64:10). https://doi.org/10.1145/2603088.2603150

Malacaria, P., & Regnier, L. (1991). Some results on the interpretation of lambda -calculus in operator algebras. In *Proceedings 1991 Sixth Annual IEEE Symposium on Logic in Computer Science* (pp. 63, 64, 65, 66, 67, 68, 69, 70, 71, 72). Los Alamitos, CA, USA: IEEE Computer Society. https://doi.org/10.1109/LICS.1991.151631

Nickau, H. (1994). Hereditarily sequential functionals. In *Logical Foundations of Computer Science, Third International Symposium, LFCS'94, St. Petersburg, Russia, July 11–14, 1994, Proceedings* (pp. 253–264). https://doi.org/10.1007/3-540-58140-5_25

Ong, C.-H. L. (1988). *The Lazy lambda calculus: An investigation into the foundations of functional programming*. Ph.D. Dissertation. https://spiral.imperial.ac.uk/handle/10044/1/47211

Ong, C.-H. L. (1992). Lazy lambda calculus: Theories, models and local structure characterization (extended abstract). In *Automata, Languages and Programming, 19th International Colloquium, ICALP92, Vienna, Austria, July 13–17, 1992, Proceedings* (pp. 487–498). https://doi.org/10.1007/3-540-55719-9_98

Panangaden, P. (2009). *Labelled Markov processes*. GBR: Imperial College Press.

Pous, D. (2016). Coinduction all the way up. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science (New York, NY, USA) (LICS '16)* (pp. 307–316). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/2933575.2934564

Sangiorgi, D. (1994). The lazy lambda calculus in a concurrency scenario. *Information and Computation, 111*(1), 120–153. https://doi.org/10.1006/inco.1994.1042

Sangiorgi, D., & Vignudelli, V. (2019). Environmental bisimulations for probabilistic higher-order languages. *ACM Transactions on Programming Languages and Systems, 41*(4), Article 22, 64. https://doi.org/10.1145/3350618

Segala, R., & Lynch, N. (1995). Probabilistic simulations for probabilistic processes. *Nordic Journal of Computing, 2*, 2.

Støvring, K., & Lassen, S. B. (2009). A complete, co-inductive syntactic theory of sequential control and state. In *Semantics and Algebraic Specification, Essays Dedicated to Peter D. Mosses on the Occasion of His 60th Birthday* (pp. 329–375). https://doi.org/10.1007/978-3-642-04164-8_17

Vákár, M., Kammar, O., & Staton, S. (2019). A domain theory for statistical probabilistic programming. *Proceedings of the ACM on Programming Languages, 3*, POPL, Article 36, 29. https://doi.org/10.1145/3290349

van Breugel, F., Mislove, M. W., Ouaknine, J., & Worrell, J. (2003). An intrinsic characterization of approximate probabilistic bisimilarity. In *Foundations of Software Science and Computational Structures, 6th International Conference, FOSSACS 2003 Held as Part of the Joint European Conference on Theory and Practice of Software, ETAPS 2003, Warsaw, Poland, April 7–11, 2003, Proceedings* (pp. 200–215). https://doi.org/10.1007/3-540-36576-1_13

van de Meent, J.-W., Paige, B., Yang, H., & Wood, F. (2018). An introduction to probabilistic programming. arxiv:1809.10756. Comment: Under review at Foundations and Trends in Machine Learning.

Chapter 29
Multisets and Distributions, in Drawing and Learning

Bart Jacobs

Abstract Multisets are 'sets' in which elements may occur multiple times. Discrete probability distributions capture states in which elements may occur with probabilities that add up to one. This paper describes how the interaction between multisets and distributions lies at the heart of some basic constructions in probability theory, especially in distributions arising from drawing from an urn with multiple balls and in learning distributions from multiple occurrences of data. Drawing multiple balls from an urn is described uniformly in terms of Kleisli iteration for a monad, covering the four standard distinctions of ordered/unordered draws, with/without replacement. In probabilistic learning the paper distinguishes two forms of likelihood, based on also on iteration, with corresponding forms of learning. Both of these forms occur in the literature, but they are not clearly distinguished, even though they lead to different outcomes.

29.1 Introduction

When we wish to combine elements from a certain set there are several mechanisms to do so, depending on whether or not the order of the elements is relevant and on how to count these elements. Concretely, one can use:

– *subsets*, in which neither the order nor the multiplicity of the elements matters;
– *lists*, in which the order matters, and in which elements may occur multiple times;
– *multisets*, in which the order of elements is irrelevant, but elements may occur multiple times;
– *distributions*, in which the order does not matter, but where elements may occur with a certain probability, taken from the unit interval [0, 1], in such a way that all probabilities add up to one.

B. Jacobs (✉)
Institute for Computing and Information Sciences, Radboud University,
Nijmegen, The Netherlands
e-mail: bart@cs.ru.nl

A. Palmigiano and M. Sadrzadeh (eds.), *Samson Abramsky on Logic and Structure in Computer Science and Beyond*, Outstanding Contributions to Logic 25,
https://doi.org/10.1007/978-3-031-24117-8_29

All these collection mechanisms can be described in terms of monads on the category of sets.

This article concentrates on the latter two collection types, namely multisets and distributions, and in particular on their interaction. This interaction will be studied in the following two typical situations.

1. Suppose we have an urn with three red (R) and two green (G) balls, represented as a multiset $3|R\rangle + 2|G\rangle$. Then the probability of drawing a red ball is $\frac{3}{5}$. Thus, one can sample a multiset by drawing elements—as one draws elements from an urn—with a certain probability determined by the multiplicities in the multiset. Such drawing may be repeated, where the distinction is important whether or not a drawn ball is replaced to the urn, and whether or not the order of multiple draws matters.
2. Suppose we have a coin with unknown bias, and flipping it three times gives outcomes head (H), tail (T), and head again. What is then the probability that the next flip is head? In this situation the data form a multiset $2|H\rangle + 1|T\rangle$, which can be used to update the prior bias distribution—which we assume to be uniform, since there is no prior knowledge. Then one can calculate the probability of head in the updated, newly learned distribution—which, in this case, is $\frac{3}{5}$.

The two topics of this paper are thus: drawing and learning, both starting from a multiset, and both yielding a probability distribution. These two topics form the two main parts of the paper: Sect. 29.3 is on drawing from an urn, and Sects. 29.6–29.9 are on learning. Section 29.5 forms the glue between drawing and learning. In between, Sects. 29.2 and 29.4 provide background information about multisets and distributions and about predicates and probabilistic conditioning/updating.

In many textbooks on probability, see e.g. Pishro-Nik (2014); Ross (2018), one finds the physical model of an urn with coloured balls, from which balls can be drawn with a certain probability. This can be done in four different ways, depending on whether the order matters in a draw of multiple balls, and whether withdrawn balls are replaced into the urn or deleted. The first part of this article on drawing introduces four 'transition' operations for drawing from an urn, corresponding to these four distinctions. All the transitions form Kleisli (endo)maps for the monad $\mathcal{D}(M \times -)$, combining the distribution monad $\mathcal{D}$ with the writer monad $M \times -$, for a monoid M. This M is the (non-commutative) list monoid for ordered draws, and the (commutative) multiset monoid for unordered draws. By iterating the transition map, using Kleisli composition for the combined monad, and then taking the first marginal, one obtains appropriate distributions on draws. In this way we reconstruct the familiar multinomial and hypergeometric distributions, for unordered draws, and two more distributions for ordered draws. This part reorganises existing material into a canonical form. It forms a topic of its own that could be used for instance in a course on the use of categorical methods, especially in probability theory.

The second part on probabilistic learning elaborates the idea that learning is about finding a probability distribution that best fits given data. In general, such learning is described as consisting of small steps that need to be repeated in order to reach a certain optimum. These steps can be used to increase the likelihood of data or to decrease errors. The latter approach is generally based on gradient descent and occurs for instance in logistic regression (see e.g. Bishop (2006)). Here we concentrate on the first approach, increasing likelihood, but we do relate it to decreasing divergence at some point (name in Proposition 20).

We organise the data from which we learn as multisets. Learning involves finding the distribution, possibly in an iterative process, that gives highest likelihood to the data. Our approach leads to *two* forms of likelihood, called 'external' and 'internal'. It is shown how both forms of likelihood arise from repeated transitions, like for drawing. Associated with these two likelihoods there are two techniques for learning. Both forms of learning occur in the literature, but they are not clearly distinguished, even though they can lead to quite different outcomes. In times where learning from huge amounts of data has become common, proper understanding of the basic concepts is not only scientifically but also practically (societally) urgent. Here, the difference is illustrated in several examples (from the literature), including coin bias learning (internal), and parameter learning and Expectation-Maximisation (both external). In the end we do not offer a mathematical criterion for when to use internal/external likelihood (and learning); for now we only have an intuitive perspective, see Sect. 29.9.

This second part extends material from an earlier conference publication (Jacobs, 2019), for instance with the new descriptions of conjugate priorship in Corollary 25 and of Expectation-Maximisation in Theorem 26, with several examples, and with the discussion about 'external' and 'internal' in Sect. 29.9.

The formalisations and results in this paper demonstrate that at a very elementary level there is categorical (esp. monadic) structure in probability theory. Of course, this observation is well-known by now, starting with the early work of Giry (1982) and of Kozen (1981, 1985) in the 1980s. The research contributions of Samson Abramsky fit in this line of work, as inspiration in unveiling fundamental mathematical (often categorical) structure in many areas, including e.g. physics and economy. Abramsky has not worked so much in (easy) classical probability theory; his work concentrates on the much more difficult field of quantum probability, with a focus on its categorical structure (Abramsky et al., 2000; Abramsky & Coecke, 2009; Abramsky, 2013; Abramsky & Heunen, 2012; Abramsky et al., 2019), on its inherent limitations (Abramsky, 2010, 2015) and especially on contextuality (Abramsky & Brandenburger, 2011; Abramsky, 2014; Bruza & Abramsky, 2016). Abramsky has been influential for my own ERC advanced grant (2012–2017) in this area, which laid the foundation for the current work.

29.2 Multisets and Distributions

This section briefly introduces (finite) multisets and (finite discrete probability) distributions. They both are collection types that can have elements occurring multiple times or with certain probabilities.

First we like to fix our notation for lists/sequences. We write $\mathcal{L}(X)$ for the set of finite sequences $[x_1, \ldots, x_n]$ of elements $x_i \in X$, of length n. This set forms a monoid, with concatenation $+\!\!+$ as binary operation and empty list $[]$ as neutral element. As is well-known, the operation $\mathcal{L}$ forms a monad on the category of sets.

29.2.1 Multisets

A multiset (or bag) is a 'set' in which (finitely many) elements may occur multiple times, with natural numbers as multiplicities. We write $\mathcal{M}(X)$ for the set of such multisets over a set X, defined as:

$$\mathcal{M}(X) := \{\phi \colon X \to \mathbb{N} \mid \mathrm{supp}(\phi) \text{ is finite}\},$$

where $\mathrm{supp}(\phi) \subseteq X$ is the support of ϕ, i.e. the subset $\{x \in X \mid \phi(x) \neq 0\}$. We often write concrete multisets as finite formal sums, using a 'ket' notation: $\phi = \sum_i n_i|x\rangle$, where $\mathrm{supp}(\phi) = \{x_1, \ldots, x_n\}$ and $n_i = \phi(x_i) \in \mathbb{N}$. Taking multisets on a set is functorial: for $f \colon X \to Y$ we get $\mathcal{M}(f) \colon \mathcal{M}(X) \to \mathcal{M}(Y)$ via $\mathcal{M}(f)(\phi)(y) = \sum_{x \in f^{-1}(y)} \phi(x)$. Alternatively, in terms of formal sums: $\mathcal{M}(f)(\sum_i n_i|x_i\rangle) = \sum_i n_i$ $|f(x_i)\rangle$. In fact, $\mathcal{M}$ is a monad on the category of sets. Moreover, $\mathcal{M}(X)$ with pointwise addition and empty multiset $\mathbf{0}$, is the free commutative monoid on X. This monoid is ordered: for $\phi, \psi \in \mathcal{M}(X)$ we write $\phi \leq \psi$ if $\phi(x) \leq \psi(x)$ for all $x \in X$. This implies $\mathrm{supp}(\phi) \subseteq \mathrm{supp}(\psi)$. In that case we write $\psi - \phi \in \mathcal{M}(X)$ for the obvious multiset, with multiplicities $(\psi - \phi)(x) = \psi(x) - \phi(x)$. The situation $\phi \leq \psi$ arises for instance when ψ is an urn with balls, and ϕ is a handful of balls drawn from the urn.

One can associate several numbers with a multiset. The next definition gives an overview.

Definition 1 Let $\phi \in \mathcal{M}(X)$ be a multiset on a set X.

1. The *size* $\|\phi\| \in \mathbb{N}$ is the total number of elements occurring in a multiset, taking multiplicities into account:

$$\|\phi\| := \sum_{x \in X} \phi(x).$$

2. The *factorial* $\phi\mathbb{!} \in \mathbb{N}$ the product of factorials of multiplicities:

$$\phi\mathbb{!} := \prod_{x \in X} \phi(x)!.$$

3. The *multiset coefficient*, or simply *coefficient* (ϕ) of ϕ is a multinomial coefficient:

$$(\phi) := \frac{\|\phi\|!}{\phi\,\text{‼}} = \frac{\|\phi\|!}{\prod_x \phi(x)!} = \binom{\|\phi\|}{\phi(x_1)\ \cdots\ \phi(x_n)},$$

The latter multinomial coefficient formulation assumes that ϕ's support is $\{x_1, \ldots, x_n\}$.

4. Finally, when $\phi \leq \psi \in \mathcal{M}(X)$ we use a binomial coefficient of multisets as product of binomial coefficients of multiplicities:

$$\begin{aligned} \binom{\psi}{\varphi} &:= \frac{\psi\,\text{‼}}{\varphi\,\text{‼} \cdot (\psi - \varphi)\,\text{‼}} \\ &= \frac{\prod_x \psi(x)!}{\left(\prod_x \varphi(x)!\right) \cdot \left(\prod_x (\psi(x) - \varphi(x))!\right)} = \prod_{x \in X} \binom{\psi(x)}{\varphi(x)}. \end{aligned}$$

Frequently we like to have some grip on the total number of elements occurring in a multiset, taking multiplicities into account. We write for $K \in \mathbb{N}$,

$$\mathcal{M}[K](X) := \{\phi \in \mathcal{M}(X) \mid \|\phi\| = K\}.$$

Clearly, $\mathcal{M}[0](X)$ is a singleton, containing only the empty multiset $\mathbf{0}$. This $\mathcal{M}[K]$ is a functor, but not a monad. However, it has the structure of a *graded* monad with respect to the monoid of natural numbers with multiplication.

There is an accumulation map $acc\colon \mathcal{L} \Rightarrow \mathcal{M}$, turning lists into multisets, given by $acc([x_1, \ldots, x_n]) = 1|x_1\rangle + \cdots + 1|x_n\rangle$. Thus, e.g., $acc([a, a, b, b, a] = 3|a\rangle + 2|b\rangle$. This accumulation forms a map of monads. We often use accumulation for a fixed size $K \in \mathbb{N}$, and then write it as $acc[K]\colon X^K \to \mathcal{M}[K](X)$. The parameter $K \in \mathbb{N}$ is omitted when it is clear for the context.

A basic question is: how many lists accumulate to a given multiset ϕ? The (standard) answer is: (ϕ). For instance, for $\phi = 2|a\rangle + 3|b\rangle$ there are $(\phi) = \frac{5!}{2!\cdot 3!} = 10$ sequences with length 5 of a's and b's that accumulate to ϕ. It is not hard to see that $(-)$ satisfies the following recurrence equation:

$$(\phi) = \sum_{y \in \mathrm{supp}(\phi)} (\phi - 1|y\rangle). \tag{29.1}$$

The following result is a generalisation of Vandermonde's formula. We include a proof, for convenience.

Lemma 2 *Let $\psi \in \mathcal{M}(X)$ be a multiset of size $L = \|\psi\|$, with a number $K \leq L$. We write $\phi \leq_K \psi$ as short-hand for: $\phi \in \mathcal{M}[K](X)$ with $\phi \leq \psi$. Then:*

$$\sum_{\phi \leq_K \psi} \binom{\psi}{\phi} = \binom{L}{K} \qquad \textit{so that} \sum_{\phi \leq_K \psi} \frac{\binom{\psi}{\phi}}{\binom{L}{K}} = 1.$$

Proof We use induction on the number of elements in $\mathrm{supp}(\psi)$. We go through some initial values explicitly. If the number is 0, then $\psi = \mathbf{0}$ and so $L = 0 = K$ and $\phi \leq_K \psi$ means $\phi = \mathbf{0}$, so that the result holds. Similarly, if $\mathrm{supp}(\psi)$ is a singleton, say $\{x\}$, then $L = \psi(x)$. For $K \leq L$ and $\phi \leq_K \psi$ we get $\mathrm{supp}(\phi) = \{x\}$ and $K = \phi(x)$. The result then obviously holds.

The case where $\mathrm{supp}(\psi) = \{x, y\}$ captures the ordinary form of Vandermonde's formula. We reformulate it for numbers $B, G \in \mathbb{N}$ and $K \leq B + G$. Then:

$$\binom{B+G}{K} = \sum_{b \leq B,\, g \leq G,\, b+g=K} \binom{B}{b} \cdot \binom{G}{g}. \tag{29.2}$$

Intuitively: if you select K children out of B boys and G girls, the number of options is given by the sum over the options for $b \leq B$ boys times the options for $g \leq G$ girls, with $b + g = K$.

The Eq. (29.2) can be proven by induction on G. When $G = 0$ both sides amount to $\binom{B}{K}$ so we proceed to the induction step. The case $K = 0$ is trivial, so we may assume $K > 0$. We use what's called Pascal's rule $\binom{n+1}{m} = \binom{n}{m} + \binom{n}{m-1}$ for binomials.

$$\begin{aligned}
&\sum_{b \leq B,\, g \leq G+1,\, b+g=K} \binom{B}{b} \cdot \binom{G+1}{g} \\
&= \binom{B}{K} \cdot \binom{G+1}{0} + \binom{B}{K-1} \cdot \binom{G+1}{1} + \cdots + \binom{B}{K-G} \cdot \binom{G+1}{G} \\
&\quad + \binom{B}{K-G-1} \cdot \binom{G+1}{G+1} = \binom{B}{K} \cdot \binom{G}{0} + \binom{B}{K-1} \cdot \binom{G}{1} + \binom{B}{K-1} \cdot \binom{G}{0} \\
&\qquad + \cdots + \binom{B}{K-G} \cdot \binom{G}{G} + \binom{B}{K-G} \cdot \binom{G}{G-1} + \binom{B}{K-G-1} \cdot \binom{G}{G} \\
&= \sum_{b \leq B,\, g \leq G,\, b+g=K} \binom{B}{b} \cdot \binom{G}{g} + \sum_{b \leq B,\, g \leq G,\, b+g=K-1} \binom{B}{b} \cdot \binom{G}{g} \\
&\overset{\text{(IH)}}{=} \binom{B+G}{K} + \binom{B+G}{K-1} \\
&= \binom{B+G+1}{K}.
\end{aligned}$$

We now turn to the (first) equation in Lemma 2. For the induction step, let $\mathrm{supp}(\psi) = \{x_1, \ldots, x_n, y\}$, for $n \geq 2$. Writing $\ell = \psi(y)$, $L' = L - \ell$ and $\psi' = \psi - \ell|y\rangle \in \mathcal{M}[L'](X)$ gives:

$$\begin{aligned}\sum_{\phi \leq_K \psi} \binom{\psi}{\phi} = \sum_{\phi \leq_K \psi} \prod\nolimits_x \binom{\psi(x)}{\phi(x)} &= \sum_{n \leq \ell} \sum_{\phi \leq_{K-n} \psi'} \binom{\ell}{n} \cdot \prod\nolimits_i \binom{\psi(x_i)}{\phi(x_i)} \\ &\overset{\text{(IH)}}{=} \sum_{n \leq \ell,\, K-n \leq L-\ell} \binom{\ell}{n} \cdot \binom{L-\ell}{K-n} \overset{(2)}{=} \binom{L}{K}.\end{aligned} \qquad \square$$

For completeness we also include the *multinomial theorem*, without proof.

Lemma 3 *For $K \in \mathbb{N}$ and $a_1, \ldots, a_n \in \mathbb{R}$,*

$$\big(a_1 + \cdots + a_n\big)^K = \sum_{\phi \in \mathcal{M}[K](\{1,\ldots,n\})} (\phi) \cdot a_1^{\phi(1)} \cdot \cdots \cdot a_n^{\phi(n)}. \qquad \square$$

29.2.2 Distributions

A distribution (or a state, or multinomial) is like a multiset but where its multiplicities are taken from the unit interval [0, 1] and add up to one. We thus define the set $\mathcal{D}(X)$ of distribution on a set X as:

$$\mathcal{D}(X) \coloneqq \{\phi \colon X \to [0,1] \mid \mathrm{supp}(\phi) \text{ is finite, and } \textstyle\sum_x \phi(x) = 1\}.$$

This $\mathcal{D}$ is also monad on the category of sets.

A *channel* $f : X \rightsquigarrow Y$ is a probabilistic computation from X to Y. Notice that it is written with a small circle on the shaft of the arrow. A channel can be understood as an X-indexed collection of states on Y, or alternatively, as a conditional probability $f(y \mid x)$. We look at a channel more categorically, as a 'Kleisli' map $f : X \to \mathcal{D}(Y)$. Such a channel can 'push' a state $\omega \in \mathcal{D}(X)$ forward to a state $f \gg \omega \in \mathcal{D}(Y)$, via 'Kleisli extension' or 'state transformation', where $(f \gg \omega)(y) = \sum_x \omega(x) \cdot f(x)(y)$. Via $\gg$ we can define composition $g \circ f$ of channels as $(g \circ f)(x) = g \gg f(x)$.

An example of a channel is what we call *arrangement* $arr : \mathcal{M}(X) \rightsquigarrow \mathcal{L}(X)$. It maps a multiset ϕ to a (uniform) distribution over the sequences $\vec{x}$ that accumulate to ϕ. As we have seen before, there are (ϕ)-many such sequences. Hence:

$$arr(\phi) \coloneqq \sum_{\vec{x} \in acc^{-1}(\phi)} \frac{1}{(\phi)} \big|\vec{x}\big\rangle. \tag{29.3}$$

This channel restricts to $arr : \mathcal{M}[K](X) \rightsquigarrow X^K$. The composite $acc \circ arr$ is the identity channel $\mathcal{M}[K](X) \rightsquigarrow \mathcal{M}[K](X)$. In the other direction, $arr \circ acc : X^K \rightsquigarrow X^K$ sends a sequence to the uniform distribution of all its permutations.

We shall use the parallel product $\sigma \otimes \tau$ of distributions $\sigma \in \mathcal{D}(X)$ and $\tau \in \mathcal{D}(Y)$. It is a distribution on the product space $X \times Y$ defined as:

$$\big(\sigma \otimes \tau\big)(x, y) = \sigma(x) \cdot \tau(y).$$

We write $iid[K] \colon \mathcal{D}(X) \rightsquigarrow X^K$ for the channel that maps a state ω to the K-fold tensor: $iid(\omega) = \omega^K = \omega \otimes \cdots \otimes \omega \in \mathcal{D}(X^K)$. This gives the so-called identical and independent distribution.

Similarly, for channels $f \colon A \rightsquigarrow X$ and $g \colon B \rightsquigarrow Y$ we get $f \otimes g \colon A \times B \rightsquigarrow X \times Y$ via $(f \otimes g)(a, b) = f(a) \otimes g(b)$. This makes the Kleisli category $\mathcal{K}\ell(\mathcal{D})$ of the distribution monad symmetric monoidal.

An ordinary function $f \colon X \to Y$ is often implicitly promoted to a channel $f \colon X \rightsquigarrow Y$ via $x \mapsto 1|\, f(x)\,\rangle$. This is used in particular for diagonals $\Delta = \langle \mathrm{id}, \mathrm{id}\rangle \colon X \to X \times X$ and projections $\pi_i \colon X_1 \times X_2 \to X_i$. Marginalisation of $\omega \in \mathcal{D}(X_1 \times X_2)$ can then be described as $\pi_i \gg \omega \in \mathcal{D}(X_i)$. One has $\pi_i \gg (\sigma_1 \otimes \sigma_2) = \sigma_i$, but in general:

$$(\pi_1 \gg \omega) \otimes (\pi_2 \gg \omega) \neq \omega \qquad \text{and } \Delta \gg \sigma \neq \sigma \otimes \sigma.$$

For two channels $c \colon A \rightsquigarrow X$ and $d \colon A \rightsquigarrow Y$ we shall write $\langle c, d\rangle = (c \otimes d) \circ \Delta \colon A \rightsquigarrow X \times Y$ for their tuple.

As is well-known in probability theory, from a joint state $\tau \in \mathcal{D}(X \times Y)$ one can extract a channel (conditional probability) $c \colon X \rightsquigarrow Y$, given by:

$$c(x)(y) = \frac{\omega(x, y)}{(\pi_1 \gg \omega)(x)} \qquad \text{so that } \langle \mathrm{id}, c\rangle \gg (\pi_1 \gg \omega) = \omega. \tag{29.4}$$

Clearly, this channel extraction is a partial operation, since the first marginal needs to be non-zero. The latter equation is commonly written as $\omega(y \mid x) \cdot \omega(x) = \omega(x, y)$. This extraction of a channel is called ‘disintegration’, see Cho and Jacobs (2019); Fritz (2020); Clerc et al. (2017).

We include two classic examples, that play an important role later on: multinomial and hypergeometric distributions. Informally, they assign a probability to taking a handful of coloured balls from an urn. These distributions are most common in *binary form*, for an urn with two colours only. Here we look at the *multivariate* form, with an arbitrary set X of colours. We shall describe this “handful”, say with K balls, as a multiset of size K. Thus, multinomial and hypergeometric distributions produce outcomes in the set $\mathcal{D}\big(\mathcal{M}[K](X)\big)$ of distributions on multisets of size K. The difference between multinomial and hypergeometric distributions lies in whether drawn balls are replaced into the urn or not. When the balls are replaced, in the case of multinomials, the urn itself may be represented abstractly as a distribution. When the drawn balls are actually deleted, the urn changes with every draw, and is represented as a multiset. We shall describe the multinomial and geometric distributions as channels, of the form:

$$\mathcal{D}(X) \xrightarrow{\ mulnom[K]\ } \mathcal{M}[K](X) \qquad \mathcal{M}[L](X) \xrightarrow{\ hypgeom[K]\ } \mathcal{M}[K](X), \tag{29.5}$$

where K is the number of drawn balls. In the hypergeometric case one needs $K \leq L$, where L is the number of balls in the urn. The channels are defined on a distribution $\omega \in \mathcal{D}(X)$ and multiset $\psi \in \mathcal{M}[L](X)$ as:

$$\begin{aligned} mulnom[K](\omega) &:= \sum_{\phi \in \mathcal{M}[K](X)} (\phi) \cdot \prod\nolimits_x \omega(x)^{\phi(x)} \big|\phi\big\rangle \\ hypgeom[K](\psi) &:= \sum_{\phi \leq_K \psi} \frac{\binom{\psi}{\phi}}{\binom{L}{K}} \big|\phi\big\rangle. \end{aligned} \tag{29.6}$$

Recall that we write $\phi \leq_K \psi$ as shorthand for: $\phi \in \mathcal{M}[K](X)$ with $\phi \leq \psi$. The multinomial definition yields a distribution via Lemma 3. In the hypergeometric case we use Lemma 2.

29.2.3 Frequentist Learning

In general in probabilistic learning, one learns from 'data'. A perspective that underlies this paper is that such data are naturally organised as multisets. For instance, if we wish to learn about the bias of an arbitrary coin, we need data in the form of coin flips. If we have seen 10 heads and 9 tails, we will organise these flips as a single multiset of the form $10|H\rangle + 9|T\rangle$ over the set $\{H, T\}$, whose elements represent head and tail. In a multiset the order of elements does not matter. This corresponds to the fact that the order of data elements does not matter in probabilistic learning.

One basic form of learning starts by counting. This is what we call frequentist learning *Flrn*; it amounts to normalisation. For instance $Flrn(10|H\rangle + 9|T\rangle) = \frac{10}{19}|H\rangle + \frac{9}{19}|T\rangle$. In general, for a non-empty multiset $\phi \in \mathcal{M}_*(X)$,

$$Flrn(\phi) := \sum_{x \in \mathrm{supp}(\phi)} \frac{\phi(x)}{\|\phi\|} \big|x\big\rangle \quad \text{where, recall, } \|\phi\| = \sum\nolimits_x \phi(x) > 0. \tag{29.7}$$

This result $Flrn(\phi) \in \mathcal{D}(X)$ is often called the empirical distribution. It is typical of such frequentist learning that learning from more of the same does not have any effect. We can make this precise via the equation:

$$Flrn\big(K \cdot \phi\big) = Flrn(\phi) \qquad \text{for } K > 0. \tag{29.8}$$

It is not hard to see that *Flrn* is a natural transformation $\mathcal{M}_* \Rightarrow \mathcal{D}$. This means in particular that it commutes with marginalisation. Thus, if one applies frequentist learning *Flrn* to a multi-dimensional table $\tau \in \mathcal{M}_*(X_1 \times \cdots \times X_n)$ it does not matter if one learns from the entire table first and then marginalises to $\mathcal{D}(X_i)$, or if one first adds up totals in columns $\mathcal{M}(X_i)$ and then applies frequentist learning.

When one has already learned the distribution $Flrn(\phi)$ and a new batch of data ψ arrives, all probabilities have to be re-adjusted, as in the convex sum of distributions:

$$Flrn\big(\phi+\psi\big) = \frac{\|\phi\|}{\|\phi\|+\|\psi\|}\cdot Flrn(\phi) + \frac{\|\psi\|}{\|\phi\|+\|\psi\|}\cdot Flrn(\psi).$$

29.3 Drawing from an Urn

The very basic concepts of probability theory are often explained in terms of urns: containers of objects of a certain kind, typically coloured balls. One can then draw a ball from the urn, whose colour probability is determined by the different numbers of the various balls in the urn. Such drawing can be repeated, where drawn balls are either replaced, or not. Also, the order of drawn balls may be taken into account, or not. The four cases are commonly described in terms ordered/unordered draws with/without replacement. They can be represented in a 2×2 table, see (29.11) below.

Our aim is to describe these four cases in a principled manner via probabilistic channels. In order to do so we first look at single-draw transition mappings, which may be described informally as:

$$Urn \longmapsto \Big(single\text{-}draw,\ Urn'\Big) \tag{29.9}$$

We use the ad hoc notation Urn' of an urn with an accent, to describe the urn after the draw. It may be the same urn as before, in case of a draw with replacement, or it may be a different urn, with one ball missing, namely the original urn without the single ball that was drawn.

The above transition arrow will be described as a probabilistic channel. It gives for each single draw the associated probability. In this description we shall combine multisets and distributions. For instance, an urn with three red balls and two blue ones will be described as a multiset $3|R\rangle + 2|B\rangle$. The transition associated with drawing a single ball *without* replacement gives a mapping:

$$3|R\rangle + 2|B\rangle \longmapsto \tfrac{3}{5}\big|R,\ 2|R\rangle + 2|B\rangle\big\rangle + \tfrac{2}{5}\big|B,\ 3|R\rangle + 1|B\rangle\big\rangle$$

It gives the $\frac{3}{5}$ probability of drawing a red ball, together with the remaining urn, and a $\frac{2}{5}$ probability of drawing a blue one, with a different new urn.

The situation *with* replacement is given by:

$$3|R\rangle + 2|B\rangle \longmapsto \tfrac{3}{5}\big|R,\ 3|R\rangle + 2|B\rangle\big\rangle + \tfrac{2}{5}\big|B,\ 3|R\rangle + 2|B\rangle\big\rangle$$

Here we see that the urn/multiset does not change. An important first observation is that in that case we may as well use a distribution as urn, instead of a multiset. The distribution represents an abstract urn. In the above example we would use the distribution $\frac{3}{5}|R\rangle + \frac{2}{5}|B\rangle$ as abstract urn, when we draw with replacement. The distribution contains all the relevant information. Clearly, it is obtained via frequentist learning from the original multiset. Using distributions instead of multisets gives more flexibility, since not all distributions are obtained via frequentist learning—in particular when the probabilities are proper real numbers and not fractions.

We formulate this approach explicitly.

- In a situation *without* replacement, an urn is a (non-empty, natural) multiset, which changes with every draw, via removal of the drawn ball. This no-replacement scenario will also be described in terms of deletion.
- In a situation *with* replacement, an urn is a probability distribution; it does not change when balls are drawn.

This covers the first distinction, between draws with and without replacement. The second distinction between ordered and unordered draws cannot be made for single draw transitions. Hence we need to suitably iterate the single-draw transition (29.9) to:

$$\mathit{Urn} \longmapsto \Big(\mathit{multiple\text{-}draws},\ \mathit{Urn}'\Big) \tag{29.10}$$

Now we can make the distinction between ordered and unordered draws explicit. Let X be the set of colours, for the balls in the urn—so $X = \{R, B\}$ in the above illustration.

- An *ordered* draw of multiple balls, say K many, is represented via a list $X^K = X \times \cdots \times X$ of length K.
- An *unordered* draw of K-many balls is represented as a K-sized multiset, in $\mathcal{M}[K](X)$.

Thus, in the latter case, both the urn and the handful of balls drawn from it, are represented as a multiset.

In the end we are interested in assigning probabilities to draws, ordered or not, with replacement or without. These probabilities on draws are obtained by taking the first marginal/projection of the iterated transition map (29.10). It yields a mapping from an urn to multiple draws. The following table gives an overview of the types of these operations, where X is the set of colours of the balls.

K-sized draws	with replacement	with deletion
ordered	$\mathcal{D}(X) \xrightarrow{OdR} X^K$	$\mathcal{M}[L](X) \xrightarrow{OdD} X^K$
unordered	$\mathcal{D}(X) \xrightarrow{UdR} \mathcal{M}[K](X)$	$\mathcal{M}[L](X) \xrightarrow{UdD} \mathcal{M}[K](X)$

(29.11)

We see that in the replacement scenario the inputs of these channels are distributions in $\mathcal{D}(X)$, as abstract urns. In the deletion scenario (without replacements) the input (urns) are multisets in $\mathcal{M}[L](X)$, of size L. In the ordered case the outputs are tuples in X^K of length K and in the unordered case they are multisets in $\mathcal{M}[K](X)$ of size K. Implicitly in this table we assume that $L \geq K$, so that the urn is full enough for K single draws.

We see that the table (29.11) combines the basic data types of lists, multisets and distributions. The names of the channels in the table reflect the two distinctions Below we explain these short names and relate them to commonly used names.

- *UdR* = unordered-draw-with-replacement; we will show that it is the multinomial channel, on the left in (29.5);
- *UdD* = unordered-draw-with-deletion; this will turn out to be the hypergeometric channel, on the right in (29.5);
- *OdR* = ordered-draw-with-replacement; it is the identical and independent (iid) channel;
- *OdD* = ordered-draw-with-deletion.

In the last situation there is no established name, so we shall simply use the short name *OdD* from Table 29.11.

Below we elaborate how the channels in Table 29.11 actually arise. It makes the earlier informal descriptions in (29.9) and (29.10) mathematically precise. We use that for any monoid M, the mapping $X \mapsto M \times X$ is a monad, called the writer monad. This can be combined with the distribution monad $\mathcal{D}$, giving a combined monad $X \mapsto \mathcal{D}(M \times X)$. It comes with an associated Kleisli composition $\odot$. It is precisely this composition that we use for iterating a single draw. Moreover, for ordered draws we use the monoid $M = \mathcal{L}(X)$ of lists, and for unordered draws we use the monoid $M = \mathcal{M}(X)$ of multisets. It is rewarding, from a formal perspective, to see that from this abstract principled approach, common distributions for different sorts of drawing arise, including the well-known multinomial and hypergeometric distributions.

Lemma 4 *Let $M = (M, 0, +)$ be a monoid. The mapping $X \mapsto \mathcal{D}\big(M \times X\big)$ is a monad on* **Sets**, *with unit $\eta\colon X \to \mathcal{D}\big(M \times X\big)$ given by:*

$$\eta(x) = 1|0, x\rangle \qquad \textit{where } 0 \in M \textit{ is the zero element.}$$

For Kleisli maps $f\colon A \to \mathcal{D}(M \times B)$ and $g\colon B \to \mathcal{D}(M \times C)$ there is the Kleisli composition $g \odot f\colon A \to \mathcal{D}(M \times C)$ given by:

$$\big(g \odot f\big)(a) = \sum_{m,m',c} \Big(\textstyle\sum_b f(a)(m,b) \cdot g(b)(m',c)\Big)\,\big|m + m', c\big\rangle. \tag{29.12}$$

Notice the occurrence of the sum $+$ of the monoid M in the first component of the ket $\big|-,-\big\rangle$ in (29.12). When M is the list monoid, this sum is the (non-commutative) concatenation $+\!\!+$ of lists, producing an ordered list of drawn elements. When M is the multiset monoid, this sum is the (commutative) $+$ of multisets, so that the accumulation of drawn elements yields a multiset, in which the order of elements is irrelevant.

If we have an 'endo' Kleisli map for the combined monad of Lemma 4, of the form $t\colon A \to \mathcal{D}(M \times A)$, we can iterate it K times, giving $t^K\colon A \to \mathcal{D}(M \times A)$. This iteration is defined via the above unit and Kleisli composition:

$$t^0 = \eta \qquad \text{and } t^{K+1} = t^K \odot t = t \odot t^K.$$

Below in (29.13) we define the four transition channels for drawing a single element from an urn. In the "with replacement" column on the left the distribution ω acts as abstract urn and remains unchanged. In the "without replacement" column on the right, the drawn element x is actually removed from the urn/multiset ψ via subtraction $\psi - 1|x\rangle$. Implicitly it is assumed that the multiset ψ is non-empty.

$$\begin{array}{ll}
\mathcal{D}(X) \xrightarrow{OtR} \mathcal{D}\big(\mathcal{L}(X) \times \mathcal{D}(X)\big) & \mathcal{M}(X) \xrightarrow{OtD} \mathcal{D}\big(\mathcal{L}(X) \times \mathcal{M}(X)\big) \\
\quad \omega \longmapsto \displaystyle\sum_{x \in \mathrm{supp}(\omega)} \omega(x)\big|[x], \omega\big\rangle & \quad \psi \longmapsto \displaystyle\sum_{x \in \mathrm{supp}(\psi)} \frac{\psi(x)}{\|\psi\|}\big|[x], \psi - 1|x\rangle\big\rangle \\[2em]
\mathcal{D}(X) \xrightarrow{UtR} \mathcal{D}\big(\mathcal{M}(X) \times \mathcal{D}(X)\big) & \mathcal{M}(X) \xrightarrow{UtD} \mathcal{D}\big(\mathcal{M}(X) \times \mathcal{M}(X)\big) \\
\quad \omega \longmapsto \displaystyle\sum_{x \in \mathrm{supp}(\omega)} \omega(x)\big|1|x\rangle, \omega\big\rangle & \quad \psi \longmapsto \displaystyle\sum_{x \in \mathrm{supp}(\psi)} \frac{\psi(x)}{\|\psi\|}\big|1|x\rangle, \psi - 1|x\rangle\big\rangle
\end{array} \tag{29.13}$$

In the subsections below we analyse what iteration means for these four channels. Subsequently, we can describe the associated K-sized draw channels, as first projection $\pi_1 \odot t^K$, going from urns to drawn elements. Notice that we use a letter 't' in a name like *OtR* to denote the *transition* channel $\mathcal{D}(X) \rightsquigarrow \mathcal{L}(X) \times \mathcal{D}(X)$, for Ordered transitions with Replacement. Similarly, we use the letter 'd' for the associated K-fold *draw* channel $OdR[K]\colon \mathcal{D}(X) \rightsquigarrow X^K$, in Table 29.11, where $OdR[K] = \pi_1 \odot OtR^K$. The same convention is used for the other forms of drawing.

29.3.1 Ordered Draws from an Urn

We start to look at the upper two 'ordered' transition channels $OtR\colon \mathcal{D}(X) \to \mathcal{D}\big(\mathcal{L}(X) \times \mathcal{D}(X)\big)$ and $OtD\colon \mathcal{M}(X) \to \mathcal{D}\big(\mathcal{L}(X) \times \mathcal{M}(X)\big)$ in (29.13). Towards a general formula for their iteration, let's look first at the easiest case, namely ordered transitions with replacement. By definition we have as first iteration.

$$OtR^1(\omega) = OtR(\omega) = \sum_{x_1 \in \mathrm{supp}(\omega)} \omega(x_1)\,\big|[x_1], \omega\big\rangle.$$

Accumulation of drawn elements in the first coordinate of $\big|-,-\big\rangle$ starts in the second iteration:

$$\begin{aligned} OtR^2(\omega) &= OtR \gg OtR(\omega) \\ &= \sum_{\ell \in \mathcal{L}(X),\, x_1 \in \mathrm{supp}(\omega)} \omega(x_1) \cdot OtR(\omega)(\ell, \omega) \, \big| [x_1] \mathbin{+\!\!+} \ell, \omega \big\rangle \\ &= \sum_{x_1, x_2 \in \mathrm{supp}(\omega)} \omega(x_1) \cdot \omega(x_2) \, \big| [x_1] \mathbin{+\!\!+} [x_2], \omega \big\rangle \\ &= \sum_{x_1, x_2 \in \mathrm{supp}(\omega)} (\omega \otimes \omega)(x_1, x_2) \, \big| [x_1, x_2], \omega \big\rangle. \end{aligned}$$

The formula for subsequent iterations is beginning to appear.

Theorem 5 *Consider in (29.13) the ordered-transition-with-replacement channel* $OtR \colon \mathcal{D}(X) \rightsquigarrow \mathcal{L}(X) \times \mathcal{D}(X)$*, with distribution* $\omega \in \mathcal{D}(X)$*.*

1. *Iterating* $K \in \mathbb{N}$ *times yields:*

$$OtR^K(\omega) = \sum_{\vec{x} \in X^K} \omega^K(\vec{x}) \, \big| \vec{x}, \omega \big\rangle.$$

2. *The associated* K*-draw channel* $OdR[K] := \pi_1 \circ OtR^K : \mathcal{D}(X) \rightsquigarrow X^K$ *satisfies*

$$OdR[K](\omega) = \omega^K = iid[K](\omega),$$

where iid is the identical and independent channel. □

The situation for ordered transition with deletion is less straightforward. We look at two iterations explicitly, starting from a multiset $\psi \in \mathcal{M}(X)$.

$$\begin{aligned} OtD^1(\psi) &= \sum_{x_1 \in \mathrm{supp}(\psi)} \frac{\psi(x_1)}{\|\psi\|} \, \big| x_1, \psi - 1|x_1\rangle \big\rangle \\ OtD^2(\psi) &= OtD \gg OtD(\psi) \\ &= \sum_{\substack{x_1 \in \mathrm{supp}(\psi), \\ x_2 \in \mathrm{supp}(\psi - 1|x_1\rangle)}} \frac{\psi(x_1)}{\|\psi\|} \cdot \frac{(\psi - 1|x_1\rangle)(x_2)}{\|\psi\| - 1} \, \big| x_1, x_2, \psi - 1|x_1\rangle - 1|x_2\rangle \big\rangle. \end{aligned}$$

Etcetera. We first collect some basic observations in an auxiliary result.

Lemma 6 *Let* $\psi \in \mathcal{M}[L](X)$ *be a multiset/urn of size* $L = \|\psi\|$*.*

1. *Iterating* $K \leq L$ *times satisfies:*

$$OtD^K(\psi) = \sum_{\vec{x} \in X^K,\, \mathrm{acc}(\vec{x}) \leq \psi} \; \prod_{0 \leq i < K} \frac{\big(\psi - \mathrm{acc}(x_1, \ldots, x_i)\big)(x_{i+1})}{L - i} \, \big| \vec{x}, \psi - \mathrm{acc}(\vec{x}) \big\rangle.$$

2. *For $\vec{x} \in X^K$ write $\phi = acc(\vec{x})$. Then:*

$$\prod_{0 \le i < K} \Big(\psi - \mathrm{acc}(x_1, \ldots, x_i)\Big)(x_{i+1}) = \prod\nolimits_y \frac{\psi(y)!}{(\psi(y) - \phi(y))!} = \frac{\psi !!}{(\psi - \phi) !!}.$$

The right-hand-side is thus independent of the sequence $\vec{x}$.

This independence means that any order of the elements of the same multiset of balls gets the same (draw) probability. This is not entirely trivial.

Proof 1. Directly from the definition of the transition channel *OtD*, using Kleisli composition (29.12).

2. Write $\phi = acc(\vec{x})$ as $\phi = \sum_j n_j | y_j \rangle$. Then each element $y_j \in X$ occurs n_j times in the sequence $\vec{x}$. The product

$$\prod_{0 \le i < K} \Big(\psi - \mathrm{acc}(x_1, \ldots, x_i)\Big)(x_{i+1})$$

does not depend on the order of the elements in $\vec{x}$: each element y_j occurs n_j times in this product, with multiplicities $\psi(y_j), \ldots, \psi(y_j) - n_j + 1$, independently of the exact occurrences of the y_j in $\vec{x}$. Thus:

$$\begin{aligned}
\prod_{0 \le i < K} \Big(\psi - \mathrm{acc}(x_1, \ldots, x_i)\Big)(x_{i+1}) &= \prod\nolimits_j \psi(y_j) \cdot \ldots \cdot (\psi(y_j) - n_j + 1) \\
&= \prod\nolimits_j \psi(y_j) \cdot \ldots \cdot (\psi(y_j) - \phi(y_j) + 1) \\
&= \prod\nolimits_j \frac{\psi(y_j)!}{(\psi(y_j) - \phi(y_j))!} \\
&= \prod_{y \in X} \frac{\psi(y)!}{(\psi(y) - \phi(y))!}.
\end{aligned}$$

We can extend the product over j to a product over all $y \in X$ since if $y \notin \mathrm{supp}(\phi)$, then, even if $\psi(y) = 0$,

$$\frac{\psi(y)!}{(\psi(y) - \phi(y))!} = \frac{\psi(y)!}{\psi(y)!} = 1. \qquad \square$$

Theorem 7 *Consider the ordered-transition-with-deletion channel OtD on $\psi \in \mathcal{M}[L](X)$.*

1. *For $K \le L$,*

$$OtD^K(\psi) = \sum_{\phi \le_K \psi} \sum_{\vec{x} \in \mathrm{acc}^{-1}(\phi)} \frac{(\psi - \phi)}{(\psi)} \big| \vec{x}, \psi - \phi \big\rangle.$$

2. *The associated K-draw channel $OdD[K] \coloneqq \pi_1 \circ OtD^K \colon \mathcal{M}[L](X) \rightsquigarrow X^K$ satisfies:*

$$
\begin{aligned}
OdD[K](\psi) &= \sum_{\phi \leq_K \psi} \sum_{\vec{x} \in acc^{-1}(\phi)} \frac{(\psi - \phi)}{(\psi)} \big|\vec{x}\big\rangle \\
&= \sum_{\vec{x} \in X^K,\, acc(\vec{x}) \leq \psi} \frac{(\psi - acc(\vec{x}))}{(\psi)} \big|\vec{x}\big\rangle.
\end{aligned}
$$

As mentioned in the beginning of this section, the latter ordered-draw-deletion distribution does not have its own name.

Proof 1. By combining the two points of Lemma 6 and using:

$$
\prod_{0 \leq i < K} (L - i) = L \cdot (L-1) \cdot \ldots \cdot (L - K + 1) = \frac{L!}{(L-K)!},
$$

we get:

$$
\begin{aligned}
OtD^K(\psi) &= \sum_{\varphi \leq_K \psi} \sum_{\vec{x} \in acc^{-1}(\varphi)} \frac{(L-K)!}{L!} \cdot \prod\nolimits_y \frac{\psi(y)!}{(\psi(y) - \varphi(y))!} \big|\vec{x}, \psi - \varphi\big\rangle \\
&= \sum_{\varphi \leq_K \psi} \sum_{\vec{x} \in acc^{-1}(\varphi)} \frac{(L-K)!}{\prod_y (\psi(y) - \varphi(y))!} \cdot \frac{\prod_y \psi(y)!}{L!} \big|\vec{x}, \psi - \varphi\big\rangle \\
&= \sum_{\varphi \leq_K \psi} \sum_{\vec{x} \in acc^{-1}(\varphi)} \frac{(L-K)!}{(\psi - \varphi)!!} \cdot \frac{\psi!!}{L!} \big|\vec{x}, \psi - \varphi\big\rangle \\
&= \sum_{\varphi \leq_K \psi} \sum_{\vec{x} \in acc^{-1}(\varphi)} \frac{(\psi - \varphi)}{(\psi)} \big|\vec{x}, \psi - \varphi\big\rangle.
\end{aligned}
$$

2. Directly by the previous point. □

29.3.2 Unordered Draws from an Urn

We now concentrate on the transition channels $UtR \colon \mathcal{D}(X) \rightsquigarrow \mathcal{M}(X) \times \mathcal{D}(X)$ and $UtD \colon \mathcal{M}(X) \rightsquigarrow \mathcal{M}(X) \times \mathcal{M}(X)$ in (29.13), for unordered draws. Notice that we are now using $M = \mathcal{M}(X)$ as commutative monoid in the setting of Lemma 4. We immediately formulate a characterisation of iteration. We immediately recognise the resemblance with multinomial and hypergeometric distributions.

Lemma 8 *1. For $\omega \in \mathcal{D}(X)$ and $K \in \mathbb{N}$,*

$$
UtR^K(\omega) = \sum_{\phi \in \mathcal{M}[K](X)} (\phi) \cdot \prod\nolimits_x \omega(x)^{\phi(x)} \big|\phi, \omega\big\rangle.
$$

2. *For $\psi \in \mathcal{M}[L+K](X)$,*

$$UtD^K(\psi) = \sum_{\phi \leq_K \psi} \frac{\prod_x \binom{\psi(x)}{\phi(x)}}{\binom{L+K}{K}} \big|\phi, \psi - \phi\big\rangle = \sum_{\phi \leq_K \psi} \frac{\binom{\psi}{\phi}}{\binom{L+K}{K}} \big|\phi, \psi - \phi\big\rangle.$$

This result shows how the relatively complicated expressions with binomial coefficients $\binom{x}{y}$ in the multinomial and hypergeometric distributions arise from the structure of the monad in Lemma 4.

Proof 1. We use induction on $K \in \mathbb{N}$. For $K = 0$ we have $\mathcal{M}[K](X) = \{\mathbf{0}\}$ and so:

$$UtR^0(\omega) = \eta(\omega) = 1|\mathbf{0}, \omega\rangle = \sum_{\phi \in \mathcal{M}[0](X)} (\phi) \cdot \prod_x \omega(x)^{\phi(x)} \big|\phi, \omega\big\rangle.$$

For the induction step:

$$\begin{aligned}
&UtR^{K+1}(\omega) \\
&= \big(UtR^K \odot UtR\big)(\omega) \\
&\overset{(12)}{=} \sum_{\psi \in \mathcal{M}[1](X),\, \phi \in \mathcal{M}[K](X)} UtR^K(\omega)(\phi, \omega) \cdot UtR(\omega)(\psi, \omega) \big|\psi{+}\phi, \omega\big\rangle \\
&\overset{(\mathrm{IH})}{=} \sum_{y \in X,\, \phi \in \mathcal{M}[K](X)} (\phi) \cdot \Big(\prod_x \omega(x)^{\phi(x)}\Big) \cdot \omega(y) \big|1|y\rangle{+}\phi, \omega\big\rangle \\
&= \sum_{\psi \in \mathcal{M}[K+1](X)} \Big(\sum_y (\psi - 1|y\rangle)\Big) \cdot \prod_x \omega(x)^{\psi(x)} \big|\psi, \omega\big\rangle \\
&\overset{(1)}{=} \sum_{\psi \in \mathcal{M}[K+1](X)} (\psi) \cdot \prod_x \omega(x)^{\psi(x)} \big|\psi, \omega\big\rangle.
\end{aligned}$$

2. For $K = 0$ both sides are equal to the empty multiset $\mathbf{0}$. Next, for a multiset $\psi \in \mathcal{M}[L+K+1](X)$ we have:

$$\begin{aligned}
&UtD^{K+1}(\psi) \\
&= \big(UtD^K \odot UtD\big)(\psi) \\
&\overset{(12)}{=} \sum_{\substack{y \in \mathrm{supp}(\psi),\, \chi \in \mathcal{M}[L](X), \\ \phi \leq_K \psi - 1|y\rangle}} UtD^K(\psi - 1|y\rangle)(\phi, \chi) \cdot \frac{\psi(y)}{L+K+1} \big|\phi + 1|y\rangle, \chi\big\rangle \\
&\overset{(\mathrm{IH})}{=} \sum_{\substack{y \in \mathrm{supp}(\psi), \\ \phi \leq_K \psi - 1|y\rangle}} \frac{\binom{\psi - 1|y\rangle}{\phi}}{\binom{L+K}{K}} \cdot \frac{\psi(y)}{L+K+1} \big|\phi + 1|y\rangle, \psi - 1|y\rangle - \phi\big\rangle \\
&\overset{(*)}{=} \sum_{\substack{y \in \mathrm{supp}(\psi), \\ \phi \leq_K \psi - 1|y\rangle}} \frac{\phi(y) + 1}{K+1} \cdot \frac{\binom{\psi}{\phi + 1|y\rangle}}{\binom{L+K+1}{K+1}} \big|\phi + 1|y\rangle, \psi - (\phi + 1|y\rangle)\big\rangle \\
&= \sum_{\chi \leq_{K+1} \psi,\, y} \frac{\chi(y)}{K+1} \cdot \frac{\binom{\psi}{\chi}}{\binom{L+K+1}{K+1}} \big|\chi, \psi - \chi\big\rangle \\
&= \sum_{\chi \leq_{K+1} \psi} \frac{\binom{\psi}{\chi}}{\binom{L+K+1}{K+1}} \big|\chi, \psi - \chi\big\rangle.
\end{aligned}$$

The equation marked (*) holds, firstly because:

$$(n+1)\cdot\binom{n}{m} = (m+1)\cdot\binom{n+1}{m+1},$$

and thus:

$$\psi(y)\cdot\binom{\psi - 1|\,y\,\rangle}{\phi} = (\phi(y)+1)\cdot\binom{\psi}{\phi + 1|\,y\,\rangle}.$$ □

We are now in a position to describe the multinomial and hypergeometric distributions (29.6) using iterations of the *UtR* and *UtD* maps.

Theorem 9 *1. The K-draw multinomial is the first marginal of the K-iteration of the unordered-with-replacement transition:*

$$\mathit{mulnom}[K] = \pi_1 \odot \mathit{UtR}^K =: \mathit{UdR}[K].$$

2. Similarly the hypergeometric distribution arises from iterated unordered-with-deletion:

$$\mathit{hypgeom}[K] = \pi_1 \odot \mathit{UtD}^K =: \mathit{UdD}[K].$$

Proof Directly by Lemma 8, see the definitions of multinomial and hypergeometric distribution in (29.6). □

Theorems 5, 7 and 9 provide a principled account of the four drawing operations in Table 29.11. This concludes the first part of this paper, on drawing balls from urns.

29.4 Intermezzo on Predicates, Validity and Conditioning

This section recalls the basic constructions associated with (fuzzy) predicates. Predicates play a role as *evidence*, notably in updating and learning.

29.4.1 Predicates

In the current setting of discrete probability we define a predicate on an arbitrary set X to be a function $p\colon X \to [0,1]$. Thus, predicates are fuzzy, taking values in the unit interval $[0,1]$. Such a predicate is called *sharp* if it restricts to $X \to \{0,1\}$, that is, if 0 and 1 are the only possible outcomes. Sharp predicates can be identified with subsets of X and are often called *events*. In general, for a subset $U \subseteq X$ we write $\mathbf{1}_U\colon X \to \{0,1\}$ for the sharp predicate with $\mathbf{1}_U(x) = 1$ iff $x \in U$. We simply write $\mathbf{1}_x$ for $\mathbf{1}_{\{x\}}$ and call $\mathbf{1}_x$ a point predicate.

The set $\mathrm{Pred}(X) := [0,1]^X$ of predicates on X carries a pointwise order. We shall write $\mathbf{0} = \mathbf{1}_\emptyset$ and $\mathbf{1} = \mathbf{1}_X$ for the least and greatest predicates (falsum and truth), with $\mathbf{0}(x) = 0$ and $\mathbf{1}(x) = 1$ for each $x \in X$. Predicates form a commutative monoid via truth $\mathbf{1}$ and conjunction &, where $(p \,\&\, q)(x) = p(x) \cdot q(x)$ involves pointwise multiplication. We have $\mathbf{1}_U \,\&\, \mathbf{1}_V = \mathbf{1}_{U \cap V}$ and thus in particular $\mathbf{1}_U \,\&\, \mathbf{1}_U = \mathbf{1}_U$. However, for properly fuzzy (i.e. non-sharp) predicates p one has $p \,\&\, p \neq p$.

There is also scalar multiplication $r \cdot p$, for $r \in [0,1]$, with $(r \cdot p)(x) = r \cdot p(x)$, and orthocomplement (negation) $p^\perp$ with $p^\perp(x) = 1 - p(x)$. Then: $p^{\perp\perp} = p$ and $\mathbf{1}^\perp = \mathbf{0}$, so that $\mathbf{0}^\perp = \mathbf{1}$. In addition there is a partial sum operation written as $\ovee$. For predicates $p, q \in \mathrm{Pred}(X)$ with $p(x) + q(x) \leq 1$ for all x, one has $p \ovee q \in \mathrm{Pred}(X)$ given by $(p \ovee q)(x) = p(x) + q(x)$. Then, for instance, $p \ovee p^\perp = \mathbf{1}$ and $\mathbf{1}_U \ovee \mathbf{1}_V = \mathbf{1}_{U \cup V}$ if U, V are disjoint subsets. Thus, on a finite set X one can write a predicate $p \in \mathrm{Pred}(X)$ in a normal form as $p = \bigovee_x p(x) \cdot \mathbf{1}_x$. All this structure makes the set of predicates $\mathrm{Pred}(X)$ an effect module $(\mathbf{0}, \ovee, (-)^\perp)$ with a commutative (non-idempotent) monoid structure $(\mathbf{1}, \&)$, see e.g. Jacobs (2015, 2018) for more details.

29.4.2 Validity and Conditioning

For a state $\omega \in \mathcal{D}(X)$ and a predicate $p \in \mathrm{Pred}(X)$ on the same set X we write $\omega \models p$ for the *validity* of p in ω. It is defined as the expected value:

$$\omega \models p \;:=\; \sum_{x \in X} \omega(x) \cdot p(x).$$

Then, for instance, $\omega \models \mathbf{1} = 1$ and $\omega \models \mathbf{0} = 0$.

When this validity $\omega \models p$ is non-zero, we can *update* or *condition* the state $\omega \in \mathcal{D}(X)$ to a new state $\omega|_p \in \mathcal{D}(X)$, in the light of the evidence p. This $\omega|_p$ is a normalised inner product:

$$\omega|_p \;:=\; \sum_{x \in X} \frac{\omega(x) \cdot p(x)}{\omega \models p} \big| x \big\rangle.$$

We shall use the following basic properties, see Jacobs (2015, 2018, 2019) for more details.

Lemma 10 *Assuming the relevant validities are non-zero, one has:*

1. $\omega|_{\mathbf{1}} = \omega$ *and* $\omega|_p|_q = \omega|_{p \& q}$;
2. *Bayes' rule holds:*

$$\omega|_p \models q \;=\; \frac{\omega \models p \,\&\, q}{\omega \models p} \;=\; \frac{(\omega|_q \models p) \cdot (\omega \models q)}{\omega \models p}. \qquad \square$$

A consequence of the first point is that the order of conditioning is irrelevant: $\omega|_p|_q = \omega|_{p\&q} = \omega|_{q\&p} = \omega|_q|_p$. It is for this reason that data, as the material to learn from, are best organised as multisets—where, recall, the order of elements is irrelevant, but not there multiplicity.

Remark 11 There are two points we like to make about conditioning and drawing.

1. One obvious thought is to try and describe a draw from an urn via conditioning. What would this mean? If the urn is a multiset $\psi \in \mathcal{M}(X)$ we can turn it into a distribution $\mathit{Flrn}(\psi) \in \mathcal{D}(X)$ via frequentist learning. Then the thought can be reformulated as a question: can we write:

$$\mathit{Flrn}(\psi - 1|x\rangle) = \mathit{Flrn}(\psi)\big|_p$$

for a suitable predicate p, depending on the element x that is drawn from the urn ψ?
This does not work, as we will illustrate. Take $X = \{a, b\}$ and $\psi = 3|a\rangle + 2|b\rangle$. Then:

$$\mathit{Flrn}(\psi - 1|b\rangle) = \mathit{Flrn}(3|a\rangle + 1|b\rangle) = \tfrac{3}{4}|a\rangle + \tfrac{1}{4}|b\rangle.$$

Now assume that $p\colon \{a, b\} \to [0, 1]$ satisfies:

$$\mathit{Flrn}(\psi)|_p = \mathit{Flrn}(\psi - 1|b\rangle) = \tfrac{3}{4}|a\rangle + \tfrac{1}{4}|b\rangle.$$

This would mean:

$$\frac{3/5 \cdot p(a)}{3/5 \cdot p(a) + 2/5 \cdot p(b)} = \frac{3}{4} \quad \text{and} \quad \frac{1/5 \cdot p(b)}{3/5 \cdot p(a) + 2/5 \cdot p(b)} = \frac{1}{4}.$$

This gives two equations $12 \cdot p(a) = 9 \cdot p(a) + 6 \cdot p(b)$ and $4 \cdot p(b) = 3 \cdot p(a) + 2 \cdot p(b)$. The only solution is $p(a) = p(b) = 0$, so that $p = \mathbf{0}$. But conditioning with falsum $\mathbf{0}$ is not well-defined, since it involves a zero validity.
2. The probabilities in the ordered/unordered transitions with deletion in (29.13) can be described in terms of frequentist learning *Flrn*, as in:

$$\begin{aligned} \mathit{OtD}(\psi) &= \textstyle\sum_x \mathit{Flrn}(\psi)(x)\big|[x], \psi - 1|x\rangle\big\rangle \\ \mathit{UtD}(\psi) &= \textstyle\sum_x \mathit{Flrn}(\psi)(x)\big|1|x\rangle, \psi - 1|x\rangle\big\rangle. \end{aligned}$$

where ψ is a (non-empty) multiset over X. In case we have a predicate p on X, we can use it to describe a 'biased' (or 'non-central') draw by updating this distribution $\mathit{Flrn}(\psi)$ with p in the above expressions, as in:

$$\begin{aligned} \mathit{OtD}_p(\psi) &= \textstyle\sum_x \mathit{Flrn}(\psi)\big|_p(x)\big|[x], \psi - 1|x\rangle\big\rangle \\ \mathit{UtD}_p(\psi) &= \textstyle\sum_x \mathit{Flrn}(\psi)\big|_p(x)\big|1|x\rangle, \psi\big\rangle. \end{aligned}$$

Taking such a bias into account may be useful, for instance in a poll-by-phone, where working people may be under-represented (since they are less often at home).

29.4.3 Predicate Transformation

We have seen in Sect. 29.2.2 that a channel $c\colon X \rightsquigarrow Y$, that is, a function $c\colon X \to \mathcal{D}(Y)$, gives rise to a state transformation function $c \gg (-)\colon \mathcal{D}(X) \to \mathcal{D}(Y)$. It pushes a state forward. One can also pull a predicate backward, along a channel. This is done via a predicate transformation function $\mathrm{Pred}(Y) \to \mathrm{Pred}(X)$, acting in the opposite direction. On $q \in \mathrm{Pred}(Y) = [0, 1]^Y$ it is written as $c \ll q \in \mathrm{Pred}(X) = [0, 1]^X$, defined by:

$$\big(c \ll q\big)(x) \;\coloneqq \sum_{y\in Y} c(x)(y) \cdot q(y).$$

It is not hard to see that predicate transformation preserves $\mathbf{0}, \mathbf{1}, \varovee, (-)^{\perp}$ and scalar multiplication $r \cdot (-)$. However, it does *not* preserve conjunction &. Predicate transformation is functorial, in the sense that $\mathrm{id} \ll q = q$ and $(d \circ c) \ll q = c \ll (d \ll q)$.

State transformation $\gg$, predicate transformation $\ll$, and validity $\models$ are connected via the following fundamental relationship:

$$c \gg \omega \models q \;= \omega \models c \ll q. \tag{29.14}$$

A function $f\colon X \to Y$ is often implicitly promoted to a channel $f\colon X \rightsquigarrow Y$ via $x \mapsto 1|\,f(x)\rangle$. Then $f \ll q$ is simply $q \circ f$. Predicate transformation $\pi_i \ll q$ along a projection is *weakening*, that is moving a predicate to a bigger context. For $p_i \in \mathrm{Pred}(X_i)$ we define parallel conjunction $p_1 \otimes p_2 \in \mathrm{Pred}(X_1 \times X_2)$ as:

$$p_1 \otimes p_2 \;\coloneqq (\pi_1 \ll p_1) \,\&\, (\pi_2 \ll p_2).$$

Thus, $(p_1 \otimes p_2)(x, y) = p_1(x) \cdot p_2(y)$. Hence weakening can also be expressed as parallel conjunction with truth: $\pi_1 \ll p = p \otimes \mathbf{1}$ and $\pi_2 \ll q = \mathbf{1} \otimes q$. Also, $q_1 \,\&\, q_2 = \Delta \ll (q_1 \otimes q_2)$. Further, $(\omega_1 \otimes \omega_2) \models (q_1 \otimes q_2) = (\omega_1 \models q_1) \cdot (\omega_2 \models q_2)$.

29.4.4 Daggers of Channels

Let $c\colon X \rightsquigarrow Y$ be a channel and $\omega \in \mathcal{D}(X)$ be a state on its domain. Under a certain side-condition (see below), one can turn this channel around to obtain a 'dagger' channel $c^{\dagger}_{\omega}\colon Y \rightsquigarrow X$ in the other direction. This new channel is defined via conditioning, as:

$$c_\omega^\dagger(y) \coloneqq \omega|_{c \ll \mathbf{1}_y} = \sum_x \frac{\omega(x) \cdot c(x)(y)}{(c \gg \omega)(y)} \big| x \big\rangle. \tag{29.15}$$

This formulation reveals the side-condition for existence of the dagger: the state $c \gg \omega$ must have full support.

Almost by construction one has:

$$c_\omega^\dagger \gg (c \gg \omega) = \omega \qquad \text{and} \qquad \big(c_\omega^\dagger\big)_{c \gg \omega}^\dagger = c. \tag{29.16}$$

This dagger is the probabilistic analogue of the conjugate transpose of a bounded map between Hilbert spaces. It can be shown that this dagger is functorial, when channels are organised in a suitable category with states as objects, see Clerc et al. (2017) and Cho and Jacobs (2019); Fritz (2020) for more details. This dagger channel is called Bayesian inversion in Clerc et al. (2017) and is related to learning. It will show up later on in Proposition 23, for learning along a channel.

29.4.5 Learning as Likelihood Increase

Consider a validity expression:

$$\underset{\text{distribution / state}}{\omega} \;\models\; \underset{\text{predicate / evidence}}{p} \tag{29.17}$$

One form of learning involves increasing this validity, by changing the state ω into a new state ω' such that $\omega' \models p \;\geq\; \omega \models p$. Thus, in this validity-based learning one takes the evidence p as a given, fixed datum that one needs to adjust to. Learning happens by changing the state ω so that it better fits the evidence. Informally, this is learning by increasing what's right, in contrast to learning by decreasing what's wrong.

The above validity expression $\omega \models p$ in 29.17 may be reorganised as a function $\mathit{val}\colon \mathrm{Pred}(X) \to \mathrm{Pred}\big(\mathcal{D}(X)\big)$, namely $\mathit{val}(p)(\omega) \coloneqq (\omega \models p)$. It then becomes an instance of a *likelihood* function $\mathcal{L}$, which is typically of the form:

$$\mathit{Data} \xrightarrow{\quad \mathcal{L} \quad} \mathrm{Pred}\big(\mathcal{D}(X)\big) = [0, 1]^{\mathcal{D}(X)}.$$

The predicate $\mathcal{L}(d)\colon \mathcal{D}(X) \to [0, 1]$ sends a state $\omega \in \mathcal{D}(X)$ to the likelihood of the data d in that state. The idea in learning is to find a maximum for $\mathcal{L}(d)$, that is, to find the state that makes the data most likely. In (29.17) we use a single predicate as data. Below we shall generalise this to multisets of predicates. This corresponds to the idea that data may come in volumes of separable, possibly identical units, where the order does not matter. The predicates that we use as data/evidence may be point predicates, corresponding to data points.

One important way to learn in the above situation is to update (condition) the state ω with the evidence p, as introduced in Sect. 29.4.2.

Proposition 12 *There is an inequality:*

$$\omega|_p \models p \;\geq \omega \models p.$$

This result captures an important intuition behind conditioning with p: it changes the state so that evidence p becomes 'more true'.

Proof We first show that it suffices to prove an inequality:

$$(\omega \models p^2) \;\geq (\omega \models p)^2, \tag{*}$$

where $p^2 = p \mathbin{\&} p$. Indeed, with (*) we are done, since by Bayes' rule (Lemma 10),

$$\omega_p \models p = \frac{\omega \models p \mathbin{\&} p}{\omega \models p} \geq \frac{\big(\omega \models p\big)^2}{\omega \models p} = \omega \models p.$$

In order to prove the inequality (*) we use the standard notion of variance $\mathit{Var}(\omega, p)$ of predicate p in state ω as validity:

$$\mathit{Var}(\omega, p) \;\coloneqq \omega \models \big(p - (\omega \models p) \cdot \mathbf{1}\big)^2.$$

This number is non-negative since the predicate on right-hand-side of $\models$ is defined as square: $x \mapsto (p(x) - (\omega \models p))^2$. The inequality (*) follows from the (also standard) equation:

$$\big(\omega \models p^2\big) - (\omega \models p)^2 \;=\; \mathit{Var}(\omega, p) \;\geq 0.$$

We show how this equation is obtained in the current setting:

$$\begin{aligned}
&\mathit{Var}(\omega, p)\\
&= \omega \models \big(p - (\omega \models p) \cdot \mathbf{1}\big)^2\\
&= \sum_x \omega(x)\big(p(x) - (\omega \models p)\big)^2\\
&= \sum_x \omega(x)\Big(p(x)^2 - 2(\omega \models p)p(x) + (\omega \models p)^2\Big)\\
&= \Big(\sum_x \omega(x)p^2(x)\Big) - 2(\omega \models p)\Big(\sum_x \omega(x)p(x)\Big) + \Big(\sum_x \omega(x)(\omega \models p)^2\Big)\\
&= \big(\omega \models p^2\big) - 2\big(\omega \models p)(\omega \models p) + (\omega \models p)^2\\
&= \big(\omega \models p^2\big) - (\omega \models p)^2.
\end{aligned}$$

□

29.5 Evaluating Instead of Drawing

The two main topics of this paper are drawing from an urn and learning. This section glues these two topics together. It replaces drawing a ball from an urn, as analysed in Sect. 29.3, by evaluating a predicate via validity $\models$. Both drawing and evaluating are seen as experiments that assign probabilities to (multiple) draws/evaluations.

Our starting point is a single transition step, as used for drawing one ball from an urn, via a channel of the form $\mathcal{D}(X) \to \mathcal{D}(M \times \mathcal{D}(X))$. It can be iterated via the combined monad $\mathcal{D}(M \times -)$ from Lemma 4. Now that we wish to use predicates for making observations, we need to decide what M is in this situation. For convenience we concentrate on the unordered case, and ignore ordered scenarios.

We fix a set $P = \{p_1, \ldots, p_n\}$ of predicates and take $M = \mathcal{M}(P)$ as monoid, containing multisets $\sum_i n_i | p_i \rangle$ of predicates. We require that the predicates in P form a *test*, that is: $p_1 \otimes \cdots \otimes p_n = \mathbf{1}$. If needed, we can always force such predicates to be a test, by switching to $p'_i = \frac{p_i}{p}$, where $p = \sum_i p_i$.

The scenarios that we consider are denoted as unordered-transition-update (*UtU*) and unordered-transition-continue (*UtC*). They are described by the following two transition maps.

$$\begin{array}{ccc} \mathcal{D}(X) \xrightarrow{UtC} \mathcal{D}\big(\mathcal{M}(P) \times \mathcal{D}(X)\big) & \qquad & \mathcal{D}(X) \xrightarrow{UtU} \mathcal{D}\big(\mathcal{M}(P) \times \mathcal{D}(X)\big) \\ \omega \longmapsto \sum_i (\omega \models p_i) \big| 1 | p_i \rangle, \omega \big\rangle & & \omega \longmapsto \sum_i (\omega \models p_i) \big| 1 | p_i \rangle, \omega / p_i \big\rangle \end{array} \tag{29.18}$$

We see that in the 'continue' case *UtC* the state ω remains the same, whereas in the 'update' case *UtU* it is updated with each occurring predicate p_i. In this description we ignore undefinedness of updating, when validities are zero.

The approach of Sect. 29.3 involves iterating (similar) transitions maps and then taking the first projection, for marginalisation. This is what we shall do here as well—without once again elaborating all the details.

Lemma 13 *Fix a state $\omega \in \mathcal{D}(X)$ with set of predicates $P = \{p_1, \ldots, p_n\}$ on X, forming a test. Then for $K \in \mathbb{N}$,*

1. *By iterating the 'continue' map in (29.18) one gets:*

$$\big(\pi_1 \circ UtC^K\big)(\omega) = \sum_{\phi \in \mathcal{M}[K](P)} (\phi) \cdot \prod_i (\omega \models p_i)^{\phi(i)} \, \big| \phi \big\rangle.$$

2. *The 'update' case in (29.18) yields:*

$$\big(\pi_1 \circ UtU^K\big)(\omega) = \sum_{\phi \in \mathcal{M}[K](P)} (\phi) \cdot \big(\omega \models \&_i \; p_i^{\phi(i)}\big) \, \big| \phi \big\rangle.$$

Proof 1. This works very much like in Lemma 8 (1).
2. The fact that the state ω is updated in the UtU transition in (29.18) introduces new dynamics, where Bayes' rule, see Lemma 10 (2), starts to play a role. For instance, after two steps we get:

$$\begin{aligned}\big(\pi_1 \circ UtU^2\big)(\omega) &= \sum\nolimits_{i,j} (\omega|_{p_i} \models p_j) \cdot (\omega \models p_i)\, \big|\, 1|\, p_i\,\rangle + 1|\, p_j\,\rangle \big\rangle \\ &= \sum\nolimits_{i,j} (\omega \models p_i \;\&\; p_j)\, \big|\, 1|\, p_i\,\rangle + 1|\, p_j\,\rangle \big\rangle.\end{aligned}$$

Continuing yields the above formula in point (2). □

One interesting thing to note is that different validity expressions arise for a multiset ϕ, namely $\prod_i (\omega \models p_i)^{\phi(i)}$ with a product on the outside, and $\omega \models \&_i\; p_i^{\phi(i)}$ with a product (conjunction) on the inside. This difference will be explored further in the remainder of this article.

Also it is noteworthy that the probabilities in the two formulas in Lemma (13) add up to one because the predicates p_i form a test. For point (1) we use the Multinomial Theorem (Lemma 3) in the obvious way:

$$\begin{aligned}\sum_{\phi \in \mathcal{M}[K](P)} (\phi) \cdot \prod\nolimits_i (\omega \models p_i)^{\phi(i)} &= \big(\sum\nolimits_i\, \omega \models p_i\big)^K \\ &= \big(\omega \models \textstyle\bigovee_i\, p_i\big)^K \;=\; \big(\omega \models \mathbf{1}\big)^K \;=\; 1^K \;=\; 1.\end{aligned}$$

Again by the Multinomial Theorem, but now in slightly different form, the probabilities in point (2) add up to one:

$$\begin{aligned}\sum_{\phi \in \mathcal{M}[K](P)} (\phi) \cdot \big(\omega \models \&_i\; p_i^{\phi(i)}\big) &= \omega \models \bigovee_{\phi \in \mathcal{M}[K](P)} (\phi) \cdot \big(\&_i\; p_i^{\phi(i)}\big) \\ &= \omega \models \big(\textstyle\bigovee_i\, p_i\big)^K \;=\; \omega \models \mathbf{1}^K \;=\; \omega \models \mathbf{1} \;=\; 1.\end{aligned}$$

The two expressions $\prod_i (\omega \models p_i)^{\phi(i)}$ and $\omega \models \&_i\; p_i^{\phi(i)}$ give, in general, different outcomes—even though when multiplied with (ϕ) and summed they both add up to one. This difference will be illustrated next.

Example 14 Let's consider a political party that has to decide on its future policies. We simplify these options to left (L), centre (C), and right (R) in a space $X = \{L, C, R\}$. The party leadership leans to the right. Its position is captured by the following distribution, giving a convex combination of the three directions.

$$\omega := \tfrac{1}{5}|\, L\,\rangle + \tfrac{3}{10}|\, C\,\rangle + \tfrac{1}{2}|\, R\,\rangle.$$

The party has four factions; their positions on the three options L, C, R are expressed via the following percentages.

	Faction 1 (%)	Faction 2 (%)	Faction 3 (%)	Faction 4 (%)
Left	30	10	50	10
Centre	20	30	20	30
Right	30	20	30	20

We can read these columns as four predicates p_1, p_2, p_3, p_4 on the space X, where $p_4 = p_2$. Explicitly:

$$\begin{aligned} p_1 &= \tfrac{3}{10} \cdot \mathbf{1}_L + \tfrac{2}{10} \cdot \mathbf{1}_L + \tfrac{3}{10} \cdot \mathbf{1}_L \\ p_2 &= \tfrac{1}{10} \cdot \mathbf{1}_L + \tfrac{3}{10} \cdot \mathbf{1}_L + \tfrac{2}{10} \cdot \mathbf{1}_L \\ p_3 &= \tfrac{5}{10} \cdot \mathbf{1}_L + \tfrac{2}{10} \cdot \mathbf{1}_L + \tfrac{3}{10} \cdot \mathbf{1}_L. \end{aligned}$$

The four predicates of the table form a test: $p_1 \otimes p_2 \otimes p_3 \otimes p_4 = \mathbf{1}$. The table can be described as a multiset $\phi = 1|p_1\rangle + 2|p_2\rangle + 1|p_3\rangle$ of predicates.

The validities $\omega \models p_i$ can be interpreted as the level of support for the leadership's position within the corresponding faction. It is not hard to see that:

$$\omega \models p_1 = \tfrac{27}{100} \quad \omega \models p_2 = \tfrac{21}{100} \quad \omega \models p_3 = \tfrac{31}{100}. \tag{29.19}$$

How should the party's leadership measure the total support for its position ω within all factions?

1. In one scenario, four secretaries of the leadership visit the four factions separately and collect their separate support values (29.19). The total support can then be computed as product of the individual support values:

$$(\omega \models p_1) \cdot (\omega \models p_2)^2 \cdot (\omega \models p_3) = \tfrac{27 \cdot 21^2 \cdot 31}{10^8} = \tfrac{369.117}{10^8} \approx 0.0037.$$

   This involves computing $\prod_i (\omega \models p_i)^{\phi(i)}$, as in Lemma 13 (1).
2. Alternatively the party may hold a congress where each faction expresses its position, as percentages in the above table, in order. A mathematical savvy secretary of the leadership may then quickly start computing, after hearing the intermediary results. After announcement of the first faction's position, in the form of predicate p_1, this secretary calculates the validity $\omega \models p_1 = \frac{27}{100}$ and the updated distribution:

$$\begin{aligned} \omega|_{p_1} &= \frac{1/5 \cdot 3/10}{27/100}|L\rangle + \frac{3/10 \cdot 2/10}{27/100}|C\rangle + \frac{1/2 \cdot 3/10}{27/100}|R\rangle \\ &= \tfrac{2}{9}|L\rangle + \tfrac{2}{19}|C\rangle + \tfrac{5}{9}|R\rangle. \end{aligned}$$

   Next, after faction 2 announces its percentages p_2 the secretary computes the validity in the latest state, and also the next update:

$$\omega|_{p_1} \models p_2 = \tfrac{1}{5} \quad \omega|_{p_1}|_{p_2} = \omega|_{p_1 \& p_2} = \tfrac{1}{9}|L\rangle + \tfrac{1}{3}|C\rangle + \tfrac{5}{9}|R\rangle.$$

Continuing like this we get:

$$\omega|_{p_1}|_{p_2} \models p_3 = \tfrac{13}{45} \quad \omega|_{p_1}|_{p_2}|_{p_3} = \tfrac{5}{26}|L\rangle + \tfrac{3}{13}|C\rangle + \tfrac{15}{26}|R\rangle.$$

After hearing the percentages of faction 4 the secretary calculates the remaining validity $\omega|_{p_1}|_{p_2}|_{p_3} \models p_2 = \frac{53}{260}$, and then also the product of all these validities:

$$\begin{aligned} &(\omega \models p_1) \cdot (\omega|_{p_1} \models p_2) \cdot (\omega|_{p_1}|_{p_2} \models p_3) \cdot (\omega|_{p_1}|_{p_2}|_{p_3} \models p_2) \\ = &\tfrac{18.603}{5.850.000} \approx 0.0032. \end{aligned}$$

We see that this second calculation shows slightly less support.

Finally, via Bayes' rule we may also compute this second outcome as validity $\omega \models p_1 \;\&\; p_2 \;\&\; p_3 \;\&\; p_2 = \omega \models \&_i\, p_i^{\phi(i)}$. The latter expression occurs in Lemma 13 (2). We conclude that the two approaches of Lemma 13 are really different.

29.6 Internal and External Likelihood and Learning, with Multiset Data

The straightforward way to describe collections of data "of type X" is as multisets over X, that is, as elements of $\mathcal{M}(X)$. In line with the previous section we are going to push things to a slightly higher level of abstraction: we will use *multisets of predicates on* X as data of type X. This will include point data of type X via point predicates $\mathbf{1}_x$ for $x \in X$, giving an inclusion $\mathcal{M}(X) \hookrightarrow \mathcal{M}\big(\mathrm{Pred}(X)\big)$. Using the more general predicates instead of points is useful, as we will illustrate below, for instance when we deal with incomplete information—caused for instance by measurement or transmission errors. We can handle such situations e.g. via uniform predicates, giving each element the same probability. We shall also see that learning along a channel can be handled via multisets of predicates, even if we start from point data.

The first step that we need to take is to formulate likelihood for such data, as multisets of predicates. After all, learning is about increasing likelihood. As we have already seen in Sect. 29.5, especially in Lemma 13, there are two forms of likelihood that make sense. We call them *external* and *internal* and write them as $\models_{\mathrm{E}}$ and $\models_{\mathrm{I}}$. The distinction is first made in Jacobs (2019), but it is not explicit elsewhere—as far as we are aware. Both forms of likelihood make sense, and also the associated learning methods. We shall discuss the non-trivial, unsolved issue of when to use which likelihood (and learning method) in Sect. 29.9.

We fix the general formulation of these two forms of likelihood.[1]

Definition 15 Let $\omega \in \mathcal{D}(X)$ be a state and $\psi \in \mathcal{M}\big(\mathrm{Pred}(X)\big)$ be a multiset of *data*.

1. The *external likelihood* of the data in this state is defined as:

$$\omega \models_{\mathrm{E}} \psi := \prod\nolimits_p (\omega \models p)^{\psi(p)}.$$

2. The *internal likelihood* is:

$$\omega \models_{\mathrm{I}} \psi := \omega \models \&_p\ p^{\psi(p)}.$$

The log-likelihood is the (natural) logarithm of these expressions. In the external case it can be computed simply as sum $\sum_p \psi(p) \cdot \log(\omega \models p)$. In the internal case we can compute the log-likelihood as an iterated sum, using Bayes' rule.

This log-likelihood is useful since these likelihoods can be become very small in the presence of lots of data.

One could argue that external and internal likelihood require an additional multinomial coefficient (ϕ), like in Lemma 13, in order to accommodate all possible orderings of the data items. However, when considering likelihood, it is usually omitted. Learning aims at increasing likelihood and a constant factor is then irrelevant.

The following result is standard, see e.g. Koller and Friedman (2009, Example 17.5). A proof is in the appendix.

Proposition 16 *For point data $\phi \in \mathcal{M}_*(X)$ the predicate "external likelihood of ϕ"*

$$\mathcal{D}(X) \xrightarrow{(-) \models_{E} \phi} [0, 1]$$

takes its maximum at the distribution $\mathrm{Flrn}(\phi) \in \mathcal{D}(X)$ *that is obtained by frequentist learning.* □

The next observation is of interest mostly for categorical aficionados.

Remark 17 As briefly mentioned in Sect. 29.2.1, the set of multisets $\mathcal{M}(X)$ is the free commutative monoid on X. Both forms of likelihood $\models_{\mathrm{E}}$ and $\models_{\mathrm{I}}$ in Definition 15 can be understood as maps $\mathcal{L}_E, \mathcal{L}_I : \mathcal{M}\big(\mathrm{Pred}(X)\big) \to \mathrm{Pred}\big(\mathcal{D}(X)\big)$, arising via this freeness, but in different ways:

$$\begin{tikzcd} \mathcal{M}\big(\mathrm{Pred}(X)\big) \arrow[rr, dashed, "\overline{val}"] \arrow[dr, dashed, "\overline{\mathrm{id}}"'] & & \mathrm{Pred}\big(\mathcal{D}(X)\big) \\ & \mathrm{Pred}(X) \arrow[ur, "val"'] & \end{tikzcd} \qquad \text{as} \qquad \begin{cases} \mathcal{L}_E = \overline{val} \\ \mathcal{L}_I = val \circ \overline{\mathrm{id}}. \end{cases}$$

where $val\colon \mathrm{Pred}(X) \to \mathrm{Pred}(\mathcal{D}(X))$ is $val(p)(\omega) := \omega \models p$.

[1] In Jacobs (2019) the phrases 'multiple state' and 'copied state' are used for what we here started calling 'external' and 'internal'.

1. Predicates with conjunction $(\mathbf{1}, \&)$ form a commutative monoid. Hence the above validity map *val* can be extended uniquely to a homomorphism of monoids $\overline{val}\colon \mathcal{M}(\mathrm{Pred}(X)) \to \mathrm{Pred}(\mathcal{D}(X))$, given by:

$$\begin{aligned}\overline{val}\Big(\textstyle\sum_i n_i|p_i\rangle\Big)(\omega) &= \big(val(p_1)^{n_1} \,\&\, \cdots \,\&\, val(p_k)^{n_k}\big)(\omega) \\ &= val(p_1)(\omega)^{n_1} \cdot \ldots \cdot val(p_k)(\omega)^{n_k} \\ &= (\omega \models p_1)^{n_1} \cdot \ldots \cdot (\omega \models p_k)^{n_k} \\ &= \omega \models_{\mathrm{E}} \textstyle\sum_i n_i|p_i\rangle.\end{aligned}$$

   We use that conjunction & of predicates is defined via pointwise multiplication.
2. The identity map $\mathrm{id}\colon \mathrm{Pred}(X) \to \mathrm{Pred}(X)$ can also be extended to a homomorphims of monoids $\overline{\mathrm{id}}\colon \mathcal{M}(\mathrm{Pred}(X)) \to \mathrm{Pred}(X)$, via:

$$\overline{\mathrm{id}}\Big(\textstyle\sum_i n_i|p_i\rangle\Big) = p_1^{n_1} \,\&\, \cdots \,\&\, p_k^{n_k}.$$

   Hence, $\big(val \circ \overline{\mathrm{id}}\big)(\sum_i n_i|p_i\rangle)(\omega) \;=\; \omega \models p_1^{n_1} \,\&\, \cdots \,\&\, p_k^{n_k} \;=\; \omega \models_{\mathrm{I}} \sum_i n_i|p_i\rangle.$

We have described (a form of) learning in Sect. 29.4.5 as validity increase, whereby we immediately mentioned that validity is really used as a likelihood function. For the explicitly defined likelihood functions $\models_{\mathrm{E}}$ and $\models_{\mathrm{I}}$ introduced in this section there are also associated (different) learning steps, both as 'likelihood increase'.

Theorem 18 *For a state $\omega \in \mathcal{D}(X)$ with a data $\psi = \sum_i n_i|p_i\rangle \in \mathcal{M}_*(\mathrm{Pred}(X))$, one has:*

1. $\omega \models_{E} \psi \;\leq\; Elrn(\omega, \psi) \models_{E} \psi$, *where:*

$$Elrn(\omega, \psi) \coloneqq \sum\nolimits_i \frac{n_i}{n} \cdot \omega|_{p_i} = \sum\nolimits_p Flrn(\psi)(p) \cdot \omega|_p,$$

for $n = \|\psi\| = \sum_i n_i > 0.$

2. $\omega \models_{I} \psi \;\leq\; Ilrn(\omega, \psi) \models_{I} \psi$, *where:*

$$Ilrn(\omega, \psi) \coloneqq \omega|_{\&_i\, p_i^{n_i}}.$$

Proof The second point is an immediate consequence of Proposition 12. A proof of the first point is given in the appendix. □

We notice that frequentist learning $Flrn(\phi)$ for $\phi \in \mathcal{M}_*(X)$ is a special case of external learning form the uniform state υ with point data ϕ, namely $Elrn(\upsilon, \phi)$, since $\upsilon|_{\mathbf{1}_x} = 1|x\rangle$. In fact, one can take instead of υ any state with full support. External learning from point data immediately jumps to the maximal likelihood, see Proposition 16.

External learning also satisfies, like frequentist learning, the more-is-the-same property (29.8), namely:

$$Elrn(\omega, K \cdot \psi) = Elrn(\omega, \psi), \quad \text{for } K > 0. \tag{29.20}$$

This external learning thus combines frequentist and Bayesian learning, via the convex combination of (Bayesian) updated states, see the formulation of *Elrn* in Theorem 18 (1).

An important advantage of *internal* learning is that it can be done incrementally: when new data arrives, one can continue learning with what has been learned so far, simply by performing a new conditioning. In particular, $Ilrn(\omega, K \cdot \phi)$ is not the same as $Ilrn(\omega, \phi)$, but involves K iterations of learning from ϕ.

Proposition 19 $Ilrn(\omega, \phi + \psi) = Ilrn\big(Ilrn(\omega, \phi), \psi\big)$.

Proof Since:

$$\begin{aligned}
Ilrn(\omega, \phi + \psi) &= \omega\big|_{\&_p\, p^{(\phi+\psi)(p)}} \\
&= \omega\big|_{\&_p\, p^{\phi(p)+\psi(p)}} \\
&= \omega\big|_{\&_p\, p^{\phi(p)}\, \&\, p^{\psi(p)}} \\
&= \omega\big|_{(\&_p\, p^{\phi(p)})\, \&\, (\&_p\, p^{\psi(p)})} \\
&= \omega\big|_{\&_p\, p^{\phi(p)}}\big|_{\&_p\, p^{\psi(p)}} \qquad \text{by Lemma 10 (1)} \\
&= Ilrn\big(Ilrn(\omega, \phi), \psi\big).
\end{aligned}$$

□

More technically, this result says that the multiset monoid $\mathcal{M}(\mathrm{Pred}(X))$ acts on $\mathcal{D}(X)$ via internal learning. Indeed, we also have $Ilrn(\omega, \mathbf{0}) = \omega|_{\mathbf{1}} = \omega$.

Interestingly, in the *external* case we can express an increase in likelihood equivalently as a decrease in divergence. Informally, this means that learning from what's right coincides with learning from what's wrong. The divergence is the familiar Kullback-Leibler divergence, which is defined for two states $\sigma, \tau \in \mathcal{D}(X)$ as:

$$D_{KL}(\sigma, \tau) := \sum_{x \in X} \sigma(x) \cdot \log\left(\frac{\sigma(x)}{\tau(x)}\right).$$

The logarithm log is typically the 2-log.

For simplicity we shall assume, like in Sect. 29.5, that the predicates $P = \{p_1, \ldots, p_n\}$ at hand form a test, i.e. satisfy $\bigovee_i p_i = \mathbf{1}$. In this way we can define an evaluation channel:

$$\mathcal{D}(X) \xrightarrow{\text{ev}} P \qquad \text{by} \qquad \text{ev}(\omega) := \sum\nolimits_i (\omega \models p_i)\big|\, p_i \big\rangle. \tag{29.21}$$

Proposition 20 *Let $\phi \in \mathcal{M}(P)$ be a non-empty multiset of data, over a set of predicates $P = \{p_1, \ldots, p_n\} \subseteq \mathrm{Pred}(X)$ forming a test. Then, for two states ω, ω',*

$$\big(\omega \models_{E} \phi\big) \;\leq\; \big(\omega' \models_{E} \phi\big) \Longleftrightarrow D_{KL}\big(Flrn(\phi), \text{ev}(\omega)\big) \;\geq\; D_{KL}\big(Flrn(\phi), \text{ev}(\omega')\big).$$

Thus, increasing likelihood of data ϕ in state ω corresponds to decreasing divergence between the distributions $\mathit{Flrn}(\phi)$ and $\mathrm{ev}(\omega)$.

Proof Let $\phi = \sum_i n_i |\, p_i \,\rangle$ and $n = \|\phi\| = \sum_i n_i$. We first notice that:

$$
\begin{aligned}
D_{KL}\big(\mathit{Flrn}(\phi), \mathrm{ev}(\omega)\big) &= \sum\nolimits_i \frac{n_i}{n} \cdot \log\left(\frac{n_i/n}{\omega \models p_i}\right) \\
&= \sum\nolimits_i \frac{n_i}{n} \cdot \log\left(\frac{n_i}{n}\right) - \sum\nolimits_i \frac{n_i}{n} \cdot \log\big(\omega \models p_i\big) \\
&= \left(\sum\nolimits_i \frac{n_i}{n} \cdot \log\left(\frac{n_i}{n}\right)\right) - \frac{1}{n} \cdot \log\left(\prod\nolimits_i (\omega \models p_i)^{n_i}\right) \\
&= -H(\phi) - \frac{1}{n} \cdot \log\big(\omega \models_{\mathrm{E}} \phi\big).
\end{aligned}
$$

where $H(\phi)$ is the entropy of ϕ. The result now follows easily, using that log preserves and reflects the order:

$$
\begin{aligned}
& D_{KL}\big(\mathit{Flrn}(\phi), \mathrm{ev}(\omega)\big) \;\geq\; D_{KL}\big(\mathit{Flrn}(\phi), \mathrm{ev}(\omega')\big) \\
\Longleftrightarrow\; & -H(\phi) - \frac{1}{n} \cdot \log\big(\omega \models_{\mathrm{E}} \phi\big) \;\geq\; -H(\phi) - \frac{1}{n} \cdot \log\big(\omega' \models_{\mathrm{E}} \phi\big) \\
\Longleftrightarrow\; & \log\big(\omega \models_{\mathrm{E}} \phi\big) \;\leq\; \log\big(\omega' \models_{\mathrm{E}} \phi\big) \\
\Longleftrightarrow\; & \big(\omega \models_{\mathrm{E}} \phi\big) \;\leq\; \big(\omega' \models_{\mathrm{E}} \phi\big).
\end{aligned}
$$

□

In the remainder of this section we consider illustrations of external and internal learning.

29.6.1 External Learning with Complete and Missing Data

We examine an example from Jensen and Nielsen (2007, Sect. 6.2.1), first with complete and then with missing data. The goal is two-fold: to illustrate the external learning step of Theorem 18 (1), and also to show why it pays off to have multisets of *predicates* as data, instead of multisets of elements. These predicates will be used to deal with the uncertainty given by missing data.

The example involves pregnancy of cows, which can be deduced from a urine test and a blood test. A simple Bayesian network structure is assumed, which we write as string diagram with explicit copy:

$$
\boxed{\textit{pregnancy}} \qquad \text{with sets} \qquad \begin{cases} P = \{p, p^{\perp}\} \\ B = \{b, b^{\perp}\} \\ U = \{u, u^{\perp}\}. \end{cases} \tag{29.22}
$$

The elements p and $p^{\perp}$ represent 'pregnancy' and 'no pregnancy', respectively. Similarly, $b, b^{\perp}$ and $u, u^{\perp}$ represent a positive and negative blood/urine test.

case	Pregn	Blood	Urine
1	$p^\perp$	b	u
2	p	$b^\perp$	u
3	p	b	$u^\perp$
4	p	b	$u^\perp$
5	$p^\perp$	$b^\perp$	u
6	$p^\perp$	$b^\perp$	$u^\perp$
7	$p^\perp$	b	u
8	p	b	$u^\perp$

case	Pregn	Blood	Urine
1	?	b	u
2	p	$b^\perp$	u
3	p	b	?
4	p	b	$u^\perp$
5	?	$b^\perp$	?

Fig. 29.1 Two tables with data to learn an interpretation of the Bayesian network (29.22), with 'complete' data on the left and with 'missing' data on the right

We have two tables with data in Fig. 29.1: the one on the left below contains 'complete' data that can be used directly for learning. The table on the right (copied from Jensen and Nielsen (2007)) uses a question mark for a missing item. In both cases the aim is to learn an interpretation of the above Bayesian network. This is commonly called parameter learning. It involves learning a state on P and two channels $P \rightsquigarrow B$ and $P \rightsquigarrow U$. These channels correspond to conditional probability tables in Bayesian networks, see Jacobs and Zanasi (2021) for more details. The state and two channels can be obtained from a joint state on $P \times B \times U$ by marginalisation and disintegration (channel extraction). Our aim is thus to first learn such a joint state from the tables.

We start with the table on the left. It is translated into a multiset ϕ of point predicates on the product space $P \times B \times U$. The table translates directly into:

$$\phi = 2|\mathbf{1}_{(p^\perp,b,u)}\rangle + 1|\mathbf{1}_{(p,b^\perp,u)}\rangle + 3|\mathbf{1}_{(p,b,u^\perp)}\rangle + 1|\mathbf{1}_{(p^\perp,b^\perp,u)}\rangle + 1|\mathbf{1}_{(p^\perp,b^\perp,u^\perp)}\rangle.$$

Since there is no prior knowledge we use the uniform state $\upsilon \in \mathcal{D}(P \times B \times U)$ in external learning, giving, according to Theorem 18 (1):

$$\begin{aligned}
\omega &:= \mathit{Elrn}(\upsilon, \phi) \\
&= \tfrac{2}{8}\upsilon|_{\mathbf{1}_{(p^\perp,b,u)}} + \tfrac{1}{8}\upsilon|_{\mathbf{1}_{(p,b^\perp,u)}} + \tfrac{3}{8}\upsilon|_{\mathbf{1}_{(p,b,u^\perp)}} + \tfrac{1}{8}\upsilon|_{\mathbf{1}_{(p^\perp,b^\perp,u)}} + \tfrac{1}{8}\upsilon|_{\mathbf{1}_{(p^\perp,b^\perp,u^\perp)}} \\
&= \tfrac{2}{8}|p^\perp, b, u\rangle + \tfrac{1}{8}|p, b^\perp, u\rangle + \tfrac{3}{8}|p, b, u^\perp\rangle + \tfrac{1}{8}|p^\perp, b^\perp, u\rangle + \tfrac{1}{8}|p^\perp, b^\perp, u^\perp\rangle \\
&= \mathit{Flrn}(\phi).
\end{aligned}$$

Notice that internal learning would not work in this situation because the conjunction & of these point predicates is falsum $\mathbf{0}$.

This learned joint state $\omega \in \mathcal{D}(P \times B \times U)$ has first marginal $\pi_1 \gg \omega = \frac{1}{2}|p\rangle + \frac{1}{2}|p^\perp\rangle \in \mathcal{D}(P)$, which is used as interpretation of the Pregnancy state in (29.22). Channels $c\colon P \rightsquigarrow B$ and $d\colon P \rightsquigarrow U$ are extracted from ω as conditional probabilities, via disintegration (see Sect. 29.2.2):

$$\begin{aligned}
c(p) &= \frac{\omega(p,b,u)+\omega(p,b,u^\perp)}{\omega(p,b,u)+\omega(p,b,u^\perp)+\omega(p,b^\perp,u)+\omega(p,b^\perp,u^\perp)}\big|b\big\rangle \\
&\quad + \frac{\omega(p,b^\perp,u)+\omega(p,b^\perp,u^\perp)}{\omega(p,b,u)+\omega(p,b,u^\perp)+\omega(p,b^\perp,u)+\omega(p,b^\perp,u^\perp)}\big|b^\perp\big\rangle \\
&= \frac{3/8}{3/8+1/8}|b\rangle + \frac{1/8}{3/8+1/8}|b^\perp\rangle \;=\; \tfrac{3}{4}|b\rangle + \tfrac{1}{4}|b^\perp\rangle \\
c(p^\perp) &= \frac{1/4}{1/4+1/8+1/8}|b\rangle + \frac{1/8+1/8}{1/4+1/8+2/8}|b^\perp\rangle \;=\; \tfrac{1}{2}|b\rangle + \tfrac{1}{2}|b^\perp\rangle
\end{aligned}$$

In the same way one gets $d(p) = \frac{1}{4}|u\rangle + \frac{3}{4}|u^\perp\rangle$ and $d(p^\perp) = \frac{3}{4}|u\rangle + \frac{1}{4}|u^\perp\rangle$. The table on the left in Fig. 29.1 thus gives us an interpretation of the Bayesian network in (29.22).

We now turn to the table on the right in Fig. 29.1 with missing data. For cases 1,3,5 we don't use point predicates, like above, but predicates $p_1, p_3, p_5 \colon P \times B \times U \to [0,1]$ given by:

$$\begin{array}{c}
p_1(p,b,u) \;=\; p_1(p^\perp,b,u) \;=\; 1 \qquad\qquad p_3(p,b,u) \;=\; p_3(p,b,u^\perp) \;=\; 1 \\
p_5(p,b^\perp,u) \;=\; p_5(p^\perp,b^\perp,u) \;=\; p_5(p,b^\perp,u^\perp) \;=\; p_5(p^\perp,b^\perp,u^\perp) \;=\; 1.
\end{array}$$

These predicate are zero elsewhere—and are thus sharp. We thus translate the table on the right in Fig. 29.1 to the multiset of predicates:

$$\phi = 1|\,p_1\,\rangle + 1|\,\mathbf{1}_{(p,b^\perp,u)}\rangle + 1|\,p_3\,\rangle + 1|\,\mathbf{1}_{(p,b,u^\perp)}\rangle + 1|\,p_5\,\rangle.$$

We again follow Theorem 18 (1):

$$\begin{aligned}
\rho &\coloneqq \mathit{Elrn}\big(\upsilon,\phi\big) \\
&= \tfrac{1}{5}\upsilon|_{p_1} + \tfrac{1}{5}\upsilon|_{\mathbf{1}_{(p,b^\perp,u)}} + \tfrac{1}{5}\upsilon|_{p_3} + \tfrac{1}{5}\upsilon|_{\mathbf{1}_{(p,b,u^\perp)}} + \tfrac{1}{5}\upsilon|_{p_5} \\
&= \tfrac{1}{10}|\,p,b,u\,\rangle + \tfrac{1}{10}|\,p^\perp,b,u\,\rangle + \tfrac{1}{5}|\,p,b^\perp,u\,\rangle + \tfrac{1}{10}|\,p,b,u\,\rangle + \tfrac{1}{10}|\,p,b,u^\perp\,\rangle \\
&\qquad + \tfrac{1}{5}|\,p,b,u^\perp\,\rangle + \tfrac{1}{20}|\,p,b^\perp,u\,\rangle + \tfrac{1}{20}|\,p^\perp,b^\perp,u\,\rangle + \tfrac{1}{20}|\,p,b^\perp,u^\perp\,\rangle + \tfrac{1}{20}|\,p^\perp,b^\perp,u^\perp\,\rangle. \\
&= \tfrac{1}{5}|\,p,b,u\,\rangle + \tfrac{3}{10}|\,p,b,u^\perp\,\rangle + \tfrac{1}{4}|\,p,b^\perp,u\,\rangle + \tfrac{1}{20}|\,p,b^\perp,u^\perp\,\rangle \\
&\qquad + \tfrac{1}{10}|\,p^\perp,b,u\,\rangle + \tfrac{1}{20}|\,p^\perp,b^\perp,u\,\rangle + \tfrac{1}{20}|\,p^\perp,b^\perp,u^\perp\,\rangle.
\end{aligned}$$

This yields a different interpretation for the Bayesian network (string diagram) in (29.22): the first marginal of ρ is $\frac{5}{8}|\,b\,\rangle + \frac{3}{8}|\,b^\perp\,\rangle$. The extracted channels $c \colon P \rightsquigarrow B$ and $d \colon P \rightsquigarrow U$ from ρ are obtained as before:

$$\begin{aligned}
c(p) &= \frac{1/5+3/10}{1/5+3/10+1/4+1/20}|\,b\,\rangle + \frac{1/4+1/20}{1/5+3/10+1/4+1/20}|\,b^\perp\,\rangle \;=\; \tfrac{5}{8}|\,b\,\rangle + \tfrac{3}{8}|\,b^\perp\,\rangle. \\
c(p^\perp) &= \frac{1/10}{1/10+1/20+1/20}|\,b\,\rangle + \frac{1/20+1/20}{1/10+1/20+1/20}|\,b^\perp\,\rangle \;=\; \tfrac{1}{2}|\,b\,\rangle + \tfrac{1}{2}|\,b^\perp\,\rangle.
\end{aligned}$$

Similarly $d(p) = \frac{9}{16}|u\rangle + \frac{7}{16}|u^{\perp}\rangle$ and $d(p^{\perp}) = \frac{3}{4}|u\rangle + \frac{1}{4}|u^{\perp}\rangle$. These outcomes are precisely as described in Jensen and Nielsen (2007, Sect. 6.2.1); there, the computation is presented as an instance of the (E-part of the) EM-algorithm (see Sect. 29.8 below).

29.7 Learning Coin Bias, Along a Channel

So far we have looked at likelihood of data in a state and at how to increase this likelihood by adapting the state. We have considered the situation where the state and data are on *the same* set X. In practice, it often happens that there is a difference, like in:

$$\begin{array}{ccc} X & \overset{e}{\longrightarrow\!\!\circ} & Y \\ \uparrow & & \uparrow \\ \text{state to be learned} & & \text{data} \end{array} \tag{29.23}$$

We will assume that there is a channel between the two spaces—as in the above picture—that can be used to mediate between the given data and the state that we wish to learn. This is what we call 'learning along a channel'. This learning challenge is often described in terms of 'hidden' or 'latent' variables, since the elements of the space X are not directly accessible, but only indirectly via the 'emission' channel e. This forms the E-part of what is called Expectation-Maximisation (EM), see Sect. 29.8, where, in the M-part, the channel e becomes a learning goal in itself. In Expectation-Maximisation these E- and M-parts are alternated. But here we first concentrate on the E-part only and assume that the channel e is given and remains fixed. This E-part typically uses what we call external learning.

Now suppose, in the setting (29.23) we have data $\psi \in \mathcal{M}(\mathrm{Pred}(Y))$ in the form of multiset of predicates on the codomain Y of the channel. We can easily turn this multiset on $\mathrm{Pred}(Y)$ into a multiset on $\mathrm{Pred}(X)$, via predicate transformation (and functoriality of $\mathcal{M}$). Then we can both externally and internally learn 'along a channel', using the formulations of Theorem 18:

$$\begin{array}{rcl} \mathit{Elrn}(\omega, e, \psi) \coloneqq \mathit{Elrn}\big(\omega, \sum_q \psi(q)\big|e \ll q\big\rangle\big) & = & \sum_q \frac{\psi(q)}{\|\psi\|} \cdot \omega|_{e \ll q} \\ \mathit{Ilrn}(\omega, e, \psi) \coloneqq \mathit{Ilrn}\big(\omega, \sum_q \psi(q)\big|e \ll q\big\rangle\big) & = & \omega\big|_{\&_q\, (e \ll q)^{\psi(q)}}. \end{array} \tag{29.24}$$

Notice that we overload the notation *Elrn*/*Ilrn*, since on the left of $\coloneqq$ it is used with three arguments, for learning along a channel, which is defined in terms of the original notion, on the right of $\coloneqq$, with two arguments.

Proposition 21 *The above definitions (29.24) give the following likelihood increases. For $\omega' \coloneqq \mathit{Elrn}(\omega, e, \psi)$ one gets:*

$$e \gg \omega' \models_E \psi \;\geq e \gg \omega \models_E \psi.$$

And for $\omega' := Ilrn(\omega, e, \psi)$ one simply has:

$$\omega' \models_I \sum_q \psi(q)\big|e \ll q\big\rangle \ \geq \omega \models_I \sum_q \psi(q)\big|e \ll q\big\rangle.$$

Proof Both likelihood inequalities follow from Theorem 18. The first one also involves (29.14). □

Internal learning along a channel also works incrementally, analogously to Proposition 19. We now use the terminology of actions, as already briefly mentioned after the proof of Proposition 19.

Proposition 22 *For a fixed channel $e : X \rightsquigarrow Y$, internal learning along e forms an action of the multiset monoid of data on states:*

$$\mathcal{D}(X) \times \mathcal{M}(\mathrm{Pred}(Y)) \xrightarrow{Ilrn(-,e,-)} \mathcal{D}(X)$$

The same works for point data $\mathcal{M}(Y)$ instead of predicates $\mathcal{M}(\mathrm{Pred}(Y))$.

Alternatively, one can say that internal learning forms an algebra for the writer monad $(-) \times \mathcal{M}(\mathrm{Pred}(Y))$ on the category of sets.

Proof As before we have $Ilrn(\omega, e, \mathbf{0}) = Ilrn(\omega, \mathbf{0}) = \omega$ and:

$$\begin{aligned}
Ilrn\big(\omega, e, \phi + \psi\big) &= Ilrn\big(\omega, \textstyle\sum_q (\phi + \psi)(q)\big|e \ll q\big\rangle\big) \\
&= Ilrn\big(\omega, \textstyle\sum_q (\phi(q) + \psi(q))\big|e \ll q\big\rangle\big) \\
&= Ilrn\big(\omega, (\textstyle\sum_q \phi(q)\big|e \ll q\big\rangle) + (\textstyle\sum_q \psi(q)\big|e \ll q\big\rangle)\big) \\
&= Ilrn\big(Ilrn\big(\omega, \textstyle\sum_q \phi(q)\big|e \ll q\big\rangle\big), \textstyle\sum_q \psi(q)\big|e \ll q\big\rangle\big) \\
&\qquad \text{by Proposition 19} \\
&= Ilrn\big(Ilrn(\omega, e, \phi), e, \psi\big).
\end{aligned}$$ □

External learning with point data can also be captured via the dagger of a channel, see Sect. 29.4.4.

Proposition 23 *When the data in the above situation consists of point data $\phi \in \mathcal{M}(Y)$, then external learning along channel e can be described via the dagger of e, as in:*

$$Elrn(\omega, e, \psi) = e^{\dagger}_{\omega} \gg Flrn(\phi).$$

Proof Since:

$$\begin{aligned}
Elrn(\omega, e, \phi) \overset{(24)}{=} \sum_x \frac{\phi(x)}{\|\phi\|} \cdot \omega|_{e \ll \mathbf{1}_x} &\overset{(16)}{=} \sum_x \frac{\phi(x)}{\|\phi\|} \cdot e^{\dagger}_{\omega}(x) \\
&= \sum_x Flrn(\phi)(x) \cdot e^{\dagger}_{\omega}(x) \\
&= e^{\dagger}_{\omega} \gg Flrn(\phi).
\end{aligned}$$ □

Notice that in this result we start with point data $\phi \in \mathcal{M}(Y)$. But the actual learning happens via transformed data $\sum_y \frac{\phi}{\|\phi\|}\big| e \ll \mathbf{1}_y \big\rangle$. The latter multiset no longer involves sharp point predicates, but fuzzy predicates $e \ll \mathbf{1}_y$. This is another reason why we have formulated data as multisets of *predicates* and not simply as multisets of *points*.

In the remainder of this section we illustrate learning along a channel in the classical situation where one wishes to learn the bias of an unknown coin form a given number of coin flips. In this situation one typically uses the flip channel describing a biased coin:

$$[0,1] \xrightarrow{\;flip\;} \{H, T\} \qquad \text{with} \qquad \mathit{flip}(r) = r|H\rangle + (1-r)|T\rangle.$$

In order to keep things simple we avoid continuous probability on the unit interval [0, 1]. Instead, we discretise it and use instead the 21-point domain:

$$D \coloneqq \{0, \tfrac{1}{20}, \tfrac{2}{20}, \ldots, \tfrac{19}{20}, 1\} \;\subseteq\; [0,1] \qquad \text{with} \qquad D \xrightarrow{\;flip\;} \{H, T\}.$$

The codomain $\{H, T\}$ of the flip channel carries two sharp point predicates $\mathbf{1}_H$ and $\mathbf{1}_T$ describing head and tail evidence. Predicate transformation turns them into two fuzzy predicates on D, namely:

$$\mathit{flip} \ll \mathbf{1}_H,\; \mathit{flip} \ll \mathbf{1}_T \;\in\; \mathrm{Pred}(D) \qquad \text{with} \qquad \begin{cases} (\mathit{flip} \ll \mathbf{1}_H)(r) = r \\ (\mathit{flip} \ll \mathbf{1}_T)(r) = 1 - r. \end{cases}$$

If we start from the uniform state $\upsilon = \sum_{0 \le i \le 20} \frac{1}{21} | \frac{i}{20} \rangle$ on D, then the probability of getting head is:

$$\upsilon \models \mathit{flip} \ll \mathbf{1}_H = \sum_{0 \le i \le 20} \tfrac{1}{21} \cdot \tfrac{i}{20} = \tfrac{1}{21} \cdot \tfrac{1}{20} \cdot \Bigg(\sum_{0 \le i \le 20} i \Bigg) = \tfrac{1}{21} \cdot \tfrac{1}{20} \cdot \tfrac{20 \cdot 21}{2} = \tfrac{1}{2}.$$

Similarly $\upsilon \models \mathit{flip} \ll \mathbf{1}_T = \frac{1}{2}$.

The questions we now ask ourselves are:

> What is the probability of seeing head after observing n heads and m tails, that is after learning from the multiset of predicates $n\big|\mathit{flip} \ll \mathbf{1}_H\big\rangle + m\big|\mathit{flip} \ll \mathbf{1}_T\big\rangle$. In which updated/learned state should the predicate $c \ll \mathbf{1}_H$ be evaluated to answer this question? Should we use external or internal learning along the flip channel?

Concretely, should we use the external variant or the internal version below, as newly learned state on D:

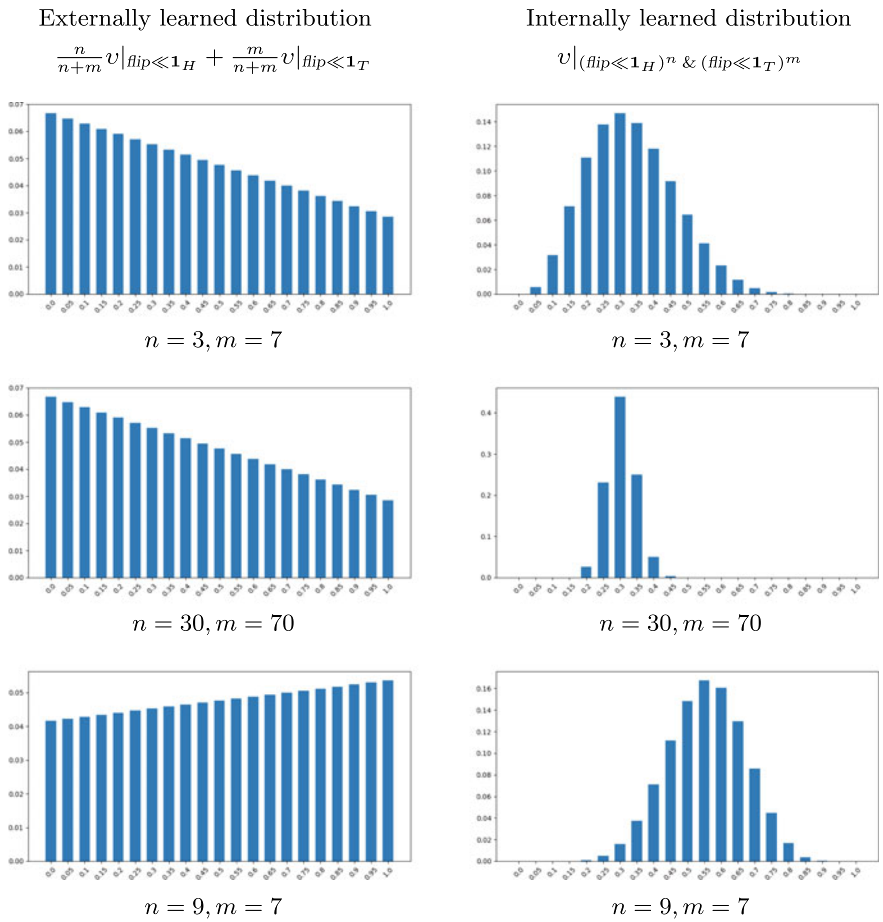


Fig. 29.2 Bar charts of learned distributions for coin bias, for different numbers n (of heads) and m (of tails)

$$\begin{aligned}
\mathit{Elrn}\big(\upsilon, \mathit{flip}, n|H\rangle + m|T\rangle\big) &= \tfrac{n}{n+m}\upsilon|_{\mathit{flip} \ll \mathbf{1}_H} + \tfrac{m}{n+m}\upsilon|_{\mathit{flip} \ll \mathbf{1}_T} \\
&= \mathit{flip}^{\dagger}_{\upsilon} \gg \big(\tfrac{n}{n+m}|H\rangle + \tfrac{m}{n+m}|T\rangle\big) \\
\mathit{Ilrn}\big(\upsilon, \mathit{flip}, n|H\rangle + m|T\rangle\big) &= \upsilon|_{(\mathit{flip} \ll \mathbf{1}_H)^n \,\&\, (\mathit{flip} \ll \mathbf{1}_T)^m} \\
&= \upsilon\, \underbrace{|_{\mathit{flip} \ll \mathbf{1}_H} \cdots |_{\mathit{flip} \ll \mathbf{1}_H}}_{n \text{ times}} \underbrace{|_{\mathit{flip} \ll \mathbf{1}_T} \cdots |_{\mathit{flip} \ll \mathbf{1}_T}}_{m \text{ times}} .
\end{aligned}$$

The order of the single updates in the last line does not matter.

Figure 29.2 contains bar charts describing these learned distributions, for various numbers n, m. We see, from these charts, and from the above formula, that the externally learned distribution for n, m is the same as for $K \cdot n$, $K \cdot m$. As we notices before, this is characteristic for external/frequentist learning. In contrast, internal learning is truly Bayesian: in the charts for the internally learned distribution one can recognise the a discretised version of the continuous beta distribution—to be precise, $\beta(n+1, m+1)$. Its variance becomes small with rising n, m, so that a higher precision is reached. It is well-known that these distributions have $\frac{n+1}{(n+1)+(m+1)}$ as mean. This is at the same time the validity $\mathit{Ilrn}(\upsilon, \mathit{flip}, n|H\rangle + m|T\rangle) \models c \ll \mathbf{1}_H$.

The interested reader may wish to check that the mean of the externally learned distribution $\frac{n}{n+m}\upsilon|_{\mathit{flip} \ll \mathbf{1}_H} + \frac{m}{n+m}\upsilon|_{\mathit{flip} \ll \mathbf{1}_T}$ is $\frac{41 \cdot n + 19 \cdot m}{60 \cdot (n+m)}$.

What to make of this? In every textbook treatment of coin bias learning one finds the internal approach. It presents an intuitively clear picture, with decreasing variance as the numbers n, m of heads and tails go up, and the 'expected' expected value $\frac{n+1}{(n+1)(m+1)}$. Is there an intrinsic reason why external learning is appropriate in Sect. 29.6.1 (and in the next section) and not here? See Sect. 29.9 for a perspective.

We like to conclude our description of coin bias learning with the conjugate prior property for internal learning. Our presentation here is different from traditional descriptions in two ways:

- it works in discrete, not continuous, probability, with a discretised version of the standard β distribution on $[0, 1]$;
- it formulates conjugate priorship in terms of a homomorphism of actions, building on Proposition 22.

Recall that we use $D = \{0, \frac{1}{20}, \dots, 1\} \subseteq [0, 1]$ as sample space, with uniform distribution υ on D. For $n, m \in \mathbb{N}$ we define on D the discretised beta distribution:

$$\beta_D(n, m) := \sum_{r \in D} \frac{r^n \cdot (1-r)^m}{\sum_{s \in D} s^n \cdot (1-s)^m} \big| r \big\rangle.$$

Proposition 24 *1. For $n, m \in \mathbb{N}$ the above distribution $\beta_D(n, m)$ satisfies:*

$$\begin{aligned} \beta_D(n, m) &= \upsilon\big|_{(\mathit{flip} \ll \mathbf{1}_H)^n \,\&\, (\mathit{flip} \ll \mathbf{1}_T)^m} \\ &= \mathit{Ilrn}(\upsilon, \mathit{flip}, n|H\rangle + m|T\rangle). \end{aligned}$$

2. For additional numbers $n', m' \in \mathbb{N}$ one has:

$$\beta_D(n, m)\big|_{(\mathit{flip} \ll \mathbf{1}_H)^{n'} \,\&\, (\mathit{flip} \ll \mathbf{1}_T)^{m'}} = \beta_D(n + n', m + m').$$

Proof 1. For $r \in D$ we have:

$$\begin{aligned} \upsilon\big|_{(\mathit{flip} \ll \mathbf{1}_H)^n \,\&\, (\mathit{flip} \ll \mathbf{1}_T)^m}(r) &= \frac{\upsilon(r) \cdot (\mathit{flip} \ll \mathbf{1}_H)^n(r) \cdot (\mathit{flip} \ll \mathbf{1}_T)^m(r)}{\upsilon \models (\mathit{flip} \ll \mathbf{1}_H)^n \,\&\, (\mathit{flip} \ll \mathbf{1}_T)^m} \\ &= \frac{1/21 \cdot r^n \cdot (1-r)^m}{\sum_{s \in D} 1/21 \cdot s^n \cdot (1-s)^m} \\ &= \beta_D(n, m)(r). \end{aligned}$$

2. We use this result and Lemma 10 (1) in:

$$\begin{aligned}
&\beta_D(n, m)\big|_{(\mathit{flip} \ll \mathbf{1}_H)^{n'} \& (\mathit{flip} \ll \mathbf{1}_T)^{m'}} \\
&= \upsilon\big|_{(\mathit{flip} \ll \mathbf{1}_H)^{n} \& (\mathit{flip} \ll \mathbf{1}_T)^{m}}\big|_{(\mathit{flip} \ll \mathbf{1}_H)^{n'} \& (\mathit{flip} \ll \mathbf{1}_T)^{m'}} \\
&= \upsilon\big|_{(\mathit{flip} \ll \mathbf{1}_H)^{n} \& (\mathit{flip} \ll \mathbf{1}_T)^{m} \& (\mathit{flip} \ll \mathbf{1}_H)^{n'} \& (\mathit{flip} \ll \mathbf{1}_T)^{m'}} \\
&= \upsilon\big|_{(\mathit{flip} \ll \mathbf{1}_H)^{n+n'} \& (\mathit{flip} \ll \mathbf{1}_T)^{m+m'}} \\
&= \beta_D(n + n', m + m').
\end{aligned}$$

□

The equation in the above second point shows that β_D is closed under updating with point predicates transformed along *flip*. It is the reason for calling β_D *conjugate prior* to *flip*. This is convenient because it means that we don't have to perform all the state updates explicitly; instead we can just adapt the inputs n, m of the channel β_D. These inputs are often called hyperparameters.

This conjugate priorship property is described at an abstract level in Jacobs (2020). Below we give a novel alternative description in terms of monoid actions—or equivalently, algebras of the writer monad. It again expresses that internal learning can be done incrementally.

Corollary 25 *The discretised beta channel* $\beta_D \colon \mathbb{N} \times \mathbb{N} \to \mathcal{D}(X)$ *forms a map of monoid actions in:*

$$\begin{tikzcd}
(\mathbb{N} \times \mathbb{N}) \times \mathcal{M}(\{H, T\}) \arrow[r, "\beta_D \times \mathrm{id}"] \arrow[d, "\mathit{add}"'] & \mathcal{D}(D) \times \mathcal{M}(\{H, T\}) \arrow[d, "\mathit{Ilrn}(-, \mathit{flip}, -)"] \\
\mathbb{N} \times \mathbb{N} \arrow[r, "\beta_D"] & \mathcal{D}(D)
\end{tikzcd}$$

The monoid action add *on the left is given by:*

$$\mathsf{add}\big(n,\ m,\ n'|\,H\,\rangle + m'|\,T\,\rangle\big) = (n + n', m + m').$$

The action on the right comes for internal learning along the channel flip: $D \rightsquigarrow \{H, T\}$, *see Proposition 22.* □

29.8 Expectation-Maximisation

Recall the situation (29.23) where we have a channel $X \rightsquigarrow Y$ and data on Y. The goal we have considered in the previous section is learning a state on X 'along the channel'. Within the so-called Expectation Maximisation (EM) algorithm, see Dempster et al. (1977) (and also McLachlan and Krishnan (1997)) this is called the E-step. There is an additional M-step which involves learning a better channel, so as to increase the (external) likelihood of the data. This section contains a fresh description of the EM-algorithm in which the two steps (E and M) are combined in a single learning step. This alternative approach uses a combination of the state on X and the channel $X \rightsquigarrow Y$ into a joint state on $X \times Y$, which is improved via external learning; subsequently, a new state on X and channel $X \rightsquigarrow Y$ are extracted. This re-description of the EM mechanism is applied to a standard EM example from the literature.

We first recall that a state $\omega \in \mathcal{D}(X)$ and a channel $e \colon X \rightsquigarrow Y$ can be combined into a joint state $\tau = \langle \mathrm{id}, e\rangle \gg \omega$, where $\langle \mathrm{id}, e\rangle = (\mathrm{id} \otimes e) \circ \Delta \colon X \rightsquigarrow X \times Y$. Then: $\tau(x, y) = \omega(x) \cdot e(x)(y)$. The marginals of τ are:

$$\pi_1 \gg \tau = \omega \qquad \text{and } \pi_2 \gg \tau = e \gg \omega.$$

When we extract a channel $X \rightsquigarrow Y$ from τ we rediscover the original channel $e \colon X \rightsquigarrow Y$, via the formula (29.4).

Now assume we have data $\psi \in \mathcal{M}(\mathrm{Pred}(Y))$ on Y. We can transform (weaken) ψ to data on $X \times Y$, written as:

$$\mathbf{1} \otimes \psi := \sum\nolimits_q \psi(q)\big|\pi_2 \ll q\big\rangle = \sum\nolimits_q \psi(q)\big|\mathbf{1} \otimes q\big\rangle.$$

Then $\tau \models_{\mathrm{E}} \mathbf{1} \otimes \psi \; = \; e \gg \omega \models_{\mathrm{E}} \psi$.

The next result gives our combined description of the E- and M-steps of the EM-algorithm via a single external learning step on a joint state. It used the conditioning $e|_q$ of a channel, which is defined pointwise as: $e|_q(x) = e(x)|_q$.

Theorem 26 *Let $\omega \in \mathcal{D}(X)$ be state with a channel $e \colon X \rightsquigarrow Y$, and with data $\psi \in \mathcal{M}(\mathrm{Pred}(Y))$. Write:*

$$\tau := \langle \mathrm{id}, e\rangle \gg \omega \;\in\; \mathcal{D}(X \times Y) \qquad \textit{and} \qquad \tau' := \mathit{Elrn}(\tau, \mathbf{1} \otimes \psi).$$

1. *The first marginal $\omega' = \pi_1 \gg \tau'$ is then the outcome of external learning from the data ψ along e:*

$$\omega' = \mathit{Elrn}(\omega, e, \psi) \overset{(24)}{=} \sum\nolimits_q \frac{\psi(q)}{\|\psi\|} \cdot \omega|_{e \ll q}.$$

2. *The channel $e' \colon X \rightsquigarrow Y$ extracted from $\tau' \in \mathcal{D}(X \times Y)$ by disintegration is:*

$$e'(x) = \sum\nolimits_q \frac{\psi(q)}{\|\psi\|} \cdot \frac{\omega|_{e \ll q}(x)}{\omega'(x)} \cdot e|_q(x).$$

   Then $e' \gg \omega' \models_E \psi \;\geq\; e \gg \omega \models_E \psi$.
3. *In the special case where the data is given by points, so $\psi \in \mathcal{M}(Y)$, we know from Proposition 23 that the newly learned state $\omega' = \pi_1 \gg \tau'$ can be expressed via a dagger, as: $\omega' = e^{\dagger}_{\omega} \gg \mathit{Flrn}(\psi)$; the newly learned channel e' is then a double dagger:*

$$e' = \big(e^{\dagger}_{\omega}\big)^{\dagger}_{\mathit{Flrn}(\psi)} \,:\, X \rightsquigarrow Y.$$

It satisfies $e' \gg \omega' = \mathit{Flrn}(\psi)$ *by (29.16), so that a second channel-learning step with the same data has no effect:* $\big(e'^{\dagger}_{\omega'}\big)^{\dagger}_{\mathit{Flrn}(\psi)} = e'$.

Proof 1. We get as first marginal of the newly learned joint state τ',

$$\begin{aligned}
\big(\pi_1 \gg \tau'\big)(x) &= \sum\nolimits_y \tau'(x,y) \\
&\overset{(24)}{=} \sum\nolimits_y \sum\nolimits_q \frac{\psi(q)}{\|\psi\|} \cdot \tau|_{\mathbf{1}\otimes q}(x,y) \\
&= \sum\nolimits_q \frac{\psi(q)}{\|\psi\|} \cdot \sum\nolimits_y \frac{\tau(x,y)\cdot q(y)}{\tau \models \mathbf{1}\otimes q} \\
&= \sum\nolimits_q \frac{\psi(q)}{\|\psi\|} \cdot \sum\nolimits_y \frac{\omega(x)\cdot e(x)(y)\cdot q(y)}{e \gg \omega \models q} \\
&= \sum\nolimits_q \frac{\psi(q)}{\|\psi\|} \cdot \frac{\omega(x)\cdot (e \ll q)(x)}{\omega \models e \ll q} \\
&= \sum\nolimits_q \frac{\psi(q)}{\|\psi\|} \cdot \omega|_{e \ll q}(x) \\
&\overset{(24)}{=} \mathit{Elrn}(\omega, e, \psi)(x).
\end{aligned}$$

2. The channel $e' \colon X \rightsquigarrow Y$ extracted from $\tau' \in \mathcal{D}(X \times Y)$ is:

$$\begin{aligned}
e'(x)(y) &\overset{(4)}{=} \frac{\tau'(x,y)}{(\pi_1 \gg \tau')(x)} \\
&= \sum\nolimits_q \frac{\psi(q)}{\|\psi\|} \cdot \frac{\tau|_{\mathbf{1}\otimes q}(x,y)}{\omega'(x)} \\
&= \sum\nolimits_q \frac{\psi(q)}{\|\psi\|} \cdot \frac{1}{\omega'(x)} \cdot \frac{\omega(x)\cdot e(x)(y)\cdot q(y)}{\omega \models e \ll q} \\
&= \sum\nolimits_q \frac{\psi(q)}{\|\psi\|} \cdot \frac{1}{\omega'(x)} \cdot \frac{\omega(x)\cdot (e \ll q)(x)}{\omega \models e \ll q} \cdot \frac{e(x)(y)\cdot q(y)}{e(x) \models q} \\
&= \sum\nolimits_q \frac{\psi(q)}{\|\psi\|} \cdot \frac{1}{\omega'(x)} \cdot \omega|_{e \ll q}(x) \cdot e|_q(x)(y).
\end{aligned}$$

3. Since $e|_{\mathbf{1}_z}(x) = e(x)|_{\mathbf{1}_z} = 1|z\rangle$ we get:

$$\begin{aligned}
e'(x)(y) &= \sum\nolimits_z \frac{\psi(z)}{\|\psi\|} \cdot \frac{\omega|_{e \ll \mathbf{1}_z}(x)}{\omega'(x)} \cdot e|_{\mathbf{1}_z}(x)(y) && \text{by point (2)} \\
&= \sum\nolimits_z \frac{\mathit{Flrn}(\psi)(z)\cdot e^{\dagger}_{\omega}(z)(x)}{(e^{\dagger}_{\omega} \gg \mathit{Flrn}(\psi))(x)} \cdot 1\big|y\big\rangle(z) && \text{by (15) and Proposition (23)} \\
&= \frac{\mathit{Flrn}(\psi)(y)\cdot e^{\dagger}_{\omega}(y)(x)}{(e^{\dagger}_{\omega} \gg \mathit{Flrn}(\psi))(x)} \\
&= \big(e^{\dagger}_{\omega}\big)^{\dagger}_{\mathit{Flrn}(\psi)}(x)(y) && \text{by (15).}
\end{aligned}$$

□

The joint-state learning approach, followed by marginalisation and extraction, of Theorem 26 can in principle also be used for internal learning. However, then we don't get a correspondence with learning along a channel—like in Theorem 26 (1). Hence the internal approach fails at this point.

29.8.1 Candy Examples

The textbook (Russell & Norvig, 2003) contains a chapter titled *Statistical learning methods*, with candy examples in two forms.

First, there is a situation with five different bags, numbered 1, …, 5, each containing its own mixture of cherry (C) and lime (L) candies. This situation can be described via a candy channel:

$$B \overset{c}{\multimap} \{C, L\} \quad \text{where} \quad B = \{1, 2, 3, 4, 5\} \quad \text{and} \quad \begin{cases} c(1) = 1|C\rangle \\ c(2) = \frac{3}{4}|C\rangle + \frac{1}{4}|L\rangle \\ c(3) = \frac{1}{2}|C\rangle + \frac{1}{2}|L\rangle \\ c(4) = \frac{1}{4}|C\rangle + \frac{3}{4}|L\rangle \\ c(5) = 1|L\rangle. \end{cases}$$

The initial bag distribution is $\omega = \frac{1}{10}|1\rangle + \frac{1}{5}|2\rangle + \frac{2}{5}|3\rangle + \frac{1}{5}|4\rangle + \frac{1}{10}|5\rangle$.

In the situation described in Russell and Norvig (2003, Sect. 20.1) the space of bags B is regarded as hidden (not directly observable), in a scenario where a new bag $i \in B$ is given and candies are drawn from it. It turns out that 10 consecutive draws yield a lime candy.[2] Transforming the lime point predicate along channel c yields the fuzzy predicate $c \ll \mathbf{1}_L : B \to [0, 1]$ given by:

$$c \ll \mathbf{1}_L = \bigovee_i c(i)(L) \cdot \mathbf{1}_i = \frac{1}{4} \cdot \mathbf{1}_2 \ovee \frac{1}{2} \cdot \mathbf{1}_3 \ovee \frac{3}{4} \cdot \mathbf{1}_4 \ovee 1 \cdot \mathbf{1}_5.$$

The question is what we learn about the bag distribution after observing this predicate 10 consecutive times? Fig. 20.1 in Russell and Norvig (2003) gives a plot of (Bayesian) internal learning along the channel c; it is reconstructed in Fig. 29.3 via internal learning.

This leads for $n = 1, 2, 3$ to distributions of bags:

$$\begin{aligned} Ilrn(\omega, c, 1|L\rangle) &= \tfrac{1}{10}|2\rangle + \tfrac{2}{5}|3\rangle + \tfrac{3}{10}|4\rangle + \tfrac{1}{5}|5\rangle \\ Ilrn(\omega, c, 2|L\rangle) &= \tfrac{1}{26}|2\rangle + \tfrac{4}{13}|3\rangle + \tfrac{9}{26}|4\rangle + \tfrac{4}{13}|5\rangle \\ &\approx 0.0385|2\rangle + 0.308|3\rangle + 0.346|4\rangle + 0.308|5\rangle \\ Ilrn(\omega, c, 3|L\rangle) &= \tfrac{1}{76}|2\rangle + \tfrac{4}{19}|3\rangle + \tfrac{27}{76}|4\rangle + \tfrac{8}{19}|5\rangle \\ &\approx 0.0132|2\rangle + 0.211|3\rangle + 0.355|4\rangle + 0.421|5\rangle. \end{aligned}$$

[2] The bags are described in Russell and Norvig (2003) as very large, so that withdrawing one candy does not change the distribution of candies in the bag. This amounts to replacing the drawn candy.

$$\begin{aligned} &Ilrn(\omega, c, n|L\rangle) \\ &\quad = \omega\big|_{(c \ll \mathbf{1}_L)^n} \end{aligned}$$

for $n = 0, 1, \ldots, 10$.

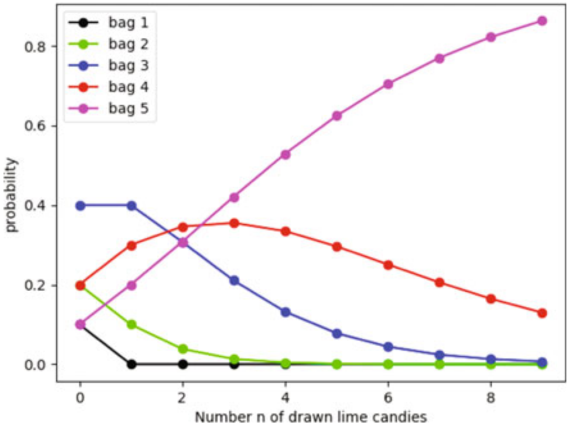


Fig. 29.3 Bag distributions, aligned vertically, after multiple lime candy draws

We see, here and in Fig. 29.3, that bag 5 quickly becomes more likely—as expected because it contains most lime candies—and that bag 1 is impossible after drawing the first lime.

As an aside, applying external learning in this candy situation gives, for $n \geq 1$,

$$Elrn(\omega, c, n|L\rangle) = Elrn(\omega, c, 1|L\rangle) = \omega|_{c \ll \mathbf{1}_L}.$$

The outcome is then the same for each number $n > 0$ of drawn lime candies: multiple lime-draws give no further information, as in (29.20). This shows that external learning is not appropriate here, in the first candy example.

The second candy example in Russell and Norvig (2003, Sect. 20.3) is used as an illustration of the Expectation-Maximisation (EM) algorithm. Therefor it now uses *external* learning. It involves the Bayesian network described below, with (two) bags of candies, named 0 and 1, each described by three features, namely their flavour (cherry or lime), their wrapper (red or green), and whether or not they have holes. The interpretation of the Bayesian network in terms of channels (conditional probability tables) $f: \{0, 1\} \rightsquigarrow \{C, L\}$, $w: \{0, 1\} \rightsquigarrow \{R, G\}$, $h: \{0, 1\} \rightsquigarrow \{H, H^\perp\}$, and initial state $\rho \in \mathcal{D}(\{0, 1\})$ is on the left.

$$\boxed{candy} \text{ with } \left\{ \begin{aligned} f(0) &= \tfrac{6}{10}|C\rangle + \tfrac{4}{10}|L\rangle \\ f(1) &= \tfrac{4}{10}|C\rangle + \tfrac{6}{10}|L\rangle \\ w(0) &= \tfrac{6}{10}|R\rangle + \tfrac{4}{10}|G\rangle \\ w(1) &= \tfrac{4}{10}|R\rangle + \tfrac{6}{10}|G\rangle \\ h(0) &= \tfrac{6}{10}|H\rangle + \tfrac{4}{10}|H^\perp\rangle \\ h(1) &= \tfrac{4}{10}|H\rangle + \tfrac{6}{10}|H^\perp\rangle \\ \rho &= \tfrac{6}{10}|0\rangle + \tfrac{4}{10}|1\rangle. \end{aligned} \right.$$

The three channels f, w, h are combined into a single (three-)tuple channel $\langle f, w, h\rangle\colon \{0,1\} \multimap \{C, L\} \times \{R, G\} \times \{H, H^{\perp}\}$. At 0 it is:

$$\begin{aligned}\langle f, w, h\rangle(0) \;&=\; f(0) \otimes w(0) \otimes h(0) \\ &= \frac{216}{1000}|C, R, H\rangle + \frac{144}{1000}|C, R, H^{\perp}\rangle + \frac{144}{1000}|C, G, H\rangle + \frac{96}{1000}|C, G, H^{\perp}\rangle \\ &\quad + \frac{144}{1000}|L, R, H\rangle + \frac{96}{1000}|L, R, H^{\perp}\rangle + \frac{96}{1000}|L, G, H\rangle + \frac{64}{1000}|L, G, H^{\perp}\rangle.\end{aligned}$$

The point-data $\psi \in \mathcal{M}\big(\{C, L\} \times \{R, G\} \times \{H, H^{\perp}\}\big)$ is given by the multiset:

$$\begin{aligned}\psi = {} & 273|C, R, H\rangle + 93|C, R, H^{\perp}\rangle + 104|C, G, H\rangle + 90|C, G, H^{\perp}\rangle \\ & + 79|L, R, H\rangle + 100|L, R, H^{\perp}\rangle + 94|L, G, H\rangle + 167|L, G, H^{\perp}\rangle,\end{aligned}$$

containing $\|\psi\| = 1000$ items. We are now set to learn a better state and channel, via EM as described in Russell and Norvig (2003). Here we use the description of external learning with a dagger channel, from Proposition 23. The newly learned distribution on $\{0, 1\}$ is:

$$\begin{aligned}\mathit{Elrn}(\rho, \langle f, w, h\rangle, \psi) &= \langle f, w, h\rangle^{\dagger}_{\rho} \gg \mathit{Flrn}(\psi) \\ &= \tfrac{273}{1000} \cdot \rho|_{\langle f,w,h\rangle \ll \mathbf{1}_{(C,R,H)}} + \cdots + \tfrac{167}{1000} \cdot \rho|_{\langle f,w,h\rangle \ll \mathbf{1}_{(L,G,H^{\perp})}} \\ &= \tfrac{30891}{50440}|0\rangle + \tfrac{19549}{50440}|1\rangle \\ &\approx 0.6124|0\rangle + 0.3876|1\rangle.\end{aligned}$$

This probability 0.6124 is exactly as computed in Russell and Norvig (2003, Sect. 20.3). The newly learned channel is obtained like in Theorem 26 (3) as a 'double dagger', which we abbreviate as:

$$dd := \big(\langle f, w, h\rangle^{\dagger}_{\rho}\big)^{\dagger}_{\mathit{Flrn}(\psi)} \colon \{0, 1\} \multimap \{C, L\} \times \{R, G\} \times \{H, H^{\perp}\}.$$

We then obtain the individually learned channels f', w', h' via marginalisation of the channels:

$$\begin{aligned}f' &:= \pi_1 \circ dd \;:\; \{0,1\} \multimap \{C, L\} \\ w' &:= \pi_2 \circ dd \;:\; \{0,1\} \multimap \{R, G\} \\ h' &:= \pi_3 \circ dd \;:\; \{0,1\} \multimap \{H, H^{\perp}\}\end{aligned}$$

This yields precisely the values reported in Russell and Norvig (2003):

$$\begin{aligned}f'(0) &= 0.6684|C\rangle + 0.3316|L\rangle & f'(1) &= 0.3887|C\rangle + 0.6113|L\rangle \\ w'(0) &= 0.6483|R\rangle + 0.3517|G\rangle & w'(1) &= 0.3817|R\rangle + 0.6183|G\rangle \\ h'(0) &= 0.6558|H\rangle + 0.3442|H^{\perp}\rangle & h'(1) &= 0.3827|H\rangle + 0.6173|H^{\perp}\rangle.\end{aligned}$$

In two adjacent sections on learning, Sects. 20.1 and 20.3, in the same textbook (Russell & Norvig, 2003), two different learning methods are used, for similar examples (bags of candies). The book makes neither that difference explicit, nor what is actually improved (like increase of some form of likelihood) by these different forms of learning.

29.9 Discussion About Likelihood and Learning

In (likelihood-based) probabilistic learning one seeks a distribution (state) that better fits given data. In this paper we have argued that such data form multisets, of points, or, more generally, of predicates. These data give rise to a likelihood function on states, assigning a numerical value in [0, 1], to a state. Learning may happen in multiple steps, where each step increases the likelihood of the data, by changing a given state ω to ω' which fits better, in the sense that it gives higher likelihood to the data.

This paper has described two likelihood functions, namely the 'external' version $\models_{\mathrm{E}}$ and the 'internal' version $\models_{\mathrm{I}}$, with two associated learning methods. In Sect. 29.5 it is shown that both forms of likelihood arise naturally from repeated transitions on states, with or without updates. Both forms of learning are used in the literature, but implicitly: the difference is not made explicit—as far as we have seen.

This final section tries to develop a perspective on this matter, with as underlying question: when, under which circumstances, should we use external likelihood and external learning and when internal likelihood and internal learning? No mathematically precise answer is formulated. Instead, an intuition is developed, see esp. points (6) and (7) below, in terms of a combination of external and internal, using batches of data that can be handled separately externally, and jointly internally.

It is unlikely that this admittedly vague answer will settle the matter. Therefor the points below are best seen as a first step in further research and debate.

1. In our approach we have consistently used fuzzy predicates, taking values in [0, 1]. This is unusual in probability theory (with exceptions e.g. in Chan and Darwiche (2005); Darwiche (2009); Mrad et al. (2015); Valtorta et al. (2002)), where people standardly use sharp predicates (with values in $\{0, 1\}$), also called events. Conjunction of events is simply intersection: $\mathbf{1}_U \;\&\; \mathbf{1}_V = \mathbf{1}_{U \cap V}$ and taking powers of events has no effect: $(\mathbf{1}_U)^n = \mathbf{1}_U \;\&\; \cdots \;\&\; \mathbf{1}_U = \mathbf{1}_U$. Thus, the internal likelihood formulation—$\omega \models_{\mathrm{I}} \sum_i n_i | p_i \rangle \;=\; \omega \models \&_i\; p_i^{n_i}$—only really makes sense in a context with fuzzy predicates p_i. This might explain why internal likelihood has not been made explicit before, and then also why the distinction between external and internal likelihood is absent in the literature.

2. Let's make things concrete and recall the coin bias learning situation in Sect. 29.7, with uniform state υ on the discretised unit interval D. The probability of seeing head (or tail) is:

$$\upsilon \models \mathit{flip} \ll \mathbf{1}_H \;= \upsilon \models \mathit{flip} \ll \mathbf{1}_T \;= \tfrac{1}{2}.$$

Now suppose we have data saying: both head and tail. How should this be interpreted? What is the likelihood of these data? We formalise it as a multiset of predicates $\psi = 1|\,\mathit{flip} \ll \mathbf{1}_H\,\rangle + 1|\,\mathit{flip} \ll \mathbf{1}_T\,\rangle$. Then:

$$\begin{aligned}
\upsilon \models_{\mathrm{E}} \psi &= (\upsilon \models \mathit{flip} \ll \mathbf{1}_H) \cdot (\upsilon \models \mathit{flip} \ll \mathbf{1}_T) \;=\; \tfrac{1}{2} \cdot \tfrac{1}{2} \;=\; 0.25 \\
\upsilon \models_{\mathrm{I}} \psi &= \upsilon \models (\mathit{flip} \ll \mathbf{1}_H) \,\&\, (\mathit{flip} \ll \mathbf{1}_T) \\
&= \sum_{0 \le i \le 20} \tfrac{1}{21} \cdot \tfrac{i}{20} \cdot (1 - \tfrac{i}{20}) \;=\; \tfrac{1}{2} - \tfrac{41}{120} \;=\; \tfrac{19}{120} \;\approx\; 0.16.
\end{aligned}$$

What is now the 'right' likelihood of seeing both head and tail: 25 or 16%? This question challenges our basic probabilistic intuitions. The internal perspective offers a reasonable interpretation by Bayes' rule: both head and tail means, first seeing head, and updating, and then seeing tail (or the other way around):

$$\begin{aligned}
&\big(\upsilon \models \mathit{flip} \ll \mathbf{1}_H\big) \cdot \big(\upsilon|_{\mathit{flip} \ll \mathbf{1}_H} \models \mathit{flip} \ll \mathbf{1}_T\big) \\
&= \upsilon \models (\mathit{flip} \ll \mathbf{1}_H) \,\&\, (\mathit{flip} \ll \mathbf{1}_T) \\
&= \big(\upsilon \models \mathit{flip} \ll \mathbf{1}_T\big) \cdot \big(\upsilon|_{\mathit{flip} \ll \mathbf{1}_T} \models \mathit{flip} \ll \mathbf{1}_H\big).
\end{aligned}$$

3. Maybe the fuzzy predicates in the previous example over-complicate the situation. So let's move to sharp predicates: we take a fair dice $\omega = \frac{1}{6}|\,1\,\rangle + \frac{1}{6}|\,2\,\rangle + \frac{1}{6}|\,3\,\rangle + \frac{1}{6}|\,4\,\rangle + \frac{1}{6}|\,5\,\rangle + \frac{1}{6}|\,6\,\rangle$, with events $E = \{2, 4, 6\}$ for 'even' and $H = \{4, 5, 6\}$ for 'high', corresponding to sharp predicates $\mathbf{1}_E$ and $\mathbf{1}_H$. Clearly, $\omega \models \mathbf{1}_E = \omega \models \mathbf{1}_H = \frac{1}{2}$. We take as data $\phi = 1|\,\mathbf{1}_E\,\rangle + 2|\,\mathbf{1}_H\,\rangle$, representing that we observe 'even' once and 'high' twice. What is ϕ's likelihood in state ω?

$$\begin{aligned}
\omega \models_{\mathrm{E}} \phi &= (\omega \models \mathbf{1}_E) \cdot (\omega \models \mathbf{1}_H) \cdot (\omega \models \mathbf{1}_H) \;=\; \tfrac{1}{2} \cdot \tfrac{1}{2} \cdot \tfrac{1}{2} \;=\; \tfrac{1}{8} \\
\omega \models_{\mathrm{I}} \phi &= \omega \models \mathbf{1}_E \,\&\, \mathbf{1}_H \,\&\, \mathbf{1}_H \;=\; \omega \models \mathbf{1}_{E \cap H} \;=\; \tfrac{1}{3}.
\end{aligned}$$

What is now the right likelihood? It depends … But on what?

4. We can try to explain the difference between $\models_{\mathrm{E}}$ and $\models_{\mathrm{I}}$ in terms of different observers, who operate *separately* or *jointly*, like in Example 14. Suppose we have a state ω and data in the form of a multiset of predicates $\psi = \sum_i n_i|\,p_i\,\rangle$, with $n = \|\psi\| = \sum_i n_i$. We assume that there are n individual observers, and each predicate occurring in the multiset ψ is assigned to one observer. Hence n_1 of them obtain p_1, n_2 observers have p_2 etc.

 - In the external likelihood perspective each observer, say with predicate p, gets an instance of the state ω, via some 'external' copy mechanism. These observers now ask: what is the probability that we are all right *separately*? Each of them determines the validity $\omega \models p$ of their own predicate. Then, all observers get together and multiply their validities, giving the external likelihood $\omega \models_{\mathrm{E}} \psi$.
 - In the internal likelihood perspective each observer is looking at the same state ω. These observers ask: what is the probability that we are *jointly* right? This joint view is obtained by putting all their predicates together in a single

conjunction $\&_i\, p_i^{n_i}$. The validity of this conjunction predicate gives internal likelihood $\omega \models_{\mathrm{I}} \psi$.

5. In the approach to quantum logic described in Jacobs (2015) the operation & on predicates is called *sequential* conjunction, whereas $\otimes$ is *parallel* conjunction. This phrase 'sequential' makes sense there, since & is not commutative in a quantum setting. One could use the term 'sequential' also in the current setting of classical (non-quantum) probability, for instance by understanding 'joint' in the previous point in a sequential manner—one after the other—but then in such a way that the order does not matter.
6. One can combine external and internal likelihood by moving one more step up the abstraction ladder and introduce multisets of multisets of predicates $\Psi \in \mathcal{M}(\mathcal{M}(\mathrm{Pred}(X)))$ as data. This may be useful when data in $\mathcal{M}(\mathrm{Pred}(X))$, as used before, comes in batches: the multisets $\phi \in \mathrm{Pred}(\mathcal{M}(X))$ occurring as elements of Ψ. For a state $\omega \in \mathcal{D}(X)$ one can now define external-internal likelihood $\models_{\mathrm{EI}}$ as:

$$\omega \models_{\mathrm{EI}} \Psi \;\coloneqq \sum\nolimits_{\phi} \big(\omega \models_{\mathrm{I}} \phi\big)^{\Psi(\phi)} = \sum\nolimits_{\phi} \big(\omega \models \&_p\; p^{\phi(p)}\big)^{\Psi(\phi)}.$$

This uses external likelihood on the outside and internal likelihood on the inside. A different order does not make sense. One can then develop an associated form of external-internal learning. This actually occurs in the literature, namely in the leading example used in Do and Batzoglou (2008) to describe Expectation-Maximisation. Elaborating the details goes beyond the current setting, but we can briefly sketch the essentials (of a single E-step).
Two coins are given in Do and Batzoglou (2008) via a channel $c\colon \{1,2\} \rightsquigarrow \{H,T\}$ with different biases: $c(1) = \frac{3}{5}|H\rangle + \frac{2}{5}|T\rangle$ and $c(2) = \frac{1}{2}|H\rangle + \frac{1}{2}|T\rangle$. There are 5 batches of point data with 10 coin flips each: $\phi_1 = 5|H\rangle + 5|T\rangle$, $\phi_2 = 9|H\rangle + 1|T\rangle$, $\phi_3 = 8|H\rangle + 2|T\rangle$, $\phi_4 = 4|H\rangle + 6|T\rangle$, $\phi_5 = 7|H\rangle + 3|T\rangle$. Starting from the uniform state $\upsilon \in \mathcal{D}(\{1,2\})$ one learns a better fit via 'external-internal' learning: take externally the weighted average (convex sum) over all batches, of the internally learned states per batch, giving:

$$\sum\nolimits_i \tfrac{1}{5} \cdot \mathit{Ilrn}(\upsilon, c, \phi_i) = 0.597|1\rangle + 0.403|2\rangle.$$

Thus, these data indicate that the first coin is a bit more likely.
7. The previous point, with combined likelihood $\models_{\mathrm{EI}}$, can offer a perspective on the question when to use external or internal likelihood. When we have data $\psi \in \mathcal{M}(\mathrm{Pred}(X))$ we can view it in two ways:

   - as multiset of batches of *separate* single data items $\psi_E \coloneqq \sum_p \psi(p)\big|1|p\rangle\big\rangle$ in $\mathcal{M}(\mathcal{M}(\mathrm{Pred}(X)))$, whose external-internal likelihood equals external likelihood: $\omega \models_{\mathrm{EI}} \psi_E \;=\; \omega \models_{\mathrm{E}} \psi$;
   - as a single batch of *joint* data items $\psi_I \coloneqq 1|\psi\rangle$ in $\mathcal{M}(\mathcal{M}(\mathrm{Pred}(X)))$, with external-internal likelihood equal to internal likelihood: $\omega \models_{\mathrm{EI}} \psi_I \;=\; \omega \models_{\mathrm{I}} \psi$.

This distinction is consistent with the one we made earlier in point (4) in terms of observers.
Concretely, in the earlier setting of coin bias learning, suppose we have two separate batches of data, in the form of two multisets of predicates:

$$\phi_1 = 1|\mathit{flip} \ll \mathbf{1}_H\rangle + 1|\mathit{flip} \ll \mathbf{1}_T\rangle \quad \phi_2 = 2|\mathit{flip} \ll \mathbf{1}_H\rangle + 1|\mathit{flip} \ll \mathbf{1}_T\rangle.$$

Then $\Psi = 1|\phi_1\rangle + 1|\phi_2\rangle$ has likelihood (in the uniform state υ):

$$\upsilon \models_{\mathrm{EI}} \Psi = (\upsilon \models_{\mathrm{I}} \phi_1) \cdot (\upsilon \models_{\mathrm{I}} \phi_2) = \tfrac{19}{120} \cdot \tfrac{19}{240} = \tfrac{361}{28800}.$$

8. Along the way we have noticed several times—e.g. in (29.8) and (29.20)—that frequentist and external learning satisfy the more-is-the-same property: repeating the same data as input has no effect. In contrast, internal learning behaves like an action—see Lemma 10 (1) and Propositions 19 and 22—where repeated (and multiple) data inputs do have effect and are processed sequentially via multiple Bayesian updates. This is a significant difference between 'internal' and 'external'.

9. Finally, we like to point at an analogy with the distinction between Pearl's updating and Jeffrey's adaptation along a channel, as described in Jacobs (2019). The description of the externally learned state $e^{\dagger}_{\omega} \gg \mathit{Flrn}(\phi)$ via a dagger channel in Proposition 23 is an instance of Jeffrey's adaptation rule. Internally learning along a channel is an instance of Pearl's updating rule. In follow-up work it will be shown that Jeffrey's rule is about decreasing divergence and Pearl's rule is about increasing validity. Proposition 20 shows that external learning can also be expressed in terms of decreasing divergence. Hence it seems that external learning and Jeffrey's rule belong to the same "decreasing divergence" school, whereas internal learning and Pearl's rule are in the "increasing likelihood" school.

Appendix

We provide the missing proofs of Theorem 18 (1) and of Proposition 16. The proof of the latter proposition is standard, but is included because it forms a proper preparation for the proofs of the two theorems—which are new results. All proofs rely on some basic real analysis for finding the maximum of functions with constraints on their inputs. This is done via the Lagrange multiplier method, see e.g. Bishop (2006, Sect. 2.2). This will be illustrated first. Subsequently we make use of a 'sum-increase' lemma to prove the other results.

Proof (*of Proposition* 16) Let $\phi \in \mathcal{M}(X)$ be a fixed non-empty multiset. We need to prove that the external likelihood function $(-) \models_{\mathrm{E}} \phi \colon \mathcal{D}(X) \to [0, 1]$ takes its maximum at $\mathit{Flrn}(\phi)$. We will thus seek the maximum of the function $\omega \mapsto \omega \models_{\mathrm{E}} \phi$ by taking the derivative with respect to $\omega \in \mathcal{D}(X)$. We will work with the 'log-validity', that is, with the function $\omega \mapsto \ln(\omega \models_{\mathrm{E}} \phi)$, where ln is the monotone (natural) logarithm function. It reduces the product $\prod$ of powers in the definition of $\models_{\mathrm{E}}$ to a sum $\sum$ of multiplications.

Assume that the support of $\phi = \sum_i n_i|x_i\rangle$ is $\{x_1, \dots, x_n\} \subseteq X$. We look at distributions $\omega \in \mathcal{D}(\{x_1, \dots, x_n\})$; they may be identified with numbers $\vec{v} = v_1, \dots, v_n \in \big(\mathbb{R}_{\geq 0}\big)^n$ with $\sum_i v_i = 1$. We thus seek the maximum of the log-validity function:

$$k(\vec{v}) := \ln\Big(\sum_i v_i|x_i\rangle \models_{\mathrm{E}} \sum_i n_i|x_i\rangle\Big) = \ln\Big(\prod_i v_i^{n_i}\Big) = \sum_i n_i \cdot \ln(v_i).$$

Since we have a constraint $(\sum_i v_i) - 1 = 0$ on the inputs, we can use the Lagrange multiplier method for finding the maximum. We thus take another parameter λ in a new function:

$$K(\vec{v}, \lambda) := k(\vec{v}) - \lambda \cdot \big((\sum_i v_i) - 1\big) = \big(\sum_i n_i \ln(v_i)\big) - \lambda \cdot \big((\sum_i v_i) - 1\big).$$

The partial derivatives of K are:

$$\frac{\partial K}{\partial v_i}(\vec{v}, \lambda) = \frac{n_i}{v_i} - \lambda \qquad \frac{\partial K}{\partial \lambda}(\vec{v}, \lambda) = 1 - \sum_i v_i.$$

Setting all of these to 0 and solving gives the required maximum. First, we have:

$$1 = \sum_i v_i = \sum_i \frac{n_i}{\lambda} = \frac{\sum_i n_i}{\lambda}.$$

Hence $\lambda = \sum_i n_i$ and thus:

$$v_i = \frac{n_i}{\lambda} = \frac{n_i}{\sum_i n_i} \overset{(7)}{=} \mathit{Flrn}(\phi)(x_i). \qquad \square$$

We now come to an auxiliary result which we shall call the sum-increase lemma. It is a special (discrete) case of a more general result (Baum et al., 1970, Theorem 2.1). It describes how to find increases for sum expressions in general.

Lemma 27 *Let X, Y be finite sets, and let $F\colon X \times Y \to \mathbb{R}_{\geq 0}$ be a given function. For each $x \in X$, write $F_1(x) := \sum_{y \in Y} F(x, y)$ for the sum that we wish to increase. Assume that there is an $x' \in X$ with:*

$$x' = \underset{z}{\operatorname{argmax}}\ G(x, z) \qquad \textit{where} \qquad G(x, z) := \sum_{y \in Y} F(x, y) \cdot \ln\big(F(z, y)\big).$$

Then $F_1(x') \geq F_1(x)$.

The proof uses Jensen's inequality: for $a_1, \dots, a_n \in \mathbb{R}_{>0}$ and $r_1, \dots, r_n \in [0, 1]$ with $\sum_i r_i = 1$ one has $\ln(\sum_i r_i a_i) \geq \sum_i r_i \ln(a_i)$. This gives a strict increase, except in 'corner' cases. The same holds for the above sum-increase lemma. The actual maximum x' in that lemma can in many situation be determined analytically—using the Lagrange multiplier method—but it need not be unique.

Proof Let x' be the element where $G(x,-)\colon Y \to \mathbb{R}_{\geq 0}$ takes its maximum. This x' satisfies $F_1(x') \geq F_1(x)$, since:

$$\begin{aligned}
\ln\left(\frac{F_1(x')}{F_1(x)}\right) &= \ln\left(\sum\nolimits_y \frac{F(x',y)}{F_1(x)}\right) \\
&= \ln\left(\sum\nolimits_y \frac{F(x,y)}{F_1(x)} \cdot \frac{F(x',y)}{F(x,y)}\right) \\
&\geq \sum\nolimits_y \frac{F(x,y)}{F_1(x)} \cdot \ln\left(\frac{F(x',y)}{F(x,y)}\right) \quad \text{by Jensen's inequality} \\
&= \frac{1}{F_1(x)} \cdot \sum\nolimits_y F(x,y) \cdot \Big(\ln\big(F(x',y)\big) - \ln\big(F(x,y)\big)\Big) \\
&= \frac{1}{F_1(x)} \cdot \Big(G(x,x') - G(x,x)\Big) \geq 0.
\end{aligned}$$

□

Proof (*Theorem* 18 (1)) Let $\omega \in \mathcal{D}(X)$ be state on a finite set X and let $p_1, \ldots, p_n$ be predicates on X, all with non-zero validity $\omega \models p_i$. We claim that the state $\omega' = \sum_i \frac{1}{n} \cdot \omega|_{p_i}$ then satisfies:

$$\prod\nolimits_i (\omega' \models p_i) \geq \prod\nolimits_i (\omega \models p_i). \tag{29.25}$$

The inequality in Theorem 18 (1) is a direct consequence of (29.25). We shall prove (29.25) for $n = 2$. The generalisation to arbitrary n should then be obvious, but involves much more book-keeping of additional variables.

We use Lemma 27 with function $F\colon \mathcal{D}(X) \times X \times X \to \mathbb{R}_{\geq 0}$ given by:

$$F(\omega, x, y) := \omega(x) \cdot p_1(x) \cdot \omega(y) \cdot p_2(y).$$

Then by distributivity of multiplication over addition:

$$\sum\nolimits_{x,y} F(\omega,x,y) = \big(\sum\nolimits_x \omega(x) \cdot p_1(x)\big) \cdot \big(\sum\nolimits_y \omega(y) \cdot p_2(y)\big) = (\omega \models p_1) \cdot (\omega \models p_2).$$

Let $X = \{x_1, \ldots, x_n\}$ and let the function H be given by:

$$H(\vec{v}, \lambda) := \sum\nolimits_{i,j} F(\omega, x_i, x_j) \cdot \ln\big(v_i \cdot p_1(x_i) \cdot v_j \cdot p_2(x_j)\big) - \lambda \cdot \big((\sum\nolimits_i v_i) - 1\big).$$

Then:

$$\frac{\partial H}{\partial v_k}(\vec{v}, \lambda) = \sum\nolimits_i \frac{F(\omega, x_k, x_i) + F(\omega, x_i, x_k)}{v_k} - \lambda \qquad \frac{\partial H}{\partial \lambda}(\vec{v}, \lambda) = 1 - \sum\nolimits_i v_i.$$

Setting these to zero gives:

$$1 = \sum_k v_k = \frac{\sum_{k,i} F(\omega, x_k, x_i) + F(\omega, x_i, x_k)}{\lambda} = \frac{2 \cdot (\omega \models p_1) \cdot (\omega \models p_2)}{\lambda}.$$

Hence $\lambda = 2 \cdot (\omega \models p_1) \cdot (\omega \models p_2)$ so that:

$$\begin{aligned} v_k &= \frac{\sum_i F(\omega, x_k, x_i) + F(\omega, x_i, x_k)}{\lambda} \\ &= \tfrac{1}{2} \cdot \frac{\omega(x_k) \cdot p_1(x_k) \cdot (\omega \models p_2)}{(\omega \models p_1) \cdot (\omega \models p_2)} + \tfrac{1}{2} \cdot \frac{(\omega \models p_1) \cdot \omega(x_k) \cdot p_2(x_k)}{(\omega \models p_1) \cdot (\omega \models p_2)} \\ &= \tfrac{1}{2} \cdot \frac{\omega(x_k) \cdot p_1(x_k)}{\omega \models p_1} + \tfrac{1}{2} \cdot \frac{\omega(x_k) \cdot p_2(x_k)}{\omega \models p_2} \\ &= \tfrac{1}{2} \cdot \omega|_{p_1}(x_k) + \tfrac{1}{2} \cdot \omega|_{p_2}(x_k). \end{aligned}$$ □

References

Abramsky, S. (2010). No-cloning in categorical quantum mechanics. In S. Gay, & I. Mackie (Eds.), *Semantical techniques in quantum computation* (pp. 1–28). Cambridge University Press.

Abramsky, S. (2013). Coalgebras, Chu spaces, and representations of physical systems. *Journal of Philosophical Logic, 42*(3), 551–574.

Abramsky, S. (2014). Contextual semantics: From quantum mechanics to logic, databases, constraints, and complexity. *EATCS Bulletin, 113*.

Abramsky, S. (2015). Arrow's theorem by Arrow theory. In Å. Hirvonen, J. Kontinen, R. Kossak, & A. Villaveces (Eds.), *Logic without borders—Essays on set theory, model theory, philosophical logic and philosophy of mathematics* (pp. 15–30). De Gruyter.

Abramsky, S., Barbosa, R., Karvonen, M., & Mansfield, S. (2019). A comonadic view of simulation and quantum resources. In *Logic in computer science* (pp. 1–12). IEEE.

Abramsky, S., Blute, R., & Panangaden, P. (2000). Nuclear and trace ideals in tensored *-categories. *Journ. of Pure & Appl. Algebra,143*, 3–47.

Abramsky, S., & Brandenburger, A. (2011). The sheaf-theoretic structure of non-locality and contextuality. *New Journal of Physics, 13,* 113036.

Abramsky, S., & Coecke, B. (2009). A categorical semantics of quantum protocols. In K. Engesser, D. M. Gabbay, & D. Lehmann (Eds.), *Handbook of quantum logic and quantum structures: Quantum logic* (pp. 261–323). North-Holland, Elsevier: Computer Science Press.

Abramsky, S., & Heunen, C. (2012). H*-algebras and nonunital Frobenius algebras: First steps in infinite-dimensional categorical quantum mechanics. *Clifford Lectures, AMS Proceedings of Symposia in Applied Mathematics, 71,* 1–24.

Baum, L., Petrie, T., Soules, G., & Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Mathematical Statistics, 41*, 164–171.

Bishop, C. (2006). *Pattern recognition and machine learning*. Information science and statistics. Berlin: Springer.

Bruza, P., & Abramsky, S. (2016). Probabilistic programs: Contextuality and relational database theory. In *Quantum interaction* (pp. 163–174).

Chan, H., & Darwiche, A. (2005). On the revision of probabilistic beliefs using uncertain evidence. *Artificial Intelligence, 163,* 67–90.

Cho, K., & Jacobs, B. (2019). Disintegration and Bayesian inversion via string diagrams. *Mathematical Structures in Computer Science, 29*(7), 938–971.

Clerc, F., Dahlqvist, F., Danos, V., & Garnier, I. (2017). Pointless learning. In J. Esparza, & A. Murawski (Eds.), *Foundations of software science and computation structures* (Vol. 10203, pp. 355–369). Lecture notes in computer science. Berlin: Springer.

Darwiche, A. (2009). *Modeling and reasoning with Bayesian networks*. Cambridge: Cambridge University Press.

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *ournal of the Royal Statistical Society, 39*(1), 1–38.

Do, C., & Batzoglou, S. (2008). What is the expectation maximization algorithm? *Nature Biotechnology, 26,* 897–899.

Fritz, T. (2020). A synthetic approach to Markov kernels, conditional independence, and theorems on sufficient statistics. *Advances in Mathematics, 370,* 107239.

Giry, M. (1982). A categorical approach to probability theory. In B. Banaschewski (Ed.), *Categorical aspects of topology and analysis* (Vol. 915, pp. 68–85). Lecture notes in mathematics. Berlin: Springer.

Jacobs, B. (2015). New directions in categorical logic, for classical, probabilistic and quantum logic. *Logical Methods in Computer Science, 11*(3).

Jacobs, B. (2018). From probability monads to commutative effect uses. *Journal of Logical and Algebraic Methods in Programming, 94*, 200–237.

Jacobs, B. (2019). Learning along a channel: The expectation part of expectation-maximisation. In B. König (Ed.), *Mathematical foundations of programming semantics* (Vol. 347, pp. 143–160). *Electronic Notes in Theoretical Computer Science*. Elsevier, Amsterdam.

Jacobs, B. (2019). The mathematics of changing one's mind, via Jeffrey's or via Pearl's update rule. *The Journal of Artificial Intelligence Research, 65,* 783–806.

Jacobs, B. (2020). A channel-based perspective on conjugate priors. *Mathematical Structures in Computer Science, 30*(1), 44–61.

Jacobs, B., & Zanasi, F. (2021). The logical essentials of Bayesian reasoning. In G. Barthe, J. -P. Katoen, & A. Silva (Eds.), *Foundations of probabilistic programming* (pp. 295–331). Cambridge University Press.

Jensen, F., & Nielsen, T. (2007). *Bayesian networks and decision graphs*. Statistics for engineering and information science (2nd rev. edn.). Berlin: Springer.

Koller, D., & Friedman, N. (2009). *Probabilistic graphical models*. Principles and techniques. Cambridge, MA: MIT Press.

Kozen, D. (1981). Semantics of probabilistic programs. *Journal of Computer and System Sciences, 22*(3), 328–350.

Kozen, D. (1985). A probabilistic PDL. *Journal of Computer and System Sciences, 30*(2), 162–178.

McLachlan, G., & Krishnan, T. (1997). *The EM algorithm and extensions*. New York: Wiley.

Mrad, A., Delcroix, V., Piechowiak, S., Leicester, P., & Abid, M. (2015). An explication of uncertain evidence in Bayesian networks: Likelihood evidence and probabilistic evidence. *Applied Intelligence, 23*(4), 802–824.

Pishro-Nik, H. (2014). *Introduction to probability, statistics, and random processes*. Kappa Research LLC. https://www.probabilitycourse.com.

Ross, S. (2018). *A first course in probability*. Pearson Education, 10th edition edition.

Russell, S., & Norvig, P. (2003). *Artificial intelligence*. A Modern Approach. Englewood Cliffs, NJ: Prentice Hall.

Valtorta, M., Kim, Y.-G., & Vomlel, J. (2002). Soft evidential update for probabilistic multiagent systems. *International Journal of Approximate Reasoning, 29*(1), 71–106.

Chapter 30
Structure in Machine Learning

Prakash Panangaden

Abstract I give some examples of instances of ideas from semantics being useful in machine learning.

Keywords Reinforcement learning · Semantics · Fixed-point theory

30.1 Introduction

I am delighted to be able to contribute this short piece to celebrate yet another milestone in Samson Abramsky's illustrious career. Samson is known for many things and I don't want to compile another list of accomplishments here. Rather, I would like to focus on one issue where he has been a forceful advocate: bringing together the two wings of theoretical computer science. These are often referred to as Theory A (algorithms, complexity, combinatorics) and Theory B (logic, semantics, verification). There are some points of contact between the two, finite model theory is a notable example, and some areas that have drifted from one wing to the other, perhaps automata theory is an example. However, culturally and socially, there is a big gap as can be seen at once from a glance at the accepted papers at STOC, FOCS, LICS, POPL and other conferences.

Samson was one of the prime movers in the Simons Institute thematic semester in the Autumn of 2016 where, along with Anuj Dawar and Phokion Kolaitis, we organised a programme specifically aimed at bringing the two branches of theoretical computer science together. Apart from these organisational initiatives he took a leading rôle in pushing research topics which resulted in papers that gave a structural account of important topics in combinatorics (Abramsky et al., 2017; Abramsky & Winschel, 2017; Abramsky & Shah, 2018; Abramsky et al., 2019).

P. Panangaden (✉)
Department of Computer Science, McGill University, Montreal, Canada
e-mail: prakash@cs.mcgill.ca

A. Palmigiano and M. Sadrzadeh (eds.), *Samson Abramsky on Logic and Structure in Computer Science and Beyond*, Outstanding Contributions to Logic 25,
https://doi.org/10.1007/978-3-031-24117-8_30

Another theme that has been pursued by Samson and his collaborators is the use of topological ideas in explaining features of quantum mechanics (Abramsky et al., 2011; Abramsky & Brandenburger, 2011). Thus, one understands contextuality as a manifestation of the nonexistence of global sections of an appropriate sheaf, ideas like had appeared earlier Isham and Butterfield (1998), but the paper by Abramsky and Brandenberger (2011) made this particularly clear in a simple setting. It turned out that this viewpoint could be brought to bear in other areas like logic or the theory of relational databases (Abramsky, 2014; Abramsky et al., 2015). While topology is not geometry, it does lead to visual thinking and intuition.

The most important influence on the present article is a recent paper called "Whither[1] semantics" (Abramsky, 2020) which appeared in a tribute volume to Maurice Nivat. This paper is essentially a call to arms to those of us working with structural approaches to use our methods to attack "hard" problems: here "hard" means concrete quantitative problems. I am not going to summarise that paper here but instead I will take up the challenge and talk about structural methods in machine learning.

The siren song of machine learning has been seducing people from all sorts of intellectual disciplines: probability theory, statistics, optimization, convexity theory and other areas of mathematics, but also algorithms, complexity theory, pattern recognition and numerical linear algebra from computer science and even many branches of physics: Hamiltonian mechanics, statistical mechanics and even quantum field theory.

I cannot resist recounting an incident that happened during the FLoC conference in Oxford in 2018. I was in a pub and bumped into a friend from graduate school days whom I had not seen in 30 years. We were both physics graduate students at Chicago in the 1970s s and he, unlike me, stayed in physics and had become a well-known condensed-matter physicist. He was with a group of Oxford physicists who graciously invited me to join them for a drink. I was asked about my area of research and I replied, "machine learning", whereupon one of the physicists exclaimed, "oh yes, we invented that!"

30.2 Probabilistic Programming Languages

One of the earliest papers to consider a probabilistic programming language is a study of probabilistic LCF by Saheb-Djahromi (1978). This led to work on the interplay between probability theory and domain theory Saheb-Djahromi (1980) and ultimately to probabilistic powerdomains (Jones & Plotkin, 1989) and integration on domains (Abbas, 1995). Shortly after (Kozen, 1981) considered a typical imperative programming augmented with probabilistic choice and gave a formal semantics in terms of measure theory.

[1] Spelled with two h's.

In the world of verification and process algebra papers appeared in the 1980s (Vardi, 1985) but the most significant step was a paper by Larsen and Skou (1991) which appeared as a conference paper in 1989. They introduced a modal logic and proved a logical characterisation theorem. The latter was subsequently refined and extended to continuous state spaces by Desharnais et al. (2002).

The idea of modelling a stochastic process as a program in a programming language is due to Gupta et al. (1999). In this paper the language was in the family of concurrent constraint programming languages (Saraswat, 1989). What was significant here is the important role played by conditional probability in the semantics.

It took about 10 years before these ideas really flowered. This happened in a spectacular result due to Ackerman et al. (2011) where they showed that one could have a computable probability distribution and impose computable constraints but the resulting conditional distribution is *non-computable*. This raised the question about what could be the included in a probabilistic programming language. Clearly one needs some kind of control on conditioning.

But why were machine learning people interested in probabilistic programming languages? For precisely the same reasons that the semantics community had been advocating all along: *compositionality*. It is clear that graphical formalisms like Bayes nets will get hopelessly clumsy as they get larger. Perhaps this lesson has not sunk in to the wider machine learning community as graphical models are widely used and taught but at least the idea of compositional construction of models has indeed entered the subject and the explicit role played by programming language theory is acknowledged.

At present one cannot say that the main stream of machine learning has embraced these ideas, but there are flourishing groups at the intersection of programming languages and machine learning that are developing higher-order probabilistic languages (Goodman et al., 2008; Tolpin et al., 2016) and the theory needed to understand the combination of probability distributions, higher-type programs and conditioning (Staton et al., 2017; Heunen et al., 2017; Cho & Jacobs, 2019).

30.3 Reinforcement Learning and Fixed-Point Theory

In statistical machine learning one is typically trying to learn some structural information from randomly sampled data. A typical example is where one has an unknown labelling function i.e. a map from $L : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X}$ is the space from which samples are drawn and $\mathcal{Y}$ is the space of labels. The machine-learning algorithm is presented with m samples, correctly labelled, and tries to "learn" L. Reinforcement learning (RL), by contrast, is a more active process.

The basic set up is a Markov decision process (MDP). One has an agent that interacts with this process and is rewarded (or penalised) as actions are chosen. The agent tries to learn a policy that will optimise the cumulative reward. We formalise this a bit more precisely now. We write $\mathcal{D}(S)$ for the set of probability distributions on a set S.

Definition 30.1 A **Markov decision process** M is a 4-tuple

$$(S, \mathcal{A}, \forall a \in \mathcal{A}\ \tau_a : S \to \mathcal{D}(S), R : S \times \mathcal{A} \to \mathcal{D}(\mathbf{R})),$$

where, S is a finite set of *states*, $\mathcal{A}$ is a set of *actions*, τ_a is a *transition probability function* and R is the *reward*.

All kinds of minor variations may be seen in the literature. If the system is in a state s and the action a is chosen, there is a transition to a new state governed by the probability distribution $\tau_a(s)$ and a reward is assigned according to the probability distribution $R(s, a)$. A numerical parameter $\gamma \in (0, 1)$, called the *discount factor* is often given as part of the description of the MDP. In RL, the agent does not know the internal structure or dynamics of the MDP and by choosing actions tries to learn a *policy*, a map $\pi : S \to \mathcal{D}(\mathcal{A})$, which will maximise the expected discounted reward:

$$\sum_{t=0}^{\infty} \gamma^t r_t,$$

where t represents the time step and r_t is the reward at time t. Rewards in the future are discounted, which ensures that the sum is finite if R is reasonably well behaved. The basic paradigm is due to Bellman (2019) who invented the basic algorithms, namely dynamic programming, to determine optimal policies. A concise modern exposition is given in Szepesvári (2010) including new algorithms invented since that time.

This subject is not an example of the semantics community influencing the machine learning community, rather it is a case of two communities converging onto a common mathematical paradigm: fixed-point theory. For semanticists, fixed-point theory is a basic and beloved tool imported into programming language theory by Dana Scott, Jaco de Bakker and others from its roots in computability (Kleene) and lattice theory (Tarski). In RL the existence of an optimal policy, *and an algorithm to compute it* is based on the Banach fixed-point theorem.

Returning to the world of MDPs, we define a *value function* as a map from states to expected rewards. The value function taken as parameter the policy π. More precisely, we define $V^\pi : S \to \mathbf{R}$, for a policy π by

$$V^\pi(s) = \mathbb{E}_{a \sim \pi(s)}[\mathbb{E}_{r \sim R(s,a)}[r] + \gamma \sum_{s' \in S} \tau_a(s)(s') V^\pi(s').]$$

Here the notation $\mathbb{E}_{x \sim D}$ means the expectation value when x is sampled according to the distribution D. This can clearly be seen to be a fixed-point equation for the *function* V; so it is even a higher-type entity.

The basic theorem that ensures the existence of V is the Banach fixed-point theorem.

Theorem 30.2 *Let* (X, d) *be a* complete *metric space and let* $f : X \to X$ *be a contractive function:* $\exists \gamma \in (0, 1),\ \forall x, y \in X,\ d(f(x), f(y)) \leq \gamma d(x, y)$. *Then there is a* unique *fixed point for* f*: a point* x_0 *such that* $f(x_0) = x_0$.

The proof is easy, start from any point and keep iterating f. The contractiveness condition ensures that the sequence is Cauchy and the completeness ensures that the sequence has a limit. It is easy to verify that this limit is the unique fixed point. Indeed this iterative approach is the basis of many of the algorithms for finding optimal policies. Here we have only talked about the value function associated with a policy, but it is not hard to define an optimal value function and show that it too satisfies a fixed-point equation.

Here is a case of a parallel evolution of similar ideas. The use of fixed-point theory in these cases is more than a coincidence. The communities did not interact and there was no cross fertilisation. It is also the case that fixed-point theory was used in various areas of mathematics and economics and there were many other fixed-point theorems available. However, next we shall see a case where there was interaction with tangible results.

30.4 A Role for Bisimulation Metrics

Bisimulation is one of the triumphs of semantics. Its origins in process algebra have been well described by Sangiorgi (2009) and the books (Sangiorgi, 2011; Sangiorgi & Rutten, 2011) give an up-to-date account of the state of the art including probabilistic bisimulation introduced by Larsen and Skou (1991). The machine learning community working with MDPs independently invented probabilistic bisimulation (Givan et al., 2003) though they did acknowledge the priority of Larsen and Skou; unfortunately, everyone else in machine learning seems to be unaware of the work of Larsen and Skou.

Shortly after the introduction of probabilistic bisimulation, it became clear that the concept was not robust (Giacalone et al., 1990). Bisimulation is a binary relation: it holds or not, whereas the parameters on which it depends vary continuously. The idea of a *metric* measuring behavioural equivalence was mooted by Giacalone et al. Unfortunately, it is not easy to define such a concept: a naive “up to ε” definition does not work. The definition was finally given by Desharnais et al. (1999; 2004) based on an earlier logical characterisation result (Desharnais & Panangaden, 1998). A fixed-point definition more in the spirit of the bisimulation relation as a greatest-fixed-point was then given by van Breugel and Worrell (2001; 2001). This version of the definition is what has spurred most of the subsequent developments and in particular the use of ideas from optimal transport theory and Kantorovich-Rubinstein duality (Villani, 2008).

These metrics were defined for labelled Markov processes: transition systems with no notion of “reward” or “optimal policy.” A bisimulation metric for MDPs was forthcoming shortly thereafter (Ferns et al., 2004, 2005). In these papers the definition of bisimulation is given by adding a term that reflects the reward. This tiny change had a significant impact: it was proven that the difference in the optimal value function between two different states: $|V(s) - V(s')|$ is bounded by the bisimulation metric $d(s, s')$. This showed a very tight relation between the bisimulation metric and the quantities of interest in reinforcement learning. Indeed later Ferns and Precup (2014) showed that bisimulation metrics *are* optimal value functions for an appropriate MDP.

30.5 A Recent Success Story

Recently a striking use of metric arguments is in a recent paper in The 23rd International Conference on Artificial Intelligence and Statistics (AISTATS 2020) conference (Amortila et al., 2020). A number of algorithms in RL proceed by sampling from the MDP: so the assumption is that the dynamics of the MDP are not known but one can obtain samples of trajectories from the MDP. The algorithms then proceed by iteratively computing an approximation to the fixed point. There are many such algorithms, the general class of such algorithms is called *stochastic approximation algorithms*.

Proofs of convergence of these algorithms can be quite difficult; see the references in Amortila et al. (2020) for details. In the AISTATS paper however, we took the perspective that one should view the algorithms themselves as Markov processes of *higher type*: that means the state space is the space of distributions. Then using coupling techniques we showed that many of the algorithms define contractive maps on the space of distributions equipped with the Kantorovich-Wasserstein[2] metric. Thus many of the difficult proofs found in the literature can be done in a simple and more uniform way. One has to find a coupling; this may be more or less difficult, but all our examples (there are seven algorithms that we treat) are dealt with simple-minded couplings. This is, in my opinion, a pleasing application of ideas from the semantics world to RL and one which is appreciated by the RL community.

We proceed to some of the details of this work. The first innovation, which is due to the last-named author of Amortila et al. (2020) and other co-workers (Bellemare et al., 2017), is to view reinforcement learning as working with the distribution of possible rewards rather than the expected reward. This is called "distributional reinforcement learning" and there seems to be experimental support that one gets better performance using this type of learning.

In Amortila et al. (2020) the basic idea is to try to show that the update rule defined on the distributions is a contraction in suitable metric. This metric is the Kantorovich metric obtained by lifting the metric induced by the infinity norm.

Given an MDP as defined above with a discount factor γ, we introduce the *Bellman operator* T^{π} which is dependent on a choice of policy π by:

[2] This should just be called the Kantorovich metric. It is usually called just the Wasserstein metric; this is a historical mistake.

$$T^{\pi}(V) = \lambda s.\mathbb{E}_{a\sim\pi(s),r\sim R(s,a)}[r + \gamma\mathbb{E}_{s'\sim P(s,a)}[V(s')]]$$

The fixed-point of T^{π} is the value function V^{π} described above. The value function of the optimal policy π^* is the fixed-point of what is called the *Bellman optimality operator*:

$$T(V) = \lambda s.\max_a \mathbb{E}_{r\sim R(s,a)}[r + \gamma\mathbb{E}_{s'\sim P(s,a)}[V(s')]].$$

These operators are contractions on the space $\mathbf{R}^{|S|}$ with the metric induced by the infinity norm.

We now give a more explicit description of the Kantorovich metric (Villani, 2008). Suppose that we wish to compare two probability measures P, Q defined on a metric space (X, ρ) with the Borel sets induced by the metric. What can one "see" about a measure? Integrals! So we can think of something like

$$\sup_f |\int f\mathrm{d}P - \int f\mathrm{d}Q|.$$

But over what class of functions should one take the sup? If we don't restrict the functions, it is easy to see that the sup can be made infinite always. Kantorovich restricted to *non-expansive* functions or 1-Lipschitz functions: $(f(x) - f(y))| \leq \rho(x, y)$. This is where the metric ρ plays a crucial rôle. We call the set of such functions $1 - \mathrm{Lip}$. Then we define

$$K(P, Q) = \sup_{f\in 1-\mathrm{Lip}} |\int f\mathrm{d}P - \int f\mathrm{d}Q|.$$

There is a beautiful duality theorem known as Kantorovich-Rubinstein duality, which gives a description of K as a minimum.

Definition 30.3 A **coupling** between probability distributions P, Q on a space (X, ρ) is a probability measure γ defined on $X \times X$ such that the marginals coincide with P and Q. This means that for any Borel set $A \subset X$ we have $\gamma(A \times X) = P(A)$ and $\gamma(X \times A) = Q(A)$.

Intuitively, one thinks of this as a transport plan. Think of the measures as defining piles of sand. One wishes to move the sand so that the pile P becomes the pile Q. The quantity $\gamma(A \times B)$ defines how much must be moved from region A to region B. The cost of moving the sand depends on how far the sand is to be moved. Thus we are led to define the total cost as

$$\int \rho(x, y)\mathrm{d}\pi.$$

Let us write $\Gamma(P, Q)$ for the set of all possible couplings between P and Q. The Kantorovich-Rubinstein duality theorem states:

$$W_1(P, Q) = \inf_{\gamma \in \Gamma(P,Q)} \int_{X \times X} \rho(x, y) \mathrm{d}\gamma = K(P, Q).$$

The notation W_1 is much more common and in view of the duality theorem it is the same as K. The key point is that the W_1 metric is defined using optimal couplings. Thus any old coupling gives an upper bound on the W_1 distance.

Now let us consider a typical RL algorithm, called $TD(0)$ (Szepesvári, 2010). The TD stands for "temporal difference" and the 0 need not concern us. It has a simple update rule for the value function. If the value function after n steps is written as V_n we have

$$V_{n+1}(s) := (1 - \alpha)V_n(s) + \alpha(R(s, a) + \gamma V_n(s'))$$

which should be viewed as a distributional equation in our setting. Here $R(s, a)$ is the distribution over the possible rewards defined by the MDP, a is sampled from the actions according to the distribution defined by the policy and s' is the new state sampled according to the MDP dynamics. The parameter α, called the "step size", tells us the extent we use the old value function as opposed to the one generated after one transition. The coinductive character of these kind of value function updates makes it a natural to compare it with a bisimulation metric and, indeed, in Ferns et al. (2004) such bounds are given.

We take the view that a learning algorithm is itself a Markov chain whose state space is the space of probability distributions over the underlying MDP. The update rule for a particular learning algorithm like $TD(0)$ then defines a Markov kernel. A Markov kernel K can be viewed as a map from a probability distribution P to a new distribution $K(P)$ given by

$$K(P) = A \mapsto \int_X K(x, A) \mathrm{d}P(x)$$

where A is a Borel set. In our paper (Amortila et al., 2020) we proved that the $TD(0)$ kernel, call it K_α for step size α, defines a contractive map:

$$W_1(K_\alpha(P), K_\alpha(Q)) \leq (1 - \alpha + \alpha\gamma)W_1(P, Q)$$

for any step size $\alpha \in (0, 1]$.

The idea of the proof is to start with an optimal coupling of the initial distributions and then we use this to construct a coupling for the updated distributions. This latter coupling may not be optimal but it suffices to produce an upper bound and prove contractiveness.

Using essentially the same method we were able to cover 6 more algorithms in the same way. The technique depends on one's ability to find suitable couplings for the updates. What is particularly pleasing is that the construction of the couplings was *ridiculously easy*. Of course, we then encountered examples where it was not at all clear that we could find suitable couplings for the updates. However, the example that we did do were all notable examples in the literature and had been proved to converge earlier by much more painful methods. A little structural thinking can go a long way.

30.6 Conclusions

This short note is a response to a question raised by Samson in "Whither Semantics": do we want to lead or follow? I would like to suggest that indeed there is a role for semantics-based ideas in machine learning. I hope that the examples described above will give the semantics community some motivation to interact with the machine learning community. I must take this opportunity to say that in my case machine learning people, Doina Precup and Joelle Pineau, came knocking on my door and sought me out rather than the other way around.

There is plenty more to be done. One of the big mysteries of machine learning is why do neural networks work so well? All the theory of optimization is geared towards convex situations but in deep neural networks are very non-convex. They are also massively overparametrised so why do they seem to generalise so well? People are invoking ideas from physics and high-dimensional geometry to try to explain these things: most of these ideas are in flux and largely experimental so it would not do much good to cite a lot of papers here. As far as I can see, any decent ideas are worth investigating: higher-order programming, monoidal categories and the algebra of string diagrams, ideas from causality and many other things.

References

Abramsky, S., & Brandenburger, A. (2011). A unified sheaf-theoretic account of non-locality and contextuality. *CoRR*. arXiv:1102.0264.

Abramsky, S., Barbosa, R. S., Kishida, K., Lal, R., & Mansfield, S. (2015). Contextuality, cohomology and paradox. In S. Kreutzer (Ed.), *24th EACSL Annual Conference on Computer Science Logic, CSL 2015, 7–10 Sept 2015, Berlin, Germany* (Vol. 41, pp. 211–228). *LIPIcs*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Abramsky, S., Barbosa, R. S., Karvonen, M., & Mansfield, S. (2019). A comonadic view of simulation and quantum resources. In *34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2019, Vancouver, BC, Canada, 24–27 June 2019* (pp. 1–12). IEEE.

Abramsky, S. (2014).Contextual semantics: From quantum mechanics to logic, databases, constraints, and complexity. *CoRR*. arXiv:1406.7386.

Abramsky, S. (2020). Whither semantics? *Theoretical Computer Science, 807,* 3–14.

Abramsky, S., Dawar, A., & Wang, P. (2017). The pebbling comonad in finite model theory. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, 20–23 June 2017* (pp. 1–12). IEEE Computer Society.

Ackerman, N. L., Freer, C. E., & Roy, D. M. (2011). Noncomputable conditional distributions. In *2011 26th Annual IEEE Symposium on Logic in Computer Science (LICS)* (pp. 107–116). IEEE.

Abramsky, S., Mansfield, S., & Barbosa, R. S. (2011). The cohomology of non-locality and contextuality. In B. Jacobs, P. Selinger, & B. Spitters (Eds.), *Proceedings 8th International Workshop* *on Quantum Physics and Logic, QPL 2011, Nijmegen, Netherlands, 27–29 Oct 2011* (Vol. 95, pp. 1–14). *EPTCS*.

Amortila, P., Precup, D., Panangaden, P., & Bellemare, M. G. (2020). A distributional analysis of sampling-based reinforcement learning algorithms. In *The 23rd International Conference on Artificial Intelligence and Statistics*.

Abramsky, S., & Shah, N. (2018). Relating structure and power: Comonadic semantics for computational resources. In D. R. Ghica, & A. Jung (Eds.), *27th EACSL Annual Conference on Computer Science Logic, CSL 2018, 4–7 Sept 2018, Birmingham, UK* (Vol. 119, pp. 2:1–2:17). *LIPIcs*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Abramsky, S., & Winschel, V. (2017). Coalgebraic analysis of subgame-perfect equilibria in infinite games without discounting. *Mathematical Structures in Computer Science, 27*(5), 751–761.

Bellemare, M. G., Dabney, W., & Munos, R. (2017). A distributional perspective on reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning* (Vol. 70. pp. 449–458) JMLR.org.

Bellman, R. E. (1954). The theory of dynamic programming. *Bulletin of the American Mathematical Society, 60*(6), 503–516.

Cho, K., & Jacobs, B. (2019). Disintegration and Bayesian inversion via string diagrams. *Mathematical Structures in Computer Science, 29*(7), 938–971.

Desharnais, J., Edalat, A., & Panangaden, P. (1998) A logical characterization of bisimulation for labelled Markov processes. In *proceedings of the 13th IEEE Symposium On Logic In Computer Science, Indianapolis* (pp. 478–489). IEEE Press.

Desharnais, J., Edalat, A., & Panangaden, P. (2002). Bisimulation for labeled Markov processes. *Information and Computation, 179*(2), 163–193.

Desharnais, J., Gupta, V., Jagadeesan, R., & Panangaden, P. (1999). Metrics for labeled Markov systems. In *Proceedings of CONCUR99* (Vol, 1664). Lecture notes in computer science. Berlin: Springer.

Desharnais, J., Gupta, V., Jagadeesan, R., & Panangaden, P. (2004). A metric for labelled Markov processes. *Theoretical Computer Science, 318*(3), 323–354.

Edalat, A. (1995). Domain theory and integration. *Theoretical Computer Science, 151,* 163–193.

Ferns, N., & Precup, D. (2014). Metrics are optimal value functions. In *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence*. Article 67.

Ferns, N., Panangaden, P., & Precup, D. (2004). Metrics for finite Markov decision precesses. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence* (pp. 162–169).

Ferns, N., Panangaden, P., & Precup, D. (2005). Metrics for Markov decision processes with infinite state spaces. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence* (pp. 201–208).

Givan, R., Dean, T., & Greig, M. (2003). Equivalence notions and model minimization in Markov decision processes. *Artificial Intelligence, 147*(1–2), 163–223.

Gupta, V., Jagadeesan, R., & Panangaden, P. (1999). Stochastic processes as concurrent constraint programs. In *Proceedings of the 26th Proceedings of the Annual ACM Symposium on Principles of Programming Languages* (pp. 189–202).

Giacalone, A., Jou, C., & Smolka, S. (1990). Algebraic reasoning for probabilistic concurrent systems. In *Proceedings of the Working Conference on Programming Concepts and Methods*. IFIP TC2.

Goodman, N., Mansinghka, V., Roy, D. M., Bonawitz, K., & Tenenbaum, J. B. (2008). Church: A language for generative models. In *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence* (pp. 220–229).

Heunen, C., Kammar, O., Staton, S., & Yang, H. (2017). A convenient category for higher-order probability theory. In *Proceedings of the Thirty-second Annual ACM-IEEE Symposium on Logic in Computer Science*.

Isham, C. J., & Butterfield, J. (1998). Topos perspective on the kochen-specker theorem: I. quantum states as generalized valuations. *International journal of theoretical physics, 37*(11), 2669–2733.

Jones, C., & Plotkin, G. D. (1989). A probabilistic powerdomain of evaluations. In *Proceedings of the Fourth Annual IEEE Symposium On Logic In Computer Science* (pp. 186–195).

Kozen, D. (1981). Semantics of probabilistic programs. *Journal of Computer and Systems Sciences, 22,* 328–350.

Larsen, K. G., & Skou, A. (1991). Bisimulation through probablistic testing. *Information and Computation, 94,* 1–28.

Sangiorgi, D. (2009). On the origins of bisimulation and coinduction. *ACM Transactions on Programming Languages and Systems (TOPLAS), 31*(4), 15.

Sangiorgi, D. (2011). *Introduction to Bisimulation and Coinduction*. Cambridge University Press.

Saraswat, V. A. (1989). *Concurrent constraint programming languages*. Ph.D. thesis, Carnegie-Mellon University.

Saheb-Djahromi, N. (1978) Probabilistic LCF. In *Mathematical foundations of computer science* (Vol. 64). Lecture notes in computer science. Berlin: Springer.

Saheb-Djahromi, N. (1980). Cpos of measures for nondeterminism. *Theoretical Computer Science, 12*(1), 19–37.

Sangiorgi, D., & Rutten, J. (Eds.), *Advanced topics in bisimulation and coinduction* (Vol. 52). Cambridge tracts in theoretical computer science. Cambridge University Press.

Staton, S., Yang, H., Wood, F., Heunen, C., & Kammar, O. (2017). Semantics for probabilistic programming: Higher-order functions, continuous distributions, and soft constraints. In *Proceedings of the 31st Annual ACM-IEEE Symposium On Logic In Computer Science* (pp. 525–534).

Szepesvári, C. (2010). Algorithms for reinforcement learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning, 4*(1), 1–103.

Tolpin, D., van de Meent, J. W., Yang, H., & Wood, F. (2016). Design and implementation of probabilistic programming language Anglican. In *Proceedings of the 28th Symposium on the Implementation and Application of Functional Programming Languages* (p. 6.) ACM.

Vardi, M. (1985). Automatic verification of probabilistic concurrent finite-state programs. In *26th IEEE Symposium On Foundations Of Computer Science* (pp. 327–338).

van Breugel, F., & Worrell, J. (2001). An algorithm for quantitative verification of probabilistic systems. In K. G. Larsen, & M. Nielsen (Eds.), *Proceedings of CONCUR'01* (Vol. 2154, pp. 336–350). Lecture notes in computer science. Berlin: Springer.

van Breugel, V., & Worrell, J. (2001). Towards quantitative verification of probabilistic systems. In *Proceedings of the Twenty-Eighth International Colloquium on Automata, Languages and Programming* (Berlin: Springer).

Villani, C. (2008). *Optimal transport: Old and new*. Berlin: Springer.