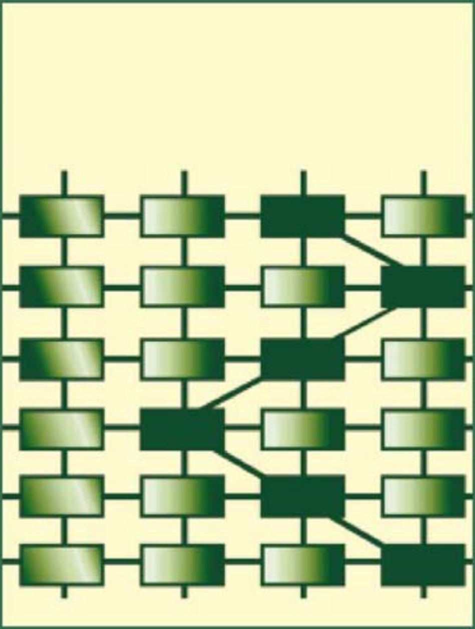
International Mathematical Series • Volume 4

MATHEMATICAL PROBLEMS FROM APPLIED LOGIC I

Logics for the XXIst Century

Dov M. Gabbay
Sergei S. Goncharov
Michael Zakharyaschev
EDITORS



Mathematical Problems from Applied Logic I

INTERNATIONAL MATHEMATICAL SERIES

Series Editor: Tamara Rozhkovskaya
Sobolev Institute of Mathematics of the Siberian Branch
of the Russian Academy of Sciences, Novosibirsk, Russia

5. **Mathematical Problems from Applied Logic II. New Logics for the XXIst Century** • D. Gabbay, S. Goncharov, M. Zakharyaschev Eds. • 2006
4. **Mathematical Problems from Applied Logic I. New Logics for the XXIst Century** • D. Gabbay, S. Goncharov, M. Zakharyaschev Eds. • 2005
3. **Different Faces of Geometry** • S. Donaldson, Ya. Eliashberg, M. Gromov Eds. • 2004
2. **Nonlinear Problems in Mathematical Physics and Related Topics II. In Honor of Professor O. A. Ladyzhenskaya** • M.Sh. Birman, S. Hildebrandt, V.A. Solonnikov, N.N. Uraltseva Eds. • 2003
1. **Nonlinear Problems in Mathematical Physics and Related Topics I. In Honor of Professor O. A. Ladyzhenskaya** • M.Sh. Birman, S. Hildebrandt, V.A. Solonnikov, N.N. Uraltseva Eds. • 2002

Mathematical Problems from Applied Logic I

Logics for the XXIst Century

Edited by

Dov M. Gabbay
King's College London
London, UK

Sergi S. Goncharov
SB Russian Academy of Sciences
Novosibirsk, Russia

and

Michael Zakharyaschev
King's College London
London, UK

Springer

Dov M. Gabbay
Department of Computer Science
King’s College London
Strand, London WC2R 2LS
UK
dg@dcs.kcl.ac.uk

Michael Zakharyaschev
Department of Computer Science
King’s College London
Strand, London WC2R 2LS
UK
mz@dcs.kcl.ac.uk

Sergei S. Goncharov
Sobolev Institute of Mathematics
SB Russian Academy of Sciences
Novosibirsk 630090
Russia
gonchar@math.nsc.ru

This series was founded by Kluwer/Plenum Publishers (now Springer) and the Russian publisher Tamara Rozhkovskaya (Novosibirsk, Russia, rozhk@math.nsc.ru) in 2002. Each volume is simultaneously published in English and in Russian and presents contributions from volume editors and authors exclusively invited by the Series editor. The English Camera-Ready Manuscript was prepared by Tamara Rozhkovskaya

Library of Congress Control Number: 2005931830

ISBN-10: 0-387-28688-8 Printed on acid-free paper.
ISBN-13: 978-0387-28688-4
ISSN 1571-5485

Printed in the United States of America. (IBT)

9 8 7 6 5 4 3 2 1

springeronline.com

Mathematical Problems from Applied Logic I
New Logics for the XXIst Century

Two volumes of the *International Mathematical Series* present the most important thematic topics of logic confronting us in this century, including problems arising from successful applications areas such as Computer Science, AI language, etc. etc.

Invited authors — world-known specialists in the field of logic — were asked to write a chapter (in the form of a survey, a specific problem, or a point of view) basically outlining

WHAT IS ON MY MIND AS MOST
STRIKING/IMPORTANT/PRESSING
NEED TO BE DONE?

Main Topics

- Nonstandard inferences in description logics; an overview of the modern state, open problems, and perspectives for future research
- Logic of provability and a list of open problems in informal concepts of proof, intuitionistic arithmetic, bounded arithmetic, bimodal and polymodal logics, Magari algebras and Lindenbaum Heyting algebras, interpretability logic and its kin, graded provability algebras
- Logical dynamics: a survey of conceptual issues and open mathematical problems emanating from the recent development of various "dynamic-epistemic logics" for information update and belief revision. These systems put many-agent activities at the center stage of logic, such as speech acts, communication, and general interaction
- The continuing relevance of Turing's approach to real-world computability and incomputability, and the mathematical modeling of emergent phenomena. Related open questions of a research interest in computability theory.
- Door to open: Mathematical logic and cognitive science
- Door to open: Semantics of medieval Arab linguists
- What logics do we need? What are logical systems and what should they be? What is a proof? What foundations do we need?
- Applied logic: characterization and relation with other trends in logic, computer science, and mathematics

Editors

Dov M Gabbay

King's College London
London, UK

Sergei S Goncharov

Sobolev Institute of Mathematics
SB Russian Academy of Sciences

Novosibirsk State University
Novosibirsk, Russia

Michael Zakharyaschev

Department of Computer Science
King's College London
London, UK

Dov M Gabbay

Augustus De Morgan Professor of Logic
Department of Computer Science
King's College London
Strand, London WC2R 2LS
UK

dg@dcs.kcl.ac.uk
www.dcs.kcl.ac.uk/staff/dg

- Author of the books ○ *Temporal Logics, Vols. 1,2*, OUP, 1994, 2000 ○ *Fibred Semantics*, OUP, 1998 ○ *Elementary Logic*, Prentice-Hall, 1998, ○ *Fibring Logics*, OUP, 1998 ○ *Neural-Symbolic Learning Systems* (with A. Garcez and K. Broda), Springer, 2002 ○ *Agenda Relevance* (with J. Woods), Elsevier, 2003, etc.
- Editor-in-Chief of journals ○ *Journal of Logic and Computation* ○ *Logic Journal of the IGPL* ○ *Journal of Applied Logic* (with J. Siekmann and A. Jones) ○ *Journal of Language and Computation* (with T. Fernando, U. Reyle, and R. Kempson) ○ *Journal of Discrete Algorithms*
- Editor of journals and series ○ *Autonomous Agents* ○ *Studia Logica* ○ *Journal of Applied Non-Classical Logics* ○ *Journal of Logic, Language & Information* ○ *F News* ○ *Handbook of Logic in Computer Science* (with S. Abramsky and T. Maibaum), ○ *Handbook of Logic in AI and Logic Programming* (with C. Hogger and J.A. Robinson), ○ *Handbook of Philosophical Logic* (with F. Guenthner) ○ *Handbook of the History of Logic* ○ *Handbook of the Philosophy of Science* ○ *Studies in Logic and the Foundations of Mathematics*, etc.

Scientific interests: Logic and computation, dynamics of practical reasoning, proof theory and goal-directed theorem proving, non-classical logics and non-monotonic reasoning, labelled deductive systems, fibring logics, logical modelling of natural language

Sergei S Goncharov

Sobolev Institute of Mathematics
SB Russian Academy of Sciences
4, Prospekt Koptyuga
Novosibirsk 630090
Russia

gonchar@math.nsc.ru
www.math.nsc.ru

- Head of Mathematical Logic Department and Laboratory of Computability and Applied Logic, Sobolev Institute of Mathematics SB Russian Academy of Sciences
- Council Member of the *Association for Symbolic Logic*
- Professor of Logic and Dean of Mathematical Department of the Novosibirsk State University
- Vice-Chairman of *Siberian Fund of Algebra and Logic*
- Author of the books ○ *Countable Boolean Algebras and Decidability*, Consultants Bureau, New York, 1997 ○ *Constructive Models* (with Yu.L. Ershov), Kluwer Academic/ Plenum Publishers, 2000
- Editor-in-Chief of ○ *Bulletin of the Novosibirsk State University. Ser. Mathematics, Mechanics, and Informatics*
- Editor of journals and series ○ *Algebra and Logic* (Associate Ed.) ○ *Siberian Mathematical Journal* ○ *Siberian School of Algebra and Logic* (monograph series), Kluwer Academic / Plenum Publishers ○ *Handbook of Recursive Mathematics* (with Yu.L. Ershov, A. Nerode, J.B. Remmel, and V.W. Marek), Vols. 1,2, Elsevier, 1998 ○ *Computability and Models (Perspectives East and West)* (with S.B. Cooper), Kluwer Academic/ Plenum Publishers, 2003

Scientific interests: Theory of computability, computable and decidable models, abstract data types, model theory and algebra, computer science and logic programming, applied logic

Michael Zakharyaschev

Department
of Computer Science
King's College London
Strand, London WC2R 2LS
UK

mz@dcs.kcl.ac.uk
www.dcs.kcl.ac.uk/staff/mz

- Professor of logic and computation, Department of Computer Science, King's College London
- Logic coordinator of *Group of Logic, Language and Computation* www.dcs.kcl.ac.uk/research/groups/gllc
- Member of the Steering Committee of *Advances in Modal Logic* (a bi-annual workshop and book series in Modal Logic), www.aiml.net
- Author of the books ∘ *Modal Logic* (with A. Chagrov), Oxford Logic Guides: 35, Clarendon Press, Oxford, 1997 ∘ *Many-Dimensional Modal Logics: Theory and Applications*, (with D. Gabbay, A. Kurucz, and F. Wolter), Series in Logic and the Foundation of Mathematics, 148, Elsevier, 2003
- Editor of journals and series ∘ *Studia Logica* (Associate Ed.) ∘ *Journal of Applied Logic* ∘ *Journal of Logic and Computation* ∘ *Advances in Modal Logic*

Scientific interests: Knowledge representation and reasoning, modal and temporal logics, description logics, spatial and temporal reasoning, automated theorem proving, intuitionistic and intermediate logics

Authors

Franz Baader

Technische Universität Dresden
Dresden, Germany

Lev Beklemishev

Steklov Mathematical Institute RAS
Moscow, Russia

Universiteit Utrecht
Utrecht, The Netherlands

Johan van Benthem

University of Amsterdam
Amsterdam, The Netherlands

Stanford University
Stanford, USA

S Barry Cooper

University of Leeds
Leeds, UK

John N Crossley

Monash University
Melbourne, Australia

Wilfrid A Hodges

Queen Mary University of London
London, UK

Ralf Küsters

Institut für Informatik
und Praktische Mathematik
Christian-Albrechts-Universität zu Kiel
Kiel, Germany

Lawrence S Moss

Indiana University
Bloomington, USA

Albert Visser

Universiteit Utrecht
Utrecht, The Netherlands

Franz Baader

Faculty of Computer Science
Technische Universität Dresden
D-01062 Dresden, Germany

baader@tcs.inf.tu-dresden.de
lat.inf.tu-dresden.de/~baader

- Fellow of the European Coordinating Committee for Artificial Intelligence
- Author (with T. Nipkow) of the textbook *Term Rewriting and All That*, Cambridge Univ. Press, 1998
- Editor of ○ *Journal of Applied Non-Classical Logics* ○ *The European Journal on Artificial Intelligence* ○ *The Journal of Artificial Intelligence Research* ○ *Journal of Applied Logic* ○ *Logical Methods in Computer Science* ○ *The Electronic Transactions on Artificial Intelligence* ○ *The Description Logic Handbook* (Cambridge, 2003)

Scientific interests: Knowledge representation (description logics, modal logics), automated deduction (term rewriting, unification theory)

Lev Beklemishev

Steklov Mathematical Institute RAS
8, Gubkina Str. Moscow 119991
Russia

Universiteit Utrecht
8 Heidelberglaan, 3584 CS Utrecht
The Netherlands

bekl@mi.ras.ru, lev@phil.uu.nl
www.phil.uu.nl/~lev

- Member of the Committee on Logic in Europe
- Author of the book *Provability, Complexity, Grammars* (with M. Pentus and N. Vereshchagin) AMS, 1999
- Rewiews editor of *Bulletin of Symbolic Logic* and Member of Translation Committee of *Association for Symbolic Logic*

Scientific interests: Proof theory, formal arithmetic and its fragments, provability logics, modal logics

Johan van Benthem

University of Amsterdam
Plantage Muidergracht 24
1018 TV Amsterdam
The Netherlands

Stanford University
Stanford, California 94305, USA

staff.science.uva.nl/~johan

johan@science.uva.nl, johan@csli.stanford.edu

- First director (in the 1990s) of the Institute for Logic, Language, and Computation (ILLC), www.illc.uva.nl
- First chairman (in the 1990s) and honory member of FoLLI, the *Association of Logic, Language and Information*, www.folli.org
- Member of Dutch Royal Academy of Science, European Academy of Science, Institute Internaional de Philosophie
- Recipient of a five-year national NWO Spinoza Award for the project *Logic in Action* (www.illc.uva.nl/lia)
- Author of the books ○ *A Manual of Intensional Logic*, 1985, 1988 ○ *The Logic of Time*, Kluwer, 1983, 1991 ○ *Modal Logic and Classical Logic*, Bibliopolis, 1985 ○ *Essays in Logical Semantics*, D. Reidel, 1986 ○ *Language in Action*, North-Holland, 1991; MIT Press, 1995 ○ *Exploring Logical Dynamics* CSLI Publications, 1996 ○ *Logic in Games*, 2002, etc.
- Editor of Handbooks and series ○ *Handbook of Logic and Language* (with A. ter Meulen), Elsevier, 1997 ○ *Handbook of Modal Logic* (with P. Blackburn and F. Wolter), Elsevier [to appear] ○ *Handbook of the Philosophy of Information* (with P. Adriaans), Elsevier [to appear] ○ *Handbook of Spatial Reasoning* (with M. Aiello and I. Pratt-Hartmann), Springer [to appear] ○ *Studies in Logic and Practical Reasoning*, Elsevier
- Editor of ○ *Journal of Logic and Computation* ○ *Logic Journal of the IGPL* ○ *Studia Logica*, etc.

Scientific interests: Logic and its applications to language, information, and cognition; modal logic, dynamic logic, logical semantics, games

S Barry Cooper

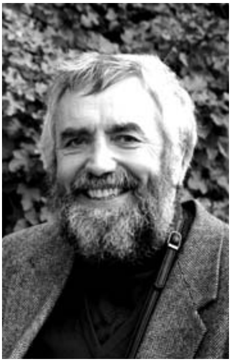
Department of Pure Mathematics
University of Leeds
Leeds LS2 9JT
UK

s.b.cooper@leeds.ac.uk
www.amsta.leeds.ac.uk/pure/staff/cooper

- Coordinator of "Computability in Europe," and co-organiser of the CiE international conference series.
- Author of the book *Computability Theory*, CRC, 2004
- Editor of selected volumes in *London Math. Soc. Lect. Note Series*, ◦ *Computability, Enumerability, Unsolvability* (with T.A. Slaman and S.S. Wainer), 1996 ◦ *Sets and Proofs* (with J.K. Truss), 1999 ◦ *Models and Computability* (with J.K. Truss), 1999

Scientific interests: Computability theory and applications to science and the humanities, complexity theory, combinatorics, and graph theory

John N Crossley



School of Computer Science
and Software Engineering
Faculty of Information Technology
Monash University
Clayton, Victoria 3800
Australia

John.Crossley@infotech.monash.edu.au
www.csse.monash.edu.au/~jnc

- Author of the books ◦ *Constructive Order Types*, North-Holland, 1969 ◦ *What is Mathematical Logic?* (with C.J. Ash, C.J. Brickhill, J.C. Stillwell, N.H. Williams), OUP, 1972 Dover Pub. Inc., 1990 ◦ *Combinatorial Functors* (with A. Nerode), Springer, 1974 ◦ *The Emergence of Number*, World Scientific, 1987 ◦ *The Nine Chapters on the Mathematical Art. Companion and Commentary* (with K.-S. Shen and A.W.-C. Lun), OUP, 1999

Scientific interests: Combining logic with state, program extraction from proofs, history of mathematics before 1600.

Wilfrid A Hodges

School of Mathematical Sciences
Queen Mary University of London
Mile End Road
London E1 4NS
UK

w.hodges@qmul.ac.uk
www.maths.qmul.ac.uk/~wilfrid

- Honorary member of FoLLI, the *Association of Logic, Language and Information* (www.folli.org)
- Council Member of *Association for Symbolic Logic*
- Author of the books ○ *A Shorter Model Theory*, Cambridge Univ. Press, 1997 ○ *Model Theory*, Cambridge Univ. Press, 1993 ○ *Building Models by Games*, Cambridge Univ. Press, 1985 ○ *Logic*, Penguin Books, 2001
- Editor of ○ *Perspectives in Logic* (Managing Ed.) ○ *Logic and Its Applications* ○ *Logic Journal of the IGPL* ○ *Journal of Logic and Computation* ○ *Journal of Applied Logic*

Scientific interests: Model theory

Ralf Küsters

Institut für Informatik
und Praktische Mathematik
Christian-Albrechts-Universität zu Kiel
Olshausenstraße 40
24098 Kiel
Germany

kuesters@ti.informatik.uni-kiel.de
www.ti.informatik.uni-kiel.de/~kuesters

- Author of the book ○ *Non-Standard Inferences in Description Logics*, Lecture Notes in Computer Science, **2100**, Springer, 2001

Scientific interests: Cryptography and computer security (analysis of cryptographic protocols), Logics in computer science and artificial intelligence (description logics)

Lawrence S Moss

Department of Mathematics
Indiana University
831 East Third Street
Bloomington, IN 47405-7106
USA

lmoss@indiana.edu
math.indiana.edu/home/moss

- Director of the Indiana University *Program in Pure and Applied Logic* (www.indiana.edu/~iulg)
- Editor of ○ *Journal of Logic, Language, and Information* ○ *The Notre Dame Journal of Formal Logic* ○ *Research on Language and Computation* ○ *The Annals of Mathematics, Computing and Teleinformatics* ○ *Logical Methods in Computer Science* ○ *Logic and Logical Philosophy*

Scientific interests: Applied logic; the study of mathematical and conceptual tools for use in computer science, lignuistics, artificial intelligence

Albert Visser

Department of Philosophy
Universiteit Utrecht
Heidelberglaan 8
3584 CS Utrecht
The Netherlands

albert.visser@phil.uu.nl
www.phil.uu.nl/~albert/

- Executive Committee Member of *Association for Symbolic Logic*
- Member of Committee on Prizes and Awards of *Association for Symbolic Logic*
- Director of the Educational Institute on AI CKI, Utrecht University and Scientific director of the Research School on Logic OzsL
- Editor of ○ *Journal of Philosophical Logic* ○ *The Notre Dame Journal of Formal Logic*

Scientific interests: Provability logics, modal logics, arithmetical theories, dynamic semantics, philosophy of language

Content

Franz Baader and Ralf Küsters

Lev Beklemishev and Albert Visser

Johan van Benthem

S Barry Cooper

John N Crossley

[†] The endless cycle of death and rebirth to which life in the material world is bound. (OED)

Wilfrid Hodges

Lawrence S Moss

Nonstandard Inferences in Description Logics: The Story So Far

Franz Baader

Technischer Universität Dresden
Dresden, Germany

Ralf Küsters

Institut für Informatik
und Praktische Mathematik
Christian-Albrechts-Universität zu Kiel
Kiel, Germany

Description logics (DLs) are a successful family of logic-based knowledge representation formalisms that can be used to represent the terminological knowledge of an application domain in a structured and formally well-founded way. DL systems provide their users with inference procedures that allow to reason about the represented knowledge. Standard inference problems (such as the subsumption and the instance problem) are now well-understood.

Mathematical Problems from Applied Logics. New Logics for the XXIst Century.
Edited by Dov M Gabbay *et al.*/ International Mathematical Series, Springer, 2005

Their computational properties (such as decidability and complexity) have been investigated in detail, and modern DL systems are equipped with highly optimized implementations of these inference procedures, which—in spite of their high worst-case complexity—perform quite well in practice.

In applications of DL systems it has turned out that building and maintaining large DL knowledge bases can be further facilitated by procedures for other, nonstandard inference problem, such as computing the least common subsumer and the most specific concept, and rewriting and matching of concepts. While the research concerning these nonstandard inferences is not as mature as the one for the standard inferences, it has now reached a point where it makes sense to motivate these inferences within a uniform application framework, give an overview of the results obtained so far, describe the remaining open problems, and give perspectives for future research in this direction.

1. Introduction

Description logics (DLs) [**12**] are a family of knowledge representation languages which can be used to represent the terminological knowledge of an application domain in a structured and formally well-understood way. The name *description logics* is motivated by the fact that, on the one hand, the important notions of the domain are described by *concept descriptions*, i.e., expressions that are built from atomic concepts (unary predicates) and atomic roles (binary predicates) using the concept and role constructors provided by the particular DL. For example, the concept of “a man that is married to a doctor, and has only happy children” can be expressed using the concept description

$$\mathsf{Man} \sqcap \exists \mathsf{married}.\mathsf{Doctor} \sqcap \forall \mathsf{child}.\mathsf{Happy}.$$

On the other hand, DLs differ from their predecessors, such as semantic networks and frames [**84, 79**], in that they are equipped with a formal, *logic*-based semantics, which can, for example, be given by a translation into first-order predicate logic. For example, the above concept description can be translated into the following first-order formula (with one free variable x):

$$\begin{aligned}\mathsf{Man}(x) \wedge \exists y.(\mathsf{married}(x,y) \wedge \mathsf{Doctor}(y))\\ \wedge \forall y.(\mathsf{child}(x,y) \rightarrow \mathsf{Happy}(y)).\end{aligned}$$

In addition to the formalism for describing concepts, DLs usually also provide their users with means for describing individuals by stating to which concepts they belong and in which role relationships they participate. For example, the assertions

$$\mathsf{Man}(\mathsf{JOHN}), \quad \mathsf{child}(\mathsf{JOHN},\mathsf{MARY}), \quad \mathsf{Happy}(\mathsf{MARY})$$

state that the individual John has a child Mary, who is happy.

Knowledge representation systems based on description logics (DL systems or DL reasoners) [**95, 81**] provide their users with various inference capabilities that deduce implicit knowledge from the explicitly represented knowledge. Standard inference services are *subsumption* and *instance checking*. Subsumption allows the user to determine subconcept-superconcept relationships, and hence, compute a subconcept-superconcept hierarchy: C is subsumed by D if and only if all instances of C are also instances of D, i.e., the first description is always interpreted as a subset of the second description. Instance checking asks whether a given individual necessarily belongs to a given concept, i.e., whether this instance relationship logically follows from the descriptions of the concept and of the individual.

In order to ensure a reasonable and predictable behavior of a DL reasoner, these inference problems should at least be decidable for the DL employed by the reasoner, and preferably of low complexity. Consequently, the expressive power of the DL in question must be restricted in an appropriate way. If the imposed restrictions are too severe, however, then the important notions of the application domain can no longer be expressed. Investigating this trade-off between the expressivity of DLs and the complexity of their inference problems has been one of the most important issues of DL research in the 1990s. As a consequence of this research, the complexity of reasoning in various DLs of different expressive power is now well-investigated (see [**49**] for an overview of these complexity results). In addition, there are highly optimized implementations of reasoners for very expressive DLs [**61, 54, 62**], which—despite their high worst-case complexity—behave very well in practice [**60, 53**].

DLs have been applied in many domains, such as medical informatics, software engineering, configuration of technical systems, natural language processing, databases, and web-based information systems (see Part III of [**12**] for details on these and other applications). A recent success story is the use of DLs as ontology languages [**15, 16**] for the Semantic Web [**33**]. In particular, the W3C recommended ontology web language OWL [**64**] is based on an expressive description logic [**67, 66**].

Editors—such as OilEd [**32**] and the OWL plug-in of Protègè [**69**]—supporting the design of ontologies in various application domains usually allow their users to access a DL reasoner, which realizes the aforementioned *standard inferences* such as subsumption and instance checking. Reasoning is not only useful when working with "finished" ontologies, it can also support the ontology engineer while building an ontology, by pointing out inconsistencies and unwanted consequences. The ontology engineer can thus use reasoning to check whether the definition of a concept or the description of an individual makes sense.

However, these standard DL inferences—subsumption and instance checking—provide only little support for actually coming up with a first version of the definition of a concept. The non-standard inferences considered in this paper were introduced to overcome this deficit, by allowing the user to construct new knowledge from the existing one. Our own motivation for investigating these novel inferences comes from an application in chemical process engineering where a knowledge base has been built by different knowledge engineers over a rather long period of time [**87, 71, 80, 44, 35, 77, 94**].

The goal of this paper is

(i) to motivate nonstandard inferences by means of a simple application scenario,
(ii) to provide an overview of the results that have been obtained for nonstandard inferences so far, and
(iii) to explain the main techniques employed for solving these novel inference problems.

In order to be able to describe the latter in detail, the exposition of the techniques is mainly restricted to the DL $\mathcal{ALE}$. However, we also provide references to results for other DLs.

Structure of the paper

In Section 2, we introduce typical DL constructors and the most important standard inference problems. In addition, we give a brief review of the different approaches for solving these inference problems, and of their complexity in different DLs. In Section 3, we first motivate the need for nonstandard inferences in a typical application scenario, and then formally define the most important nonstandard inferences in description logics. Then, we briefly introduces the techniques used to solve these problems. Since these techniques depend on a syntactic characterization of the subsumption problem, Section 3 is followed by a section that describes such a characterization for the DL $\mathcal{ALE}$, which we use as a prototypical example (Section 4). The next four sections consider the four most important nonstandard inference problems: computing the *least common subsumer* and the *most specific concept*, *rewriting*, and *matching*. Related nonstandard inferences are briefly discussed in the respective sections as well. We explain the results on these four nonstandard inferences in $\mathcal{ALE}$ in detail, whereas results for other DLs are reviewed only briefly. Finally, Section 9 summarizes the results on nonstandard inferences obtained so far, and gives perspectives for further research.

2. Description Logics and Standard Inferences

In order to define concepts in a DL knowledge base, one starts with a set N_C of concept names (unary predicates) and a set N_R of role names (binary predicates), and defines more complex *concept descriptions* using the *concept constructors* provided by the concept description language of the particular system. In this paper, we consider the DL $\mathcal{ALCN}$ and some of its sublanguages. *Concept descriptions* of $\mathcal{ALCN}$ are built using the constructors shown in the first part of Table 1. In this table, r stands for a role name, n for a nonnegative integer, A for a concept name, and C, D for arbitrary concept descriptions.

A *concept definition* $A \equiv C$ (as shown in the second part of Table 1) assigns a concept name A to a complex description C. A finite set of such definitions is called a *TBox* if and only if it is unambiguous, i.e., each name has at most one definition. The concept names occurring on the left-hand side of a concept definition are called *defined* concepts, and the others *primitive*. In many cases, one restricts the attention to *acyclic* TBoxes, where the definition of a defined concept A cannot (directly or indirectly) refer to A itself.

A (concept or role) *assertion* is of the form shown in the last part of Table 1. Here, a, b belong to an additional set N_I of individual names. A finite set of such assertions is called an *ABox*.

The *sublanguages* of $\mathcal{ALCN}$ that will be considered in this paper are shown in Table 2. The first column explains the naming scheme for the members of the $\mathcal{AL}$-family.

The *semantics* of concept descriptions is defined in terms of an *interpretation* $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$. The domain $\Delta^\mathcal{I}$ of $\mathcal{I}$ is a non-empty set and the interpretation function $\cdot^\mathcal{I}$ maps each concept name $A \in N_C$ to a set $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$, each role name $r \in N_R$ to a binary relation $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$, and each individual name $a \in N_I$ to an element $a^\mathcal{I} \in \Delta^\mathcal{I}$. The extension of $\cdot^\mathcal{I}$ to arbitrary concept descriptions is inductively defined, as shown in the third column

Name	Syntax	Semantics
top-concept	$\top$	$\Delta^{\mathcal{I}}$
bottom-concept	$\bot$	$\emptyset$
negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
atomic negation	$\neg A$	$\Delta^{\mathcal{I}} \setminus A^{\mathcal{I}}$
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
disjunction	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
value restriction	$\forall r.C$	$\{x \in \Delta^{\mathcal{I}} \mid \forall y : (x,y) \in r^{\mathcal{I}} \to y \in C^{\mathcal{I}}\}$
existential restriction	$\exists r.C$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y : (x,y) \in r^{\mathcal{I}} \land y \in C^{\mathcal{I}}\}$
at-least restriction	$\geqslant n\, r$	$\{x \in \Delta^{\mathcal{I}} \mid \sharp\{y \mid (x,y) \in r^{\mathcal{I}}\} \geqslant n\}$
at-most restriction	$\leqslant n\, r$	$\{x \in \Delta^{\mathcal{I}} \mid \sharp\{y \mid (x,y) \in r^{\mathcal{I}}\} \leqslant n\}$
concept definition	$A \equiv C$	$A^{\mathcal{I}} = C^{\mathcal{I}}$
concept assertion	$C(a)$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
role assertion	$r(a,b))$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$

TABLE 1. Syntax and semantics of concept descriptions, definitions, and assertions

of Table 1. In the rows treating at-least and at-most number restrictions, $\sharp M$ denotes the cardinality of a set M.

The interpretation $\mathcal{I}$ is a *model* of the TBox $\mathcal{T}$ if it satisfies all its concept definitions, i.e., $A^{\mathcal{I}} = C^{\mathcal{I}}$ for all $A \equiv C$ in $\mathcal{T}$, and it is a *model* of the ABox $\mathcal{A}$ if it satisfies all its assertions, i.e., $a^{\mathcal{I}} \in C^{\mathcal{I}}$ for all concept assertions $C(a)$ in $\mathcal{A}$ and $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$ for all role assertions $r(a,b)$ in $\mathcal{A}$.

Based on this semantics, we can now formally introduce the standard inference problems in description logics.

Definition 2.1. Let $\mathcal{A}$ be an ABox, $\mathcal{T}$ a TBox, C, D concept descriptions, and a an individual name.

- C is *satisfiable* w.r.t. $\mathcal{T}$ if there is a model $\mathcal{I}$ of $\mathcal{T}$ such that $C^{\mathcal{I}} \neq \emptyset$.
- D *subsumes* C w.r.t. $\mathcal{T}$ ($C \sqsubseteq_{\mathcal{T}} D$) if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{T}$.

Symbol	Syntax	$\mathcal{ALC}$	$\mathcal{ALEN}$	$\mathcal{ALE}$	$\mathcal{ALN}$	$\mathcal{EL}$	$\mathcal{FL}_0$
$\mathcal{AL}$	$\top$	x	x	x	x	x	x
	$\bot$	x	x	x	x		
	$\sqcap$	x	x	x	x	x	x
	$\neg A$	x	x	x	x		
	$\forall r.C$	x	x	x	x		x
$\mathcal{C}$	$\neg C$	x					
$\mathcal{E}$	$\exists r.C$	x	x	x		x	
$\mathcal{U}$	$C \sqcup D$	x					
$\mathcal{N}$	$(\leqslant n\, r), (\geqslant n\, r)$		x		x		

TABLE 2. The relevant sublanguages of $\mathcal{ALCN}$

- $\mathcal{A}$ is *consistent* w.r.t. $\mathcal{T}$ if there is a model $\mathcal{I}$ of $\mathcal{T}$ that is also a model of $\mathcal{A}$.
- a is an *instance* of C in $\mathcal{A}$ w.r.t. $\mathcal{T}$ ($\mathcal{A}, \mathcal{T} \models C(a)$) if $a^{\mathcal{I}} \in C^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{T}$ and $\mathcal{A}$.

In case the TBox $\mathcal{T}$ is empty, we omit the appendage "w.r.t. $\emptyset$." In particular, we say that D *subsumes* C and write this as $C \sqsubseteq D$. Two concept descriptions are *equivalent* ($C \equiv D$) if they subsume each other (w.r.t. the empty TBox), i.e., if $C \sqsubseteq D$ and $D \sqsubseteq C$. We write $C \sqsubset D$ to express that $C \sqsubseteq D$ but $D \not\sqsubseteq C$.

If the DL under consideration allows for full negation ($\mathcal{C}$), then subsumption and satisfiability are interreducable, and the same is true for the instance and the consistence problem. In addition, satisfiability (subsumption) can always be reduced to ABox-consistency (instance checking). This follows from the following equivalences:

- $C \sqsubseteq_{\mathcal{T}} D$ if and only if $C \sqcap \neg D$ is unsatisfiable w.r.t. $\mathcal{T}$;
- C is unsatisfiable w.r.t. $\mathcal{T}$ if and only if $C \sqsubseteq_{\mathcal{T}} \bot$;
- $\mathcal{A}, \mathcal{T} \models C(a)$ if and only if $\mathcal{A} \cup \{\neg C(a)\}$ is inconsistent w.r.t. $\mathcal{T}$;

- $\mathcal{A}$ is inconsistent w.r.t. $\mathcal{T}$ if and only if $\mathcal{A}, \mathcal{T} \models \{\bot(a)\}$ where a is an arbitrary individual name;
- C is satisfiable w.r.t. $\mathcal{T}$ if and only if $\{C(a)\}$ is consistent where a is an arbitrary individual name;
- $C \sqsubseteq_{\mathcal{T}} D$ if and only if $\{C(a)\}, \mathcal{T} \models D(a)$ where a is an arbitrary individual name.

If the TBox $\mathcal{T}$ is acyclic, then reasoning w.r.t. $\mathcal{T}$ can be reduced to reasoning w.r.t. the empty TBox by expanding concept definitions, i.e., by replacing defined concept by their definitions until all defined concepts have been replaced. This can, however, result in an exponential blow-up of the problem [**82**].

Most of the early research on reasoning in DLs concentrated on the subsumption problem for concept descriptions (i.e., w.r.t. the empty TBox). For the DLs introduced above, the worst-case complexity of this problem is well-investigated. Subsumption in $\mathcal{ALCN}$, $\mathcal{ALC}$, and $\mathcal{ALEN}$ is PSPACE-complete, whereas subsumption in $\mathcal{ALE}$ is NP-complete. The subsumption problem for $\mathcal{ALN}$, $\mathcal{EL}$, and $\mathcal{FL}_0$ is polynomial (see [**49**] for references and additional complexity results for other DLs).

In the presence of an acyclic TBox, the complexity of subsumption may increase, but not in all cases. For example, subsumption w.r.t. an acyclic TBox in $\mathcal{FL}_0$ is coNP-complete [**82**], but it remains polynomial in $\mathcal{EL}$ [**8**] and PSPACE-complete in $\mathcal{ALCN}$ [**75**]. Cyclic TBoxes may increase the complexity of the subsumption problem even further (for example, for $\mathcal{FL}_0$ to PSPACE [**4, 68**]), but again not in all cases (for example, for $\mathcal{EL}$, subsumption w.r.t. cyclic TBoxes remains polynomial [**8**]).

In most cases, the complexity of the instance problem is the same as the complexity of the subsumption problem (for example, in $\mathcal{ALCN}$ [**57**] and $\mathcal{EL}$ [**7**]), but in some cases it may be harder (for example, in $\mathcal{ALE}$, where it is PSPACE-complete [**51**]).

The original KLone system [**40**] as well as its early successor systems (such as Back [**83**], KRep [**78**], and Loom [**76**]) employed so-called *structural subsumption algorithms*, which first normalize the concept descriptions, and then recursively compare the syntactic structure of the normalized descriptions. These algorithms are usually very efficient (polynomial), but they have the disadvantage that they are complete only for very inexpressive DLs, i.e., for more expressive DLs they cannot detect all the existing subsumption relationships. The DL $\mathcal{ALN}$ is an example of a DL where this structural approach yields a polynomial-time subsumption algorithm (see [**27**] for a sketch of such an algorithm and [**38**] for a detailed description of a structural subsumption algorithm for an extension of $\mathcal{ALN}$).

The syntactic characterization of subsumption in $\mathcal{EL}$ and $\mathcal{ALE}$ given in Section 4 can in principle also be used to obtain a structural subsumption algorithm for these DLs. It should be noted, however, that in the case of $\mathcal{ALE}$ the normalization phase is not polynomial. For $\mathcal{EL}$, the normalization phase is void, but a naive top-down structural comparison would not result in a deterministic polynomial-time algorithm. To obtain a polynomial subsumption algorithm, one must use a dynamic programming approach, i.e., work bottom-up. Overall, structural subsumption does not seem to be the right tool for solving standard inferences for expressive DLs. However, as we will see, structural subsumption plays an important role for solving nonstandard inferences.

For expressive DLs (in particular, DLs allowing for disjunction and/or negation), for which the structural approach does not lead to complete subsumption algorithms, *tableau algorithms* have turned out to be useful: they are complete and often behave quite well in practice. The first such algorithm was proposed by Schmidt-Schauß and Smolka [**89**] for the DL $\mathcal{ALC}$.[1] It quickly turned out that this approach for deciding subsumption can be extended to various other DLs [**59, 58, 13, 2, 55, 46, 11, 28, 65, 67, 29**] and also to other inference problems such as the instance problem [**56, 51, 57**]. Early on, DL researchers started to

[1] Actually, at that time the authors were not aware of the close connection between their rule-based algorithm working on constraint systems and tableau procedures for modal and first-order predicate logics.

call the algorithms obtained this way “tableau-based algorithms” since they observed that the original algorithm by Schmidt-Schauß and Smolka for $\mathcal{ALC}$, as well as subsequent algorithms for more expressive DLs, could be seen as specializations of the tableau calculus for first-order predicate logic (the main problem to solve was to find a specialization that always terminates, and thus yields a decision procedure).

After Schild [**88**] showed that $\mathcal{ALC}$ is a syntactic variant of multi-modal K, it turned out that the algorithm by Schmidt-Schauß and Smolka was actually a re-invention of the tableau algorithm for K known from modal logics [**34**].

The first DL systems employing tableau-based algorithms (Kris [**14**] and Crack [**45**]) demonstrated that (in spite of the high worst-case complexity of the underlying DL $\mathcal{ALCN}$) such algorithms can be implemented in a practical way. The complexity barrier has been pushed even further back by the seminal system FaCT [**61**]. Although FaCT employs the very expressive DL $\mathcal{SHIQ}$, which has an EXPTIME-complete subsumption problem, its highly optimized tableau-based subsumption algorithm outperforms the early systems based on structural subsumption algorithms and Kris by several orders of magnitude [**63**]. The equally well-performing system Racer [**54**] also provides for a highly-optimized implementation of the ABox-consistency and instance test for an extension of $\mathcal{SHIQ}$.

3. Nonstandard Inferences—Motivation and Definitions

In this section, we will first motivate the nonstandard inferences considered in this paper within a uniform application scenario, in which these inferences are used to support the design of DL knowledge bases. Then, we give formal definitions of the relevant nonstandard inferences, and briefly sketch different techniques for solving them. Each nonstandard inference will be considered in more detail in a separate section, where we concentrate on the DL $\mathcal{ALE}$.

3.1. Motivation

As mentioned in the introduction, the standard DL inferences introduced in Section 2 can already be employed during the design phase of a DL knowledge base since they allow the knowledge engineer to check whether the definition of a concept make senses (i.e., whether the defined concept is satisfiable) and whether it behaves as expected (i.e., whether the computed subsumption relationships are the ones intuitively expected).

However, inferences such as subsumption provide no support for actually coming up with a first version of the definition of a concept.

The nonstandard inferences introduced in this section can be used to overcome this deficit, basically by providing two ways of re-using "old" knowledge when defining new one:

(i) constructing concepts by generalizing from examples, and
(ii) constructing concepts by modifying "similar" ones.

The first approach was introduced as *bottom-up construction* of description logic knowledge bases in [**17, 22**]. Instead of defining the relevant concepts of an application domain from scratch, this methodology allows the user to give typical examples of individuals belonging to the concept to be defined. These individuals are then generalized to a concept by first computing the most specific concept (msc) of each individual (i.e., the least concept description in the available description language that has this individual as an instance), and then computing the least common subsumer (lcs) of these concepts (i.e., the least concept description in the available description language that subsumes all these concepts). The knowledge engineer can then use the computed concept as a starting point for the concept definition.

As a simple example, assume that the knowledge engineer has already defined the concept of a man and a woman as

$$\mathsf{Man} \equiv \mathsf{Human} \sqcap \mathsf{Male} \quad \text{and} \quad \mathsf{Woman} \equiv \mathsf{Human} \sqcap \mathsf{Female},$$

and now wants to define the concept of a parent, but does not know how to do this within the available DL (which we assume to be $\mathcal{EL}$ in this example). However, the available ABox

$$\begin{array}{lll} \mathsf{Man(JACK)}, & \mathsf{child(JACK, CAROLINE)}, & \mathsf{Woman(CAROLINE)}, \\ \mathsf{Woman(JACKIE)}, & \mathsf{child(JACKIE, JOHN)}, & \mathsf{Man(JOHN)} \end{array}$$

contains the individuals JACK and JACKIE, of whom the knowledge engineer knows that they are parents. The most specific concepts of JACK and JACKIE in the given ABox are

$$\mathsf{Man} \sqcap \exists \mathsf{child.Woman} \quad \text{and} \quad \mathsf{Woman} \sqcap \exists \mathsf{child.Man}$$

respectively and the least common subsumer (in $\mathcal{EL}$) of these two concepts w.r.t. the definitions of Man and Woman is

$$\mathsf{Human} \sqcap \exists \mathsf{child.Human},$$

which looks like a good starting point for a definition of parent.

In contrast to standard inferences such as subsumption and instance checking, the output of the nonstandard inferences we have mentioned until now (computing the msc and the lcs) is a concept description rather than a yes/no answer. In such a setting, it is important that the returned descriptions are as readable and comprehensible as possible. Unfortunately, the descriptions that are produced by the known algorithms for computing the lcs and the msc do not satisfy this requirement. The reason is that—like most algorithms for the standard inference problems—these algorithms work on expanded concept descriptions, i.e., concept descriptions that do not contain names defined in the underlying TBox. Consequently, the descriptions that the algorithms produce also do not use defined concepts, which makes them in many cases large and hard to read and comprehend. In the above example, this means that the definitions of Man and Woman are expanded before applying the lcs algorithm. If Human also had a definition, then it would also be expanded, and instead of the concept description containing Human shown above, the algorithm would return its expanded version.

This problem can be overcome by *rewriting* the resulting concept w.r.t. the given TBox. Informally, the problem of rewriting a concept given a terminology can be stated as follows: given an acyclic TBox $\mathcal{T}$ and a concept description C that does not contain concept names defined in $\mathcal{T}$, can this description be rewritten into an equivalent shorter description E by using (some of) the names defined in $\mathcal{T}$?

For example, w.r.t. the TBox

$$\begin{array}{rcl} \mathsf{Woman} & \equiv & \mathsf{Human} \sqcap \mathsf{Female}, \\ \mathsf{Man} & \equiv & \mathsf{Human} \sqcap \mathsf{Male}, \\ \mathsf{Parent} & \equiv & \mathsf{Human} \sqcap \exists \mathsf{child}.\mathsf{Human}, \end{array}$$

the concept description

$$\mathsf{Human} \sqcap \forall \mathsf{child}.\mathsf{Female} \sqcap \exists \mathsf{child}.\top \sqcap \forall \mathsf{child}.\mathsf{Human}$$

can be rewritten to the equivalent concept

$$\mathsf{Parent} \sqcap \forall \mathsf{child}.\mathsf{Woman}.$$

In order to apply the second approach of constructing concepts by modifying existing ones, one must first find the right candidates for modification. One way of doing this is to give a partial description of the concept to be defined as a concept pattern (i.e., a concept description containing variables), and then look for concept descriptions that match this pattern.

For example, the pattern

$$\mathsf{Man} \sqcap \exists \mathsf{child}.(\mathsf{Man} \sqcap X) \sqcap \exists \mathsf{spouse}.(\mathsf{Woman} \sqcap X)$$

looks for descriptions of classes of men whose wife and son share some characteristic. An example of a concept description *matching* this pattern is

$$\mathsf{Man} \sqcap \exists \mathsf{child}.(\mathsf{Man} \sqcap \mathsf{Tall}) \sqcap \exists \mathsf{spouse}.(\mathsf{Woman} \sqcap \mathsf{Tall}).$$

We refer the reader to [**71, 44, 24**] for a description of other possible applications of nonstandard inferences.

3.2. Definitions

In the following, we formally define the most important nonstandard inferences.

Least Common Subsumer

Intuitively, the least common subsumer of a given collection of concept descriptions is a description that represents the properties that all the elements of the collection have in common. More formally, it is the most specific concept description that subsumes the given descriptions. How this most specific description looks like, whether it really captures the intuition of representing the properties common to the input descriptions, and whether it exists at all strongly depends on the DL under consideration.

Definition 3.1. Let $\mathcal{L}$ be a DL. A concept description E of $\mathcal{L}$ is a *least common subsumer* (lcs) of the concept descriptions $C_1, \ldots, C_n$ in $\mathcal{L}$ ($lcs_{\mathcal{L}}(C_1, \ldots, C_n)$ for short) if and only if it satisfies

(1) $C_i \sqsubseteq E$ for all $i = 1, \ldots, n$, and

(2) E is the least $\mathcal{L}$ concept description with this property, i.e., if E' is an $\mathcal{L}$ concept description satisfying $C_i \sqsubseteq E'$ for all $i = 1, \ldots, n$, then $E \sqsubseteq E'$.

As an easy consequence of this definition, the lcs is unique up to equivalence, which justifies talking about *the* lcs. In addition, the n-ary lcs as defined above can be reduced to the binary lcs (the case $n = 2$ above). Indeed, it is easy to see that

$$lcs_{\mathcal{L}}(C_1, \ldots, C_n) \equiv lcs_{\mathcal{L}}(C_1, \ldots, lcs_{\mathcal{L}}(C_{n-1}, C_n) \cdots).$$

Thus, it is enough to devise algorithms for computing the binary lcs.

It should be noted, however, that the lcs does not always need to exist. This can have several reasons:

(a) there may not exist a concept description in $\mathcal{L}$ satisfying (1) of the definition (i.e., subsuming $C_1, \ldots, C_n$);
(b) there may be several subsumption incomparable minimal concept descriptions satisfying (1) of the definition;
(c) there may be an infinite chain of more and more specific descriptions satisfying (1) of the definition.

Obviously, (a) cannot occur for DLs containing the top concept. It is easy to see that, for DLs allowing for conjunction, (b) cannot occur. Case (c) is also rare to occur for DLs allowing for conjunction, but this is less obvious to see. Basically, for many DLs one can use the role depth of the concepts $C_1, \ldots, C_n$ to restrict the role depth of (relevant) common subsumers. The existence of the lcs then follows from the presence of conjunction and the fact that, up to equivalence, there are only finitely many concepts over a finite vocabulary having a fixed role depth (see [**30**] for more details). An example where case (c) actually occurs is the DL $\mathcal{EL}$ with cyclic terminologies interpreted with descriptive semantics [**5**] (see also Section 5.3).

It is clear that in DLs allowing for disjunction, the lcs of $C_1, \ldots, C_n$ is their disjunction $C_1 \sqcup \ldots \sqcup C_n$. In this case, the lcs is not of interest. In fact, as we have said above, the lcs is supposed to make explicit the properties that the input concepts have in common. This is, of course, not achieved by writing down their disjunction. Hence the lcs appears to be useful only in cases where the DL does not allow for disjunction.

Definition 3.1 is formulated for concept descriptions, i.e., it does not take a TBox into account. For acyclic TBoxes, this is not a real restriction since one can first expand the definitions before computing the lcs, and then apply rewriting to the lcs to obtain an equivalent shorter description containing defined concepts. For cyclic TBoxes, expansion is not possible. In addition, it may be advantageous to use cycles within the definition of the lcs, i.e., to allow the TBox to be extended by additional (possibly cyclic) concept definitions [**17, 7**]. The use of cyclic TBoxes in this context is also motivated by the most specific concept (see below).

Most Specific Concept

The most specific concept of a given ABox individual captures all the properties of the individual that are expressible by a concept description of the DL under consideration. Again, the form of the most specific concept and its existence strongly depend on this DL.

Definition 3.2. Let $\mathcal{L}$ be a DL. The $\mathcal{L}$ concept description E is the *most specific concept* (msc) in $\mathcal{L}$ of the individual a in the $\mathcal{L}$ ABox $\mathcal{A}$ ($msc_{\mathcal{L}}(a)$ for short) if and only if

(1) $\mathcal{A} \models E(a)$, and

(2) E is the least $\mathcal{L}$ concept description satisfying (i), i.e., if E' is an $\mathcal{L}$ concept description satisfying $\mathcal{A} \models E'(a)$, then $E \sqsubseteq E'$.

As with the lcs, the msc is unique up to equivalence, if it exists. In contrast to the lcs, which always exists for the DLs shown in Table 2, the msc does not always exist in these DLs. This is due to the presence of so-called role cycles in the ABox.

For example, w.r.t. the ABox

$$\{\mathsf{loves}(\mathsf{NARCIS}, \mathsf{NARCIS}),\ \mathsf{Vain}(\mathsf{NARCIS})\},$$

the individual NARCIS does not have an msc in $\mathcal{EL}$. In fact, assume that E is the msc of NARCIS. Then E has a finite role depth, i.e., a finite maximal number of nestings of existential restrictions. If this role depth is smaller than n, then E is not subsumed by the $\mathcal{EL}$ concept description

$$E' := \underbrace{\exists \mathsf{loves}.\cdots\exists \mathsf{loves}.}_{n \text{ times}}\mathsf{Vain},$$

in spite of the fact that NARCIS is an instance of E'.

One way to overcome this problem is to allow for cyclic TBoxes interpreted with greatest fixpoint semantics. In the above example, the defined concept

$$\mathsf{Narcis} \equiv \mathsf{Vain} \sqcap \exists \mathsf{loves}.\mathsf{Narcis}$$

is an msc of the individual NARCIS in $\mathcal{EL}$ w.r.t. cyclic TBoxes with greatest fixpoint semantics. In order to employ this approach in the bottom-up construction of DL knowledge bases, the impact of such cyclic definitions on the subsumption problem and the problem of computing the lcs must also be dealt with. In [**17**] this is done for $\mathcal{ALN}$, and in [**8, 7**] for $\mathcal{EL}$.

Another possibility is to approximate the msc by restricting the attention to concept descriptions whose role depth is bounded by a fixed number k [**48, 73**] (see Section 6 for details).

Rewriting

In [**23**], a very general framework for rewriting in DLs is introduced, which has several interesting instances. In order to introduce this framework, we fix a set N_R of role names and a set N_P of primitive concept names.

Definition 3.3. Let $\mathcal{L}_s$, $\mathcal{L}_d$, and $\mathcal{L}_t$ be three DLs (the source-, destination, and TBox-DL respectively). A *rewriting problem* is given by

- an $\mathcal{L}_t$ TBox $\mathcal{T}$ containing only role names from N_R and primitive concepts from N_P; the set of defined concepts occurring in $\mathcal{T}$ is denoted by N_D;
- an $\mathcal{L}_s$ concept description C using only the names from N_R and N_P;

- a binary relation ρ between $\mathcal{L}_s$ and $\mathcal{L}_d$ concept descriptions.

An $\mathcal{L}_d$ *rewriting of* C *using* $\mathcal{T}$ is an $\mathcal{L}_d$ concept description E built using role names from N_R and concept names from $N_P \cup N_D$ such that $C\rho E$.

Given an appropriate ordering $\preceq$ on $\mathcal{L}_d$ concept descriptions, a rewriting E is called $\preceq$*-minimal* if and only if there does not exist a rewriting E' such that $E' \prec E$.

In this paper, we consider one instances of this general framework in more detail, the minimal rewriting problem [**23**], and briefly discuss another instance, the approximation problem [**42**]. The *minimal rewriting problem* is the instance of the framework where

- all three DLs are the same language $\mathcal{L}$;
- the TBox $\mathcal{T}$ is acyclic;
- the binary relation ρ corresponds to equivalence modulo the TBox;
- $\mathcal{L}$ concept descriptions are ordered by size, i.e., $E \preceq E'$ if and only if $|E| \leqslant |E'|$, where the size $|E|$ of a concept description E is defined to be the number of occurrences of concept and role names in E.

The *approximation problem* is the instance of the framework where

- $\mathcal{T}$ is empty, and thus $\mathcal{L}_t$ is irrelevant;
- both ρ and $\preceq$ are the subsumption relation $\sqsubseteq$.

Given two DLs $\mathcal{L}_s$ and $\mathcal{L}_d$, an $\mathcal{L}_d$ *approximation* of an $\mathcal{L}_s$ concept description C is thus an $\mathcal{L}_d$ concept description D such that $C \sqsubseteq D$ and D is minimal (w.r.t. subsumption) in $\mathcal{L}_d$ with this property. Typically, $\mathcal{L}_d$ is a less expressive DL than $\mathcal{L}_s$, and hence, D is the best approximation from above of C in $\mathcal{L}_d$. One motivation for approximation is to be able to translate a knowledge base expressed in an expressive DL into a knowledge base expressed in a less expressive DL [**23, 42**].

Matching

Before we can define matching, we must define the notion of a pattern. *Concept patterns* are concept descriptions in which *concept variables* (usually denoted by X, Y, etc.) may occur in place of concept names. However, concept variables may not occur in the scope of a negation. The main difference between concept names and concept variables is that the latter can be replaced by concept descriptions when applying a substitution.

For example,

$$D := P \sqcap X \sqcap \forall r.(Y \sqcap \forall r.X)$$

is a concept pattern containing the concept variables X and Y. By applying the substitution $\sigma := \{X \mapsto Q,\ Y \mapsto \forall r.P\}$ to it, we obtain the concept description

$$\sigma(D) = P \sqcap Q \sqcap \forall r.(\forall r.P \sqcap \forall r.Q).$$

Definition 3.4. Let C be a concept description and D a concept pattern. Then $C \equiv^? D$ is called a *matching problem modulo equivalence* and $C \sqsubseteq^? D$ is called a *matching problem modulo subsumption.* The substitution σ is a *matcher* of the matching problem $C \equiv^? D$ ($C \sqsubseteq^? D$) if and only if $C \equiv \sigma(D)$ ($C \sqsubseteq \sigma(D)$).

Since $C \sqsubseteq \sigma(D)$ if and only if $C \sqcap \sigma(D) \equiv C$, the matching problem modulo subsumption $C \sqsubseteq^? D$ can be reduced to the following matching problem modulo equivalence: $C \equiv^? C \sqcap D$. However, in many cases, matching modulo subsumption is simpler than matching modulo equivalence since it can be reduced to the subsumption problem. If the DL contains $\top$ and all its constructors are monotonic, then $C \sqsubseteq^? D$ has a matcher if and only if the substitution $\sigma_\top$ that replaces all variables by $\top$ is a matcher, i.e., if $C \sqsubseteq \sigma_\top(D)$.

However, in the context of matching modulo subsumption, one is usually not interested in an arbitrary solution, but rather in certain “interesting” ones. One criterion for being interesting is that the matcher should bring D as close to C as possible, i.e., an interesting matcher σ of $C \sqsubseteq^? D$ should be minimal in that there does not exist another substitution δ such that $C \sqsubseteq \delta(D) \sqsubset \sigma(D)$ [**37**]. Other criteria for defining interesting matchers are discussed in Section 8.2.

In Section 8, we will briefly mention an extension of matching modulo equivalence, namely *unification*, where besides D also C may contain variables. Given a unification problem of the form $C \equiv^? D$, a substitution σ is a *unifier* of this problem if and only if $\sigma(C) \equiv \sigma(D)$.

3.3. Techniques

The approaches for solving nonstandard inferences in DLs developed so far are based on appropriate structural characterizations of the subsumption or the instance problem. Based on these characterizations, the nonstandard inferences can be characterized as well, and from these characterizations, approaches solving these inferences can be deduced.

In the literature, two different approaches for developing structural characterizations of subsumption have been considered: the language-based and the tree-based approach.

In the *language based approach*, one first computes a normal form that is based on finite or regular sets of words over the alphabet of role names, and then characterizes subsumption by inclusion relationships between these languages (see, for example, [**3, 70**]). In the *tree-based approach*, concept descriptions are turned into so-called description trees, and subsumption is then characterized via the existence of certain homomorphisms between these trees (see Section 4).

Since the tree-based approach to characterizing subsumption and solving nonstandard inferences will be considered in detail in the next sections, we briefly illustrate the language-based approach for the simple DL $\mathcal{FL}_0$ and the nonstandard inferences lcs and matching.

Using the equivalence

$$\forall r.(C \sqcap D) \equiv \forall r.C \sqcap \forall r.D$$

as a rewrite rule from left to right, any $\mathcal{FL}_0$ concept description can be transformed into an equivalent description that is a conjunction of descriptions of the form $\forall r_1.\cdots\forall r_m.A$ for $m \geqslant 0$ (not necessarily distinct) role names $r_1, \ldots, r_m$ and a concept name A. We abbreviate $\forall r_1.\cdots\forall r_m.A$ by $\forall r_1 \ldots r_m.A$, where $r_1 \ldots r_m$ is viewed as a word over the alphabet of all role names. In addition, instead of $\forall w_1.A \sqcap \ldots \sqcap \forall w_\ell.A$ we write $\forall L.A$ where $L := \{w_1, \ldots, w_\ell\}$ is a finite set of words over Σ. The term $\forall\emptyset.A$ is considered to be equivalent to the top concept $\top$, which means that it can be added to a conjunction without changing the meaning of the concept. Using these abbreviations, any pair of $\mathcal{FL}_0$ concept descriptions C, D containing the concept names $A_1, \ldots, A_k$ can be rewritten as

$$C \equiv \forall U_1.A_1 \sqcap \ldots \sqcap \forall U_k.A_k \quad \text{and} \quad D \equiv \forall V_1.A_1 \sqcap \ldots \sqcap \forall V_k.A_k,$$

where U_i, V_i are finite sets of words over the alphabet of all role names. This normal form provides us with the following *characterization of subsumption* of $\mathcal{FL}_0$ concept descriptions [**26**]:

$$C \sqsubseteq D \quad \text{iff} \quad U_i \supseteq V_i \ \text{ for all } i, 1 \leqslant i \leqslant k.$$

Since the size of the language-based normal forms is polynomial in the size of the original descriptions, and since the inclusion tests $U_i \supseteq V_i$ can also be realized in polynomial time, this yields a polynomial-time decision procedure for subsumption in $\mathcal{FL}_0$.

As an easy consequence of this characterization we find that the lcs E of C, D is of the form

$$E \equiv \forall(U_1 \cap V_1).A_1 \sqcap \ldots \sqcap \forall(U_k \cap V_k).A_k,$$

and thus can also be computed in polynomial time.

In order to treat matching in $\mathcal{FL}_0$ using the language-based approach, the language-based normal form of $\mathcal{FL}_0$ concept descriptions is extended in the obvious way to patterns. Let C be an $\mathcal{FL}_0$ concept description and D an $\mathcal{FL}_0$ concept pattern. We can write C, D in the form

$$C \equiv \forall S_{0,1}.A_1 \sqcap \ldots \sqcap \forall S_{0,k}.A_k,$$

$$D \equiv \forall T_{0,1}.A_1 \sqcap \ldots \sqcap \forall T_{0,k}.A_k \sqcap \forall T_1.X_1 \sqcap \ldots \sqcap \forall T_n.X_n,$$

where $A_1, \ldots, A_k$ are the concept names and $X_1, \ldots, X_n$ the concept variables occurring in C, D, and $S_{0,i}, T_{0,i}, T_j$ with $i = 1, \ldots, k$, $j = 1, \ldots, n$ are finite sets of words over the alphabet of all role names.

In [**26**] it is shown that the matching problem modulo equivalence $C \equiv^? D$ has a matcher if and only if for all $i = 1, \ldots, k$, the *linear language equation*

$$S_{0,i} = T_{0,i} \cup T_1 X_{1,i} \cup \cdots \cup T_n X_{n,i}$$

has a solution, i.e., we can substitute the variables $X_{j,i}$ by finite languages such that the equation holds.

Solvability of this linear language equation can be decided in polynomial time since it is sufficient to check whether the following substitution θ is a solution:

$$\theta(X_{j,i}) := \bigcap_{u \in T_j} u^{-1} S_{0,i},$$

where $u^{-1} S_{0,i} = \{v \mid uv \in S_{0,i}\}$.

We have used $\mathcal{FL}_0$ to sketch how the language based approach for characterizing subsumption can be used to solve nonstandard inferences. In the rest of this paper, we will concentrate on the tree based approach.

4. A Structural Characterization of Subsumption

As explained in the previous section, the basis for solving nonstandard inferences is an appropriate structural characterization of subsumption. In this section, we present the characterization for the DL $\mathcal{ALE}$ given in [**22**] in detail. Characterizations for other DLs are discussed only briefly.

The idea underlying the characterization of subsumption between $\mathcal{ALE}$ concept description is as follows. First, the concept descriptions are presented as edge- and node-labeled trees—called description trees—in which certain implicit facts have been made explicit. Then, we show that subsumption between $\mathcal{ALE}$ concept descriptions corresponds to the existence of homomorphisms between description trees.

As a warming-up, in Section 4.1, we first present the characterization of subsumption for the sublanguage $\mathcal{EL}$ of $\mathcal{ALE}$, with the extension to $\mathcal{ALE}$ presented in Section 4.2. We then briefly discuss characterizations of subsumption for extensions of $\mathcal{ALE}$ and other families of DLs (Section 4.3).

4.1. Getting started — The characterization for $\mathcal{EL}$

We first introduce $\mathcal{EL}$ description trees, and then present the characterization of subsumption.

Definition 4.1. *$\mathcal{EL}$ description trees* are of the form $\mathcal{G} = (V, E, v_0, \ell)$ where $\mathcal{G}$ is a tree with root v_0 whose edges $vrw \in E$ are labeled with role names $r \in N_R$, and whose nodes $v \in V$ are labeled with sets $\ell(v)$ of concept names from N_C. The empty label corresponds to the top-concept.

Intuitively, such a tree is merely a graphical representation of the syntax of the concept description. More formally, every $\mathcal{EL}$ concept description C can be written (modulo equivalence) as $C \equiv P_1 \sqcap \ldots \sqcap P_n \sqcap \exists r_1.C_1 \sqcap \ldots \sqcap \exists r_m.C_m$ with $P_i \in N_C \cup \{\top\}$. This description can now be translated into an $\mathcal{EL}$ description tree $\mathcal{G}_C = (V, E, v_0, \ell)$ as follows. The set of all concept names occurring in the top-level conjunction of C yields the label $\ell(v_0)$ of the root v_0,

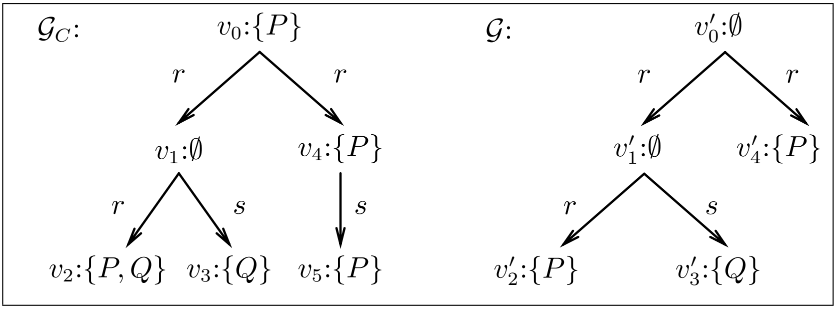


FIGURE 1. Two $\mathcal{EL}$ description trees

and each existential restriction $\exists r_i.C_i$ in this conjunction yields an r_i-successor that is the root of the tree corresponding to C_i. For example, the $\mathcal{EL}$ concept description

$$C := P \sqcap \exists r.(\exists r.(P \sqcap Q) \sqcap \exists s.Q) \sqcap \exists r.(P \sqcap \exists s.P)$$

yields the tree $\mathcal{G}_C$ depicted on the left-hand side of Figure 1.

Conversely, every $\mathcal{EL}$ description tree $\mathcal{G} = (V, E, v_0, \ell)$ can be translated into an $\mathcal{EL}$ concept description $C_\mathcal{G}$. Intuitively, the concept names in the label of v_0 yield the concept names in the top-level conjunction of $C_\mathcal{G}$, and each r-successor v of v_0 yields an existential restriction $\exists r.C$ where C is the $\mathcal{EL}$ concept description obtained by translating the subtree of $\mathcal{G}$ rooted at v. For a leaf $v \in V$, the empty label is translated into the top-concept. For example, the $\mathcal{EL}$ description tree $\mathcal{G}$ in Figure 1 yields the $\mathcal{EL}$ concept description

$$C_\mathcal{G} = \exists r.(\exists r.P \sqcap \exists s.Q) \sqcap \exists r.P.$$

These translations preserve the semantics of concept descriptions in the sense that $C \equiv C_{\mathcal{G}_C}$ holds for all $\mathcal{EL}$ concept descriptions C.

Definition 4.2. A *homomorphism* from an $\mathcal{EL}$ description tree $\mathcal{H} = (V_H, E_H, w_0, \ell_H)$ into an $\mathcal{EL}$ description tree $\mathcal{G} = (V_G, E_G, v_0, \ell_G)$ is a mapping $\varphi : V_H \longrightarrow V_G$ such that

(1) $\varphi(w_0) = v_0$,
(2) $\ell_H(v) \subseteq \ell_G(\varphi(v))$ for all $v \in V_H$, and
(3) $\varphi(v)r\varphi(w) \in E_G$ for all $vrw \in E_H$.

Subsumption in $\mathcal{EL}$ can be characterized in terms of homomorphisms between $\mathcal{EL}$ description trees.

Theorem 4.3. *Let C, D be $\mathcal{EL}$ concept descriptions and $\mathcal{G}_C, \mathcal{G}_D$ be the corresponding $\mathcal{EL}$ description trees. Then $C \sqsubseteq D$ if and only if there exists a homomorphism from $\mathcal{G}_D$ into $\mathcal{G}_C$.*

In our example (see Figure 1), the $\mathcal{EL}$ concept description $C_\mathcal{G}$ subsumes C. Indeed, mapping v_i' to v_i for all $0 \leqslant i \leqslant 4$ yields a homomorphism from $\mathcal{G} = \mathcal{G}_{C_\mathcal{G}}$ to $\mathcal{G}_C$.

Theorem 4.3 may look like a special case of the characterization of subsumption between simple conceptual graphs [**47**], and of the characterization of containment of conjunctive queries [**1**]. In the more general setting of simple conceptual graphs and conjunctive queries, one considers homomorphisms between *graphs*, and thus testing for the existence of a homomorphism is an NP-complete problem [**52**]. If one restricts the attention to graphs that are *trees*, then testing for the existence of a homomorphism can be realized in polynomial time using dynamic programming techniques [**86**]. Thus, Theorem 4.3 implies that subsumption between $\mathcal{EL}$ concept descriptions is a tractable problem, as already mentioned in Section 2. The fact that both subsumption in $\mathcal{EL}$ and subsumption of conceptual graphs (containment of conjunctive queries) corresponds to the existence of homomorphisms suggests a stronger connection between these problems than is actually the case. In fact, the nodes in conceptual graphs (the variables in conjunctive queries) stand for individuals whereas the nodes of $\mathcal{EL}$ description trees stand for concepts (i.e., sets of individuals). This semantic difference becomes relevant if one considers cyclic $\mathcal{EL}$ TBoxes, which can be represented by description graphs. In this case, however, subsumption no longer corresponds to the existence of a homomorphism, but to the existence of a so-called simulation relation [**8**]. Whereas the existence of a homomorphism is an NP-complete problem, the existence of a simulation relation can still be checked in polynomial time. It is only for trees that both problems are identical, i.e., on trees the existence of a simulation relation implies the existence of a homomorphism and vice versa, whereas this does not hold on general graphs.

4.2. Extending the characterization to $\mathcal{ALE}$

To obtain a characterization of subsumption for $\mathcal{ALE}$, we must first extend $\mathcal{EL}$ description trees to $\mathcal{ALE}$ description trees. Since $\mathcal{ALE}$ concept descriptions may contain value restrictions in addition to existential restrictions, $\mathcal{ALE}$ description trees have two types of edges, namely those labeled with a role name $r \in N_R$, which correspond to existential restrictions of the form $\exists r.C$, and those labeled with $\forall r$, which correspond to value restrictions of the form $\forall r.C$. Also, we have to allow negated concept names $\neg P$ and the bottom concept $\bot$ in the labels of nodes, in addition to concept names $P \in N_C$. As in the case of $\mathcal{EL}$, there is a one-to-one correspondence between $\mathcal{ALE}$ concept descriptions and $\mathcal{ALE}$ description trees.

It might be tempting to think that the notion of a homomorphism can also be extended in such a straightforward way to $\mathcal{ALE}$ description trees as well by just adding the following requirement to Definition 4.2:

4. $\varphi(v)\,\forall r\,\varphi(w) \in E_G$ *for all* $v\,\forall r\,w \in E_H$.

Now, using this notion of a homomorphism between $\mathcal{ALE}$ description trees, one could try to characterize subsumption as before. However, this fails for several reasons.

First, we need to take into account implicit facts that are implied by interactions among value restrictions and among value restrictions and existential restrictions. Consider, for instance,

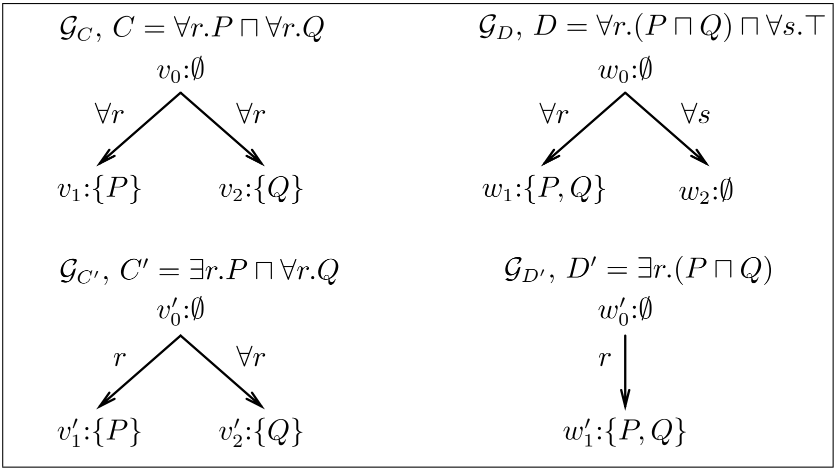


FIGURE 2. Examples illustrating that implicit facts induced by value and existential restrictions must be taken into account

the $\mathcal{ALE}$ concept descriptions and their translations into $\mathcal{ALE}$ description trees depicted in Figure 2. It is easy to see that $C \sqsubseteq D$ and $C' \sqsubseteq D'$, but that there exist neither a homomorphism from $\mathcal{G}_D$ to $\mathcal{G}_C$ nor one from $\mathcal{G}_{D'}$ to $\mathcal{G}_{C'}$. The problem is that C and D are actually equivalent to $\forall r.(P \sqcap Q)$ and that C' is equivalent to $\exists r.(P \sqcap Q) \sqcap \forall r.Q$, but that this is not reflected in the description trees.

To make such implicit facts explicit, we have to normalize the $\mathcal{ALE}$ concept descriptions before translating them into $\mathcal{ALE}$ description trees. For this purpose, the following *normalization rules* are exhaustively applied to the given $\mathcal{ALE}$ concept descriptions:

$$\begin{array}{rcl}
\forall r.E \sqcap \forall r.F & \longrightarrow & \forall r.(E \sqcap F), \\
\forall r.E \sqcap \exists r.F & \longrightarrow & \forall r.E \sqcap \exists r.(E \sqcap F), \\
\forall r.\top & \longrightarrow & \top, \\
E \sqcap \top & \longrightarrow & E.
\end{array}$$

Since each normalization rule preserves equivalence, the resulting $\mathcal{ALE}$ concept descriptions are equivalent to the original ones. The rules should be read modulo associativity and commutativity of conjunction. For instance, $\exists r.E \sqcap \forall r.F$ is also turned into $\exists r.(E \sqcap F) \sqcap \forall r.F$.

The above normalization rules are, however, not yet sufficient to make all implicit facts explicit. This is due to the fact that an $\mathcal{ALE}$ concept description may contain unsatisfiable subdescriptions. In addition to the above normalization rules, we need three more rules to handle this:

$$\begin{array}{rcll} P \sqcap \neg P & \longrightarrow & \bot & \text{for each } P \in N_C, \\ \exists r.\bot & \longrightarrow & \bot, & \\ E \sqcap \bot & \longrightarrow & \bot. & \end{array}$$

Starting with an $\mathcal{ALE}$ concept description C, the exhaustive application of (both groups of) rules yields an equivalent $\mathcal{ALE}$ concept description in *normal form*. Given such a normal form, the corresponding $\mathcal{ALE}$ description tree is obtain as in the case of $\mathcal{EL}$, with the obvious adaptations due to the existence of two different kinds of edges and the fact that the label of a node may be $\bot$. We refer to the *$\mathcal{ALE}$ description tree corresponding to the normal form* of C as $\mathcal{G}_C$.

Unfortunately, even after normalization, the straightforward adaptation of the notion of a homomorphism from $\mathcal{EL}$ description trees to $\mathcal{ALE}$ description trees sketched above does not yield a sound and complete characterization of subsumption in $\mathcal{ALE}$. As an example, consider the following $\mathcal{ALE}$ concept descriptions:

$$\begin{array}{rcl} C & := & (\forall r.\exists r.(P \sqcap \neg P)) \sqcap (\exists s.(P \sqcap \exists r.Q)), \\ D & := & (\forall r.(\exists r.P \sqcap \exists r.\neg P)) \sqcap (\exists s.\exists r.Q). \end{array}$$

The description D is already in normal form, and the normal form of C is

$$C' := \forall r.\bot \sqcap \exists s.(P \sqcap \exists r.Q).$$

The corresponding $\mathcal{ALE}$ description trees $\mathcal{G}_C$ and $\mathcal{G}_D$ are depicted in Figure 3.

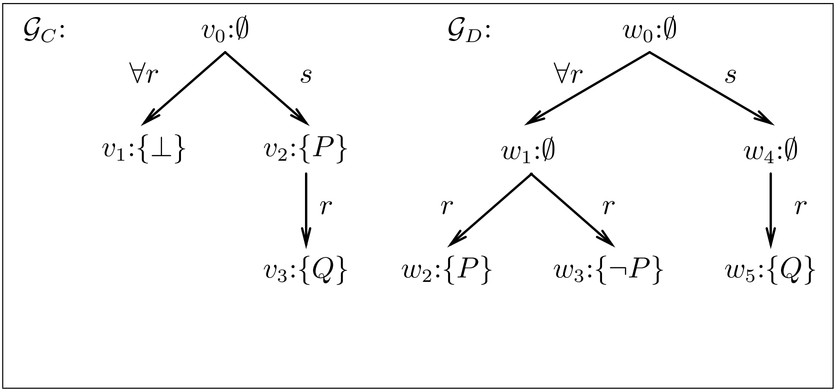


Figure 3. Example illustrating that the notion of a homomorphism must be adapted

It is easy to see that there does not exist a homomorphism in the above sense from $\mathcal{G}_D$ into $\mathcal{G}_C$, although we have $C \sqsubseteq D$. In particular, the $\mathcal{ALE}$ concept description $\exists r.P \sqcap \exists r.\neg P$ corresponding to the subtree with root w_1 of $\mathcal{G}_D$ subsumes $\bot$, which is the concept description corresponding to the subtree with root v_1 in $\mathcal{G}_C$. Therefore, a homomorphism from $\mathcal{G}_D$ into $\mathcal{G}_C$ should be allowed to map the whole tree corresponding to $\exists r.P \sqcap \exists r.\neg P$, i.e., the nodes w_1, w_2, w_3, onto the tree corresponding to $\bot$, i.e., onto v_1.

The example suggests the following new notion of a homomorphism on $\mathcal{ALE}$ description trees.

Definition 4.4. A *homomorphism* from an $\mathcal{ALE}$ description tree $\mathcal{H} = (V_H, E_H, w_0, \ell_H)$ into an $\mathcal{ALE}$ description tree $\mathcal{G} = (V_G, E_G, v_0, \ell_G)$ is a mapping $\varphi : V_H \longrightarrow V_G$ such that

(1) $\varphi(w_0) = v_0$,
(2) $\ell_H(v) \subseteq \ell_G(\varphi(v))$ or $\ell_G(\varphi(v)) = \{\bot\}$ for all $v \in V_H$,
(3) for all $vrw \in E_H$, either $\varphi(v)r\varphi(w) \in E_G$, or $\varphi(v) = \varphi(w)$ and $\ell_G(\varphi(v)) = \{\bot\}$, and

(4) for all $v\forall rw \in E_H$, either $\varphi(v)\forall r\varphi(w) \in E_G$, or $\varphi(v) = \varphi(w)$ and $\ell_G(\varphi(v)) = \{\bot\}$.

In Figure 3, if we map w_0 onto v_0; w_1, w_2, and w_3 onto v_1; w_4 onto v_2; and w_5 onto v_3, then the above conditions are satisfied, i.e., this mapping yields a homomorphism from $\mathcal{G}_D$ into $\mathcal{G}_C$. With this new notion of a homomorphism between $\mathcal{ALE}$ description trees, we can characterize subsumption in $\mathcal{ALE}$ in a sound and complete way.

Theorem 4.5. *Let C, D be two $\mathcal{ALE}$ concept descriptions and $\mathcal{G}_C$, $\mathcal{G}_D$ the corresponding $\mathcal{ALE}$ description trees. Then $C \sqsubseteq D$ if and only if there exists a homomorphism from $\mathcal{G}_D$ into $\mathcal{G}_C$.*

It should be noted that there is a close relationship between the normalization rules introduced above and some of the rules employed by tableaux-based subsumption algorithms, as e.g. introduced in [**50**]. As shown in [**50**], the propagation of value restrictions on existential restrictions may lead to an exponential blow-up (see the concept descriptions C_n introduced below Theorem 5.5). Consequently, the size of the normal forms, and thus also of the description trees, may grow exponentially in the size of the original $\mathcal{ALE}$ concept descriptions. It is easy to see that this exponential blow-up cannot be avoided: On the one hand, as for $\mathcal{EL}$, the existence of a homomorphism between $\mathcal{ALE}$ description trees can still be tested in polynomial time. On the other hand, subsumption in $\mathcal{ALE}$ is an NP-complete problem [**50**].

4.3. Characterization of subsumption for other DLs

The characterization of subsumption for $\mathcal{ALE}$ has been extended to $\mathcal{ALEN}$ in [**74**]. There, description trees are not used explicitly. Subsumption is rather characterized directly for the normalized concept descriptions, by using induction on the role depth of the descriptions.

For the sublanguage $\mathcal{ALNS}$ of the DL employed by the CLASSIC system [**36**], which extends $\mathcal{ALN}$ by the so-called same-as operator, subsumption has been characterized in [**72**]. Due to the presence of the same-as operator in $\mathcal{ALNS}$, description graphs instead of description trees are used in this characterization.

For DLs with cyclic TBoxes, subsumption has been characterized for $\mathcal{FL}_0$ [**4**], $\mathcal{ALN}$ [**70**], and $\mathcal{EL}$ [**8**] w.r.t. the three types of semantics employed for cyclic TBoxes: descriptive semantics, and greatest and least fixed point semantics. For $\mathcal{FL}_0$ and $\mathcal{ALN}$, subsumption has been characterized using the language-based approach (see Section 3.3). For $\mathcal{EL}$, the characterization extends the approach for $\mathcal{EL}$ concept descriptions presented in Section 4.1. However, instead of homomorphisms between description trees, simulation relationships on description graphs are employed.

5. The Least Common Subsumer

In this section, we study the existence of the lcs and how it can be computed (if it exists). Our exposition again concentrates on $\mathcal{ALE}$. It is based on the results shown in [**22**]. In addition, we briefly mention results for other DLs.

As we will see, once the structural characterization of subsumption is in place, it is rather easy to derive algorithms for computing the lcs. As a warming up exercise, in the following subsection, we present an lcs algorithm for $\mathcal{EL}$. Its extension to $\mathcal{ALE}$ is described in Section 5.2. An overview of results for other DLs is provided in Section 5.3.

5.1. The LCS for $\mathcal{EL}$

The characterization of subsumption by homomorphisms allows us to characterize the lcs by the product of $\mathcal{EL}$ description trees.

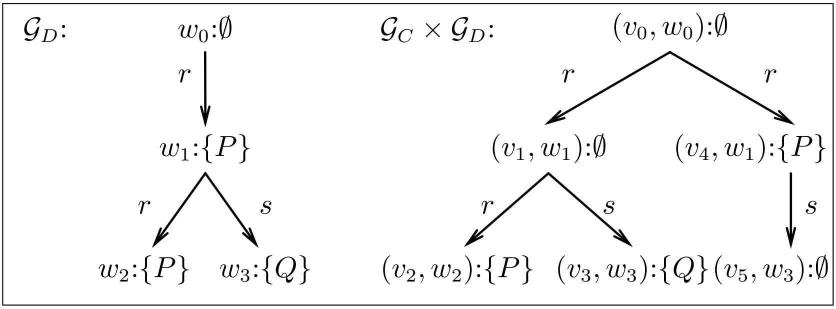


Figure 4. The product of $\mathcal{EL}$ description trees

Definition 5.1. The *product* $\mathcal{G} \times \mathcal{H}$ of two $\mathcal{EL}$ description trees $\mathcal{G} = (V_G, E_G, v_0, \ell_G)$ and $\mathcal{H} = (V_H, E_H, w_0, \ell_H)$ is defined inductively on the depth of the trees. Let $\mathcal{G}(v)$ denote the subtree of $\mathcal{G}$ rooted at v. We define (v_0, w_0) to be the root of $\mathcal{G} \times \mathcal{H}$, labeled with $\ell_G(v_0) \cap \ell_H(w_0)$. For each r-successor v of v_0 in $\mathcal{G}$ and w of w_0 in $\mathcal{H}$, we obtain an r-successor (v, w) of (v_0, w_0) in $\mathcal{G} \times \mathcal{H}$ that is the root of the product of $\mathcal{G}(v)$ and $\mathcal{H}(w)$.

For example, consider the $\mathcal{EL}$ description tree $\mathcal{G}_C$ (Figure 1) and the $\mathcal{EL}$ description tree $\mathcal{G}_D$ (Figure 4), where $\mathcal{G}_D$ corresponds to the $\mathcal{EL}$ concept description $D := \exists r.(P \sqcap \exists r.P \sqcap \exists s.Q)$. The product $\mathcal{G}_C \times \mathcal{G}_D$ is depicted on the right-hand side of Figure 4.

Theorem 5.2. *Let C, D be two $\mathcal{EL}$ concept descriptions and $\mathcal{G}_C$, $\mathcal{G}_D$ the corresponding $\mathcal{EL}$ description trees. Then $C_{\mathcal{G}_C \times \mathcal{G}_D}$ is the lcs of C and D. In particular, the lcs of $\mathcal{EL}$ concept descriptions always exists.*

In our example, we thus find that the lcs of $C = P \sqcap \exists r.(\exists r.(P \sqcap Q) \sqcap \exists s.Q) \sqcap \exists r.(P \sqcap \exists s.P)$ and $D = \exists r.(P \sqcap \exists r.P \sqcap \exists s.Q)$ is

$$r.(\exists r.P \sqcap \exists s.Q) \sqcap \exists r.(P \sqcap \exists s.\top).$$

The size of the lcs of two $\mathcal{EL}$ concept descriptions C, D can be bounded by the size of $\mathcal{G}_C \times \mathcal{G}_D$, which is polynomial in the size of $\mathcal{G}_C$ and $\mathcal{G}_D$. Since the size of the description tree corresponding to a given description is linear in the size of the description, we obtain:

Proposition 5.3. *The size of the lcs of two $\mathcal{EL}$ concept descriptions C, D is polynomial in the size of C and D, and the lcs can be computed in polynomial time.*

In many applications, however, one is interested in the lcs of $n > 2$ concept descriptions $C_1, \ldots, C_n$. This lcs can be obtained from the n-ary product $\mathcal{G}_{C_1} \times \cdots \times \mathcal{G}_{C_n}$ of their corresponding $\mathcal{EL}$ description trees. Therefore, the size of the lcs can be bounded by the size of this product. It is not hard to show that in general this size cannot be polynomially bounded [**22, 31**].

Proposition 5.4. *The size of the lcs of n $\mathcal{EL}$ concept descriptions $C_1, \ldots, C_n$ of size linear in n may grow exponentially in n.*

5.2. The LCS for $\mathcal{ALE}$

Just as for $\mathcal{EL}$, the lcs of $\mathcal{ALE}$ concept descriptions can be obtain as the product of the corresponding $\mathcal{ALE}$ description trees. However, the definition of the product must be adapted to the modified notion of a homomorphism. In particular, this definition must treat leaves with label $\{\bot\}$ in a special manner. Such a leaf corresponds to the bottom-concept, and since $\bot \sqsubseteq C$ for all $\mathcal{ALE}$ concept descriptions C, we have $lcs(\bot, C) \equiv C$. Thus, our product operation should be defined such that $C_{\mathcal{G}_\bot \times \mathcal{G}_C} \equiv C$.

More precisely, the *product $\mathcal{G} \times \mathcal{H}$ of two $\mathcal{ALE}$ description trees* $\mathcal{G} = (V_G, E_G, v_0, \ell_G)$ and $\mathcal{H} = (V_H, E_H, w_0, \ell_H)$ is defined as follow s. If $\ell_G(v_0) = \{\bot\}$ ($\ell_H(w_0) = \{\bot\}$), then we define $\mathcal{G} \times \mathcal{H}$ by replacing each node w in $\mathcal{H}$ (v in $\mathcal{G}$) by (v_0, w) ((v, w_0)). Otherwise, we define $\mathcal{G} \times \mathcal{H}$ by induction on the depth of the trees analogously to the definition of the product of $\leq$ description trees.

For the $\mathcal{ALE}$ description trees depicted in Figure 3, $\mathcal{G}_C \times \mathcal{G}_D$ is obtained from $\mathcal{G}_D$ by replacing w_0 by (v_0, w_0), w_i by (v_1, w_i) for $i = 1, 2, 3$, w_4 by (v_2, w_4), and w_5 by (v_3, w_5).[2]

Theorem 5.5. *Let C, D be two $\mathcal{ALE}$ concept descriptions and $\mathcal{G}_C$, $\mathcal{G}_D$ their corresponding $\mathcal{ALE}$ description trees. Then $C_{\mathcal{G}_C \times \mathcal{G}_D}$ is the lcs of C and D. In particular, the lcs of $\mathcal{ALE}$ concept descriptions always exists.*

Unlike $\mathcal{EL}$, the size of the lcs of two $\mathcal{ALE}$ concept descriptions may already grow exponentially. To see this, consider the following example. Let C_n, $n \geqslant 1$, be defined inductively as

$$C_1 := \exists r.P \sqcap \exists r.Q \text{ and } C_n := \exists r.P \sqcap \exists r.Q \sqcap \forall r.C_{n-1}$$

and let D_n, $n \geqslant 1$, be defined as

$$D_1 := \exists r.(P \sqcap Q) \text{ and } D_n := \exists r.(P \sqcap Q \sqcap D_{n-1}).$$

Note that the size of the normal form of C_n grows exponentially in n. It is easy to verify that the lcs of C_n and D_n is equivalent to the concept description E_n where

$$E_1 := \exists r.P \sqcap \exists r.Q \text{ and } E_n := \exists r.(P \sqcap E_{n-1}) \sqcap \exists r.(Q \sqcap E_{n-1}).$$

The size of E_n grows exponentially in n. It is not hard to check that there does not exist a smaller concept description equivalent to the lcs of C_n and D_n. Hence we obtain:

Proposition 5.6. *The size of the lcs of two $\mathcal{ALE}$ concept descriptions C, D may be exponential in the size of C, D.*

The above example suggests that, by employing structure sharing, the size of the lcs can be reduced. However, in general this is not the case. More specifically, it was shown in [**31**] that even if equivalent sub-concept descriptions of the lcs can be represented as defined concepts in an acyclic TBox, the representation of the lcs may still grow exponentially.

[2] Note that this is a somewhat atypical example since in this case C is subsumed by D, and thus the lcs is equivalent to D.

5.3. The LCS for other DLs

Based on the structural characterization of subsumption for the DLs mentioned in Section 4.3, algorithms for computing the lcs have been employed in a similar manner as illustrated above [**74, 72, 17, 7, 5**]. Interestingly, for the DL $\mathcal{ALNS}$ it has turned out that the existence of the lcs depends on whether features, i.e., roles that are restricted to be functional, are interpreted as partial or total functions. While in the former case, the lcs always exists, this is not true in the latter case [**72**]. As mentioned above, for the DL $\mathcal{EL}$ with cyclic TBoxes interpreted with descriptive semantics the lcs also does not need to exist [**5**].

6. The Most Specific Concept

As illustrated in Section 3.2, most specific concepts need not exist for DLs with number restrictions or existential restrictions. There are two ways to overcome this problem. First, the languages can be extended to allow for cyclic TBoxes interpreted with the greatest fixed point semantics. Second, one can resort to approximating the most specific concept. In the following subsection, we consider the latter approach in more detail, mainly concentrating on the simple DL $\mathcal{EL}$. Besides introducing methods for computing approximations, we will also characterize the existence of the msc. In Section 6.2, we will summarize results obtained following the first approach.

6.1. Existence and approximation of the MSC

The example presented in Section 3.2 illustrates that describing an msc may require a concept description with infinite role depth. Such a concept description can be approximated by restricting the role depth to a fixed constant k. This leads to the notion of a k-approximation. In Section 3.3 we have pointed out that the basis for solving nonstandard inferences is an appropriate characterization of the underlying standard inference. For the lcs, this standard inference is the subsumption problem. In order characterize the msc and to design algorithms for computing (approximations of) it, an appropriate characterization of the instance problem is needed.

After defining k-approximations in Section 6.1.1, we first present a characterization of the instance problem in Section 6.1.2 and then, in Section 6.1.3, apply this characterization to compute k-approximations. All this is done for the simple case that the DL is $\mathcal{EL}$. Extensions to more expressive DLs are discussed in Section 6.1.4. The results presented in this section are mainly based on [**73**].

6.1.1. ***Defining k-Approximations.*** To give a formal definition of k-approximations of the msc, we first need to define the role depth of concept descriptions. The role depth $\mathit{depth}(C)$ of an $\mathcal{EL}$ concept description C is defined as the maximum number of nested quantifiers in C:

$$\begin{array}{rcl} \mathit{depth}(\top) & = & \mathit{depth}(P) \;=\; 0, \\ \mathit{depth}(C \sqcap D) & = & \max(\mathit{depth}(C), \mathit{depth}(D)), \\ \mathit{depth}(\exists r.C) & = & \mathit{depth}(C) + 1. \end{array}$$

Definition 6.1. Let $\mathcal{A}$ be an $\mathcal{EL}$-ABox, b an individual in $\mathcal{A}$, C an $\mathcal{EL}$ concept descriptions, and $k \in \mathbb{N}$. Then, C is a *k-approximation* of b w.r.t. $\mathcal{A}$ ($\mathit{msc}^k_{\mathcal{EL}}(b)$) if and only if

(1) $\mathcal{A} \models C(b)$,
(2) $\mathit{depth}(C) \leqslant k$, and
(3) $C \sqsubseteq C'$ for all $\mathcal{EL}$ concept descriptions C' with $\mathcal{A} \models C'(b)$ and $\mathit{depth}(C') \leqslant k$.

It is an easy consequence of this definition that k-approximations are unique up to equivalence (if they exist). Thus, we can talk about *the* k-approximation of a given individual.

The k-approximation of the individual Narcis in the example presented in Section 3.2 is the $\mathcal{EL}$ concept description

$$\underbrace{\exists\mathsf{loves}.\ldots.\exists\mathsf{loves}}_{k\text{ times}}.\mathsf{Vain}.$$

6.1.2. ***Characterizing the instance problem in $\mathcal{EL}$.*** In order to characterize instance relationships, we need to introduce description graphs (representing ABoxes) as a generalization of description trees (representing concept descriptions). An $\mathcal{EL}$ *description graph* is a labeled graph of the form $\mathcal{G} = (V, E, \ell)$ whose edges $vrw \in E$ are labeled with role names $r \in N_R$ and whose nodes $v \in V$ are labeled with sets $\ell(v)$ of concept names from N_C. The empty label corresponds to the top-concept.

Similarly to the translation of $\mathcal{EL}$ concept descriptions into $\mathcal{EL}$ description trees, an $\mathcal{EL}$-ABox $\mathcal{A}$ is translated into an $\mathcal{EL}$ description graph $\mathcal{G}(\mathcal{A})$ in the following way. Let $\mathit{Ind}(\mathcal{A})$ denote the set of all individuals occurring in $\mathcal{A}$. For each $a \in \mathit{Ind}(\mathcal{A})$, let

$$C_a = \begin{cases} \bigsqcap\limits_{D(a)\in\mathcal{A}} D & \text{if there exists a concept assertion of the form} \\ & \quad D(a) \in \mathcal{A}, \\ \top & \text{otherwise.} \end{cases}$$

Let $\mathcal{G}_{C_a} = (V_a, E_a, a, \ell_a)$ denote the $\mathcal{EL}$ description tree obtained from C_a.[3] Without loss of generality we assume that the sets V_a for $a \in \mathit{Ind}(\mathcal{A})$ are pairwise disjoint. Then, $\mathcal{G}(\mathcal{A}) = (V, E, \ell)$ is defined as

- $V = \bigcup_{a\in \mathit{Ind}(\mathcal{A})} V_a$,
- $E = \{arb \mid r(a, b) \in \mathcal{A}\} \cup \bigcup_{a\in \mathit{Ind}(\mathcal{A})} E_a$, and

[3] Note that the individual a is defined to be the root of $\mathcal{G}(C_a)$; in particular, this means that $a \in V_a$.

- $\ell(v) = \ell_a(v)$ for all $v \in V_a$.

As an example, consider the $\mathcal{EL}$ ABox

$$\begin{array}{rcl}\mathcal{A} & = & \{(P \sqcap \exists s.(Q \sqcap \exists r.P \sqcap \exists s.\top))(a), \quad (P \sqcap Q)(b), \quad (\exists r.P)(c), \\ & & r(a,b), \quad r(a,c), \quad s(b,c)\}.\end{array}$$

The corresponding $\mathcal{EL}$ description graph $\mathcal{G}(\mathcal{A})$ is depicted on the right-hand side of Figure 6.1.2. Later on we will also consider the $\mathcal{EL}$ description tree of

$$C = \exists s.(Q \sqcap \exists r.\top) \sqcap \exists r.(Q \sqcap \exists s.\top),$$

which is depicted on the left-hand side of this figure.

Now, an instance relationship $\mathcal{A} \models C(a)$ in $\mathcal{EL}$ can be characterized via the existence of a homomorphism from the description tree of C into the description graph of $\mathcal{A}$, where such *homomorphisms* are defined analogously to the case of homomorphisms between $\mathcal{EL}$ description trees. Given an individual a, we must require that homomorphism maps the root of the description tree to the node a in $\mathcal{G}(\mathcal{A})$.

Theorem 6.2. *Let $\mathcal{A}$ be an $\mathcal{EL}$-ABox, $a \in Ind(\mathcal{A})$ be an individual in $\mathcal{A}$, and C be an $\mathcal{EL}$ concept description. Then, $\mathcal{A} \models C(a)$ if and only if there exists a homomorphism φ from $\mathcal{G}_C$ into $\mathcal{G}(\mathcal{A})$ such that $\varphi(v_0) = a$, where v_0 is the root of $\mathcal{G}_C$.*

In our example (Figure 6.1.2), a is an instance of C, since mapping v_0 on a, v_i on w_i, $i = 1, 2$, and v_3 on b and v_4 on c yields a homomorphism from $\mathcal{G}(C)$ into $\mathcal{G}(\mathcal{A})$.

As mentioned in Section 4, existence of a homomorphism between graphs is an NP-complete problem. In the restricted case of testing for the existence of homomorphisms mapping trees into graphs, the problem is polynomial [**52**]. Thus, as a corollary of Theorem 6.2 , we obtain the following complexity result.

Corollary 6.3. *The instance problem for $\mathcal{EL}$ can be decided in polynomial time.*

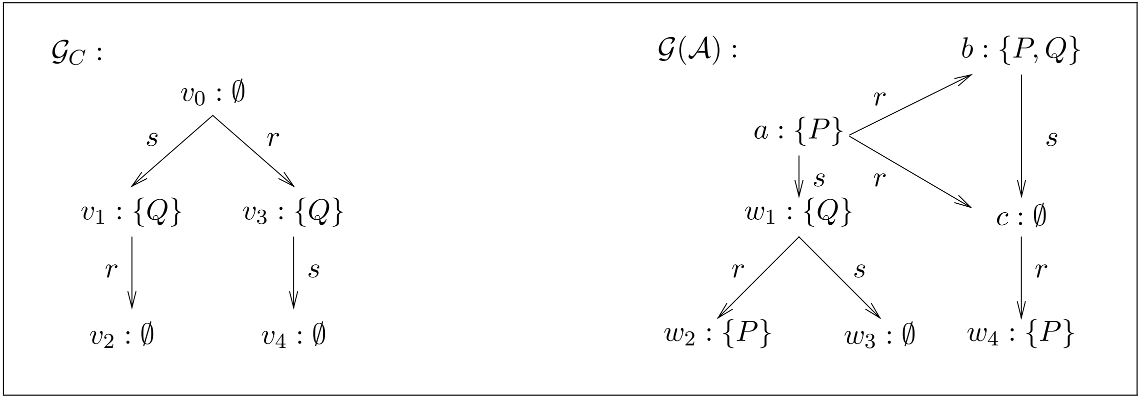


FIGURE 5. The $\mathcal{EL}$ description tree of C and the $\mathcal{EL}$ description graph of $\mathcal{A}$

6.1.3. *Computing k-approximations*

Our algorithm for computing $msc^k_{\mathcal{EL}}(a)$ is based on the following idea. Let $\mathcal{T}(a, \mathcal{G}(\mathcal{A}))$ denote the tree with root a obtained from the graph $\mathcal{G}(\mathcal{A})$ by unraveling. This tree has a finite branching factor, but possibly infinitely long paths. Pruning all paths to length k yields an $\mathcal{EL}$ description tree $\mathcal{T}_k(a, \mathcal{G}(\mathcal{A}))$ of depth $\leqslant k$. Using Theorem 6.2 and Theorem 4.3, it is easy to show that the $\mathcal{EL}$ concept description $C_{\mathcal{T}_k(a,\mathcal{G}(\mathcal{A}))}$ is equivalent to $msc^k_{\mathcal{EL}}(a)$. In addition, we obtain a characterization of the existence of the msc. The following theorem summarizes the results.

Theorem 6.4. *Let $\mathcal{A}$ be an $\mathcal{EL}$-ABox, $a \in Ind(\mathcal{A})$, and $k \in \mathbb{N}$. Then, $C_{\mathcal{T}_k(a,\mathcal{G}(\mathcal{A}))}$ is the k-approximation of a w.r.t. $\mathcal{A}$. If, starting from a, no cyclic can be reached in $\mathcal{A}$ (i.e., $\mathcal{T}(a, \mathcal{G}(\mathcal{A}))$ is finite), then $C_{\mathcal{T}(a,\mathcal{G}(\mathcal{A}))}$ is the msc of a w.r.t. $\mathcal{A}$; otherwise no msc exists.*

As a corollary of this theorem we obtain:

Corollary 6.5. *For an $\mathcal{EL}$-ABox $\mathcal{A}$, an individual $a \in Ind(\mathcal{A})$, and $k \in \mathbb{N}$, the k-approximation of a w.r.t. $\mathcal{A}$ always exists and it can be computed in time polynomial in the size of $\mathcal{A}$ if k is assumed to be constant, and in exponential time otherwise. The existence of the msc can be decided in polynomial time. If the msc exists, then it can be computed in time exponential in the size of $\mathcal{A}$.*

Taking the ABox $\mathcal{A} = \{r(a,a),\ s(a,a)\}$ as an example, it is easy to see that the size of the k-approximation of $\mathcal{A}$ may grow exponentially in k if no structure sharing is employed. However, this exponential blow-up can be avoided when the k-approximations are defined by acyclic TBoxes. The same is true for the msc in case it exists: Consider, for instance, the ABox which consists of a sequence $a_1, \ldots, a_n$ of n individuals where there is an r and an s edge from a_i to a_{i+1} for every i.

6.1.4. *Extensions to more expressive DLs*

So far, not much is known about computing k-approximations of the msc for DLs more expressible than $\mathcal{EL}$. In [**73**], the approach presented above is extended to the DL $\mathcal{EL}_\neg$, which extends $\mathcal{EL}$ by the bottom concept $\bot$ and primitive negation $\neg P$. In the following, we briefly present the ideas behind computing k-approximations of the msc in $\mathcal{EL}_\neg$, and discuss the problems that arise when considering more expressive DLs.

The following example illustrates that a naïve extension of the approach for $\mathcal{EL}$ does not work for $\mathcal{EL}_\neg$. Consider, for instance, the following $\mathcal{EL}_\neg$ concept description C and $\mathcal{EL}_\neg$ ABox $\mathcal{A}$:

$$C = P \sqcap \exists r.(P \sqcap \exists r.\neg P)$$
$$\mathcal{A} = \{P(a), P(b_1), \neg P(b_3), r(a, b_1), r(a, b_2), r(b_1, b_2), r(b_2, b_3)\}.$$

The corresponding description tree and graph are depicted in Figure 6.1.2. Obviously, there does not exist a homomorphism φ from $\mathcal{G}_C$ into $\mathcal{G}(\mathcal{A})$ with $\varphi(w_0) = a$, because neither $P \in \ell(b_2)$ nor $\neg P \in \ell(b_2)$. For each model $\mathcal{I}$ of $\mathcal{A}$, however, either $b_2^\mathcal{I} \in P^\mathcal{I}$ or $b_2^\mathcal{I} \in (\neg P)^\mathcal{I}$, and thus $a^\mathcal{I} \in C^\mathcal{I}$.

Hence a is an instance of C w.r.t. $\mathcal{A}$ even though there does not exist a homomorphism φ from $\mathcal{G}_C$ into $\mathcal{G}(\mathcal{A})$ with $\varphi(w_0) = a$.

The reason for the problem illustrated by the example is that for the individuals in the ABox it is not always fixed whether they are instances of a given atomic concept or of its negation. In order to obtain a sound and complete characterization analogous to Theorem 6.2, we therefore consider all so-called atomic completions of $\mathcal{G}(\mathcal{A})$. An atomic completion of $\mathcal{G}(\mathcal{A})$ is obtained from $\mathcal{G}(\mathcal{A})$ by adding, for all concept names P and all nodes whose label contains neither P nor $\neg P$, either P or $\neg P$ to the label of this node.

In [**73**], it is shown that an individual a of the consistent $\mathcal{EL}_\neg$-ABox $\mathcal{A}$ is an instance of the $\mathcal{EL}_\neg$ concept description C if and only if *for every atomic completion $\mathcal{G}'$ of $\mathcal{G}(\mathcal{A})$* there exists a homomorphism from $\mathcal{G}_C$ into $\mathcal{G}'$ that maps the root of $\mathcal{G}_C$ onto a.

Using this characterization of the instance problem, it is possible to show that the instance problem for $\mathcal{EL}_\neg$ is coNP-complete.

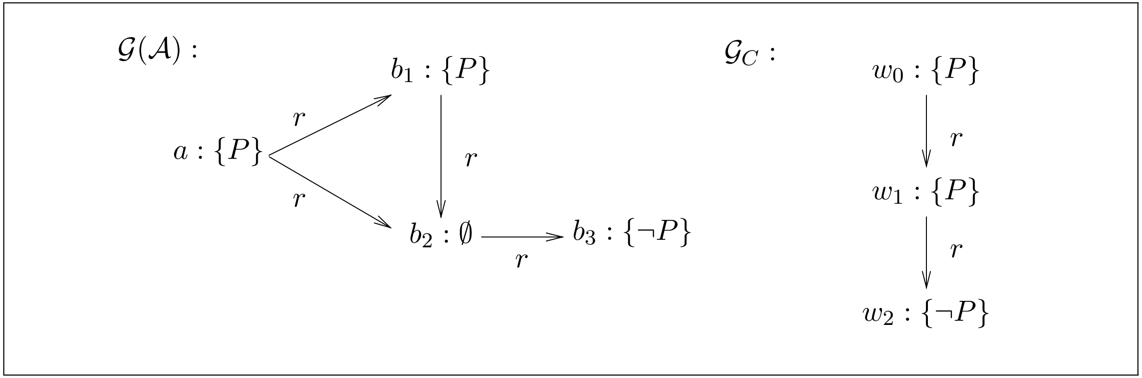


FIGURE 6. The $\mathcal{EL}_\neg$ description graph and $\mathcal{EL}_\neg$ description tree of our example

Also, k-approximations can be obtained by unraveling the completions up to depth k and then taking the lcs of these completions.

Unfortunately, for $\mathcal{ALE}$ (or even more expressive DLs), analogous characterizations of the instance problem are not known. However, given finite sets of concept and role names, the set of all $\mathcal{ALE}$ concept descriptions of depth $\leqslant k$ is finite (up to equivalence) and can be computed effectively. The fact that $\mathcal{ALE}$ allows for conjunction implies that a k-approximation always exists: it can be obtained as the conjunction of all concepts (up to equivalence) of depth $\leqslant k$ that have the individual a as an instance. Obviously, this generic argument also carries over to more expressive DLs, including $\mathcal{ALCN}$ and beyond. However, such an enumeration algorithm is clearly to complex, and thus of no practical use.

6.2. The most specific concept in the presence of cyclic TBoxes

It has first been shown for cyclic $\mathcal{ALN}$ TBoxes [**17**] and more recently for cyclic $\mathcal{EL}$ TBoxes [**7**] that the msc always exists if the TBoxes are interpreted with the greatest fixed point semantics. In addition, this msc can effectively be computed. In contrast, the msc need not exist if the TBoxes are interpreted with the least fixed-point semantics or descriptive semantics. The problem of computing the msc w.r.t. $\mathcal{EL}$ TBoxes interpreted with descriptive semantics is investigated in interpreted with descriptive semantics, a polynomial [**6, 9**].

7. Rewriting

In this section, we review results obtained for computing minimal rewritings, as defined in Section 3.2. As before, in our exposition we concentrate on $\mathcal{ALE}$ and sublanguages thereof, and comment on results for other DLs only briefly. As introduced in Section 3.2, the minimal rewriting problem is one instance of a more general rewriting framework. Another instance is approximation, which is also briefly discussed here.

In the following subsection, we consider the minimal rewriting decision problem. This will provide us with complexity lower bounds for the problem of computing minimal rewritings. The minimal rewriting computation problem itself is covered in Section 7.2. Approximation is discussed in Section 7.3. The results for minimal rewriting presented in this section are based mainly on the results of [**23**].

7.1. The minimal rewriting decision problem

Formulated for $\mathcal{ALE}$, the *minimal rewriting decision problem* is concerned with the following question: given an $\mathcal{ALE}$ concept description C, an $\mathcal{ALE}$ TBox $\mathcal{T}$, and a nonnegative integer $\varkappa$, does there exist an $\mathcal{ALE}$-rewriting E of C using $\mathcal{T}$ such that $|E| \leqslant \varkappa$.

Clearly, this problem is decidable in nondeterministic polynomial time using an oracle for deciding equivalence modulo TBoxes by the following algorithm. First, guess an $\mathcal{ALE}$ concept description E of size $\leqslant \varkappa$. Then check whether E is equivalent to C modulo $\mathcal{T}$.

This simple algorithm yields the following complexity upper bounds for the minimal rewriting decision problem in $\mathcal{ALE}$. If $\mathcal{T}$ is unfolded, i.e., the right-hand sides of the concept definitions do not contain defined concepts, we know that equivalence in $\mathcal{ALE}$ is in NP (see Section 2). Otherwise, if we do not assume $\mathcal{T}$ to be unfolded, equivalence is in PSPACE[4] since this is even the case for the larger DL $\mathcal{ALC}$ (see Section 2). Hence, for unfolded TBoxes the minimal rewriting decision problem for $\mathcal{ALE}$ is in NP, and otherwise it is in PSPACE.

[4] This is only an upper bound. The exact complexity of the equivalence problem in $\mathcal{ALE}$ with acyclic TBoxes is not known.

Conversely, it is easy to see that the minimal rewriting decision problem is at least as hard as deciding subsumption. Let C and D be $\mathcal{ALE}$ concept descriptions, and A, P_1, P_2 be three different concept names not occurring in C, D. It is easy to see that $C \sqsubseteq D$ if and only if there exists a minimal rewriting of size $\leqslant 1$ of the $\mathcal{ALE}$ concept description $P_1 \sqcap P_1 \sqcap C$ using the TBox $\mathcal{T} = \{A \doteq P_1 \sqcap P_2 \sqcap C \sqcap D\}$. Since subsumption in $\mathcal{ALE}$ w.r.t. an (unfolded or non-unfolded) $\mathcal{ALE}$ TBox is NP-hard, it follows that the minimal rewriting decision problem is NP-hard for $\mathcal{ALE}$.

Theorem 7.1. *In $\mathcal{ALE}$, the minimal rewriting decision problem is NP-complete for unfolded $\mathcal{ALE}$ TBoxes. With respect to arbitrary acyclic TBoxes, this problem is NP-hard and in PSPACE.*

Clearly, the above arguments also apply to other DLs. For example, we can use these arguments and the known complexity results for subsumption and equivalence in $\mathcal{ALC}$ to show that the minimal rewriting decision problem is PSPACE-complete for $\mathcal{ALC}$ (independently of whether the $\mathcal{ALC}$ TBox is unfolded or not). It should be noted, however, that the complexity of the subsumption problem is not the only source of complexity for the minimal rewriting decision problem. As an optimization problem, the minimal rewriting decision problem may also be intractable even if the subsumption problem is tractable. For example, subsumption w.r.t. unfolded TBoxes in $\mathcal{FL}_0$ and $\mathcal{ALN}$ is in P, but the NP-hardness of the minimal rewriting decision problem can nevertheless be shown by a reduction from SETCOVER (see [**23**] for details).

7.2. The minimal rewriting computation problem

Whereas the previous subsection was concerned with deciding whether there exists a rewriting within a given size bound, this subsection considers the problem of actually computing minimal rewritings. This is called the *minimal rewriting computation problem*. Since the minimal rewriting decision problem can obviously be reduced in polynomial time to the minimal rewriting computation problem, the lower bounds shown above immediately carry over to the computation problem.

To be more precise, there are actually two different variants of the computation problem. For a given instance $(C, \mathcal{T})$ of the minimal rewriting computation problem, one can be interested in computing either (1) *one* minimal rewriting of C using $\mathcal{T}$, or (2) *all* minimal rewritings of C using $\mathcal{T}$.

The hardness results of the previous subsection imply that even computing one minimal rewriting is in general a hard problem. In addition, it is easy to see that the number of minimal rewritings of a concept description C w.r.t. a TBox $\mathcal{T}$ can be exponential in the size of C and $\mathcal{T}$. Consider, for instance, the concept description

$$C_n = P_1 \sqcap \ldots \sqcap P_n$$

and the TBox

$$\mathcal{T}_n = \{A_i \doteq P_i \mid 1 \leqslant i \leqslant n\}.$$

The minimal rewritings are of the form

$$E = P_{i_1} \sqcap \cdots P_{i_k} \sqcap A_{j_1} \sqcap \cdots A_{j_l},$$

where $l+k=n$ and $\{1, \ldots, n\} = \{i_1, \ldots, i_k, j_1, \ldots, j_l\}$. Obviously, there are exponentially many such rewritings.

It is very easy to come up with an algorithm for computing one or all minimal rewritings of a concept description C w.r.t. the TBox $\mathcal{T}$. Since the size of the minimal rewritings is bounded by the size of C, one can simply enumerate all concept descriptions of size less than or equal to the size of C, and check which of them are equivalent to C w.r.t. $\mathcal{T}$. Those of minimal size are the minimal rewritings. Clearly, this algorithm works for all DLs where equivalence w.r.t. a TBox is decidable. However, such a brute-force enumeration algorithm is clearly too inefficient to be of any practical interest.

In what follows, we present a more source-driven algorithm for $\mathcal{ALE}$ which uses the form of C (rather than only the size of C) to prune the search space.[5] The algorithm assumes the concept description C to be in $\forall$-*normal form.* This normal form is obtained from C (in polynomial time) by exhaustively applying the rule $\forall r.E \sqcap \forall r.F \longrightarrow \forall r.(E \sqcap F)$ to C. As a result, every conjunction in C contains at most one value restriction $\forall r.D$ for a given role $r \in N_R$.

Given an $\mathcal{ALE}$ concept description C in $\forall$-normal form and an $\mathcal{ALE}$ TBox $\mathcal{T}$, the algorithm for computing minimal rewritings works as follows:

(1) Compute an extension C^* of C w.r.t. $\mathcal{T}$, which adds some defined concepts to C without changing its meaning.
(2) Compute a reduction $\widehat{C}$ of C^* w.r.t. $\mathcal{T}$, which removes parts of C^* without changing its meaning.
(3) Return $\widehat{C}$.

It remains to give formal definitions of the notions "extension" and "reduction."

Definition 7.2. Let C be an $\mathcal{ALE}$ concept description and $\mathcal{T}$ be an $\mathcal{ALE}$ TBox. An *extension* C^* of C w. r.t. $\mathcal{T}$ is an $\mathcal{ALE}$ concept description obtained from C by conjoining defined names at some positions in C such that C^* is equivalent to C modulo $\mathcal{T}$.

Obviously, there may exist exponentially many different extensions of C^*, which shows that this step may take exponential time. Alternatively, we could considered this to be a nondeterministic step, in which an appropriate extension is guessed.

Informally speaking, a reduction $\widehat{C}$ of C^* w.r.t. $\mathcal{T}$ is an $\mathcal{ALE}$ concept description obtained from C^* by "eliminating all redundancies in C^*" such that the resulting concept description is still

[5] A similar approach works also for the DL $\mathcal{ALN}$ [**23**].

equivalent to C^* modulo $\mathcal{T}$. A concept description may have exponentially many different reductions, and hence computing reductions may also be considered to be a nondeterministic step.

Before defining the notion of a "reduction" formally, we illustrate how our algorithm works by a simple example. Consider the $\mathcal{ALE}$ concept description

$$C = P \sqcap Q \sqcap \forall r.P \sqcap \exists r.(P \sqcap \exists r.Q) \sqcap \exists r.(P \sqcap \forall r.(Q \sqcap \neg Q)),$$

and the $\mathcal{ALE}$ TBox $\mathcal{T} = \{\ A_1 \doteq \exists r.Q,\ A_2 \doteq P \sqcap \forall r.P,\ A_3 \doteq \forall r.P\ \}$.

The concept description

$$\begin{array}{rcl} C^* & = & A_2 \sqcap P \sqcap Q \sqcap \forall r.P \sqcap \\ & & \exists r.(A_1 \sqcap P \sqcap \exists r.Q) \sqcap \exists r.(P \sqcap \forall r.(Q \sqcap \neg Q)) \end{array}$$

is an extension of C. A reduction of C^* can be obtained by eliminating

- P and $\forall r.P$ on the top-level of C^*, because they are redundant w.r.t. A_2;
- P in both of the existential restrictions on the top-level of C^*, because it is redundant due to the value restriction $\forall r.P$ on the top-level of C;
- the existential restriction $\exists r.Q$, because it is redundant w.r.t. A_1; and
- replacing $Q \sqcap \neg Q$ by $\bot$, since $\bot$ is the minimal inconsistent concept description.

The resulting concept description $\widehat{C} = A_2 \sqcap Q \sqcap \exists r.A_1 \sqcap \exists r.\forall r.\bot$ is equivalent to C modulo $\mathcal{T}$, i.e., $\widehat{C}$ is a rewriting of C using $\mathcal{T}$. Furthermore, it is easy to see that $\widehat{C}$ is in fact a *minimal* rewriting of C using $\mathcal{T}$.

Before we can define the notion of a "reduction" formally, we must formalize the notion of a "subdescription."

Definition 7.3. The $\mathcal{ALE}$ concept description $\widehat{C}$ is a *subdescription* of the $\mathcal{ALE}$ concept description C if and only if it is equivalent to

(1) $\widehat{C} = C$; or

(2) $\widehat{C} = \bot$; or

(3) $\widehat{C}$ is obtained from C by

- removing some (negated) primitive concept names, value restrictions, or existential restrictions on the top-level of C, and
- for all remaining value/existential restrictions $\forall r.D/\exists r.D$ replacing D by a subdescription $\widehat{D}$ of D.

The subdescription $\widehat{C}$ of C is a *proper* subdescription of C if and only if it is different from C.

Now, reductions can be defined as follows:

Definition 7.4. Let C^* be an $\mathcal{ALE}$ concept description and $\mathcal{T}$ be an $\mathcal{ALE}$ TBox. The $\mathcal{ALE}$ concept description $\widehat{C}$ is called a *reduction of C^* w.r.t. $\mathcal{T}$* if and only if $\widehat{C}$ is equivalent to C^* w.r.t. $\mathcal{T}$ and minimal in the following sense: there does not exist a proper subdescription of $\widehat{C}$ that is also equivalent to C^* w.r.t. $\mathcal{T}$.

Note that, in the definition of a reduction, we do not allow removal of defined concepts unless they occur within value or existential restrictions that are removed as a whole. This makes sense since such defined concepts could have been omitted in the first place when computing the extension C^* of C.

From the definition of a reduction, it is not immediately clear how to actually compute one. Intuitively, a reduction $\widehat{C}$ of an $\mathcal{ALE}$ concept C^* in $\forall$-normal form is computed in a top-down manner. If $C \equiv_{\mathcal{T}} \bot$, then $\widehat{C} := \bot$. Otherwise, let $\forall r.C'$ be the (unique!) value restriction on the role r and $A_1 \sqcap \ldots \sqcap A_n$ the conjunction of the names of defined concepts on the top-level of C. Basically, $\widehat{C}$ is then obtained from C^* as follows:

(1) Remove any (negated) primitive concept Q occurring on the top-level of C^*, if $A_1 \sqcap \ldots \sqcap A_n \sqsubseteq_{\mathcal{T}} Q$.

(2) Remove any existential restriction $\exists r.C_1$ occurring on the top-level of C^*, if
   (a) $A_1 \sqcap \ldots \sqcap A_n \sqcap \forall r.C' \sqsubseteq_{\mathcal{T}} \exists r.C_1$, or
   (b) there is another existential restriction $\exists r.C_2$ on the top-level of C^* such that $A_1 \sqcap \ldots \sqcap A_n \sqcap \forall r.C' \sqcap \exists r.C_2 \sqsubseteq_{\mathcal{T}} \exists r.C_1$.

(3) Remove the value restriction $\forall r.C'$ if $A_1 \sqcap \ldots \sqcap A_n \sqsubseteq_{\mathcal{T}} \forall r.C'$.

(4) Finally, all concept descriptions D occurring in the remaining value and existential restrictions are reduced recursively.

The formal specification of the reduction algorithm given in [**23**] is more complex than the informal description given above mainly for two reasons. First, in (2b) it could be the case that the subsumption relation also holds if the rôles of $\exists r.C_1$ and $\exists r.C_2$ are exchanged. In this case, one has a choice of which existential restriction to remove. If the (recursive) reduction of C_1 and C_2 yields descriptions of different size, then we must remove the existential restriction for the concept with the larger reduction. If, however, the reductions are of equal size, then we must make a (don't know) nondeterministic choice between removing the one or the other.

Second, in (4) we cannot reduce the descriptions D without considering the context in which they occur. The reduction of these concepts must take into account the concept C' of the top-level value restriction of C as well as all concepts D' occurring in value restrictions of the form $\forall r.D'$ on the top-level of the defining concepts for $A_1, \ldots, A_n$. For instance, in our example the removal of P within the existential restrictions on the top-level of C^* was justified by the presence of $\forall r.P$ on the top-level of C^*. For this purpose, the algorithm described in [**23**] employs a third input parameter that takes care of such contexts.

Theorem 7.5. *The rewriting algorithm for $\mathcal{ALE}$ defined in* [**23**] *has the following properties*:

(1) *Every possible output of the algorithm is a rewriting of the input concept description C using the input TBox $\mathcal{T}$, though it need not always be minimal.*

(2) *The set of all computed rewritings contains all minimal rewritings of C using $\mathcal{T}$ (modulo associativity, commutativity and idempotence of conjunction, and the equivalence $C \sqcap \top \equiv C$).*

(3) *One minimal rewriting of C w.r.t. $\mathcal{T}$ can be computed using polynomial space.*

(4) *The set of all minimal rewritings of C w.r.t. $\mathcal{T}$ can be computed in exponential time.*

In practice, it often suffices to compute one (not necessarily minimal, but "small") rewriting. The sketch of the rewriting algorithm presented above suggests the following greedy algorithm for computing such a small rewriting. First, compute the extension C^* of C in which at all positions of C all possible defined concepts are conjoined. Then compute just one reduction $\widehat{C}$ of C^*. This yields a polynomial-time algorithm—given an oracle for equivalence testing—which does not always return a minimal rewriting, but nevertheless behaves well in practice, both in terms of the quality of the returned rewritings and in terms of runtime (see [**23**] for more details).

7.3. Approximation

Given two DLs $\mathcal{L}_s$ and $\mathcal{L}_d$, an $\mathcal{L}_d$ *approximation* of an $\mathcal{L}_s$ concept description C is an $\mathcal{L}_d$ concept description D such that $C \sqsubseteq D$ and D is minimal (w.r.t. subsumption) in $\mathcal{L}_d$ with this property.

In [**42**] the case where $\mathcal{L}_s$ is $\mathcal{ALC}$ and $\mathcal{L}_d$ is $\mathcal{ALE}$ was investigated in detail. It was shown that for every $\mathcal{ALC}$ concept description there exists a unique (up to equivalence) approximation in $\mathcal{ALE}$. The size of the $\mathcal{ALE}$ approximation may grow exponentially in the size of the given $\mathcal{ALC}$ concept description, and it can be computed in double exponential time.

To measure the information that is lost by using an approximation rather than the original concept, in [**42**] the notion of the difference between concepts has been refined from an early definition by Teege [**91**]. Intuitively, the difference between a concept description C and its approximation is the concept description that needs to be conjoined to the approximation to obtain a concept description equivalent to C.

8. Matching

The matching problem has been introduced in Section 3.2. In this section, we sketch how it can be solved. As usual, our exposition concentrates on the DL $\mathcal{ALE}$. However, we will also comment on other DLs and on extensions of the basic matching problem. Most results presented here are based on [**18, 71**].

In what follows, we first consider the complexity of deciding whether a given matching problem has a solution (Section 8.1). In case a matching problem has a solution, we are also interested in computing a solution. In general, a solvable matching problem may have several (even infinitely many) solutions. Thus, the question arises what solutions are actually interesting ones. We try to answer this question in Section 8.2, where we define a precedence orderings on matchers. This ordering tells us which matchers are more interesting than others. Algorithms for computing such matchers in $\mathcal{ALE}$ are presented in Section 8.3.

	$\mathcal{EL}$	$\mathcal{ALE}$
subsumption	P	NP-complete
equivalence	NP-complete	NP-complete

TABLE 3. Deciding the solvability of matching problems

A summary of results for matching in other DLs as well as extensions of the basic matching problem is provided in Section 8.4.

8.1. Deciding matching problems

We study the question of how to decide whether a given matching problem has a matcher or not, and investigate the complexity of this problem. For the DLs $\mathcal{EL}$ and $\mathcal{ALE}$ we obtain the complexity results summarized in Table 3. The first and the second row of the table refer to matching modulo subsumption and matching modulo equivalence respectively.

These results can be obtained as follows: First, note that patterns are not required to contain variables. Consequently, matching modulo subsumption (equivalence) is at least as hard as subsumption (equivalence). Thus, NP-completeness of subsumption in $\mathcal{ALE}$ [**50**] yields hardness in the second column of Table 3. Second, for the languages $\mathcal{ALE}$ and $\mathcal{EL}$, as already mentioned in Section 3.2, matching modulo subsumption can be reduced to subsumption: $C \sqsubseteq^? D$ has a matcher if and only if the substitution $\sigma_\top$, which replaces every variable by $\top$, is a matcher of $C \sqsubseteq^? D$. Thus, the known complexity results for subsumption in $\mathcal{ALE}$ and $\mathcal{EL}$ [**50, 22**] complete the first row of Table 3. Third, NP-hardness of matching modulo equivalence for $\mathcal{EL}$ can be shown by a reduction from SAT. It remains to show that matching modulo equivalence in $\mathcal{EL}$ and $\mathcal{ALE}$ can in fact be decided in nondeterministic polynomial time. This is an easy consequence of the following (nontrivial) lemma [**71**].

Lemma 8.1. *If an $\mathcal{EL}$ or $\mathcal{ALE}$ matching problem modulo equivalence has a matcher, then it has one of size polynomially bounded in the size of the problem. Furthermore, this matcher uses only concept and role names already contained in the matching problem.*

The lemma (together with the known complexity results for subsumption) shows that the following can be realized in NP:

"guess" a substitution satisfying the given size bound, and then test whether it is a matcher.

8.2. Solutions of matching problems

As mentioned above, solvable matching problems may have infinitely many solutions. Hence it is necessary to define a class of "interesting" matchers to be presented to the user. Such a definition certainly depends on the specific application in mind. Our definition is motivated by the application in chemical process engineering mentioned before. However, it is general enough to apply also to other applications.

We use the $\mathcal{EL}$ concept description C^1_{ex} and the pattern D^1_{ex} shown in Figure 8.2 to illustrate and motivate our definitions. Along with the concept descriptions, Figure 8.2 also depicts the description trees corresponding to C^1_{ex} and D^1_{ex} as defined in Section 4.1, where concept variables are simply dealt with like concept names.

It is easy to see that the substitution $\sigma_\top$ is a matcher of $C^1_{\mathrm{ex}} \sqsubseteq^? D^1_{\mathrm{ex}}$, and thus this matching problem modulo subsumption is indeed solvable. However, the matcher $\sigma_\top$ is obviously not an interesting one. We are interested in matchers that bring us as close as possible to the description C^1_{ex}. In this sense, the matcher

$$\sigma_1 := \{X \mapsto \mathsf{W} \sqcap \exists \mathsf{hc}.\mathsf{W},\ Y \mapsto \mathsf{W}\}$$

is better than $\sigma_\top$, but still not optimal. In fact,

$$\sigma_2 := \{X \mapsto \mathsf{W} \sqcap \exists \mathsf{hc}.\mathsf{W} \sqcap \exists \mathsf{hc}.(\mathsf{W} \sqcap \mathsf{P}),\ Y \mapsto \mathsf{W} \sqcap \mathsf{D}\}$$

is better than σ_1 since it satisfies $C^1_{\mathrm{ex}} \equiv \sigma_2(D^1_{\mathrm{ex}}) \sqsubset \sigma_1(D^1_{\mathrm{ex}})$.

We formalize this intuition with the help of the following precedence ordering on matchers. For a given matching problem $C \sqsubseteq^? D$ and two matchers σ, τ we define

$$\sigma \sqsubseteq_i \tau \ \text{ iff } \ \sigma(D) \sqsubseteq \tau(D).$$

Here "i" stands for "instance". Two matchers σ, τ are *i-equivalent* ($\sigma \equiv_i \tau$) if and only if $\sigma \sqsubseteq_i \tau$ and $\tau \sqsubseteq_i \sigma$. A matcher σ is called *i-minimal* if and only if $\tau \sqsubseteq_i \sigma$ implies $\tau \equiv_i \sigma$ for every matcher τ. We are interested in *computing i-minimal matchers.* More precisely, we want to obtain at least one i-minimal matcher for each of the minimal i-equivalence classes (i.e., i-equivalence classes of i-minimal matchers). Note that, given an i-minimal matcher σ of a matching problem $C \sqsubseteq^? D$, its equivalence class, i.e., the set of all matchers that are i-equivalent to σ, consists of the matchers of the problem $\sigma(D) \equiv^? D$.

The matching problem

$$\exists r.A \sqcap \exists r.B \sqsubseteq^? \exists r.X$$

illustrates that there may in fact be different minimal i-equivalence classes: mapping X to A and mapping X to B respectively yields two i-minimal matchers which, however, do not belong to the same i-equivalence class.

Since an i-equivalence class usually contains more than one matcher, the question is which ones to prefer within this class. In our running example, σ_2 is a least and therefore i-minimal matcher. Nevertheless, it is not the one we really want to compute since it contains redundancies, i.e., expressions that are not really necessary for obtaining the instance $\sigma_2(D^1_{\text{ex}})$ (modulo equivalence). In fact, σ_2 contains two different kinds of redundancies. First, the existential restriction $\exists\mathsf{hc}.\mathsf{W}$ in $\sigma_2(X)$ is redundant since removing it still yields a concept description equivalent to $\sigma_2(X)$. Second, W in $\sigma_2(Y)$ is redundant in that the substitution obtained by deleting W from $\sigma_2(Y)$ still yields the same instance of D^1_{ex} (although the resulting concept description is no longer equivalent to $\sigma_2(Y)$). In our example, the only i-minimal matcher (modulo associativity and commutativity of concept conjunction) that is free of redundancies in this sense is

$$\sigma_3 := \{X \mapsto \mathsf{W} \sqcap \exists\mathsf{hc}.(\mathsf{W} \sqcap \mathsf{P}),\ Y \mapsto \mathsf{D}\}.$$

$$C^1_{\mathrm{ex}} := \mathsf{W} \sqcap \exists \mathsf{hc}.(\mathsf{W} \sqcap \exists \mathsf{hc}.(\mathsf{W} \sqcap \mathsf{D}) \sqcap \exists \mathsf{hc}.(\mathsf{W} \sqcap \mathsf{P})) \sqcap \exists \mathsf{hc}.(\mathsf{W} \sqcap \mathsf{D} \sqcap \exists \mathsf{hc}.(\mathsf{W} \sqcap \mathsf{P}))$$

$$D^1_{\mathrm{ex}} := \mathsf{W} \sqcap \exists \mathsf{hc}.(X \sqcap \exists \mathsf{hc}.(\mathsf{W} \sqcap Y)) \sqcap \exists \mathsf{hc}.(X \sqcap Y)$$

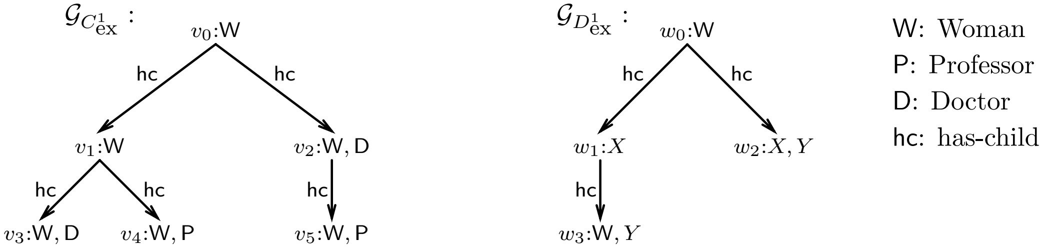


FIGURE 7. $\mathcal{EL}$ concept description and pattern, and their $\mathcal{EL}$ description trees

Summing up, we want to compute *all i-minimal matchers that are reduced*, i.e., free of redundancies. We use the notion of subdescriptions introduced above (Definition 7.3) to capture the notion "reduced" in a formal way. Given two matchers σ, τ of $C \sqsubseteq^? D$, we say that τ is a *submatcher* of σ if and only if $\tau(Y)$ is a (not necessarily strict) subdescription of $\sigma(Y)$ for all variables Y. If τ is a *submatcher* of σ and there is at least one variable X for which $\tau(X)$ is a strict subdescription of $\sigma(X)$, then we say that τ is a *strict submatcher* of σ.

Definition 8.2. The matcher σ of $C \sqsubseteq^? D$ is *i-minimal and reduced* if and only if

(1) σ is i-minimal,
(2) σ is in $\forall$-normal form, i.e., $\sigma(X)$ is in $\forall$-normal form for all variables X (see Section 7.2 for the definition of $\forall$-normal form), and
(3) there does not exist a matcher τ of $C \sqsubseteq^? D$ that is both i-equivalent to σ and a strict submatcher of σ.

8.3. Computing matchers

In the previous section, we have identified the set of all i-minimal and reduced matchers (in $\forall$-normal form) as the set of "interesting" matchers. We now show how these matchers can be computed. Given a matching problem $C \sqsubseteq^? D$, our algorithm for computing i-minimal and reduced matchers in principle proceeds as follows:

(1) Compute the set of all i-minimal matchers of $C \sqsubseteq^? D$ up to i-equivalence (i.e., one matcher for each i-equivalence class).
(2) For each i-minimal matcher σ computed in the first step, compute the set of all reduced matchers in $\forall$-normal form up to commutativity and associativity of conjunction for the problem $\sigma(D) \equiv^? D$.

If we are interested in matching modulo equivalence instead of subsumption, we just apply the second step to $C \equiv^? D$.

In the following two subsections, we illustrate the first step of the algorithm—computing i-minimal matchers—for $\mathcal{EL}$ and $\mathcal{ALE}$. For the second step, we refer the reader to [**18, 71**]. In particular, this step involves to show that every solvable $\mathcal{ALE}$ matching problem has a matcher of size polynomially bounded in the size of the matching problem.

The main results on computing matchers shown in [**18, 71**] are summarized in the following theorem. We call a set containing all i-minimal matchers up to i-equivalence *i-complete*. Such a set is called *minimal i-complete* if it contains only i-minimal matchers. Similarly, a set containing all reduced matchers in ∀-normal form (up to commutativity and associativity of conjunction) is called *complete w.r.t. reduction*, and it is called *minimal* if it contains only reduced matchers.

Theorem 8.3. (1) *For a solvable $\mathcal{ALE}$ or $\mathcal{EL}$ matching problem modulo subsumption, the cardinality of a* (*minimal*) i-*complete set can be bounded exponentially in the size of the matching problem. This upper bounds is tight. Furthermore, minimal* i-*complete sets can be computed in exponential time in case of $\mathcal{EL}$ and in exponential space in case of $\mathcal{ALE}$. If minimality is not required, such a set can be computed in exponential time also for $\mathcal{ALE}$.*

(2) *For a solvable $\mathcal{ALE}$ or $\mathcal{EL}$ matching problem modulo equivalence, the cardinality of a* (*minimal*) *complete set w.r.t. reduction may grow exponentially in the size of the matching problem. However, the size of the matchers in this set can polynomially be bounded. This immediately implies that there exists an exponential time algorithm for computing minimal complete sets w.r.t. reduction* (*both for $\mathcal{ALE}$ and $\mathcal{EL}$*).

8.3.1. *Computing i-minimal matchers in $\mathcal{EL}$*

The algorithm for computing i-minimal matchers in $\mathcal{EL}$ is based on the characterization of subsumption via homomorphisms between description trees presented in Section 4.1.

Given a matching problem of the form $C \sqsubseteq^? D$, our algorithm computes homomorphisms from the description tree $\mathcal{G}_D$ corresponding to D into the description tree $\mathcal{G}_C$ corresponding to C. Concept patters are turned into description trees in the obvious way, i.e., concept variables are dealt with as concept names (see, for example, Figure 8.2). When computing the homomorphisms from $\mathcal{G}_D$ into $\mathcal{G}_C$, the variables in $\mathcal{G}_D$ are ignored. For instance, in our example, there are six homomorphisms from $\mathcal{G}_{D^1_{\text{ex}}}$ into $\mathcal{G}_{C^1_{\text{ex}}}$. We will later consider the ones mapping w_i onto v_i for $i = 0, 1, 2$, and w_3 onto v_3 or w_3 onto v_4, which we denote by φ_1 and φ_2 respectively.

The complete algorithm is depicted in Figure 8. With $C_{\varphi(v)}$ we denote the $\mathcal{EL}$ concept description that corresponds to the $\mathcal{EL}$ description tree rooted at the node $\varphi(v)$ in $\mathcal{G}_C$. The algorithm constructs substitutions τ such that $C \sqsubseteq \tau(D)$, i.e., there is a homomorphism from $\mathcal{G}_{\tau(D)}$ into $\mathcal{G}_C$. This is achieved by first computing all homomorphisms from $\mathcal{G}_D$ into $\mathcal{G}_C$. Assume that the node v in $\mathcal{G}_D$, whose label contains X, is mapped onto the note $w = \varphi(v)$ of $\mathcal{G}_C$. The idea is then to substitute X with the concept description corresponding to the subtree of $\mathcal{G}_C$ starting with the node $w = \varphi(v)$, i.e., with $C_{\varphi(v)}$. The remaining problem is that a variable X may occur more than once in D. Thus, we cannot simply define $\tau(X)$ as $C_{\varphi(v)}$ where v is such that X occurs in the label of v. Since there may exist several nodes v with this property, we take the least common subsumer of the corresponding parts of C. The reason for taking the *least* common subsumer is that we want to compute substitutions that are as specific as possible.

In our example, the homomorphism φ_1 yields the substitution τ_1:

$$\begin{aligned} \tau_1(X) &:= lcs\{C^1_{\mathsf{ex},v_1},\ C^1_{\mathsf{ex},v_2}\} \equiv \mathsf{W} \sqcap \exists \mathsf{hc}.(\mathsf{W} \sqcap \mathsf{P}), \\ \tau_1(Y) &:= lcs\{C^1_{\mathsf{ex},v_2},\ C^1_{\mathsf{ex},v_3}\} \equiv \mathsf{W} \sqcap \mathsf{D}, \end{aligned}$$

whereas φ_2 yields the substitution τ_2:

$$\begin{aligned} \tau_2(X) &:= lcs\{C^1_{\mathsf{ex},v_1},\ C^1_{\mathsf{ex},v_2}\} \equiv \mathsf{W} \sqcap \exists \mathsf{hc}.(\mathsf{W} \sqcap \mathsf{P}), \\ \tau_2(Y) &:= lcs\{C^1_{\mathsf{ex},v_2},\ C^1_{\mathsf{ex},v_4}\} \equiv \mathsf{W}. \end{aligned}$$

```
Input: $\mathcal{EL}$ matching problem $C \sqsubseteq^? D$.
Output: $i$-complete set $\mathcal{C}$ for $C \sqsubseteq^? D$.

$\mathcal{C} := \emptyset$;
For all homomorphisms $\varphi$ from
  $\mathcal{G}_D = (V, E, v_0, \ell)$ into $\mathcal{G}_C$ do
    Define $\tau$ by $\tau(X) := lcs\{C_{\varphi(v)} \mid X \in \ell(v)\}$
      for all variables $X$ in $D$;
    $\mathcal{C} := \mathcal{C} \cup \{\tau\}$;
```

FIGURE 8. The $\mathcal{EL}$ matching algorithm

Unlike τ_1, the substitution τ_2 is not i-minimal. Therefore, τ_2 will be removed in a post-processing step, which extracts a *minimal* i-complete set from the i-complete one. By applying Theorem 4.3, the following theorem is easy to show:

Theorem 8.4. *The algorithm described in Figure 8 always computes an i-complete set of matchers for a given $\mathcal{EL}$ matching problem modulo subsumption.*

8.3.2. *Computing i-minimal matchers in $\mathcal{ALE}$*

The idea underlying the algorithm for computing i-minimal matchers in $\mathcal{ALE}$ is similar to the one for $\mathcal{EL}$. Again, we apply the characterization of subsumption by homomorphisms (Theorem 4.5). One problem is that this characterization requires the subsuming description to be normalized.[6] However, the pattern D contains variables, and hence the normalization of $\sigma(D)$ depends on what is substituted for these variables by the matcher σ. However, this matcher is exactly what we want to compute in the first place.

Fortunately, Theorem 4.5 can be relaxed as follows. To characterize the subsumption relation $C \sqsubseteq D$, it is not necessary to

[6] Recall that, in the case of $\mathcal{ALE}$, the description tree $\mathcal{G}_C$ of a concept description C is obtained from the normal form of C.

normalize D completely. Instead of $\mathcal{G}_D$, which is based on the normal form of D, it suffices to employ the tree $\mathcal{G}_D^\top$ that is obtained from the so-called $\top$-*normal form* of D. This normal form is obtained from D by exhaustively applying the rule $\forall r.\top \longrightarrow \top$. As an easy consequence of the proof of Theorem 4.5, we obtain the following corollary:

Corollary 8.5. *Let C, D be $\mathcal{ALE}$ concept descriptions. Then, $C \sqsubseteq D$ if and only if there exists a homomorphism from $\mathcal{G}_D^\top$ to $\mathcal{G}_C$.*

Given the $\mathcal{ALE}$ matching problem $C \sqsubseteq^? D$, the following example illustrates that it does not suffice to consider just all homomorphisms from $\mathcal{G}_D^\top$ to $\mathcal{G}_C$ in order to compute an i-complete set.

Example 8.6. Consider the $\mathcal{ALE}$ matching problem $C^2_{\text{ex}} \sqsubseteq^? D^2_{\text{ex}}$, where

$$C^2_{\text{ex}} := (\exists r.\forall r.Q) \sqcap (\exists r.\forall s.P)$$
$$D^2_{\text{ex}} := \exists r.(\forall r.X \sqcap \forall s.Y).$$

The description trees corresponding to C^2_{ex} and D^2_{ex} are depicted in Figure 8.3. Obviously, $\sigma := \{X \mapsto Q, Y \mapsto \top\}$ and $\tau := \{X \mapsto \top, Y \mapsto P\}$ are solutions of the matching problem. However, there is no homomorphism from $\mathcal{G}^\top_{D^2_{\text{ex}}}$ into $\mathcal{G}_{C^2_{\text{ex}}}$. Indeed, the node w_1 can be mapped either to v_1 or v_2. In the former case, w_2 can be mapped to v_3, but then there is no way to map w_3. In the latter case, w_3 must be mapped to v_4, but then there is no node w_2 can be mapped to.

The problem is that Corollary 8.5 requires the subsumer to be in $\top$-normal form. However, the $\top$-normal form of the instantiated concept pattern depends on the matcher, and thus cannot be computed in advance. Fortunately, only matchers that substitute variables by $\top$ cause problems. Thus, the problem can be fixed by first guessing which variables are replaced by $\top$. Replacing these variables in D by $\top$ yields a so-called $\top$-*pattern* E. Now, instead of computing all homomorphisms from $\mathcal{G}_D^\top$ into $\mathcal{G}_C$, our matching

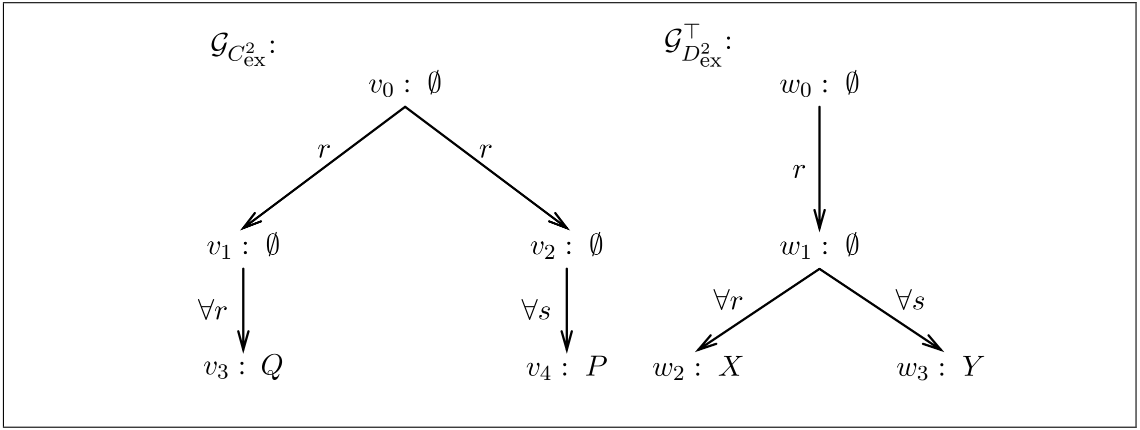


FIGURE 9. The description trees for C^2_{ex} and D^2_{ex}

algorithm computes for *all* $\top$-patterns E of D all homomorphism from $\mathcal{G}_E^\top$ into $\mathcal{G}_C$. With this modification, we obtain:

Theorem 8.7. *There is an algorithm that computes an i-complete set of matchers for a given $\mathcal{ALE}$ matching problem modulo subsumption.*

8.4. Matching in other DLs and extensions of matching

We give only a very brief overview on results for other DLs and on extensions of matching (see also [**71**] for a more detailed overview).

Matching has also been considered for the DLs $\mathcal{ALN}$ [**21**], $\mathcal{ALNS}$ [**71**], and $\mathcal{ALN}$ with cyclic TBoxes [**71**], based on the characterization of subsumption proved for these DLs.

The basic matching problem, as introduced in Section 3.2, has been extended in the following two directions. First, matching where variables are further constrained by side conditions of the form $X \sqsubseteq E$ or $X \sqsubset E$ (where E is a concept pattern and X is a concept variable) was first introduced in [**37**], and further studied in [**21, 10**] for the DL $\mathcal{ALN}$.

Second, unification, which extends matching modulo equivalence in that both sides of the equation may contain variables, has first been introduced in the context of DLs in [**25**], and studied there for the DL $\mathcal{FL}_0$. It is shown there that unification is considerably more complex than matching: even for the small DL $\mathcal{FL}_0$, deciding whether a given unification problem has a solution or not is EXPTIME-complete. Later on, these results were extended to unification in $\mathcal{FL}_{trans}$, the extension of $\mathcal{FL}_0$ by transitive closure of roles [**19**], and to the extension of this DL by atomic negation [**20**].

9. Conclusion and Future Perspectives

Compared to the large body of results for standard inferences in DLs, the investigation of nonstandard inferences is only at its beginning.

Nevertheless, for the DLs $\mathcal{ALE}$ and $\mathcal{ALN}$ and their sublanguages, we now have a relatively good understanding of how to solve nonstandard inferences like computing the least common subsumer, matching, and rewriting. For these results to be useful in practice, two more problems must be addressed, though.

First, there is a need for good implementations of the algorithms developed for nonstandard inferences, which must be able to interact with existing systems implementing standard inferences. The system Sonic [**92, 93**] is a first step in this direction. It extends the ontology editor OilEd [**32**] by implementations of the nonstandard inferences lcs and approximation, and uses the system Racer [**54**] as standard reasoner. There also exist first implementations of matching algorithms for $\mathcal{ALE}$ [**41**] and $\mathcal{ALN}$ [**43**].

Second, modern DL systems like FaCT [**61**] and Racer [**54**] are based on very expressive DLs, and there exist large knowledge bases that use this expressive power and can be processed by these systems [**85, 90, 53**]. In contrast, results for nonstandard inferences are currently restricted to rather inexpressive DLs, and some of these inferences do not even make sense for more expressive DLs.[7] In order to allow the user to re-use concepts defined in such existing expressive knowledge bases and still support the user with nonstandard inferences, one can either use approximation or consider nonstandard inferences w.r.t. a background terminology.

To explain these two options in more detail, assume that $\mathcal{L}_2$ is an expressive DL, and that $\mathcal{L}_1$ is a sublanguage of $\mathcal{L}_2$ for which we know how to compute nonstandard inferences. In the first case, one first computes the $\mathcal{L}_1$ approximation of the concepts expressed in $\mathcal{L}_2$, and then applies the nonstandard inferences in $\mathcal{L}_1$. As mentioned, first results for approximation have been obtained in [**42**]. In the second case, one considers a *background terminology* $\mathcal{T}$ defined in $\mathcal{L}_2$. When defining new concepts, the user employs

[7] For example, as pointed out before, using the lcs does not make sense in DLs allowing for disjunction.

only the sublanguage $\mathcal{L}_1$ of $\mathcal{L}_2$. However, in addition to primitive concepts and roles, the concept descriptions written in the DL $\mathcal{L}_1$ may also contain names of concepts defined in $\mathcal{T}$. The nonstandard inferences are then defined modulo the TBox $\mathcal{T}$, i.e., instead of using subsumption between $\mathcal{L}_1$ concept descriptions, one uses subsumption w.r.t. the TBox $\mathcal{T}$. First results for the lcs modulo background terminologies have been obtained in [**30**].

References

1. S. Abiteboul, R. Hull, and V. Vianu, *Foundations of Databases*, Amsterdam, Addison-Wesley, 1995.
2. F. Baader, *Augmenting concept languages by transitive closure of roles: An alternative to terminological cycles*, In: Proceedings of the 12th International Conference (Sydney/Australia 1991), 1991, pp. 446–451.
3. F. Baader, *A formal definition for the expressive power of terminological knowledge representation languages*, J. Log. Comput. **6** (1996), no. 1, 33–54.
4. F. Baader, *Using automata theory for characterizing the semantics of terminological cycles*, Ann. Math. Artif. Intell. **18** (1996), no. 2–4, 175–219.
5. F. Baader, *Computing the least common subsumer in the description logic EL w.r.t. terminological cycles with descriptive semantics*, In: Proceedings of the 11th International Conference on Conceptual Structures, Lect. Notes Artif. Intell. **2746** (2003), pp. 117–130.
6. F. Baader, *The instance problem and the most speci.c concept in the description logic EL w.r.t. terminological cycles with descriptive semantics*, In: Proceedings of the 26th Annual German Conference on Artificial Intelligence, Lect. Notes Artif. Intell. **2821** (2003), pp. 64–78.
7. F. Baader, *Least common subsumers and most specific concepts in a description logic with existential restrictions and terminological cycles*, In: G. Gottlob and T. Walsh (eds.), Proceedings of the 18th International Joint Conference on Artificial Intelligence, Morgan Kaufmann, 2003, pp. 319–324.

8. F. Baader, *Terminological cycles in a description logic with existential restrictions*, In: G. Gottlob and T. Walsh (eds.), Proceedings of the 18th International Joint Conference on Artificial Intelligence, Morgan Kaufmann, 2003, pp. 325–330.

9. F. Baader, *A graph-theoretic generalization of the least common subsumer and the most specific concept in the description logic EL*, In: J. Hromkovic and M. Nagl (eds.), Proceedings of the 30th International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2004), Lect. Notes Comput. Sci., 2004.

10. F. Baader, S. Brandt, and R. Küsters, *Matching under side conditions in description logics*, In: Proceedings of the 17th International Joint Conference on Artificial Intelligence, Morgan Kaufmann, 2001, pp. 213–218.

11. F. Baader, M. Buchheit, and B. Hollunder, *Cardinality restrictions on concepts*, Artif. Intell. **88** (1996), no. 1–2, 195–213.

12. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider (eds.), *The Description Logic Handbook: Theory, Implementation, and Applications*, Cambridge, Cambridge Univ. Press, 2003.

13. F. Baader and Ph. Hanschke, *A schema for integrating concrete domains into concept languages*, In: Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI-91), 1991, pp. 452–457.

14. F. Baader and B. Hollunder, *A terminological knowledge representation system with complete inference algorithm*, In: Proceedings of the Workshop on Processing Declarative Knowledge (PDK-91), Lect. Notes Artif. Intell. **567** (1991), pp. 67–86.

15. F. Baader, I. Horrocks, and U. Sattler, *Description logics for the semantic web*, KI – K.unstl iche Intelligenz, 4, 2002.

16. F. Baader, I. Horrocks, and U. Sattler, *Description logics*, In: S. Staab and R. Studer (eds.), Handbook on Ontologies, International Handbooks in Information Systems, pages 3–28. Springer–Verlag, Berlin, Germany, 2003.

17. F. Baader and R. Küsters, *Computing the least common subsumer and the most specific concept in the presence of cyclic ALN -concept descriptions*, In: Proceedings of the 22nd German Annual Conference

on Artificial Intelligence, Lect. Notes Comput. Sci. **1504** (1998), pp. 129–140.

18. F. Baader and R. Küsters, *Matching in description logics with existential restrictions*, In: Proceedings of the 7th International Conference on Principles of Knowledge Representation and Reasoning (KR-2000), 2000, pp. 261–272.

19. F. Baader and R. Küsters, *Unification in a description logic with transitive closure of roles*, In: R. Nieuwenhuis and A. Voronkov (eds.), Proceedings of the 8th International Conference on Logic for Programming, Artificial Intelligence and Reasoning (LPAR 2001), Lect. Notes Artif. Intell., 2001.

20. F. Baader and R. Küsters, *Unification in a description logic with inconsistency and transitive closure of roles*, In: Proceedings of the 2002 International Workshop on Description Logics (DL 2002), 2002. [http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/]

21. F. Baader, R. Küsters, A. Borgida, and D. L. McGuinness, *Matching in description logics*, J. Log. Comput. **9** (1999), no. 3, 411–447.

22. F. Baader, R. Küsters, and R. Molitor, *Computing least common subsumers in description logics with existential restrictions*, In: Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99), 1999, pp. 96–101.

23. F. Baader, R. Küsters, and R. Molitor, *Rewriting concepts using terminologies*, In: Proceedings of the 7th International Conference on Principles of Knowledge Representation and Reasoning (KR-2000), 2000, pp. 297–308.

24. F. Baader, R. Küsters, and F. Wolter, *Extensions to description logics*, In: F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider (eds.), *The Description Logic Handbook: Theory, Implementation, and Applications*, Cambridge, Cambridge Univ. Press, 2003, pp. 219–261.

25. F. Baader and P. Narendran, *Unification of concept terms in description logics*, In: H. Prade (ed.), Proceedings of the 13th Eur. Conference on Artificial Intelligence (ECAI-98), John Wiley & Sons, 1998, pp. 331–335.

26. F. Baader and P. Narendran, Unification of concepts terms in description logics. J. Symbolic Comput. **31** (2001), no. 3, 277–305.

27. F. Baader and W. Nutt, *Basic description logics*, In: F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider (eds.), *The Description Logic Handbook: Theory, Implementation, and Applications*, Cambridge, Cambridge Univ. Press, 2003, pp. 43–95.

28. F. Baader and U. Sattler, *Expressive number restrictions in description logics*, J. Log. Comput., **9** (1999), no. 3, 319–350.

29. F. Baader and U. Sattler, An overview of tableau algorithms for description logics. Studia Logica, 69:5–40, 2001.

30. F. Baader, B. Sertkaya, and A.-Ya. Turhan, *Computing the least common subsumer w.r.t. a background terminology*, In: J. J. Alferes and J. A. Leite (eds.), Proceedings of the 9th European Conference on Logics in Artificial Intelligence (JELIA 2004), Lect. Notes Artif. Intell. **3229** (2004), 400–412.

31. F. Baader and A.-Ya. Turhan, *On the problem of computing small representations of least common subsumers*, In: Proceedings of the 25th German Conference on Artificial Intelligence (KI 2002), Lect. Notes Artif. Intell. **2479** (2002), pp. 99–113.

32. S. Bechhofer, I. Horrocks, C. Goble, and R. Stevens, OilEd, *A reasonable ontology editor for the semantic web*, In: F. Baader, Gerhard Brewka, and Thomas Eiter (eds.), KI 2001: Advances in Artificial Intelligence, Berlin, Springer, Lect. Notes Artif. Intell. **2174**, 2001, pp. 396–408.

33. T. Berners-Lee, J. A. Hendler, and O. Lassila, *The semantic Web*, Sci. American, **284** (2001), no. 5, 34–43.

34. P. Blackburn, M. de Rijke, and Yde Venema, *Modal Logic*, Cambridge, Cambridge Univ. Press, 2001, Cambridge Tracts Theoret. Comput. Sci. **53**.

35. R. Bogusch, B. Lohmann, and W. Marquardt, *Computer-aided process modeling with MODKIT*, In: Proceedings of Chemputers Europe III, Frankfurt, 1996.

36. A. Borgida, R. J. Brachman, D. L. McGuinness, and L. A. Resnick, *CLASSIC: A structural data model for objects*, In: Proceedings of the ACM SIGMOD International Conference on Management of Data, 1989, pp. 59–67.

37. A. Borgida and D. L. McGuinness, *Asking queries about frames*, In: Proceedings of the 5th International Conference on the Principles of Knowledge Representation and Reasoning (KR-96), 1996, pp. 340–349.

38. A. Borgida and P. F.Patel-Schneider, *A semantics and complete algorithm for subsumption in the CLASSIC description logic*, J. Artif. Intell. Res. **1** (1994), 277–308.

39. R. J. Brachman and H. J. Levesque Eds. *Readings in Knowledge Representation*, Morgan Kaufmann, 1985.

40. R. J. Brachman and J. G. Schmolze, An overview of the KL-ONE knowledge representation system. Cognitive Science, 9(2):171–216, 1985.

41. S. Brandt, *Implementing matching in ALE –first results*, In: Proceedings of the 2003 International Workshop on Description Logics (DL2003), CEUR Electronic Workshop Proceedings, 2003. [http://CEUR-WS.org/Vol-81/]

42. S. Brandt, R. Küsters, and A.-Ya. Turhan, *Approximation and difference in description logics*, In: D. Fensel, F. Giunchiglia, D. McGuiness, and M.-A. Williams (eds.), Proceedings of the 8th International Conference on Principles of Knowledge Representation and Reasoning (KR2002), San Francisco, CA, Morgan Kaufmann, 2002, pp. 203–214.

43. S. Brandt and H. Liu, *Implementing matching in ALN*, In: Proceedings of the KI-2004 Workshop on Applications of Description Logics (KI-ADL-04). CEUR Electronic Workshop Proceedings, 2004. [http://CEUR-WS.org/Vol-115/]

44. S. Brandt and A. -Ya. Turhan, *Using nonstandard inferences in description logics – what does it buy me?* In: Proceedings of the KI-2001 Workshop on Applications of Description Logics (ADL-01). CEUR Electronic Workshop Proceedings, 2001. [http://CEUR-WS.org/Vol-44/]

45. P. Bresciani, E. Franconi, and S. Tessaris, *Implementing and testing expressive description logics: Preliminary report*, In: Proceedings of the 1995 Description Logic Workshop (DL-95), 1995, pp. 131–139.

46. M. Buchheit, F. M. Donini, and A. Schaerf, *Decidable reasoning in terminological knowledge representation systems*, J. Artif. Intell. Res. **1** (1993), 109–138.

47. M. Chein and M.-L. Mugnier, *Conceptual graphs: Fundamental notions*, Revue d'Intelligence Artificielle, **6** (1992), no. 4, 365–406.

48. W. W. Cohen and H. Hirsh, *Learning the CLASSIC description logics: Theoretical and experimental results*, In: J. Doyle, E. Sandewall, and P. Torasso (eds.), Proceedings of the 4th International Conference on the Principles of Knowledge Representation and Reasoning (KR-94), 1994, pp. 121–133.

49. F. Donini, *Complexity of reasoning*, In: F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider (eds.), The Description Logic Handbook: Theory, Implementation, and Applications, Cambridge, Cambridge Univ. Press, 2003, pp. 96–136.

50. F. M. Donini, B. Hollunder, M. Lenzerini, A. M. Spaccamela, D. Nardi, and W. Nutt, *The complexity of existential quantification in concept languages*, Artif. Intell., (1992), no. 2–3, 309–327.

51. F. M. Donini, M. Lenzerini, D. Nardi, and A. Schaerf, *Deduction in concept languages: From subsumption to instance checking*, J. Log. Comput., **4** (1994), no. 4, 423–452.

52. M. R. Garey and D. S. Johnson, Computers and Intractability – A guide to NP-completeness. W. H. Freeman and Company, San Francisco (CA, USA), 1979.

53. V. Haarslev and R. Möller, *High performance reasoning with very large knowledge bases: A practical case study*, In: Proceedings of the 17th International Joint Conference on Artificial Intelligence, 2001.

54. V. Haarslev and R. Möller, *RACER system description*, In: Proceedings of the 16th International Joint Conference on Automated Reasoning (IJCAR 2001), 2001.

55. Ph. Hanschke, *Specifying role interaction in concept languages*, In: Proceedings of the 3rd International Conference on the Principles of Knowledge Representation and Reasoning (KR-92), Morgan Kaufmann, 1992, pp. 318–329.

56. B. Hollunder, *Hybrid inferences in KL-ONE-based knowledge representation systems*, In: Proceedings of the German Workshop on Artificial Intelligence, Springer-Verlag, 1990, pp.38–47.

57. B. Hollunder, *Consistency checking reduced to satis.ability of concepts in terminological systems*, Ann. Math. Artif. Intell., **18** (1996), no. 2–4, 133–157.

58. B. Hollunder and F. Baader, *Qualifying number restrictions in concept languages*, In: Proceedings of the 2nd International Conference on the Principles of Knowledge Representation and Reasoning (KR-91), 1991, pp. 335–346.

59. B. Hollunder, W. Nutt, and M. Schmidt-Schauβ, *Subsumption algorithms for concept description languages*, In: Proceedings of the 9th Eur. Conference on Artificial Intelligence (ECAI-90), Ptiman, 1990, pp. 348–353.

60. I. Horrocks, *The FaCT system*, In: H. de Swart (ed.), Proceedings of the 2nd International Conference on Analytic Tableaux and Related Methods (TABLEAUX-98), Lect. Notes Artif. Intell. **1397** (1998), pp. 307–312.

61. I. Horrocks, *Using an expressive description logic: FaCT or fiction?* In: Proceedings of the 6th International Conference on Principles of Knowledge Representation and Reasoning (KR-98), 1998, pp. 636–647.

62. I. Horrocks, *Implementation and optimization techniques*, In: F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider (eds.), *The Description Logic Handbook: Theory, Implementation, and Applications*, Cambridge, Cambridge Univ. Press, 2003, pp. 306–346.

63. I. Horrocks and P. F. Patel-Schneider, *DL systems comparison*, In: Proceedings of the 1998 Description Logic Workshop (DL-98), 1998, pp. 55–57. CEUR Electronic Workshop Proceedings, [http://ceur-ws.org/Vol-11/]

64. I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen, *From SHIQ and RDF to OWL: The making of a web ontology language*, J. Web Semantics, **1** (2003), no. 1, 7–26.

65. I. Horrocks and U. Sattler, *A description logic with transitive and inverse roles and role hierarchies*, J. Log. Comput., **9** (1999), no. 3, 385–410.

66. I. Horrocks and U. Sattler, *Ontology reasoning in the SHOQ(D) description logic* In: Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI 2001), Morgan Kaufmann, 2001.

67. I. Horrocks, U. Sattler, and S. Tobies, *Practical reasoning for very expressive description logics*, Log. J. IGPL, **8** (2003), no. 3, 239–264.

68. Ye. Kazakov and H. de Nivelle, *Subsumption of concepts in F L0 for (cyclic) terminologies with respect to descriptive semantics is PSPACE-complete*, In: Proceedings of the 2003 Description Logic Workshop (DL 2003), 2003, CEUR Electronic Workshop Proceedings [http://CEUR-WS.org/Vol-81/].

69. H. Knublauch, R. W. Fergerson, N. F. Noy, and M. A. Musen, *The Protégé OWL plugin: An open development environment for semantic web applications*, In: Proceedings of the 3rd International Semantic Web Conference, Hiroshima, Japan, 2004.

70. R. Küsters, *Characterizing the semantics of terminological cycles in ALN using .nite automata*, In: Proceedings of the 6th International Conference on Principles of Knowledge Representation and Reasoning (KR-98), 1998, pp. 499–510.

71. R. Küsters, *Non-standard Inferences in Description Logics*, Lect. Notes Artif. Intell. **2100** (2001).

72. R. Küsters and A. Borgida, *What's in an attribute? Consequences for the least common subsumer*, J. Artif. Intell. Res. **14** (2001), 167–203.

73. R. Küsters and R. Molitor, Approximating most specific concepts in description logics with existential restrictions. In: F. Baader, G. Brewka, and T. Eiter (eds.), Proceedings of the Joint German/Austrian Conference on Artificial Intelligence (KI 2001), Lect. Notes Artif. Intell. **2174** (2001), pp. 33–47.

74. R. Küsters and R. Molitor, *Computing least common subsumers in ALE N*, In: Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI 2001), 2001, pp. 219– 224.

75. C. Lutz, *Complexity of terminological reasoning revisited*, In: Proceedings of the 6th International Conference on Logic for Programming and Automated Reasoning (LPAR-99), Lect. Notes Artif. Intell. **1705** (1999), 181–200.

76. R. MacGregor, *The evolving technology of classification-based knowledge representation systems*, In: J. F. Sowa (ed.), *Principles of Semantic Networks*, Morgan Kaufmann, 1991, pp. 385–400.

77. W. Marquardt, L. von Wedel, and B. Bayer, *Perspectives on Lifecycle Process Modeling*, In: Proceedings of the .fth International Conference on Foundations of Computer-Aided (FOCAPD-99), Breckenridge, Colorado, USA, 1999. Process Design.

78. E. Mays, R. Dionne, and R. Weida, *K-REP system overview*, SIGART Bull., **2** (1991), no. 3.

79. M. Minsky, *A framework for representing knowledge*, In: J. Haugeland (ed.), Mind Design. The MIT Press, 1981. [A longer version appeared in The Psychology of Computer Vision (1975). Republished in [39].]

80. R. Molitor, Unterst. utzung der Modellierung verfahrenstechnischer Prozesse durch Nicht-Standardinferenzen in Beschreibungslogiken (Supporting the Modelling of of Chemical Processes by Using Non-Standard Inferences in Description Logics) [In German], PhD thesis, LuFG Theoretical Computer Science, RWTH-Aachen, Germany, 2000.

81. R. Möller and V. Haarslev, *Description logic systems*, In: F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider (eds.), The Description Logic Handbook: Theory, Implementation, and Applications, Cambridge: Cambridge Univ. Press, 2003, pp. 282–305.

82. B. Nebel, *Terminological reasoning is inherently intractable*, Artif. Intell. **43** (1990), 235–249.

83. Ch. Peltason, *The BACK system – an overview*, SIGART Bull. **2** (1991), no. 3, 114–119.

84. M. R. Quillian, *Word concepts: A theory and simulation of some basic capabilities*, Behavioral Sci. **12** (1967), 410–430. [Republished in [39]]

85. A. Rector and I. Horrocks, *Experience building a large, re-usable medical ontology using a description logic with transitivity and concept inclusions*, In: Proceedings of the Workshop on Ontological Engineering, AAAI Spring Symposium (AAAI-97), Stanford, CA, 1997. AAAI Press.

86. S. W. Reyner, *An analysis of a good algorithm for the subtree problem*, SIAM J. Comput., **6** (1977), no. 4, 730–732.

87. U. Sattler, *Terminological Knowledge Representation Systems in a Process Engineering Application*, PhD thesis, LuFG Theoretical Computer Science, RWTH Aachen, Germany, 1998.

88. K. Schild, *A correspondence theory for terminological logics: Preliminary report*, In: Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI-91), 1991, pp. 466– 471.

89. M. Schmidt-Schauβ and G. Smolka, *Attributive concept descriptions with complements*, Artif. Intell. **48** (1991), no. 1, 1–26.

90. S. Schultz and U. Hahn, *Knowledge engineering by large-scale knowledge reuse–experience from the medical domain*, In: A. G. Cohn, F. Giunchiglia, and B. Selman (eds.), Proceedings of the 7th International Conference on Principles of Knowledge Representation and Reasoning (KR-2000), Morgan Kaufmann, 2000, pp.601–610.

91. G. Teege, *Making the Difference: A Subtraction Operation for Description Logics*, In: J. Doyle, E. Sandewall, and P. Torasso (eds.), Proceedings of the 4th International Conference on the Principles of Knowledge Representation and Reasoning (KR-94), Bonn, Germany, 1994. Morgan Kaufmann, 1994, pp. 540–550.

92. A. -Ya. Turhan and Ch. Kissig, *Sonic–nonstandard inferences go oiled*, In: D. Basin and M. Rusinowitch (eds.), Proceedings of the 2nd International Joint Conference on Automated Reasoning (IJCAR-04), Lect. Notes Artif. Intell. **3097** (2004), pp. 321–325.

93. A. -Ya. Turhan and Ch. Kissig, *Sonic–system description*, In: Proceedings of the 2004 International Workshop on Description Logics (DL2004). CEUR Electronic Workshop Proceedings, [http://CEUR-WS.org/Vol-104/] 2004.

94. L. von Wedel and W. Marquardt, *ROME: A Repository to Support the Integration of Models over the Lifecycle of Model-based Engineering Processes*, In: Proceedings of the 10th European Symposium on Computer Aided Process Engineering (ESCAPE-10), 2000.

95. W. A. Woods and J. G. Schmolze, *The KL-ONE family*, In: F. W. Lehmann (ed.), *Semantic Networks in Artificial Intelligence*, Pergamon Press, 1992, pp. 133–178. [Published as a special issue of Computers & Mathematics with Applications, **23**, no. 2–9.]

Problems in the Logic of Provability

Lev Beklemishev [†]

Steklov Mathematical Institute RAS
Moscow, Russia

Utrecht University
Utrecht, The Netherlands

Albert Visser

Utrecht University
Utrecht, The Netherlands

In the first part of the paper we discuss some conceptual problems related to the notion of proof. In the second part we survey five major open problems in Provability Logic as well as possible directions for future research in this area.

[†] Supported by the Russian Foundation for Basic Research and the Council for Support of Leading Scientific Schools.

Mathematical Problems from Applied Logics. New Logics for the XXIst Century.
Edited by Dov M Gabbay *et al*./ International Mathematical Series, Springer, 2005

1. Introduction

Provability logic was conceived by Kurt Gödel in 1933 [**43**], but it really took off in the seventies as a study of modal logics with provability interpretations. After about thirty years of fruitful development this area now finds itself in a transitional period. On the one hand, many of the problems originally perceived by the founders of the discipline have been successfully solved. On the other hand, new challenges are coming from the other areas of Logic and Computer Science. Presently, provability logic starts crossing the original borders of its domain and expands in several novel directions.

The original motivation for the study of provability as a modality was mainly philosophical in nature.[1] Provability logic for the first time provided a mathematically robust intended semantics of modality. Traditionally, modal logics were used to explicate informal and sometimes inherently vague notions such as necessity, belief, obligation, etc., often plagued by paradoxes. These logics were then provided with formal Kripke-style semantics along the chain

intended semantics – logic axioms – Kripke semantics,

in which only the second link was robust.[2]

In contrast, provability is a notion for which there is a formal as well as an informal understanding. Moreover, there is a widely accepted belief that the formal notion of provability in some respects adequately represents the informal one.[3] Therefore, philosophers such as Willard Van Orman Quine, Saul Kripke and George Boolos were for the first time in a position to study a modality for

[1] Therefore, it is not surprising that the axioms of provability logic first appeared in a treatise on ethics by Smiley [**79**].

[2] Sometimes also a direct link between intended semantics and Kripke semantics was established, but it had to be informal as well.

[3] For a more detailed discussion of this claim see below.

which all the three links above could be subject to a rigorous mathematical analysis.[4]

Very soon some mathematical logicians, with their own set of concerns, working in the USA, the Netherlands, Italy and the Soviet Union, started to get interested in the topic. They saw the potential usefulness of provability logic as a tool in the study of formal axiomatic theories. Perhaps the earliest example of such an application was the so-called de Jongh–Sambin fixed point theorem, which clarified the reason why certain fixed point equations in arithmetic had explicit and unique solutions. Some other applications followed (see [**4, 82, 46**]). However, most of the emphasis in the study of provability logics from the late 70s until the late 90s was on the arithmetical completeness results.

Solovay proved, in a famous paper from 1976 [**83**], that the propositional Gödel–Löb logic **GL** is complete w.r.t. the provability semantics. Subsequent research mainly concerned with the possibility of extending Solovay's theorems to more expressive languages, such as the language of predicate logic, propositional language with several modalities, the language of interpretability logic, second order propositional logic language, and some others. In this way, the natural borders of the discipline were roughly mapped. By now we have a reasonably good idea which of the above mentioned logics with provability semantics are manageable (are decidable, have nice axiomatizations, models, etc.) and which are not.[5] By the late 90s the time for autonomous development of provability logic was over. The time has come for new challenges and search for new applications.

Two such new directions of research emerged in the recent years. The first one is the so-called *logic of proofs* initiated by S. Artemov about 1994 and which since then has grown into a lively area. It was inspired by foundational concerns and the question of providing intuitionistic logic with a robust provability semantics in

[4] This analysis was particularly important for discussing the semantics of quantification in modal logic.

[5] There are some notable open questions, though, which are discussed below.

the spirit of Brouwer–Heyting–Kolmogorov interpretation.[6] The logic of proofs also has connections with several important topics in Computer Science such as lambda calculus, logics of knowledge and belief, and formal verification. Discussing numerous open questions in this interesting area would exceed the limits of the present paper. However, we refer the reader to [**3, 6**] for a comprehensive exposition and a recent survey. A list of current problems in this field can be found at the homepage of S. Artemov.[7]

The second new development, the theory of the so-called *graded provability algebras* [**12**], aims to establish links and find applications of provability logic in the mainstream proof theory tradition. Graded provability algebras reveal surprising connections between provability logic and ordinal notation systems and provide a technically simple and clean treatment of proof-theoretic results such as consistency proofs, combinatorial independent principles, etc.

In this paper we want to present some open questions in this new developing area, as well as to record some long standing problems left in traditional provability logic.

In the first, mainly philosophical, part of the paper we discuss conceptual problems related to the notion of proof, in particular the relation between formal and informal proofs. We believe that these questions are very important and interesting in their own right, and provability logic may contribute to their study in the future.

In the second, mathematical, part we formulate five major open problems left in the area of traditional provability logic and discuss some related questions. These five problems are:

(i) Provability logic of intuitionistic arithmetic.

(ii) Provability logic of bounded arithmetic.

(iii) Classification of bimodal provability logics.

[6] This problem was the main motivation for Gödel in 1933 to introduce a calculus of provability in the first place.

[7] URL `http://www.cs.gc.cuny.edu/~sartemov/`.

(iv) Decidability of the $\forall^*\exists^*$-fragment of the first order theory of Magari algebra of Peano arithmetic.

(v) Interpretability logic of all reasonable theories.

In the last section of the paper we present current problems in the area of graded provability algebras.

A compound list of the problems discussed in this paper is given in the Appendix. An online version with some additional questions is maintained by the first author.[8]

The authors would like to thank Sergei Artemov, Yuri Gurevich, Joost Joosten, and Rostik Yavorsky for useful discussions and comments.

2. Informal Concepts of Proof

The role of provability logic and its relationship with the other parts of proof theory are best to be explained by first discussing the general relations between formal and informal proofs. Our discussion will be of necessity one-sided: we concentrate on the phenomenology (representations) of proofs, but altogether ignore the questions such as validity (see, for example, [**84**] for a discussion). We are mainly interested in the aspects where a provability logic approach could be relevant.

2.1. Formal and informal provability and the problem of equivalence of proofs

The notion of axiomatic system and the associated formal notion of proof emerged at the beginning of 20th century in the hands of G. Frege, G. Peano, and D. Hilbert. Behind these notions there was an implicitly accepted thesis, emphasized by Hilbert: every sufficiently developed area of mathematics (and perhaps not only

[8] URL `http://www.phil.uu.nl/~lev/`.

of mathematics) can be axiomatized. Hence, every valid mathematical argument can be faithfully represented in a suitable formal axiomatic system.

The status, as well as the spirit, of this statement is similar to that of the Church–Turing Thesis in the theory of computation: it is a conjecture relating a robust mathematical notion (formal proof) and informal one (informal proof) that can only be verified by practice. Soon enough universal "in practice" axiomatic systems, such as Zermelo–Fraenkel set theory, were formulated, which were apparently sufficient to formalize all current mathematics. Of course, the universality of these axiom systems had to be later qualified by Gödel: no single axiom system can be universal, in an absolute sense.[9] From this point on, proof theory developed in the course of the 20th century as a study of sufficiently universal axiomatic systems and the associated concepts of proof.

What is apparent from the historical perspective is a predominant interest in one particular model of proofs (aka deductive axiomatic proofs). Of course, this is almost the only model which had a clear mathematical formulation. However, one should not forget that deductive axiomatic proofs are not the only kind of proofs around. Compare, for example, the commonsense notion of proof in natural sciences, with experiment as a major means of proof, or a well-developed concept of proof in jurisprudence having many non-traditional features such as defeasibility (see, for example, [**66**]). In contrast with these important but informal concepts of proof, probabilistic proof-like concepts encountered, in particular, in cryptography (see [**45, 44**]), do have rigorous models.[10]

[9] This contrasts with the existence of truly universal models of computation, such as Turing machines.

[10] There have also been some discussions within the logic community of the so-called "visual" proofs in geometry. It can be argued that proofs directly appealing to visual intuition form a separate class of proofs. However, it is worth remembering that getting rid of such "visual" intuitions was one of the main purposes of Hilbert's program of axiomatizing geometry, which made a strong impression on his contemporaries. There also were various interesting

On the other hand, even if one restricts attention to standard deductive axiomatic proofs only, there is a notable discrepancy between conventional, informal mathematical proofs and their formalized representations. Fully formalized proofs have become a reality with the advent of automatic provers and interactive proof assistants such as `Mizar`, `NuPrl`, and `Coq`. The difference with informal proofs becomes evident if one compares, say, a textbook proof of the fundamental theorem of algebra with its formalization (proof-object) in `Coq`. The appearance, as well as the possible uses, of both proofs are quite different. In what sense are they actually the same?

The situation is analogous with the one around the Church–Turing Thesis: there is a difference between the high-level notion of algorithm and the low-level notion of program code (or Turing machine). Therefore, both in proof theory and in computer science the problem of equivalence of proofs (respectively, of programs) arises:

Problem 1. Which derivations/programs are essentially the same, that is, represent the same informal proof (respectively, algorithm)?

Needless to say, this question is a notoriously difficult one and it may not have a unique answer. The problem of equivalence of proofs is known in proof theory for quite some time, and was advocated by Kreisel and Prawitz (see [**60, 67**]). However, remarkably little has been done on it – partly because it fell, and still falls, outside the mainstream proof theory, partly because it is a conceptual rather than a strictly mathematical problem. See Došen [**30**] for an interesting recent discussion.

attempts to look at some visual kind of proofs from the point of view of logic, see, for example, [**8, 68**].

We believe that, in the coming years, the importance of this and related questions will become more obvious to the wider community of logicians under the influences coming from Computer Science and the development of the automated deduction systems.

In principle there are two possible approaches to this problem, which we can call *bottom-up* and *top-down*. The bottom-up approach starts with a low-level notion of proof, tries to obtain more canonical representation of such proofs and looks for meaningful equivalence relations.

This is how this problem is usually perceived within the context of structural proof theory and within a related categorial proof theory approach. Attempts were made to find mathematically attractive and sufficiently broad equivalence relations on formal proofs. The first significant contribution to this problem came from Prawitz [**67**] who isolated the following notion of equivalence: two (natural deduction style) proofs are equivalent, if they normalize to the same proof. This equivalence relation is certainly very interesting, however it behaves well only for rather restricted kinds of formalisms. Already for classical propositional logic it does not really work – in fact, it identifies all proofs of a contradiction from a given hypothesis (see [**30**]).

Since the 70's the structural proof theory underwent a rapid development with the popularization of proofs-as-programs paradigm [**42**], Girard's linear logic [**39**], game semantics [**2**] and culminating in Girard's *ludics* [**41**]. It falls outside the scope of this paper to discuss possible bearings of these important and broad doctrines on the problem of equivalence of proofs.

Instead, we would like to concentrate on the opposite, top-down approach, which is how the question "'What is an algorithm?" was approached in Computer Science. Following the top-down methodology, researchers formulated more and more general models of computational processes, so that to make these abstract descriptions eventually fit the desired intuitive concept of algorithm.

2.2. Strengthening Hilbert's Thesis

It is instructive, from the point of view of the above mentioned problem in proof theory, to gain some wisdom from the debate around Church–Turing Thesis in the theory of computation.[11] Some relevant issues have been recently raised by Gurevich in connection with his Abstract State Machines (ASM) (see, for example, [**20, 19**]) and Moschovakis in connection with general recursive algorithms (see [**63**]).

Although according to the Church–Turing Thesis every computable function can be represented by a Turing machine, such a representation will not in general be faithful w.r.t. the algorithm's data and elementary steps, which, from the point of view of computational practice, is a major drawback.[12] In contrast, Blass and Gurevich in [**20**] and elsewhere convincingly argue that every algorithm can be faithfully represented *on its own level of abstraction* by a suitable ASM. Of course, this does not yet settle the equivalence of programs problem – the question is simply being translated into a similar one about ASMs. However, it reveals additional information hidden in the informal notion of algorithm, such as its "abstraction level," which narrows the gap between the algorithm's formal and informal presentations.[13]

Gurevich's Thesis, as opposed to the Church–Turing Thesis, is noticeably non-uniform: there cannot be a single ASM which could simulate any algorithm on its own level of abstraction. This is the price we pay for representing the algorithms more faithfully. Again, the situation is parallel to the non-uniform version

[11] The fact that there are challenges to the Church–Turing Thesis from various directions, including, for example, physics, shows a healthy attitude developed in Computer Science towards these matters.

[12] Some early investigations in computation theory dealt with attempts to challenge the Church–Turing Thesis by inventing computing devices working with more complex kinds of data such as labeled complexes (Kolmogorov–Uspensky machines [**58**], Schönhage storage modification machines [**73**]).

[13] What exactly is a level of abstraction remains a bit unclear. The idea is intuitive.

of Hilbert's Thesis – stating that every proof can be represented in a suitable axiomatic system – as opposed to a uniform version related to, say, a fixed system of set theory ZFC. Uniformity presupposes some kind of coding.

A close connection between Gurevich strengthening of the Church–Turing Thesis and what we dubbed Hilbert's thesis is not really accidental. They both rely on the same basic presupposition that *the data for algorithms, as well as for mathematical proofs, are faithfully representable by first order logic structures.* Blass and Gurevich [**20**] call this claim (for the case of algorithms) the *abstract state postulate.*

A further issue addressed by the ASM approach is what kind of action constitutes a possible computation step (with roughly the answer: almost anything goes). A parallel question, what constitutes an admissible proof step, received some attention in proof theory. The initial answer – only logical inference rules modus ponens and generalization are sufficient – is not really satisfactory as these steps are too restricted.

One of the reasons for Gentzen to introduce his *natural deduction* proof system was to provide a model that better fitted the actual ("natural") form of mathematical arguments. However, despite the fundamental significance and various useful applications of this approach, that particular goal was not really achieved. Partly this is due to the fact that notions such as abstraction levels (of proofs) did not play any role in his analysis. He might have uncovered a natural form of logical steps for purely propositional (in a sense, lowest level) proofs, though.

Our next problem asks whether an analog of Gurevich Thesis holds for proof systems.

Problem 2. Find a reasonable proof system, or a class of proof systems, such that every (informal) mathematical proof admits a faithful formalization *on its own level of abstraction and complexity* in a suitable system from that class.

We think that a positive answer to this question would be a significant step towards a better understanding of the problem of equivalence of proofs and similar questions relating formal and informal notions of proof.[14]

A priori it is not really clear if such a system can be formulated. The ASM approach showed its practical usefulness for the task of program specification. Similarly, what we are looking for is a language suitable for *proof specification*, with formalized proofs playing the role of implementations of informal proofs. To have convenient proof specification tools based on clear principles is of obvious importance for the development of automated deduction systems.

Specialists working in the area of automated deduction approached the problem of proof specification using the notions such as *proof-sketches* (see, for example, Barendregt [**7**]). These sketches admit more general than the elementary logical proof steps and provide a better approximation to the kind of proof format used in ordinary mathematics. At the same time, ideally, a prover must be able to automatically reconstruct a complete proof from such a proof-sketch.

2.3. Coordinate-free proof theory

As we have indicated before, standard proof systems formalizing logic, be it Hilbert-style, Gentzen sequent-style or natural deduction-style, suffer from the same drawback: they are in a sense too concrete, that is, they depend on a lot of arbitrary, and irrelevant – from the point of view of the informal proof – details of syntax. This makes methods of structural proof theory – the part of proof theory dealing with the study of concrete proof systems by essentially syntactical methods – mathematically inelegant and

[14] According to R. Thiele (*Hilbert's Twenty-Fourth Problem*, Am. Math. Month. **110**, no. 1, 1-24) Hilbert wanted to formulate the following problem as Problem 24 in his famous list: what is the simplest possible proof of a theorem? This question was found in his notes but never made it to the final list.

non-modular (see similar remarks, for example, in [**40**], although developments in linear logic, such as proof nets, do offer a real improvement in terms of elegance).

Modularity in this context means the ability to fruitfully apply a single proof-theoretic result to various formalisms. De facto, even such a basic result as the cut-elimination theorem has to be proved for every formalism anew, even if the modifications are relatively minor. In this sense, cut-elimination plays the role of a useful method rather than that of a single important result. Stating such a general cut-elimination result seems to be quite difficult, for it presupposes that one is able to formulate some kind of general conditions under which “cut-elimination” holds. Such conditions are very elusive: meaningful classifications of syntactic formalisms have not been really developed.

The problem of non-canonicity of syntax and related non-modularity problem are major methodological drawbacks of current structural proof theory – they make the methods unattractive which ultimately results in technical difficulties and lack of progress. It might be the case that these problems are caused in part by the same phenomenon: having introduced the concept of formal proof with all the non-canonical and irrelevant for the content syntactical details we are then no longer able to state proof-theoretic results in a clear, modular way.

Whatever the cause, a way out is to provide a more general, or more abstract, notion of proof. In mathematics a way to generalizations is often pointed out by an axiomatic approach.

The standard notion of formal proof is a *genetic* one – proofs are the objects constructed by certain rules from basic symbols.[15] Provability logic emerges from the idea of treating the notions of provability and proof *axiomatically* rather than genetically.

[15] The term “genetic’ as opposed to axiomatic was used by Hilbert in 1899 in his paper *On the concept of number.*

The situation when one and the same notion can be defined genetically as well as axiomatically is quite common in mathematics. A well-known example is the concept of real number, where the standard Dedekind definition is, in this sense, genetic. A well-known axiomatic definition would be the second order categorical axiomatization of the structure of reals. In an ideal situation, as in this one, axiomatic description is categorical and one obtains a perfect match between both approaches. However, this ideal cannot be always achieved.

The main advantage of axiomatic approach compared to a genetic one is that it allows more easily for generalizations of notions in question. Hence, it widens the range of applicability of the theory and clarifies its logical structure. On the other hand, genetic approach is better when it comes to questions of explicit representation and computation. A good illustration of these two different roles is the axiomatic treatment of vector algebra versus genetic matrix calculus.

As far as the study of proofs is concerned, an abstract axiomatic approach has not yet been really developed. Could there be such a thing as *coordinate-free proof theory*? We share the spirit of Hilbertian optimism and formulate a problem, which is in fact a broad program of research rather than a single question, in the form of an imperative.

Problem 3. Develop the theory of proofs on a sufficiently abstract axiomatic basis.

In particular, we hope that such a theory could potentially help to elucidate the informal notion of deductive proof and the other, non-deductive notions of proof.

The main line of development of provability logic strived to characterize axiomatically an already existing (genetically defined) notion of provability in a sufficiently strong arithmetical theory. The goal was to obtain a sound and complete system of axioms for provability. This goal was achieved by Solovay for the case of a very weak propositional language with provability modality. However, the success of propositional provability logic was undermined by non-axiomatizability results by Artemov, Vardanyan and Boolos–McGee on predicate provability logic (see [**21**]).

In hindsight it appears that the preoccupation with arithmetical completeness results in provability logic – which naturally came from the motivations discussed at the beginning of this paper and was further enhanced by the fascination with the beauty of Solovay's theorem – actually lead the researchers away from the other relevant questions such as those related to the informal concepts of proof. Hence, the idea of axiomatic reconstruction of proof theory was never pursued or posed as a problem.

Although from the very beginning there were hopes to find applications of modal logic methods in the study of formal arithmetic, yet it was never acknowledged that serious applications would require to some extent the reconstruction of the standard proof-theoretic results and that the current modal languages were way too weak for that task. An abstract approach to proof theory based on provability logic ideas would require a development of this discipline in a new direction. Of course, technical experience accumulated in this area for so many years will still be highly relevant for this program.

From this point of view, both recent developments in provability logic mentioned before – logic of proofs and provability algebras – can be seen as attempts to approach various aspects of this general problem. The logic of proofs made the first steps in characterizing axiomatically the notion of proof rather than that of provability. Provability algebraic approach, in contrast, aims at reconstructing those results in classical proof theory which are expressible in terms of the more abstract notion of provability and treating them at the same level of generality.

Having this philosophy in mind we now turn to a different topic – the long standing problems in traditional provability logic.

3. Basics of Provability Logic

In this section we briefly formulate the basic facts concerning provability logic needed to read the rest of the paper. The section is more intended to fix the notations than as a real introduction. The reader is referred to one of the introductory texts [**82, 81, 21, 22, 26**] and a survey [**6**].

The basic system of provability logic is the modal propositional Gödel-Löb logic **GL**. On top of the classical propositional calculus the modal axioms and rules of **GL** are as follows:

L1 $\vdash \varphi \Rightarrow \vdash \Box\varphi$

L2 $\vdash \Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi)$

L3 $\vdash \Box\varphi \to \Box\Box\varphi$

L4 $\vdash \Box(\Box\varphi \to \varphi) \to \Box\varphi$

The principle L4 (Löb's Principle) is interderivable with Löb's Rule:

LR $\vdash \Box\varphi \to \varphi \Rightarrow \vdash \varphi$

Löb's Rule works too when we add assumptions of the form $\boxdot\chi$, where $\boxdot\chi = (\chi \wedge \Box\chi)$. The principle L3 follows from L1, L2 and L4.

Consider a theory T into which a sufficiently strong arithmetical theory S is interpretable.[16] Specifically, we want S to be an extension of Buss's theory S^1_2 (see [**24**] or [**48**]). We assume that the axioms of T are given by a Δ^b_1-formula. We employ a fixed

[16] The formulation employing an interpretation takes care of cases like ZF which are not "really" about numbers. We need an interpretation like the von Neumann interpretation to have access to number theory.

efficient arithmetization of arithmetical concepts like the provability predicate $\mathsf{Prov}_T(x)$. These arithmetizations are employed in T via the interpreted theory S.[17]

We define a T-realization f_T of the modal language into the language of T as follows:

- $f_T(p)$ is a sentence of the language of T,
- f_T commutes with the propositional connectives,
- $f_T(\Box\varphi) := \mathsf{Prov}_T(\ulcorner f_T(\varphi)\urcorner)$.

We say that a formula φ is *arithmetically valid* in T iff, for all T-realizations f, $T \vdash f_T(\varphi)$. The *provability logic of* T, denoted $\boldsymbol{PL}_T$, is the set of all modal propositional formulas arithmetically valid in T.

It is easy to see that all theorems of **GL** are arithmetically valid in T, that is, **GL** is arithmetically sound. For a wide class of theories we also have arithmetical completeness as was shown by Solovay [**83**].

Let EA denote the Elementary Arithmetic (or $I\Delta_0 + \exp$) [**48**].

Theorem 1 (Solovay)**.** *Suppose that T interprets* EA *and that T is Σ_1-sound w.r.t. this interpretation. Then, $\boldsymbol{PL}_T = \mathbf{GL}$.*

The strength of Solovay's theorem can also be seen as a disadvantage: the provability logic of a theory gives very little information about a theory. We will see that the situation is different if we change the underlying logic, for, example, to constructive logic (see Section 4), or if we extend the modal language, for example, to the language of interpretability logic (see Section 8). Also we may build in extra variation in our notion "the logic of." For example, we may consider the principles for provability in T verifiable in a theory U. The appropriate notion here is $\boldsymbol{PL}_T(U)$, the set of all

[17] Note that, for example, in ZF we could treat syntax directly without the detour via arithmetic. Our strategy of treating syntax by *composing* a fixed arithmetization in a basic arithmetical theory with an interpretation of that theory is just a convenient design choice that goes back to Feferman's classical paper [**31**].

φ such that, for all T-realizations f, $U \vdash f_T(\varphi)$. $\boldsymbol{PL}_T(U)$ is called the *provability logic of T relative to a metatheory U*.

Solovay has also shown (the so-called *Solovay's Second Theorem*) that $\boldsymbol{PL}_T(\mathsf{TA}) = \mathbf{S}$, where TA is the set of all true arithmetical sentences, T is a sound arithmetical theory, and $\mathbf{S}$ is the extension of all theorems of $\mathbf{GL}$ by the axiom $\Box\varphi \to \varphi$ and modus ponens as the sole inference rule. A complete classification of relative provability logics (for T containing EA) was given by Beklemishev in [**9**], see also Section 6.

4. Provability Logic for Intuitionistic Arithmetic

Whereas provability logic for classical arithmetical theories turns out to be remarkably stable, as long as we restrict ourselves to the usual unimodal language, the situation for constructive arithmetical theories is spectacularly different. Different constructive theories may have different logics. Moreover, many of the principles of different logics of this kind are mutually incompatible in the sense that together they imply an iterated inconsistency statement over the minimal constructive provability logic $i\mathbf{GL}$, i.e. the Gödel–Löb logic over intuitionistic propositional logic instead of the classical one.

In our exposition we will assume that the reader is familiar to some extent with constructive logic and constructive arithmetic. The reader is referred to the excellent textbooks [**87, 88**].

The main problem we want to describe is the problem of axiomatization and decidability of the provability logic of Heyting arithmetic HA. The theory HA is defined exactly as the first order Peano arithmetic PA, with its underlying logic changed to the intuitionistic predicate logic.[18]

[18] The least element principle intuitionistically implies the law of excluded middle. So, we should employ the standard version of induction.

Problem 4. Give an axiomatization of $\boldsymbol{PL}_{\mathsf{HA}}$. Is $\boldsymbol{PL}_{\mathsf{HA}}$ decidable?

To get all the pieces on the board in a systematic way, we will backtrack a bit, to treat the simpler question: *what is the ordinary propositional logic of an arithmetical theory?* The question is, in the intuitionistic case, not trivial.

4.1. Propositional logics of arithmetical theories

The propositional logic of an arithmetical theory T is, of course, just the box-free part of $\boldsymbol{PL}_T$.

In 1969, de Jongh shows in an unpublished paper that the propositional logic of HA is precisely IPC. He uses substitutions of formulas of a complicated form. See the extended abstract [**27**]. Subsequently, the same result has been proved for many theories other than HA, see, for example, [**80, 35, 91**].

Friedman [**33**] improves de Jongh's result for propositional logic, showing that there is a substitution of Π_2-sentences σ such that, for all propositional formulas φ, $\mathsf{HA} \vdash \sigma(\varphi) \iff \mathsf{IPC} \vdash \varphi$. Thus, IPC is *uniformly complete for* Π_2*-realizations* in HA. From the algebraic point of view the result tells us that the free Heyting algebra on countably many generators can be embedded in the Lindenbaum Algebra of HA. Moreover, we may take as generators (equivalence classes of) Π_2-sentences. Visser [**93**] improves Friedman's result, employing a realization by Σ_1-sentences. The proof is verifiable in $\mathsf{HA}+\mathsf{Con}(\mathsf{HA})$. (Note that de Jongh's theorem *implies* $\mathsf{Con}(\mathsf{HA})$, so the result is, in a sense, optimal.) The proof is based on the NNIL-*algorithm*, an algorithm that is used to characterize the admissible rules for Σ_1-realizations. This last result also holds for a number of other theories, see [**92, 29**].

Let MP be Markov's Principle. Let ECT_0 be extended Church's Thesis. Smoryński has shown that the logic of $\mathsf{HA}+\mathsf{MP}$ is precisely IPC and Gavrilenko has shown that the logic of $\mathsf{HA} + \mathsf{ECT}_0$ is precisely IPC. Surprisingly, the logic of $\mathsf{HA} + \mathsf{MP} + \mathsf{ECT}_0$ turns out to be a proper extension of IPC.

Consider the following formulas χ and ρ:

- $\chi := (\neg p \vee \neg q)$,
- $\rho := [(\neg\neg\chi \to \chi) \to (\neg\neg\chi \vee \neg\chi)] \to (\neg\neg\chi \vee \neg\chi)$

Clearly, ρ is IPC-invalid. A minor adaptation of the arguments of Rose (see [**70**]) shows that ρ is in the logic of $\mathsf{HA} + \mathsf{MP} + \mathsf{ECT}_0$.

Problem 5. What is the propositional logic of

$$\mathsf{HA} + \mathsf{MP} + \mathsf{ECT}_0?$$

Let φ be an arithmetical sentence. We write $x \ \mathbf{r} \ \varphi$ for "x realizes φ" in the sense of Kleene, see [**87, 88**]. The formula "$x \ \mathbf{r} \ \varphi$" is itself an arithmetical formula. We write φ^{r} for $\exists x \, x \ \mathbf{r} \ \varphi$. By a result of Troelstra, we have that the theorems of $\mathsf{HA} + \mathsf{MP} + \mathsf{ECT}_0$ coincide with the set of φ such that $\mathsf{HA} + \mathsf{MP} \vdash \varphi^{\mathsf{r}}$. Thus, our Problem 5 can be viewed as the question what the propositional logic of the principles realized in $\mathsf{HA} + \mathsf{MP}$ is. What happens if we replace $\mathsf{HA} + \mathsf{MP}$ in this rephrased question by another theory?

Perhaps the most interesting theory for this question is the full true arithmetic TA. So, we may ask: what is the propositional logic of the set of principles realized in TA? The answer is somewhat disappointing: it is precisely classical propositional logic, since sentential excluded middle is realized over TA.[19] Upon reflection, the reason of this disappointing outcome is the fact that we did not demand a sufficiently effective connection between the realizations of propositional formulas and their realizers. To obtain more effective connections, we introduce the following notions.

- *The propositional logic of a theory T for open realizations* is the set of propositional φ such that, for all realizations f in

[19] Despite this fact, the set of sentences realized over TA is a constructive theory inconsistent with classical logic!

(possibly open) arithmetical formulas, we have $T \vdash \forall \vec{x}\ f(\varphi)$, where $\vec{x}$ consists of the free variables of $f(\varphi)$. Clearly, the propositional logic for open realizations of a theory is a sublogic of the propositonal logic of that theory for sentential realizations.

- IR, the set of *identically realizable propositional formulas* is the logic for open realizations of the set of realizable sentences of TA. I.o.w., it is the set of φ such that, for all open f, there is an n such that $n\ \mathbf{r}\ \forall \vec{x}\, f(\varphi)$.
- ER, the set of *effectively realizable propositional formulas*, is the set of φ, such that there is a recursive function F on (finite representations of) realizations such that, for all f, $F(f)\ \mathbf{r}\ f(\varphi)$.
- IER, the set of *effectively indentically realizable propositional formulas*, is the set of φ, such that there is a recursive function F on (finite representations of) open realizations such that, for all open realizations f, $F(f)\ \mathbf{r}\ \forall \vec{x}\, f(\varphi)$.
- UR, the set of *uniformely realizable propositional formulas*, is the set of φ, such that there is an n such that, for all f, $n\ \mathbf{r}\ f(\varphi)$.

It is not difficult to show that $\mathsf{UR} \subseteq \mathsf{IER} = \mathsf{ER} \subseteq \mathsf{IR}$. We are led to the following questions.

Problem 6. (*Markov*) Give a characterization of UR, ER and IR. Is UR equal to ER? Is ER equal to IR? Are these logics decidable?

We can ask analogous questions as in Problem 6 replacing Kleene's realizability by Gödel's *Dialectica* interpretation, so we arrive at the following problem, first formulated by V. Plisko.

Problem 7. (*Plisko*) Characterize the propositional logics of Gödel's *Dialectica* interpretation.

It is well known that $\mathsf{HA} + \mathsf{MP} + \mathsf{ECT}_0$ is finitely axiomatizable over the theory $\mathsf{HA} + \mathsf{ECT}_0$, to wit by primitive recursive Markov's Principle $\mathsf{MP}_{\mathsf{PR}}$ (see [**87, 88**]). By Gavrilenko's result, the propositional logic of $\mathsf{HA} + \mathsf{ECT}_0$ is IPC. By Rose's result the propositional logic of $\mathsf{HA} + \mathsf{MP} + \mathsf{ECT}_0$ is not IPC. Thus, consistent addition of one sentence may change the propositional logic of a theory. This suggest the following problem.

Problem 8. Suppose $\mathsf{HA} + A$ is consistent. Is it always the case that the propositional logic of $\mathsf{HA} + A$ is IPC?[20]

We can ask the same question for a consistent recursively enumerable T which extends HA by axioms of restricted complexity.[21]

The problems concerning the propositional logic of a theory extend to similar problems concerning the predicate logic of a theory. We will not pursue that direction of questioning in this paper.

Another extension of the questions is to ask for a description of the Lindenbaum algebra of a theory, which in this case will be a Heyting algebra. We will mention some of these questions in Section 7. Notice that a positive answer to Problem 8 implies that the Lindenbaum algebras of HA and $\mathsf{HA} + \mathsf{ECT}_0$ are not isomorphic. The problem of the admissible rules of a theory, discussed in the next subsection, is in fact a subproblem of the problem of characterizing the Lindenbaum algebra, since the admissible rules of a theory only depend on the Lindenbaum algebra.

4.2. Admissible rules

Recall that a propositional inference rule φ/ψ is admissible in a logic L, if for every substitution σ of formulas of L for propositional

[20] In this spirit, Smoryński has shown that excluded middle is not finitely axiomatizable over HA.

[21] See [**23**] for a treatment of logical complexity measures over HA.

variables, we have

$$L \vdash \sigma(\varphi) \;\Rightarrow\; L \vdash \sigma(\psi).$$

Similarly, the rule is admissible in an arithmetical theory T if, for every realization f,

$$T \vdash f(\varphi) \;\Rightarrow\; T \vdash f(\psi).$$

The simplest example of a (nontrivial) admissible rule in IPC is the *independence of premise* rule:

$$\text{IP:} \quad \mathsf{IPC} \vdash \neg\varphi \to (\psi \vee \theta) \;\Rightarrow\; \mathsf{IPC} \vdash (\neg\varphi \to \psi) \vee (\neg\varphi \to \theta).$$

A well-known result obtained by Rybakov [**71, 72**] is that the property of a rule being admissible in IPC is decidable. Visser [**97**] showed that the propositional admissible rules for HA are the same as those for IPC.[22]

It is clear that any admissible propositional inference rule φ/ψ in HA (or, equivalently, IPC) delivers a principle of the provability logic $\boldsymbol{PL}_{\mathsf{HA}}(\mathsf{TA})$ of the form $\vdash \Box\varphi \to \Box\psi$. Here TA is true arithmetic. Is the principle also in $\boldsymbol{PL}_{\mathsf{HA}}$? The answer is *yes.* Iemhoff [**50**] proved, building on work of Ghilardi [**38**], that the set of all pairs of propositional formulas φ/ψ such that $\Box\varphi \to \Box\psi$ is in $\boldsymbol{PL}_{\mathsf{HA}}$ is precisely the set of admissible rules of IPC.

Note that it follows that, corresponding to IP, we have the following principle of the provability logic of HA:

$$\vdash \Box(\neg\varphi \to (\psi \vee \theta)) \to \Box((\neg\varphi \to \psi) \vee (\neg\varphi \to \theta)).$$

[22] One can easily show that, if L is the propositional logic of a theory T, then the rules admissible in T are a subset of the rules admissible in L. Thus, Visser's result shows that the rules admissible for HA are the maximum set possible, given de Jongh's theorem. In contrast one can produce an example of an arithmetical theory for which the logic is IPC and for which the admissible rules are the *derivable rules* of IPC (see [**29**]). Thus, the admissible rules of theories T with propositional logic IPC can both be the maximal possible set and the minimal possible one.

This principle is in **GL**, but not in i**GL**, the version of **GL** with IPC as underlying logic.

Problem 9. What are the propositional admissible rules of $\mathsf{HA}+\mathsf{MP}$ and of $\mathsf{HA}+\mathsf{ECT}_0$?

For the study of the provability logic of HA it is interesting to study admissible rules for Σ_1-realizations over HA. Specifically the characterization we give below is a lemma to characterize the closed fragment of the provability logic of HA. To formulate the result we need a few preliminaries.

A NNIL-formula is a formula with no nestings of implications to the left. We take $\neg p$ to be an abbreviation of $(p \to \bot)$. So $(p \to (q \vee \neg q))$ and $\neg p$ are NNIL-formulas, and $((p \to q) \to q)$ is not a NNIL-formula. Van Benthem and Visser have shown (independently) that the NNIL-formulas are precisely the formulas preserved under taking sub-Kripke models (modulo provable equivalence). Here a submodel is a full submodel given by an arbitrary subset of the nodes, see [**93, 98, 100**]. Note that the NNIL-formulas form an analogue of the universal formulas in ordinary model theory [**100**].

The result connecting NNIL-formulas and admissible rules is as follows (see Visser [**98**]).

Theorem 2. *There is an effectively computable function* $(.)^*$ *from propositional formulas to* NNIL*-formulas such that*

(i) $(.)^*$ *is, modulo* IPC*-provable equivalence, surjective,*

(ii) φ/ψ *is admissible for* HA *w.r.t.* Σ_1*-realizations if and only if* $\mathsf{IPC} \vdash \varphi^* \to \psi$.

If we write $\varphi \mathrel{\vdash\!\!\sim}^{\Sigma_1}_{\mathsf{HA}} \psi$, for φ/ψ is admissible in HA w.r.t. Σ_1-realizations, the second part of the above result can be symbolized as follows.

$$\varphi^* \vdash_{\mathsf{IPC}} \psi \;\Leftrightarrow\; \varphi \mathrel{\vdash\!\!\sim}^{\Sigma_1}_{\mathsf{HA}} \psi.$$

In other words, the embedding $\vdash_{\mathsf{IPC}} \hookrightarrow\ \mid\!\sim^{\Sigma_1}_{\mathsf{HA}}$ has left adjoint $(.)^*$. It follows that φ^* is the best NNIL-approximation from below (in the preorder of IPC-provability) of φ. Thus, admissibility in HA w.r.t. Σ_1-realizations is, in a sense, completely characterized by the class NNIL.[23] It would be interesting to combine the results on the admissible rules of HA for arbitrary realizations and those on Σ_1-realizations.

Problem 10. Extend the language of propositional logic with a second sort of propositional variables $s_1, s_2, \ldots$ Realizations will send ordinary variables to arithmetical sentences and the new variables to Σ_1-sentences. Characterize the rules for this language admissible in HA.

4.3. The provability logic of HA and related theories

The usual process of arithmetization of syntax is constructive and therefore can be carried out in iEA. Here iEA is the intuitionistic counterpart of elementary arithmetic (Δ_0-induction plus the axiom stating that the exponentiation function is total). In particular, the provability predicate for any elementarily presented theory T can be formulated as a Σ_1-formula. Moreover, this formula satisfies the usual Löb's derivability conditions within iEA.

The definitions of provability interpretation and of provability logic of a theory relative to a metatheory carry over without any change. $\boldsymbol{PL}_{\mathsf{HA}}$ will denote the provability logic of Heyting arithmetic that we are particularly interested in.

It is not difficult to convince oneself that, once we have the derivability conditions, the proof of the fixed-point lemma, and therefore that of Löb's theorem, can be carried out in iEA. Consequently, the logic $\boldsymbol{PL}_T(i\mathsf{EA})$ contains the axioms and rules of

[23] Ordinary admissibility for HA, and, thus, IPC, also has a left adjoint. However, we know of no simple formula class characterizing this adjoint.

GL formulated over the intuitionistic propositional logic IPC. We denote this basic system by i**GL**.

It was immediately clear that $\boldsymbol{PL}_{\mathsf{HA}}$ satisfies some additional principles. A number of such independent principles were found in [**91**]. For example, HA is closed under the so-called *Markov's rule* (see [**87**]):

$$\mathsf{HA} \vdash \neg\neg\pi \Rightarrow \mathsf{HA} \vdash \pi,$$

where π is a Π_2-formula. This can be proved constructively using the so-called *Friedman–Dragalin translation.* Thus, a proof of this fact can be formalized in HA itself, therefore $\boldsymbol{PL}_{\mathsf{HA}}$ contains the principle

$$\Box\neg\neg\Box\varphi \to \Box\Box\varphi.$$

A more general provable form of the same principle is as follows:

Ma: $\Box\neg\neg(\Box\psi \to \bigvee_{i=1}^{n} \Box\varphi_i) \to \Box(\Box\psi \to \bigvee_{i=1}^{n} \Box\varphi_i)$.

The *disjunction property* for HA is the statement that, whenever $\mathsf{HA} \vdash \varphi \vee \psi$, one has $\mathsf{HA} \vdash \varphi$ or $\mathsf{HA} \vdash \psi$. This can be written down as

Dis: $\Box(\varphi \vee \psi) \to \Box\varphi \vee \Box\psi$.

However, Friedman [**34**] has shown that the proof of disjunction property cannot be formalized in HA itself.

D. Leivant found a nice weakening of the disjunction property that is already provable in HA:

Le: $\Box(\varphi \vee \psi) \to \Box(\Box\varphi \vee \psi)$.

This principle was formulated by Leivant in his PhD. thesis. For a proof of this fact see [**98**].

Leivant's principle is weakly inconsistent with excluded middle in the sense that these principles combined prove a formula of the form $\Box^n\bot$ over i**GL**. We reason in **GL** + Le. Clearly, we have $\Box(\Box\bot \vee \neg\Box\bot)$. Hence, by Le, we have that $\Box(\Box\bot \vee \Box\neg\Box\bot)$. So, by Löb's principle, we obtain $\Box^2\bot$.

The validity of Leivant's principle illustrates that $\boldsymbol{PL}_T$ is *not* monotonically increasing in T.

These sample principles do not exhaust the list of all principles valid over HA. For all we know, the $\boldsymbol{PL}_{\mathsf{HA}}$ could be complete Π_2. A list of all the principles we know at the moment of writing can be found in [**98**] or [**51**] or [**6**].

Next to the great open problem of the provability logic of HA, we may ask after the logic of all extensions of HA.

Problem 11. What is the intersection of all $\boldsymbol{PL}_T$ for recursively enumerable extensions T of HA in the language of HA?

We have precise knowledge concerning two fragments of $\boldsymbol{PL}_{\mathsf{HA}}$. In the first place, as described above, we have a precise description of all principles of the form $\Box\varphi \to \Box\psi$, for box-free φ and ψ, in $\boldsymbol{PL}_{\mathsf{HA}}$. The pairs φ, ψ occurring in these principles are precisely the pairs φ/ψ admissible in IPC. Secondly, we have a precise characterization of the closed, or letterless, fragment of $\boldsymbol{PL}_{\mathsf{HA}}$. This characterization was given by Visser in [**93**] (see also [**98**]). Visser's proof uses in an essential way the characterization given for the admissible rules of HA for Σ_1-realizations (and the fact that this result is HA-verifiable). Not much is known about other closed fragments – except in the classical case, of course.

Problem 12. What are the closed fragments of the logic $\boldsymbol{PL}_{\mathsf{HA+MP}}$ and of the logic $\boldsymbol{PL}_{\mathsf{HA+MP+ECT_0}}$?

5. Provability Logic and Bounded Arithmetic

Bounded arithmetic theories were introduced and developed by Sam Buss and others in order to capture the informal notion of feasible proof and to clarify the relationships between proof theory and computation complexity theory [**24, 48, 59**]. The most important among these theories is the system S^1_2, which corresponds to the class of polytime computable functions, and whose principal axiom schema is the induction over binary words for Σ^b_1-formulas. (Σ^b_1-formulas in the language of bounded arithmetic naturally represent NP-predicates.) There were suggestions to identify the notion of provability in S^1_2 with *feasible provability.*

Bounded arithmetic systems are important, because they allow to approach such questions as provability or unprovability of P$\neq$ NP conjecture and are related to the study of complexity of proofs. Questions of separation and axiomatizability of various bounded arithmetic theories are often highly non-trivial and depend on difficult open problems in complexity theory, for a source book see [**59**].

Several questions related to the study of provability principles in bounded arithmetic are open. The most well-known problem is whether Solovay's arithmetical completeness theorem holds for bounded arithmetic.

Problem 13. Characterize the propositional provability logic of bounded arithmetic theories such as S^1_2 and S_2.

It is known that all principles of the modal logic **GL** are valid under the arithmetical interpretation w.r.t. bounded arithmetic theories. Hence, it seems natural to conjecture that the provability logic of S^1_2 and S_2 is precisely **GL**. However, this conjecture appears to be difficult to prove.

The reason is that the standard proof of Solovay's theorem – and this is essentially the only currently known proof of that result – relies on the property of, at least sentential, *provable* $\exists\Pi^b_1$-*completeness.* Given a theory T, this property states that, for every Π^b_1-formula $\varphi(x)$,

$$T \vdash \exists x\varphi(x) \to \mathsf{Prov}_T(\ulcorner \exists x\varphi(x)\urcorner).$$

The standard formalization of the proof predicate is Σ^b_1 in the bounded arithmetic hierarchy (in fact, it is polytime computable).

Hence, $\mathsf{Prov}_T(x)$ is a $\exists\Sigma^b_1$-formula. It is known [**24**] that S^1_2 satisfies provable $\exists\Sigma^b_1$-completeness and, in particular, Löb's third derivability condition

$$T \vdash \mathsf{Prov}_T(\ulcorner\varphi\urcorner) \to \mathsf{Prov}_T(\ulcorner\mathsf{Prov}_T(\ulcorner\varphi\urcorner)\urcorner).$$

With the principle of $\exists\Pi^b_1$-completeness the situation is different. Theories proving the totality of exponentiation function, such as the elementary arithmetic EA, do satisfy this property. It is unknown whether this property holds for bounded arithmetic theories such as S^1_2 and S_2. However, there is a reason to believe that it fails, because according to a result by Razborov and Verbrugge [**90**], if it holds then P = NP$\cap$co-NP. The latter statement is one of the difficult open questions in complexity theory, but it is believed to be false.

Thus, the question whether **GL** is the logic of provability for S_2 may actually depend on complexity-theoretic assumptions. For one thing, if the answer is *no*, then S_2 does not prove the Matiyasevich–Davis–Robinson–Putnam (MDRP) theorem, that is, that every r.e. set is diophantine. This would settle a well-known difficult problem in bounded arithmetic which is open since the 80s. Indeed, MDRP theorem implies that every $\exists\Pi^b_1$-sentence is equivalent to a purely existential one, and for such sentences we have provable completeness even in S^1_2. Of course, we do not really believe that this is a plausible way to solve the MDRP problem.

A possibility remains that one can give a relatively easy proof of the arithmetical completeness theorem for **GL** without using provable $\exists\Pi^b_1$-completeness. Berarducci and Verbrugge investigated this option in their paper [**17**]. Although they involved some ingenious modifications of Solovay construction, they only succeeded in embedding very simple kinds of Kripke models into bounded arithmetic. The main question remains open.

Apart from this intriguing problem, there are some other related open questions between bounded arithmetic and provability logic.

Problem 14. Does the Friedman–Goldfarb–Harrington principle hold in S^1_2?

The *FGH principle* for an arithmetical theory T, independently proved by Friedman, Harrington and Goldfarb (see [**81, 99**]), states that for any Σ_1-sentence S there is a sentence R such that $\mathsf{Prov}_T(\ulcorner R\urcorner)$ is T-equivalent to $S \vee \mathsf{Prov}_T(\ulcorner 0 = 1\urcorner)$. In particular, for any Σ_1-sentence S such that $T \vdash \mathsf{Prov}_T(\ulcorner 0 = 1\urcorner) \to S$, S is equivalent to $\mathsf{Prov}_T(\ulcorner R\urcorner)$, for some R.

For $T = \mathsf{S}^1_2$ we deal with sentences S of the form $\exists\Sigma^b_1$, because such is the complexity of the provability predicate, and ask whether the corresponding statement holds.

The problem arises in S^1_2, since the standard proof uses a Rosser-type fixed point construction of R:

$$R \leftrightarrow \text{“}S \text{ is witnessed before } \mathsf{Prov}_T(\ulcorner R\urcorner)\text{”}$$

Such an R is $\exists\Pi^b_1$ and one then would want to apply the familiar $\exists\Pi^b_1$-completeness principle, which is not available.

A related argument was used by Friedman to prove the equivalence of the Σ_1-disjuction property and the Σ_1-reflection principle (modulo consistency). This yields a similar kind of problem in bounded arithmetic. Recall that (sentential) Γ-*reflection principle* is the schema

$$\mathsf{Rfn}_\Gamma(T)\colon\quad \mathsf{Prov}_T(\ulcorner\varphi\urcorner) \to \varphi,$$

for all sentences from a class Γ, whereas (sentential) Γ-*disjunction property* can be formalized as the schema

$$\mathsf{Dis}_\Gamma(T)\colon\quad \mathsf{Prov}_T(\ulcorner\varphi \vee \psi\urcorner) \to (\mathsf{Prov}_T(\ulcorner\varphi\urcorner) \vee \mathsf{Prov}_T(\ulcorner\psi\urcorner)),$$

for all sentences φ, ψ from Γ.

By provable $\exists\Sigma^b_1$-completeness, $\mathsf{S}^1_2 + \mathsf{Rfn}_{\exists\Sigma^b_1}(T)$ proves $\mathsf{Dis}_{\exists\Sigma^b_1}(T)$. It is also clear that the reflection schema proves $\mathsf{Con}(T)$, the consistency assertion for T. However, the opposite implication relies on $\exists\Pi^b_1$-completeness and is, thus, an open question.

Problem 15. Does $\exists\Sigma_1^b$-disjunction property for a Σ_1^b-presented theory T imply its $\exists\Sigma_1^b$-reflection principle in $\mathsf{S}_2^1 + \mathsf{Con}(T)$?

A similar question also makes sense for natural versions of the reflection and disjunction principles with free variables.

6. Classification of Bimodal Provability Logics

An obvious way to increase the expressive power of modal language is to consider several interacting provability operators, which naturally leads to *bi-* and *polymodal provability logic*.

Perhaps, the most natural provability interpretation of the polymodal language is the understanding of modalities as provability predicates in some r.e. arithmetical theories containing EA. A modal description of two such provability predicates is, in general, already a considerably more difficult task than a characterization of each one's provability logic. There is no single system that can be justifiably called *the* bimodal provability logic – rather, we know particular systems for different natural pairs of theories, and none of those systems occupies any privileged place among the others. The following question is one of the main open problems in provability logic.

Problem 16. Characterize within the lattice of bimodal logics the propositional provability logics for pairs of r.e. arithmetical theories containing a sufficiently strong fragment of arithmetic such as EA.

Although at present this problem appears to be out of reach, a lot of partial results have been obtained and it seems that a positive solution of this problem is possible. To present some details, let us first define the notion of bimodal provability logic more precisely.

The language $\mathcal{L}(\Box, \triangle)$ of bimodal provability logic is obtained from that of propositional calculus by adding two unary modal operators $\Box$ and $\triangle$. Let (T, U) be a pair of sufficiently strong r.e. arithmetical theories. An *arithmetical realization* $f_{T,U}(\varphi)$ *of a formula* φ *w.r.t.* (T, U) translates $\Box$ as provability in T and $\triangle$ as that in U while commuting with all the boolean connectives:

$$f_{T,U}(\Box\varphi) = \mathsf{Prov}_T(\ulcorner f_{T,U}(\varphi)\urcorner),$$
$$f_{T,U}(\triangle\varphi) = \mathsf{Prov}_U(\ulcorner f_{T,U}(\varphi)\urcorner).$$

The provability logic for (T, U) is the collection of all $\mathcal{L}(\Box, \triangle)$-formulas φ such that $T \cap U \vdash f_{T,U}(\varphi)$, for every arithmetical realization f. It is denoted $\boldsymbol{PL}_{T,U}$. In general, one can consider bimodal provability logics for (T, U) relative to an arbitrary metatheory V. $\boldsymbol{PL}_{T,U}(V)$ is the set of all formulas φ such that $V \vdash f_{T,U}(\varphi)$, for every arithmetical realization f. Thus, $\boldsymbol{PL}_{T,U}$ corresponds to $V = T \cap U$.

Not too much can a priori be said about $\boldsymbol{PL}_{T,U}$, for arbitrary T and U. Firstly, $\boldsymbol{PL}_{T,U}$ is closed under *modus ponens*, substitution, $\Box$- and $\triangle$-necessitation rules.[24] Secondly, $\boldsymbol{PL}_{T,U}$ has to be an extension of the following bimodal logic **CS**, whose axioms and rules (on top of the principles of **GL** formulated for $\Box$ and for $\triangle$) are as follows:

CS1 $\vdash \Box\varphi \to \triangle\Box\varphi$

CS2 $\vdash \triangle\varphi \to \Box\triangle\varphi$

CS3 $\vdash \varphi \;\Rightarrow\; \vdash \Box\varphi$

CS4 $\vdash \varphi \;\Rightarrow\; \vdash \triangle\varphi.$

[24] Notice that neither $\boldsymbol{PL}_{T,U}(T)$ nor $\boldsymbol{PL}_{T,U}(U)$ will in general be closed under both necessitation rules.

In fact, applying the so-called uniform version of Solovay's theorem Smoryński showed that **CS** is the provability logic of a particular pair of finite extensions of PA (see [**82**]).

Deeper structural information on bimodal provability logics is provided by the Classification theorem for arithmetically complete unimodal logics [**9, 14**]. With every (normal) bimodal logic L containing **CS** we can associate its □-*projection* or *type*:

$$(L)^{\Box} := \{\varphi \in \mathcal{L}(\Box) : L \vdash \triangle\varphi\}.$$

Notice that $(L)^{\Box}$ contains **GL** and is closed under modus ponens and substitution rules, but not necessarily under the necessitation.

Under the assumption of Σ_1-soundness of V the unimodal provability logic of T relative to U coincides with the type of $\boldsymbol{PL}_{T,U}(V)$:

$$\boldsymbol{PL}_T(U) = (\boldsymbol{PL}_{T,U}(V))^{\Box}.$$

The Classification theorem shows that not every extension of **GL** is materialized as the projection of a bimodal provability logic and gives us a description of all such possible projections: $\mathbf{GL}_\alpha$, $\mathbf{GL}_\beta^-$, $\mathbf{S}_\beta$, $\mathbf{D}_\beta$, $\alpha, \beta \subseteq \omega$, α r.e. and $\omega \setminus \beta$ finite (see [**14**]).

This already excludes a lot of bimodal non-provability logics and provides the first step towards a general classification. Indeed, now the problem amounts to classifying bimodal provability logics of each of these types. However, such a classification is only known for logics of type **S**: in this case a theorem due to Carlson [**25**] tells us that there is only one such provability logic. For types **D** and $\mathbf{GL}_\omega$ we know that there is more than one logic. Some further partial results in this direction were obtained by Beklemishev in [**10, 11**].

Theorem 119 of [**6**] due to the authors of this paper characterizes all possible letterless fragments of bimodal provability logics. This result is based on a related characterization of r.e. subalgebras of Magari algebras of theories due to Shavrukov (see [**76**] and [**6**] for a relationship between these problems).

Related to the general Classification problem for bimodal provability logics are the questions of characterizing the bimodal logics for specific "natural" pairs of theories. A number of results of this kind have been obtained, see [**6**] for an overview. However, some of the more exotic cases found among fragments of PA have not yet been treated. We mention the following questions.

Problem 17. Characterize the bimodal provability logics of any natural pair of theories (T, U) such that U is a Π_1-conservative extension of T, but U is not conservative over T w.r.t. boolean combinations of Σ_1-sentences.

An example of such a pair of theories (T, U) is

$$T = \mathsf{EA}_\omega = \mathsf{EA} + \mathsf{Con}(\mathsf{EA}) + \mathsf{Con}(\mathsf{EA} + \mathsf{Con}(\mathsf{EA})) + \cdots$$

and

$$U = \mathsf{EA} + \mathsf{RFN}_{\Sigma_1}(\mathsf{EA}) = I\Delta_0 + \mathsf{supexp}.$$

Problem 18. Characterize the bimodal provability logics of $(I\Sigma_1, I\Pi_2^-)$ and the other natural pairs of incomparable fragments of PA.

7. Magari Algebras

The notion of Magari algebra was introduced by Magari [**62**] under the name *diagonalizable algebra.* Given an arithmetical theory T we consider its Lindenbaum boolean algebra $\mathcal{B}_T$, that is, the set of all T-sentences modulo the equivalence relation

$$\varphi \sim_T \psi \iff T \vdash \varphi \leftrightarrow \psi.$$

The usual logical operations provide this set with the structure of a boolean algebra, in particular, the ordering relation can be defined by:

$$[\varphi]_T \leqslant [\psi]_T \iff T \vdash \varphi \to \psi,$$

where $[\varphi]_T$ denotes the $\sim_T$ equivalence class of φ.

Gödel's provability formula Prov_T correctly defines an operator

$$\Box_T : [\varphi]_T \longmapsto [\mathsf{Prov}_T(\ulcorner\varphi\urcorner)]_T$$

acting on the Lindenbaum algebra $\mathcal{B}_T$. Indeed, if $T \vdash \varphi \leftrightarrow \psi$, then $T \vdash \mathsf{Prov}_T(\ulcorner\varphi\urcorner) \leftrightarrow \mathsf{Prov}_T(\ulcorner\psi\urcorner)$, by Löb's derivability conditions. The enriched structure $\mathcal{M}_T = (\mathcal{B}_T, \Box_T)$ is called the *provability algebra* or the *Magari algebra of* T.

The language of Magari algebras generalizes that of purely propositional provability logic. In particular, by Solovay's theorem the provability logic **GL** describes the set of all identities of $\mathcal{M}_T$, for Σ_1-sound T. Solovay's second theorem implies the decidability of the purely universal theory of $\mathcal{M}_T$, under the assumption of soundness of T.

Indeed, any quantifier-free formula $A(\vec{x})$ in the language of Magari algebras can be understood as a propositional modal formula: One preserves the boolean connectives and translates equalities of terms $s = t$ as formulas $\Box(\varphi_s \leftrightarrow \varphi_t)$, where φ_s denotes a formula corresponding to the term t. Validity of the universal closure of $A(\vec{x})$ in $\mathcal{M}_T$ is, thus, equivalent to the validity of the arithmetical interpretation of $A(\vec{x})$, under every substitution of T-sentences for variables $\vec{x}$. Hence, by Solovay's theorem, the question of validity of the universal closure of A is reducible to the one whether A is provable in the logic **S**.

The problem of decidability of the full first order theory of Magari algebra of PA stood open for some time, until it was answered negatively in an important paper by Shavrukov [**78**]. In fact, for Σ_1-sound theories T the first order theory of $\mathcal{M}_T$ happens to be mutually interpretable with the theory axiomatized by all true arithmetical sentences. Hence, it is not even arithmetical.

The question remains, where the border between decidable and undecidable fragments of that theory exactly goes. The proof of Shavrukov's theorem shows that four quantifier alternations are enough to get undecidability. The most interesting question then concerns the fragment for which there is still a considerable hope for a positive result: the $\forall^*\exists^*$-fragment, in other words, the set of prenex formulas with a block of universal quantifiers followed by a block of existential ones.

Problem 19. Is the set of $\forall^*\exists^*$-formulas valid in the Magari algebra of PA decidable?

Examples of meaningful valid arithmetical principles expressed by $\forall^*\exists^*$-formulas are plentiful. The two most prominent ones are *Rosser's theorem* and the familiar *FGH-principle*.

Rosser's theorem states that for every consistent theory there is an independent sentence. Applying this to an arbitrary finite extension of T of the form $T + p$ yields the following principle:

$$\forall p\,(\Diamond p \to \exists q\,(\neg\Box(p \to q) \wedge \neg\Box(p \to \neg q))).$$

The FGH-principle implies that the set of sentences of the form $\Box_T\psi$ coincides, modulo equivalence in T, with the set of all Σ_1-sentences above $\Box_T\bot$. Whereas the set of all Σ_1-sentences does not seem to be definable in the language of Magari algebras[25], we can still formally state some nontrivial consequences of the FGH-principle, for example:

$$\forall p_1, p_2 \exists q\, \Box(\Box q \leftrightarrow (\Box p_1 \vee \Box p_2)).$$

It is worth noticing that neither in this case, nor in the case of Rosser's principle, is the Skolem function implicitly defined by these $\forall^*\exists^*$-formulas expressible by a term in $\mathcal{M}_T$. In this sense these principles are nontrivial, i.e., not expressible by $\forall^*$-formulas.

As we mentioned before, we really expect a positive solution of Problem 19. Not much is known towards a possible solution of this difficult problem. There is an interesting connection, observed in [**6**], between this problem and the problem of classification of the so-called propositional *provability logics with constants.*

Consider a tuple of arithmetical sentences $\vec{A}$. Provability logic with the constants for $\vec{A}$ is, essentially, the set of all universal

[25] In fact, it is an open question.

formulas $\forall\vec{x}\,\varphi(\vec{c},\vec{x})$ such that $\mathcal{M}_T \models \forall\vec{x}\,\varphi(\vec{A},\vec{x})$, in other words, the *universal type* of the tuple $\vec{A}$ in $\mathcal{M}_T$. A universal type is *realizable* in $\mathcal{M}_T$, if

$$\mathcal{M}_T \models \exists\vec{c}\forall\vec{x}\,\varphi(\vec{c},\vec{x}).$$

We conjecture that there is an effective description of all universal types realizable in $\mathcal{M}_T$, hence the $\forall^*\exists^*$-fragment of the first order theory of $\mathcal{M}_T$ is decidable.

Notice that the classification of universal types is basically the same kind of problem as the classification of propositional polymodal provability logics. In fact, it is very close to the classification of provability logics for a tuple of finite extensions of T. At the moment, even the simplest variant of this question – the classification of bimodal provability logics for pairs of theories of the form $(T, T + A)$, for a single sentence A – is wide open. Therefore, we formulate the following meaningful particular case as a separate problem.

Problem 20. Give an effective description of all possible provability logics with a constant for a single sentence over PA.

It is worth mentioning that Shavrukov [**76**] also essentially gave a description of all possible closed fragments of provability logics with constants for an arbitrary tuple of sentences $\vec{A}$ (or *open types* of elements $\vec{A}$ of $\mathcal{M}_T$). They can be viewed as the propositional theories in the language with variables $\vec{c}$ over **GL** which are r.e. and satisfy the so-called *strong disjunction property*, in the case T is Σ_1-sound. It is decidable whether a finitely axiomatized propositional theory satisfies this property. However, arithmetically realizable propositional theories may, in general, be infinitely axiomatized.

Apart from that, a number of particular logics for "natural" constants have been characterized [**10, 11**]. For example, if a constant c corresponds to a true Π_1-sentence that implies all finite iterations of the consistency assertion for T, then the logic of such a constant has a nice axiomatization and is decidable. Examples are the consistency statement $\mathsf{Con}(\mathsf{ZF})$ over PA or $\mathsf{Con}(I\Sigma_1)$ over EA.

Additional information on logics with constants can be extracted from the Classification theorem for unimodal provability logics [**9, 14**]. This can be done in a manner similar to our comments on the classification problem for bimodal logics of provability.

Apart from the important Problem 19 a number of other natural questions about Magari algebras remains open.

One group of questions concerns the isomorphism problem for Magari algebras. Ideally, one would want to have a complete classification of Magari algebras of theories modulo isomorphism. However, nobody believes such a sweeping classification to be possible. Rather, one is looking for interesting criteria of isomorphism and non-isomorphism of algebras. Some such criteria have been formulated by Shavrukov [**75, 77**], who has proved, roughly, that $\mathcal{M}_T$ is recursively isomorphic to $\mathcal{M}_U$, if T and U are (effectively) conservative over each other for boolean combinations of Σ_1-sentences (for example $T = I\Sigma_1$ and $U = \mathsf{PRA}$). On the other hand, the algebras will be non-isomorphic, if one of the theories proves the uniform Σ_1-reflection principle for the other (for example, $T = \mathsf{PA}$ and $U = \mathsf{ZF}$). Between these two opposite classes there are still many meaningful examples of pairs of theories, such as PA and $\mathsf{PA} + \mathsf{Con}(\mathsf{PA})$. We formulate the general question and its most obvious particular case.

Problem 21. (Isomorphism) Give sharp necessary and sufficient conditions for the isomorphism of Magari algebras of reasonable theories. In particular, are the algebras of PA and $\mathsf{PA} + \mathsf{Con}(\mathsf{PA})$ isomorphic?

Another group of questions concerns definability in the provability algebras. Very little is known about it, in particular, about the definable elements of the algebra. Some non-definability results have been obtained by Shavrukov [**77**], however the general question remains open.

Problem 22. Characterize the first order definable elements of $\mathcal{M}_{\mathsf{PA}}$.

A natural conjecture is that these are only the elements of the 0-generated subalgebra of $\mathcal{M}_{\mathsf{PA}}$, that is, the elements already definable by ground (or variable-free) terms of the structure.

As the reader may have noticed, all the questions formulated in this section, except for Problem 20, come from the deep work of Shavrukov on the theory of Magari algebras and have been formulated by him.

8. Interpretability Logic

One of the most interesting extensions of the modal language of provability logic is the extension with a binary modality which can be arithmetically interpreted as interpretability or as conservativity. The first one to consider such extensions was Švejdar in his pioneering paper [**85**]. In this paper he showed that some substantial reasoning can be represented in such logics.

The project of interpretability logic was subsequently taken up by Visser who formulated two conjectures concerning arithmetical completeness. Visser proposed a system ILM as a candidate for the interpretability logic of Peano Arithmetic and a system ILP as a candidate for the interpretability logic of Gödel–Bernays set theory. Veltman found a Kripke style semantics for these logics, which was studied in [**28**].[26] Visser's first conjecture about arithmetical completeness was proved independently by Shavrukov [**74**]

[26] The history contains an instance of the sometimes almost mystical quality of mathematico-logical research. The system for which Visser asked Veltman to produce a semantics was $\mathsf{IL(KM1)}$, a system that is weaker than ILM. During the time Veltman was inventing the semantics, Visser realized the validity of M, when analysing an argument by Montagna in a letter. (Later it would turn out that Per Lindström already knew the principle.) The miracle was that the semantics Veltman found *did* validate M, even if he didn't know this principle. It turned out that F does not imply M, even if both principles correspond to the same set of Veltman frames – an example of modal incompleteness. Later Joosten and Goris discovered the principle R when they were looking

and by Berarducci [**16**]. Visser himself proved the second conjecture, see [**94**]. The main open problem (on the arithmetical side) left open by the work of the 90's of the previous century is the question of the interpretability logic of all reasonable arithmetical theories. For a survey of the status questions, see [**56**].[27] For surveys of the whole area, the reader might consult [**26, 96**] and [**55**].

Interpretations are ubiquitous in mathematics. To mention a few examples, think of the Poincaré interpretation of two dimensional hyperbolic geometry in two dimensional Euclidean geometry, the von Neumann interpretation of number theory in set theory, the Ackermann interpretation of the theory of finite sets in arithmetic and Tarski's interpretation of arithmetic in an extension of the theory of groups with one extra constant.

The interpretations we are interested in are *relative interpretations* in the sense of Tarski, Mostowski and Robinson (see [**86**]). Consider theories U with language L_U and T with language L_T. For simplicity, we assume that L_U is a relational language. An interpretation K of U in T is given by a pair $\langle \delta(x), F\rangle$. Here $\delta(x)$ is an L_T-formula representing the *domain* of the interpretation. F is a mapping that associates to each relation symbol R of $\mathcal{L}_U$ with arity n an $\mathcal{L}_T$-formula $F(R)(x_1, \cdots, x_n)$. Here $x_1, \ldots, x_n$ are suitably chosen free variables. We translate the formulas of L_U to the formulas of L_T as follows:

- $K(R(y_1, \cdots, y_n)) := F(R)(y_1, \cdots, y_n)$
  (we do not demand that identity is translated as identity),
- K commutes with the propositional connectives,
- $K(\forall y\, \varphi) := \forall y\, (\delta(y) \to K(\varphi))$,

at a class of Veltman frames satisfying other principles. R turned out to be arithmetically valid.

[27] The survey is already slightly outdated since Joosten and Goris discovered new principles in the mean time, see [**55**].

- $K(\exists y\, \varphi) := \exists y\, (\delta(y) \wedge K(\varphi))$,

Finally, we demand that for all sentences φ which are universal closures of axioms of U, we have $T \vdash K(\varphi)$. We will write $T \rhd U$ for *K is an interpretation of U in T*. We write $T \rhd U$, for *$K : T \rhd U$, for some K*.

Interpretations are used for various purposes: to prove relative consistency, conservation results and undecidability results. The syntactical character of interpretations has the obvious advantage that it allows us to convert proofs of the interpreted theory in an efficient way into proofs of the interpreting theory.

In order to be able to reason modally about interpretability, we must "downtune" the notion to relate sentences rather than theories. This can be realized as follows. We define $\varphi \rhd_T \psi$ as: $(T + \varphi) \rhd (T + \psi)$.

To be able to iterate modalities, we must consider theories with sufficient coding potential. It is a delicate question to determine which theories are rich enough. In this section, we will not worry about finding the sharpest class. We simply will work with recursively enumerable theories which contain EA plus the Σ_1-collection principle (possibly via interpretation) and have a good coding of sequences of all objects of the domain (sequentiality). We will call these theories *reasonable.*

We consider the modal language of provability logic extended with a binary modality $\rhd$. Let a reasonable theory T be given. It is easy to see that interpretability is formalizable in T. We define $\boldsymbol{IL}_T$, the interpretability logic of T in the same way as we defined the provability logic of T. The only new feature is the clause:

- $f_T(\varphi \rhd \psi) := \ulcorner f_T(\varphi) \urcorner \rhd_T \ulcorner f_T(\psi) \urcorner$.

The following principles constitute a good starting point for both the modal and the arithmetical investigation. The logic IL is given on top of the principles of **GL** as follows:

J1 $\vdash \Box(\varphi \to \psi) \to \varphi \rhd \psi$

J2 $\vdash (\varphi \rhd \psi \ \wedge \ \psi \rhd \chi) \to \varphi \rhd \chi$

J3 $\vdash (\varphi \rhd \chi \ \wedge \ \psi \rhd \chi) \to (\varphi \vee \psi) \rhd \chi$

J4 $\vdash \varphi \rhd \psi \to (\Diamond\varphi \to \Diamond\psi)$

J5 $\vdash \Diamond\varphi \rhd \varphi$

The only surprising principle is J5. This principle is a "syntactification" of the well-known model existence lemma: if a (first order) theory is consistent then it has a model. Here "model" is replaced by "interpretation."[28]

We will name further logics by appending the names of the further principles after IL. The first principle we consider is Montagna's principle M.

M $\vdash \varphi \rhd \psi \to (\varphi \wedge \Box\chi) \rhd (\psi \wedge \Box\chi)$

This principle is valid in essentially reflexive theories. For our purposes we can simply describe these as all theories that contain Peano Arithmetic, possibly via interpretation, that have good coding of sequences and that satisfy induction *for the full language*. Examples of essentially reflexive theories are PA and ZF. Visser conjectured that $\boldsymbol{IL}_T = \mathsf{ILM}$, for all essentially reflexive Σ_1-sound T. This conjecture was proved independently by Berarducci [**16**] and Shavrukov [**74**].

The second principle we consider is the persistence principle P:

P $\vdash \varphi \rhd \psi \to \Box(\varphi \rhd \psi)$

The persistence principle is valid for interpretations in finitely axiomatized reasonable theories T. Examples of such theories are $I\Sigma_1$, ACA_0 and GB. The conjecture was proved in [**94**].

[28] To be able to verify the desired properties of the Henkin-style construction of the interpetation one needs the technology of *shortening cuts* to make up for possible lack of induction.

The most salient problem left by the history is the question concerning the interpretability principles valid in all reasonable theories.

Problem 23. Let $\boldsymbol{IL}_{\mathsf{all}}$ be the intersection of the $\boldsymbol{IL}_T$ for all reasonable T. Characterize $\boldsymbol{IL}_{\mathsf{all}}$.

A number of further principles were discovered valid in all reasonable theories. Here they are.

$$\begin{array}{ll}
\mathsf{W} & \vdash \varphi \rhd \psi \to \varphi \rhd (\psi \wedge \Box\neg\varphi) \\
\mathsf{M}_0 & \vdash \varphi \rhd \psi \to (\Diamond\varphi \wedge \Box\chi) \rhd (\psi \wedge \Box\chi) \\
\mathsf{W}^* & \vdash \varphi \rhd \psi \to (\psi \wedge \Box\chi) \rhd (\psi \wedge \Box\chi \wedge \Box\neg\varphi) \\
\mathsf{P}_0 & \vdash \varphi \rhd \Diamond\psi \to \Box(\varphi \rhd \psi) \\
\mathsf{R} & \vdash \varphi \rhd \psi \to \neg(\varphi \rhd \neg\chi) \rhd \psi \wedge \Box\chi
\end{array}$$

The principles W, M_0, W^* and P_0 were discovered by Visser, see [**95**] or [**56**]. The principle R was recently discovered by Joosten and Goris [**55, 57**].

There are many interesting particular theories which are neither reflexive nor finitely axiomatizable. For all of them the question of characterization of their interpretability logic is open. One such example is the *primitive recursive arithmetic* PRA.

Problem 24. Characterize the interpretability logic of PRA.

Beklemishev has observed that this logic is strictly stronger than $\boldsymbol{IL}_{\mathsf{all}}$ (see [**96**]). Joosten [**55**] found an appropriate frame condition for the principle found by Beklemishev and studied its relationship with the other known principles, including the so-called Zambella principle (see below).

There are various other interpretations of the $\rhd$ that give us interesting logics. We present two possibilities. First there is the notion of *local interpretability*. The theory U is locally interpretable in T, or $T \rhd^{\mathsf{loc}} U$, if every finite subtheory of U is interpretable in T. Both between finitely axiomatized reasonable theories and between essentially reflexive reasonable theories interpretability and local interpretability coincide. However, one may provide examples where the two kinds differ. We may ask the following question.

Problem 25. Let $\boldsymbol{IL}^{\mathsf{loc}}_{\mathsf{all}}$ be the logic of the principles for local interpretability valid in all reasonable theories. Characterize $\boldsymbol{IL}^{\mathsf{loc}}_{\mathsf{all}}$.

The second notion is Π_1-*conservativity.* U is Π_1-conservative over T, or $T \rhd_{\Pi_1} U$, if for all Π_1-sentences π, $U \vdash \pi \Rightarrow T \vdash \pi$. Hájek and Montagna show that ILM is complete for arithmetical interpretations for Π_1-conservativity in Σ_1-sound extensions of $I\Sigma_1$, see their paper [**47**]. This has been improved in [**15**] to extensions of the rather weak parameter-free induction schema $I\Pi_1^-$.

In the case of Primitive Recursive Arithmetic, PRA, we also get more principles. This insight is due to D. Zambella. G. Mints proved that Zambella's principle is verifiable in PRA itself, see [**15**]. So, a salient question is the following.

Problem 26. What is the Π_1-conservativity logic of PRA?

Ignatiev [**52**] studied conservativity notions for larger formula classes Π_n and Σ_n, for $n \geq 1$. He characterized the corresponding logics for almost all classes, with the exception of Σ_1 and Σ_2.

Problem 27. Characterize the logics of Σ_1- and Σ_2-conservativity over a sufficiently strong fragment of arithmetic.

9. Graded Provability Algebras

Graded provability algebras link provability logic with two other traditions in proof theory. The first one is the study of transfinite progressions of axiomatic theories by iterated reflection schemata, which goes back to Turing [**89**] and Feferman [**32**]. The second one is the ordinal analysis tradition that stems from the work of Gentzen [**36, 37**]. The goals are to gain insight into the results on proof-theoretic analysis and ordinal notation systems from a more abstract perspective, develop a framework into which different kinds of analyses naturally fit in, and ultimately approach the fundamental questions such as the problem of natural ordinal notations.

Main results in this direction achieved so far concern the analysis of Peano arithmetic and its fragments [**12, 13**]. In particular, using provability-algebraic methods a new consistency proof for Peano arithmetic by transfinite induction à la Gentzen was given. A characterization of provably total computable functions of PA and an interesting combinatorial independent principle were also derived from the graded provability algebra approach.

The main structure one deals with is the Lindenbaum algebra of a theory T enriched by operators $\langle n\rangle$, for each natural number n, which stand for *n-consistency*: $\langle n\rangle\varphi$ is the arithmetization of the statement that the theory ($T + \varphi +$ all true Π_n-sentences) is consistent. This statement is also equivalent to the uniform Π_{n+1}-reflection principle for $T + \varphi$.

The structure $\mathcal{M}_T^\infty = (\mathcal{B}_T, \langle 0\rangle, \langle 1\rangle, \ldots)$ is called the *graded provability algebra of T*.[29] Its identities constitute a polymodal provability logic **GLP** first formulated and studied by Japaridze [**54**] (see also [**53**] and [**21**]). The 0-generated subalgebra of this algebra, which can also be seen as the set of letterless formulas of **GLP** modulo provability in **GLP**, provides an ordinal notation

[29] For the purpose of the discussion below we ignore the additional sorting structure of this algebra.

system up to the ordinal ϵ_0. The associated ordering

$$\varphi <_0 \psi \iff \mathbf{GLP} \vdash \psi \to \langle 0 \rangle \varphi$$

is isomorphic to the standard one for ϵ_0, however it has a different term representation. This ordinal notation system suggested an interesting analog of Hercules–Hydra game, the so-called Worm game, which was studied in [**13**]. This game provides one of the simplest examples of combinatorial principles independent from PA.

Currently the main questions in this area concern possible generalizations of the notion of graded provability algebra to systems stronger than PA. Standard examples of such systems are: ATR_0, a fragment of the second order arithmetic with the induction axiom and arithmetical transfinite recursion schema. This system was formulated by Friedman and its proof-theoretic ordinal is Γ_0. Another prominent example is a mildly impredicative theory KP_ω, Kripke–Platek set theory with the infinity axiom. These systems were important stages in the ongoing proof-theoretic research into the ever stronger fragments of set theory and analysis currently culminating in the work of Rathjen and Arai (see [**69, 65**] for motivations and an overview).

Problem 28. Develop generalizations of the notion of graded provability algebra suitable for proof-theoretic ordinal analysis of ATR_0 and KP_ω.

The recent paper [**12**] makes the first step towards a suitable treatment of ATR_0 by developing a provability-algebraic ordinal notation system up to Γ_0. Thus, we believe that the treatment of ATR_0 and the other predicative systems by these methods is within reach.

The treatment of KP_ω seems to be a bit more problematic. The success of graded provability algebra approach in the study of formal arithmetic was based on a well-known correspondence between induction and reflection schemata in fragments of PA. In the case of fragments of KP_ω some such correspondences hold as well, however the currently known picture is far from complete. In fact, the analogy between set-theoretic and arithmetical reflection principles needs to be clarified. Therefore, as a first stage in solving Problem 28 one is confronted with the following question.

Problem 29. Develop versions of reflection principles suitable for axiomatizing fragments of KP_ω over some weak basic set theory. Clarify the analogy between set-theoretic and arithmetical reflection principles.

The latter statement deserves a comment. The idea of the analogy can be very simply explained as follows.[30] The arithmetical reflection principle asserts that if a sentence φ is provable, then φ is true. This statement could be read dually: if $\neg\varphi$ holds, then $\neg\varphi$ is consistent, i.e., $\neg\varphi$ has a model. This is the familiar form of the set-theoretic reflection principle, except for the more specialized notion of model in set theory. In fact, different kinds of models (omega-models, beta-models, etc.) yield a much richer variety of set-theoretic reflection principles. It seems very plausible that the sought generalizations of the arithmetical graded provability algebras will be obtained by considering these higher reflection principles as new modal operators.

We note in passing that analogs of Problem 29 also make sense in two different contexts: intuitionistic arithmetic and bounded arithmetic. It seems interesting to find versions of reflection principle suitable for an axiomatization of, for example, fragments S^i_2 and T^i_2 of bounded arithmetic S_2. Similarly, very little is known about fragments of Heyting arithmetic HA. Burr [**23**] suggested a hierarchy of formula classes and fragments which match classical fragments $I\Pi_n$ of PA. Can these fragments be axiomatized by suitable reflection principles over intuitionistic version of EA?

[30] Interestingly, Kreisel and Lévy [**61**] wrote that they could not agree on whether using the same term "reflection" both for the arithmetical and the set-theoretical reflection principle was a mere figure of speech and the principles had anything to do with each other.

Next we mention a simple modal logic question concerning the standard notion of (arithmetical) graded provability algebra.

Problem 30. Axiomatize the equational theory of the reduct of $\mathcal{M}_T^\infty$ in the language with only the following operations: $\top$, $\wedge$, $\langle n \rangle$, for all n.

Before presenting a motivation for this question, let us remark that the equational theory in question deals with formulas of the form $\varphi(\vec{x}) = \psi(\vec{x})$, where φ and ψ are terms in the above language. Hence, the equational theory is equivalent to a fragment of Japaridze logic **GLP** consisting of formulas of the form $\varphi(\vec{x}) \leftrightarrow \psi(\vec{x})$. We already know a lot about **GLP**, in particular that it is decidable, thus, we believe the problem to be easy. One valid principle of the equational theory is

$$\langle n \rangle (\varphi \wedge \langle m \rangle \psi) = \langle n \rangle \varphi \wedge \langle m \rangle \psi, \quad \text{for } n > m,$$

and we conjecture that it will be its principal modal axiom.

The reason to be interested in that question is twofold. Firstly, only the formulas of this restricted language seem to play a role in the provability-algebraic analysis of Peano arithmetic. Thus, using the language from the outset may further simplify some proofs.

Secondly, this language admits a wider class of arithmetical interpretations. Propositional variables can now be understood as possibly infinitely axiomatized (but elementary presented) arithmetical theories.[31] The treatment of infinitely axiomatized theories in the context of graded provability algebras seems to be useful for possible generalizations to stronger theories. For example, PA is naturally represented as a filter generated by the elements $\{\langle 0 \rangle \top, \langle 1 \rangle \top, \langle 2 \rangle \top, \ldots\}$. However, we cannot naturally define $\mathsf{Con}(\mathsf{PA})$ in the standard graded provability algebra of EA. To do so, one wants to be able to legally use expressions such as $\langle 0 \rangle \{\langle n \rangle \top : n < \omega\}$ or $\forall n \, \langle 0 \rangle \langle n \rangle \top$.

[31] In the presence of negation there is a question how to interpret negation of an infinitely axiomatized theory. However, in the restricted language all formulas are positive.

A difficulty with generalizing this basic example is that the meaning of the $\langle n \rangle$ operator applied to an infinite theory depends on a representation (numeration) of the theory rather than the theory itself. This is precisely the same problem that lead to serious difficulties with the program of classifying arithmetical sentences by transfinite progressions of iterated reflection principles. However, the advantage of graded provability algebra is that it provides numerations to sets of its elements and it has its own very weak language with particularly simple notion of definability. Thus, the problem might be resolved by allowing the use of expressions of the form $\langle n \rangle X$, where X is a definable set of elements of the graded provability algebra. The meaning of the term "definable" has to be made more precise here, but a possible candidate could be the notion of definability by purely existential formulas (in the language of graded provability algebra). Thus, we come to the following challenging question.

Problem 31. Develop a definability theory for graded provability algebras that would allow the modalities to be applicable to (definably) infinite sets of elements.

A number of other, more technical, questions concerning graded provability algebras are known. One such question concerns the topological semantics for Japaridze logic **GLP**. A nice topological semantics for **GL** was given independently by Abashidze [**1**] and Blass [**18**]. The diamond operator can be interpreted as a derived set operator in scattered topological spaces. In fact, a completeness theorem for **GL** was proved w.r.t. the order topology of the ordinal ω^ω. In the case of **GLP** the situation is more difficult, as this system is not per se Kripke complete. However, there is a closely related subsystem **GLP**$^-$ which is (see [**12**]).

Problem 32. Is there a natural topological semantics for **GLP** and **GLP**$^-$?

Another interesting question concerns the decidability of the elementary theory of the 0-generated subalgebra of the graded provability algebra of PA, which is isomorphic to the Lindenbaum algebra of the letterless fragment of **GLP**. For the language with one modality the corresponding theory is mutually interpretable with the weak monadic second order theory of order $(\omega, <)$, commonly denoted WS1S, which is decidable by a well-known theorem of Büchi [**5**]. In the case of several modal operators the question becomes more difficult, but also more interesting.

Problem 33. Is the elementary theory of the 0-generated subalgebra of the graded provability algebra of PA decidable?

The following list summarizes the problems mentioned in the paper. Their numeration is slightly different from the one used in the main text.

10. List of Problems

Informal concepts of proof

P1. ("Hilbert's 24th") What is the simplest proof of a given theorem?

P2. (Kreisel, equivalence of proofs) What proofs are essentially the same, i.e., represent the same informal proof?

P3. Find a proof system, or a class of proof systems, in which every (informal) mathematical proof can be faithfully represented on its own level of abstraction and complexity.

P4. (Coordinate-free proof theory) Develop the theory of proofs on a sufficiently abstract axiomatic basis.

P5. Develop alternative, non-deductive models of provability.

Intuitionistic arithmetic

P6. The provability logic of Heyting arithmetic HA: decidability, axiomatization.

P7. Characterize the propositional logic of $\mathsf{HA} + \mathsf{MP} + \mathsf{ECT}_0$.

P8. (Markov) Characterize the propositional logics of Kleene realizability.

P9. (Plisko) Characterize the propositional logics of Gödel's *Dialectica* interpretation.

P10. Suppose $\mathsf{HA} + A$ is consistent. Is it always the case that the propositional logic of $\mathsf{HA} + A$ is IPC?

P11. What are the propositional admissible rules of $\mathsf{HA} + \mathsf{MP}$ and of $\mathsf{HA} + \mathsf{ECT}_0$?

P12. Extend the language of propositional logic with a second sort of propositional variables $s_1, s_2, \ldots$ Realizations send ordinary variables to arithmetical sentences and the new variables to Σ_1-sentences. Characterize the rules for this language admissible in HA.

P13. What is the intersection of all provability logics for recursively enumerable extensions of HA in the language of HA?

P14. What are the closed fragments of the provability logics of $\mathsf{HA} + \mathsf{MP}$ and of $\mathsf{HA} + \mathsf{MP} + \mathsf{ECT}_0$?

Bounded arithmetic

P15. The provability logic of bounded arithmetics S^1_2 and S_2: decidability, axiomatization.

P16. Does the Friedman–Goldfarb–Harrington principle hold in S^1_2?

P17. Does $\exists\Sigma^b_1$-disjunction property for a Σ^b_1-presented theory T imply its $\exists\Sigma^b_1$-reflection principle in $\mathsf{S}^1_2 + \mathsf{Con}(T)$?

Bimodal and polymodal logics

P18. Classification of bimodal provability logics for pairs of r.e. theories containing a sufficiently strong fragment of PA.

P19. Characterize the provability logics of any natural pair of theories (T, U) such that U is a Π_1-conservative extension of T, but U is not conservative over T w.r.t. boolean combinations of Σ_1-sentences.

P20. What are the bimodal provability logics of $(I\Sigma_1, I\Pi_2^-)$ and other pairs of incomparable fragments of PA?

P21. Give an effective description of all possible provability logics with a constant for a single sentence over PA.

Magari algebras and Lindenbaum Heyting algebras

P22. Is the $\forall^*\exists^*$-fragment of the first order theory of the provability algebra of PA decidable? The same question for the $\forall^*\exists^*\forall^*$-fragment.

P23. Give sharp necessary and sufficient conditions for the isomorphism of Magari algebras of reasonable theories. In particular, are the algebras of PA and $\mathsf{PA} + \mathsf{Con}(\mathsf{PA})$ isomorphic?

P24. Characterize the first-order definable elements in the Magari algebra of PA.

P25. Characterize (r.e.) subalgebras of the Lindenbaum Heyting algebra of HA.

P26. Is the elementary theory of the Lindenbaum Heyting algebra of HA decidable? Which fragments of it are?

P27. Are the Lindenbaum Heyting algebras of HA and $\mathsf{HA} + \mathsf{RFN}(\mathsf{HA})$ isomorphic?

Interpretability logic and its kin

P28. Characterize the interpretability logic of all reasonable theories.

P29. (Ignatiev) Characterize the logics of Σ_1- and Σ_2-conservativity over PA.

P30. Characterize the interpretability logic and the Π_1-conservativity logic for PRA.

P31. Characterize the logic of the principles for local interpretability valid in all reasonable theories.

Graded Provability Algebras

P32. Develop generalizations of the notion of graded provability algebra suitable for the proof-theoretic analysis of ATR_0 and KP_ω.

P33. Develop versions of reflection principles suitable for axiomatizing fragments of KP_ω over some weak basic set theory. Clarify the analogy between set-theoretic and arithmetical reflection principles.

P34. The same question for bounded arithmetic theories S^i_2 and T^i_2 over PV and for the fragments of Heyting arithmetic HA.

P35. Axiomatize the equational theory of the reduct of graded provability algebra in the language with only the following operations: $\top$, $\wedge$, $\langle n\rangle$, for all n.

P36. Develop a definability theory for graded provability algebras that would allow the modalities to be applicable to (definably) infinite sets of elements.

P37. Find a new combinatorial independent principle that would be motivated by Kripke models for **GLP**.

P38. Is there a natural topological semantics for **GLP**?

P39. Is the elementary theory of the 0-generated subalgebra of the graded provability algebra of PA decidable?

References

1. M. A. Abashidze, *Ordinal completeness of the Gödel-Löb modal system* [in Russian], In: Intensional Logics and Logical Structure of Theories: Material from the Fourth Soviet-Finnish Symposium on Logic, Telavi, May 20-24, 1985, Tbilisi, Metsniereba, 1988, pp. 49–73.

2. S. Abramsky, *Algorithmic game semantics. A tutorial introduction*, In: H. Schwichtenberg (ed.) *et al*, Proof and System-Reliability. Proc. NATO Advanced Study Institute, Marktoberdorf, Germany, July 24 - August 5, 2001, Dordrecht, Kluwer Academic Publishers, 2002, pp. 21–47.

3. S. Artemov, *Explicit provability and constructive semantics*, Bull. Symbolic Log. **7** (2001), no. 1, 1–36.

4. S. N. Artemov, *Applications of modal logic in proof theory* [in Russian], In: Questions of Cybernetics: Nonclassical Logics and Their Applications, Moscow, Nauka, 1982, pp. 3–20.

5. S. N. Artemov and L. D. Beklemishev, *On propositional quantifiers in provability logic*, Notre Dame J. Formal Logic **34** (1993), no. 3, 401–419.

6. S. N. Artemov and L.D. Beklemishev, *Provability logic*, In: D. Gabbay and F. Guenthner (eds.), Handbook of Philosophical Logic, 2nd ed., Vol. 13, Dordrecht, Kluwer, 2004, pp. 229–403.

7. H. Barendregt, *Towards an interactive mathematical proof language*, In: F. Kamareddine (ed.), Thirty Five Years of Automath, Dordrecht, Kluwer, 2003, pp. 25–36.

8. J. Barwise and J. Etchemendy, *Visual information and valid reasoning*, In: W. Zimmerman and S. Cunningham (eds.), Visualization in Teaching and Learning Mathematics, Washington, Math. Ass. Amer. 1990, pp. 9–24.

9. L. D. Beklemishev, *On the classification of propositional provability logics* [in Russian], Izv. AN SSSR, Ser. Mat. **53** (1989), no. 5, 915–943; English transl., Math. USSR Izv. **35** (1990), 247–275.

10. L. D. Beklemishev, *On bimodal logics of provability*, Ann. Pure Appl. Logic **68** (1994), no. 2, 115–160.

11. L. D. Beklemishev, *Bimodal logics for extensions of arithmetical theories*, J. Symbolic Logic **61** (1996), no. 1, 91–124.

12. L. D. Beklemishev, *Provability algebras and proof-theoretic ordinals I*, Ann. Pure Appl. Logic **128** (2004), 103–123.

13. L. D. Beklemishev, *The Worm Principle*, Logic Group Preprint Ser. no. 219, Univ. Utrecht, March 2003. http://preprints.phil.uu.nl/lgps/.

14. L. D. Beklemishev, M. Pentus, and N. Vereshchagin, *Provability, Complexity, Grammars*, Am. Math. Soc. Transl. Series 2, **192**, 1999.

15. L. D. Beklemishev and A. Visser, *On the Limit Existence Principles in Elementary Arithmetic and Related Topics*, Tech. Report LGPS no. 224, Dept. Phil., Univ. Utrecht, 2004.

16. A. Berarducci, *The interpretability logic of Peano arithmetic*, J. Symbolic Logic **55** (1990), 1059–1089.

17. A. Berarducci and R. Verbrugge, *On the provability logic of bounded arithmetic*, Ann. Pure Appl. Logic **61** (1993), 75–93.

18. A. Blass, *Infinitary combinatorics and modal logic*, J. Symbolic Logic **55** (1990), 7611–778.

19. A. Blass and Yu. Gurevich, *Algorithms vs. Machines*, Bull. Europ. Ass. Theoret. Comput. Sci. **77** (200), June, 96–118.

20. A. Blass and Yu. Gurevich, *Algorithms: A Quest for Absolute Definitions*, Bull. Europ. Ass. Theoret. Comput. Sci. **81** (2003), Oct., 195–225.

21. G. Boolos, *The Logic of Provability*, Cambridge, Cambridge Univ. Press, 1993.

22. G. Boolos and G. Sambin, *Provability: the emergence of a mathematical modality*, Studia Logica **50** (1991), no. 1, 1–23.

23. W. Burr, *Fragments of Heyting arithmetic*, J. Symbolic Logic **65** (2000), no. 3, 1223–1240.

24. S. Buss, *Bounded arithmetic*, Napoli, Bibliopolis, 1986.

25. T. Carlson, *Modal logics with several operators and provability interpretations*, Israel J. Math. **54** (1986), 14–24.

26. D. de Jongh and G. Japaridze, *The logic of provability*, In: S. R. Buss (ed.), Handbook of Proof Theory, Stud. Logic Found. Math. **137** Amsterdam, Elsevier, 1998, pp. 475–546.

27. D. H. J. de Jongh, *The maximality of the intuitionistic predicate calculus with respect to Heyting's Arithmetic*, J. Symbolic Logic **36** (1970), 606.

28. D. H. J. de Jongh and F. Veltman, *Provability logics for relative interpretability*, In: [**64**, pp. 31–42].

29. D. H. J. de Jongh and A. Visser, *Embeddings of Heyting algebras*, In: [**49**, pp. 187–213].

30. K. Došen, *Idedntity of proofs based on normalization and generality*, Bull. Symbolic Log. **9** (2003), 477–503.

31. S. Feferman, *Arithmetization of metamathematics in a general setting*, Fundamen. Math. **49** (1960), 35–92.

32. S. Feferman, *Transfinite recursive progressions of axiomatic theories*, J. Symbolic Logic **27** (1962), 259–316.

33. H. Friedman, *Some applications of Kleene's methods for intuitionistic systems*, In: A. R. D. Mathias and H. Rogers (eds.), Cambridge Summerschool in Mathematical Logic, Berlin etc., Springer-Verlag, 1973, pp. 113–170.

34. H. Friedman, *The disjunction property implies the numerical existence property*, Proc. Nat. Acad. USA, **72** (1975), 2877–2878.

35. Yu. V. Gavrilenko, *Recursive realizability from the inuitionistic point of view*, Soviet. Math. Dokl. **23** (1981), 9–14.

36. G. Gentzen, *Die Wiederspruchsfreiheit der reinen Zahlentheorie*, Math. Ann. **112** (1936), no. 4, 493–565.

37. G. Gentzen, *Neue Fassung des Wiederspruchsfreiheitsbeweises für die reine Zahlentheorie*, Forsch. Logik Grund. Wiss. **4** (1938), 19–44.

38. S. Ghilardi, *Unification in intuitionistic logic*, J. Symbolic Logic **64** (1990), 859–880.

39. J.-Y. Girard, *Linear logic*, Theoret. Comput. Sci. **50** (1987), 1–102.

40. J.-Y. Girard, *Proof Theory and Logical Complexity.* Studies in Proof Theory. Monographs **1**, Napoli, Bibliopolis, 1987.

41. J.-Y. Girard, *Locus solum: From the rules of logic to the logic of rules*, Math. Structures Comput. Sci. **11** (2001), no. 3, 301–506.

42. J.-Y. Girard, Y. Lafont, and P. Taylor, *Proofs and Types*, Cambridge, Cambridge Univ. Press, 1989.

43. K. Gödel, *Eine Interpretation des intuitionistischen Aussagenkalkuls* Ergebnisse Math. Kolloq. (1933), 39–40.

44. O. Goldreich, *Zero-knowledge twenty years after its invention*, Tutorial, URL: http://www.wisdom.weizmann.ac.il/ oded/zk-tut02.html, 2004.

45. S. Goldwasser, S. Micali, and C. Rako, *The knowledge complexity of interactive proof systems*, SIAM J. Comput. **18** (1989), 186–208.

46. S. V. Goryachev, *On interpretability of some extensions of arithmetic* [in Russian], Mat. Zametki, **40** (1986), 561–572.

47. P. Hájek and F. Montagna, *The logic of* Π_1*-conservativity*, Arch. Math. Logik **30** (1990), 113–123.

48. P. Hájek and P. Pudlák, *Metamathematics of First Order Arithmetic*, Berlin etc., Springer-Verlag, 1993.

49. W. Hodges, M. Hyland, C. Steinhorn, and J. Truss (eds.), *Logic: from Foundations to Applications*, Oxford, Clarendon Press, 1996.

50. R. Iemhof, *On the admissible rules of intuitionistic propositional logic*, J. Symbolic Logic **66** (2001), no. 1, 281–294.

51. R. Iemhof, *Provability Logic and Admissible Rules*, PhD Thesis, Univ. Amsterdam, 2001.

52. K. N. Ignatiev, *Partial Conservativity and Modal logics*, ITLI Prepublication Ser. X-91-04, Univ. Amsterdam, 1991.

53. K. N. Ignatiev, *On strong provability predicates and the associated modal logics*, J. Symbolic Logic **58** (1993), 249–290.

54. G. K. Japaridze, *The modal logical means of investigation of provability* [in Russian], Thesis in Phil., Moscow, 1986.

55. J. J. Joosten, *Interpretability Formalized*, PhD Thesis, Univ. Utrecht, 2004.

56. J. J. Joosten and A. Visser, *The interpretability logic of all reasonable arithmetical theories*, Erkenntnis, **53** (2000), no. 1-2, 3–26.

57. J. J. Joosten and A. Visser, *How to derive principles of interpretability logic. A toolkit*, In: J. van Benthem, A. Troelstra, F. Veltman, and A. Visser (eds.), Liber Amicorum for Dick de Jongh. ILLC, hhttp://www.illc.uva.nl/D65/i, 2004.

58. A. Kolmogorov and V. Uspensky, *On the definition of algorithm* [in Russian], Uspekhi Mat. Nauk **13** (1958), no. 4, 3–28; English transl.: Am. Math. Soc. Transl. **29** (1963), 217-245.

59. J. Krajíček, *Bounded Arithmetic, Propositional Logic, and Complexity Theory*, Cambridge, Cambridge Univ. Press, 1995.

60. G. Kreisel, *A survey of proof theory. II*, In: Proc. 2nd Scandinav. Logic Symposium (Univ. Oslo, Oslo, 1970), Stud. Logic Found. Math. **63** Amsterdam, Elsevier, 1971, pp. 109–170.

61. G. Kreisel and A. Lévy, *Reflection principles and their use for establishing the complexity of axiomatic systems*, Z. Math. Logik **14** (1968), 97–142.

62. R. Magari, *The diagonalizable algebras (the algebraization of the theories which express Theor.:II)*, Boll. Unione Mat. Ital., Ser. 4, **12** (1975). Suppl. fasc. 3, 117-125.

63. Y. Moschovakis, *What is an algorithm?* In: B. Engquist and W. Schmid (eds.), Mathematics Unlimited, Berlin, Springer-Verlag, 2001, pp. 919–936.

64. P. P. Petkov (ed.), *Mathematical Logic, Proc. the Heyting 1988 summer school in Varna, Bulgaria*, Plenum Press, 1990.

65. W. Pohlers, *Subsystems of set theory and second order number theory*, In: S. R. Buss (ed.), Handbook of Proof Theory, Stud. Logic Found. Math. **137** Amsterdam, Elsevier, 1998, pp. 210–335.

66. H. Prakken, *Logical Tools for Modelling Legal Argument*, A Study of Defeasible Reasoning in Law, Dordrecht, 1997.

67. D. Prawitz, *Ideas and results in proof theory*, In: Proc. 2nd Scandinav. Logic Symposium (Univ. Oslo, Oslo, 1970), Stud. Logic Found. Math. **63** Amsterdam, Elsevier, 1971, pp. 235–307.

68. N. Preining, *Sketch-as-proof*, In: G. Gottlob, A. Leitsch, and D. Mundici (eds.), Computational Logic and Proof Theory, Lect. Notes Comput. Sci. **1289** (1997), pp. 264–277.

69. M. Rathjen, *Recent advances in ordinal analysis:* Π^1_2*−CA and related systems*, Bull. Symbolic Log. **1** (1995) no. 4, 468–485.

70. G. F. Rose, *Propositional calculus and realizability*, Trans. Am. Math. Soc. **61** (1953), 1–19.

71. V. V. Rybakov, *A criterion for admissibility of rules in the modal system S4 and intuitionistic logic*, Algebra Logic **23** (1984), 369–384.

72. V. V. Rybakov, *Admissibility of Logical Inference Rules*, Amsterdam, Elsevier, 1997.

73. A. Schönhage, *Storage modification machines*, SIAM J. Comput. **9** (1980), 490–508.

74. V. Yu. Shavrukov, *The logic of relative interpretability over Peano arithmetic* [in Russian], Tech. Report No. 5, Steklov Math. Institute, Moscow, 1988.

75. V. Yu. Shavrukov, *A note on the diagonalizable algebras of PA and ZF*, Ann. Pure Appl. Logic **61** (1993), 161-173.

76. V. Yu. Shavrukov, *Subalgebras of Diagonalizable Algebras of Theories Containing Arithmetic*, Disser. Math., no. 323, 1993.

77. V. Yu. Shavrukov, *Isomorphisms of diagonalizable algebras*, Theoria **63** (1997), no. 3, 210–221.

78. V. Yu. Shavrukov, *Undecidability in diagonalizable algebras*, J. Symbolic Logic **62** (1997), no. 1, 79–116.

79. T. Smiley, *The logical basis of ethics*, Act. Phil. Fennica **16** (1963), 237–246.

80. C. Smoryński, *Applications of Kripke models*, In: [**87**], pp. 324–391.

81. C. Smoryński, *Fifty years of self-reference*, Notre Dame J. Formal Logic **22** (1981), 357–374.

82. C. Smoryński, *Self-Reference and Modal Logic*, Berlin, Springer-Verlag, 1985.

83. R. M. Solovay, *Provability interpretations of modal logic*, Israel J. Math. *28* (1976), 33–71.

84. G. Sundholm, *Proofs as acts versus proofs as objects: Some questions for Dag Prawitz*, Theoria **64** (1998), 187–216.

85. V. Švejdar, *Modal analysis of generalized Rosser sentences*, J. Symbolic Logic **48** (1983), 986–999.

86. A. Tarski, A. Mostowski, and R. M. Robinson, *Undecidable Theories*, Amsterdam, North-Holland, 1953.

87. A. Troelstra, *Metamathematical Investigations of Intuitionistic Arithmetic and Analysis*, Berlin, Springer-Verlag, Lect. Notes **344**, 1973.

88. A. Troelstra and D. van Dalen, *Constructivism in Mathematics, Vols. 1, 2*, Amsterdam, North-Holland, 1988.

89. A. M. Turing, *System of logics based on ordinals*, Proc. London Math. Soc., Ser. 2 **45** (1939), 161–228.

90. R. Verbrugge, *Efficient Metamathematics*, PhD Thesis, Univ. Amsterdam, 1993.

91. A. Visser, *Aspects of Diagonalization and Provability*, PhD Thesis, Univ. Utrecht, 1981.

92. A. Visser, *On the completeness principle*, Ann. Math. Logic **22** (1982), 263–295.

93. A. Visser, *Evaluation, provably deductive equivalence in Heyting's Arithmetic of substitution instances of propositional formulas*, Logic Group Preprint Ser. 4, Dept. Phil., Univ. Utrecht, 1985.

94. A. Visser, *Interpretability logic*, In: [**64**, pp. 175–209].

95. A. Visser, *The formalization of interpretability*, Studia Logica **51** (1991), 81–105.

96. A. Visser, *An overview of interpretability logic*, In: M. Kracht, M. de Rijke, H. Wansing, and M. Zakhariaschev (eds.), Advances in Modal Logic, Vol. 1, CSLI Lect. Notes, **87** (1998), 307–359.

97. A. Visser, *Rules and arithmetics*, Notre Dame J. Formal Logic **40** (1999), no. 1, 116–140.

98. A. Visser, *Substitutions of Σ^0_1-sentences: Explorations between intuitionistic propositional logic and intuitionistic arithmetic*, Ann. Pure Appl. Logic **114** (2002),no. 1-3, 227–271.

99. A. Visser, *Faith and falsity*, Ann. Pure Appl. Logic, **131** (2005), 103–131.

100. A. Visser, J. van Benthem, D. de Jongh, and G. R. de Lavalette, *NNIL, a study in intuitionistic propositional logic* In: A. Ponse, M. de Rijke, and Y. Venema (eds.), Modal Logic and Process Algebra, a Bisimulation Perspective, CSLI Lect. Notes **53** (1995), 289–326.

Open Problems in Logical Dynamics

Johan van Benthem

*University of Amsterdam
Amsterdam, The Netherlands*

*Stanford University
Stanford, USA*

In recent years, a number of "dynamic epistemic logics" have been developed for dealing with information, communication, and interaction. This paper is a survey of conceptual issues and open mathematical problems emanating from this development.

1. Logical Dynamics

The traditional paradigm of logic is drawing a conclusion from some given premises. But derivation from data already at our disposal is just one way in which information can be obtained. We can also observe new facts, or just ask some better-informed person whom we trust. Concomitantly with all this information

**Mathematical Problems from Applied Logics. New Logics for the XXIst Century.
Edited by Dov M Gabbay *et al.*/ International Mathematical Series, Springer, 2005**

flow, our knowledge and beliefs change, and this adaptation process may even be triggered by further cues. Such cognitive actions are of logical interest per se, and their explicit study and its various repercussions has been described as a “Dynamic Turn” in logic [**14**]. In particular, relevant actions in this broader setting need not be single-agent tasks such as drawing a conclusion or observing some fact. After all, perhaps the simplest logical scenario for getting or giving information is *asking a question*. But this essentially involves information flow between two agents, and their mutual epistemic and “social” interactions as the question is asked and an answer is given.

An excellent framework for multi-agent dynamic behavior in communication is *epistemic logic* (introduced in Section 2), suitably “dynamified” by using ideas from the *dynamic logic* of actions. Section 3 is about the best-explored system of this kind, viz. the dynamic logic of *public announcements*. Section 4 generalizes this to general *dynamic-epistemic logic* of actions or events whose observation conveys information. The resulting technical questions here blend into issues about more classical logical systems, which are discussed in Section 5 on first-order and fixed-point logics. But knowledge and ignorance are not the only attitudes of participants in a conversation. They also have beliefs about their current situation and expectations about the future. These are revised as observation and communication take place. Thus, epistemic dynamics runs into belief revision. Section 6 is devoted to links between dynamic-epistemic logic and *belief revision theory* as developed in AI and related areas.

Next, there is also a longer term to information flow beyond individual update or revision steps. For, these do not occur in isolation. There is a history of past interactions, determining our trust in our interlocutor, as well as a future of things yet to be said. Eventually, this calls for a merge of epistemic and a *temporal logic* which allows for statements of regularities over time, often in the form of “protocols” (Section 7). Another longer-term perspective concerns the *purpose* of a question. Behind every ordinary question there is a "Why" meta-question: what is the point, and what are people trying to achieve? This leads to current *game logics* for describing strategic behavior in games, of which there exist quite a few by now. These are just some of the current bridges between logic and game theory, which deserve a separate treatment. Section 8 provides a very brief introduction with pointers. Finally, Section 9 summarizes a few general issues playing across all of the above phenomena.

A word of clarification. "Open problems" in new areas like this may be of several different sorts. Some are clear-cut mathematical questions, where all notions have crystallized out, and what remains to be done is some hard-nosed *theorem proving.* But other significant questions concern *mathematical modeling*, finding perspicuous formal mechanisms for information update, or temporal evolution, or say, the logical behavior of different types of agent. Often, questions like this are triggered by the challenge of describing some communicative practice, or some given game. Other questions at this conceptual level have to do with relating different logical paradigms trying to describe the same phenomena. Third, and finally, there are interesting and highly nontrivial questions of *computational implementation* and *cognitive realism* in studying the fit between all these logics and actual behavior of men and machines. In this survey, the main emphasis will be on the first type of question, but there are a few of the second kind as well. We only mention issues of the third kind in passing.

Finally, a caveat. This paper is not a self-contained introduction to logics for epistemic update and games. It is rather intended for readers with at least some background in the area, who can then see a coherent panorama of directions to be pursued.

2. Standard Epistemic Logic

The basis for all our further topics is standard epistemic logic, created originally by Hintikka for analyzing philosophical issues in epistemology – but linked more closely with computer science, and even economics, for a long time now. Excellent introductions to epistemic logic with a computational slant are [**48**] and [**56**]. For modal logic in general, which serves as a sort of mathematical laboratory, see [**35**].

2.1. Language

Definition 1 (*standard language*). The standard syntax of epistemic logic has a propositional base with modal operators $K_i\varphi$ ("i know that φ"), C_G ("φ is *common knowledge* in group G"):

$$p \,|\, \neg\varphi \,|\, \varphi \vee \psi \,|\, K_i\varphi \,|\, C_G\varphi.$$

We write $\langle i\rangle$ for the dual modal existential statement $\neg K_i\neg\varphi$; which says that agent i considers φ possible. The existential dual of C_G is written $\langle C_G\rangle\varphi$. □

Example 1 (*questions and answers*). Let $\boldsymbol{Q}$ ask a factual question "*P?*", where $\boldsymbol{A}$ answers truly: "*Yes.*" A presupposition for giving a normal truthful answer is that $\boldsymbol{A}$ knows that P: $K_{\boldsymbol{A}}P$. The question itself, if it is a normal co-operative one, also conveys its own presuppositions, such as

(i) $\neg K_{\boldsymbol{Q}}P \vee \neg K_{\boldsymbol{Q}}\neg P$ ("$\boldsymbol{Q}$ does not know if P") and

(ii) $\langle \boldsymbol{Q}\rangle(K_{\boldsymbol{A}}P \vee K_{\boldsymbol{A}}\neg P)$ ("$\boldsymbol{Q}$ thinks $\boldsymbol{A}$ may know the answer").

After the whole two-step communication episode, P has become common knowledge among $\boldsymbol{Q}$, $\boldsymbol{A}$: $C_{\{Q,\boldsymbol{A}\}}P$. Note the crucial role of epistemic iterations: knowledge that agents have about each others knowledge or ignorance, and also the "group knowledge" achieved at the end. □

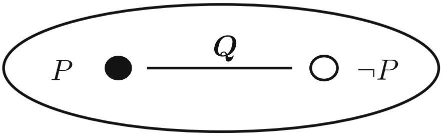
Another group notion is "distributed knowledge"' $D_G\varphi$, which holds intuitively when agents in G put their information together. Finally, epistemic logics can be extended by strengthening their operators in many ways, just as in modal logic in general (cf. [**34**]).

2.2. Semantics

Definition 2 (*models and truth definition*). Models $\boldsymbol{M}$ for the language are triples $(W, \{\sim_i \mid i \in G\}, V)$, where W is a set of worlds, the $\sim_i$ are binary accessibility relations between worlds, and V is a propositional valuation. The major epistemic truth conditions are as follows:

$$\begin{aligned} \boldsymbol{M}, s \models K_i\varphi \quad &\text{iff} \quad \text{for all } t \text{ with } s \sim_i t : \boldsymbol{M}, t \models \varphi, \\ \boldsymbol{M}, s \models C_G\varphi \quad &\text{iff} \quad \text{for all } t \text{ reachable from } s \text{ by some finite} \\ &\qquad \text{sequence of } \sim_i \text{ steps } (i \in G) : \boldsymbol{M}, t \models \varphi. \end{aligned}$$ □

Example 2. Here is a simple epistemic model:



In the black world, the following are true:

$$P, \quad K_{\boldsymbol{A}}P, \quad \neg K_{\boldsymbol{Q}}P \wedge K_{\boldsymbol{Q}}\neg P, \quad K_{\boldsymbol{Q}}(K_A P \vee K_{\boldsymbol{A}}\neg P),$$
$$C_{\{\boldsymbol{Q},\boldsymbol{A}\}}(\neg K_{\boldsymbol{Q}}P \wedge \neg K_{\boldsymbol{Q}}\neg P), \quad C_{\{\boldsymbol{Q},\boldsymbol{A}\}}(K_{\boldsymbol{A}}P \vee K_{\boldsymbol{A}}\neg P).$$ □

Common knowledge acts as the dynamic logic modality

$$\left[\left(\cup_{i \in G} \sim_i\right)^*\right]\varphi.$$

This is a reflexive transitive closure. Finally, distributed knowledge involves intersection of accessibility relations:

$$\boldsymbol{M}, s \models D_G\varphi \quad \text{iff} \quad \text{for all } t \text{ with } s \bigcap_{i \in G} \sim_i t : \boldsymbol{M}, t \models \varphi.$$ □

2.3. Basic model theory

A basic notion states when two epistemic models represent the same informational situation from the viewpoint of our language.

Definition 3 (*epistemic bisimulation*). A *bisimulation* between epistemic models $\boldsymbol{M}$, $\boldsymbol{N}$ is a binary relation $\equiv$ between states m, n in $\boldsymbol{M}$, $\boldsymbol{N}$ such that, whenever $m \equiv n$, then

(a) m, $n$satisfy the same proposition letters,

(b1) if $m\,R\,m'$, then there is a world n' with $n\,R\,n'$ and $m' \equiv n'$,

(b2) the same "zigzag clause" in the opposite direction. □

Every model $(\boldsymbol{M}, s)$ has a smallest bisimilar $(\boldsymbol{N}, s)$, its "bisimulation contraction." The latter is the simplest representation of the epistemic information in $(\boldsymbol{M}, s)$.

Fact 1 (*invariance for bisimulation*). *For every bisimulation E between two models $\boldsymbol{M}$, $\boldsymbol{N}$ with sEt, s, t satisfy the same formulas in the epistemic language with common knowledge.*

Theorem 1 (*epistemic definability of models*). *Each finite model $(\boldsymbol{M}, s)$ has an epistemic formula $\delta(\boldsymbol{M}, s)$ (involving common knowledge) such that the following assertions are equivalent for all models $\boldsymbol{N}$, t,*

(a) $\boldsymbol{N}, t \models \delta(\boldsymbol{M}, s)$,

(b) *$\boldsymbol{N}, t$ has a bisimulation $\equiv$ with $\boldsymbol{M}$, s such that $s \equiv t$.*

For a proof cf. [**15**] and [**10**]. Thus, there is a strongest epistemic assertion one can make about states in a current model. In particular, each world in a bisimulation contraction has a unique epistemic definition inside that model. Also, it follows that, at least when comparing finite models, epistemic equivalence of worlds amounts to the existence of a bisimulation connecting them. For infinite models, the situation is more complex, but this technical line of research is not relevant to us here. Bisimulation is the basic structural equivalence between epistemic models. It plays the same role as potential isomorphism in first-order logic, and like the latter, it has Ehrenfeucht-Fraïssé–style game versions. Many meta-properties of first-order logic, such as interpolation or preservation theorems, transfer to basic epistemic logic without common knowledge by bisimulation-based arguments. For the complicating role of common knowledge, cf. Section 4 below.

Remark 1. The language with distributed knowledge is richer. Standard modal bisimulations do not preserve statements with intersection modalities, such as $\langle R \cap S \rangle p$. □

2.4. Axiomatics

The complete logics for these models are the usual ones (cf. the cited references). Minimal modal $\boldsymbol{K}$ arises for knowledge if nothing is required of the accessibility relations. In that case, the K-operator is better read as *belief.* But most often in this paper, we think of accessibilities as equivalence relations, in which case the complete logic is multi-$\boldsymbol{S5}$. The axioms for $C_G\varphi$ resemble those of dynamic logic, and can be found in the literature. The axioms for $D_G\varphi$ also resemble standard modal ones.

2.5. Complexity

Model checking of epistemic formulas in a given finite models takes polynomial time $\boldsymbol{P}$, just as for modal formulas in general. The complexity of the satisfiability problem for epistemic formulas is ***NP**-complete* in the case of a single agent, but with two or more agents, it becomes ***Pspace**-complete*, just like for the minimal modal logic – and even ***Exptime**-complete* when common knowledge is added to the language.

2.6. Open problems, even here

Static epistemic logic may seem a closed chapter of research. But there are clear open ends, related to alternatives for the above semantics. Here are three of many examples.

(a) The universal quantifier truth condition for knowledge has an existential quantifier competitor, taking knowledge of φ as the *existence* of compelling *evidence* for φ. Thus, epistemic logic might include an explicit calculus of evidence and reasons (cf. [**13**]). One such system is the "logic of proofs" of [**2**], while an account of groups and epistemic actions is in [**3**].

(b) A second alternative are topological models, where a universal modality $K\varphi$ describes the interior of the set of points in a model verifying φ. Van Benthem and Sarenac [**33**] show that this gives much more freedom of epistemic expression. For example, unlike relational semantics, topological models can separate the different intuitions concerning common knowledge distinguished in [**8**].

(c) Finally, knowledge in natural usage also takes more object-like arguments, such as "know who did it," "know his telephone number" or "know how to do the job." Such settings are central to epistemic logic in philosophy. No compact useful systems of this sort have been developed, though there is, of course, "epistemic predicate logic" doing part of the job.

But now, let us move to where we really want to go: namely, the "dynamic turn" putting actions that convey and change information at center stage.

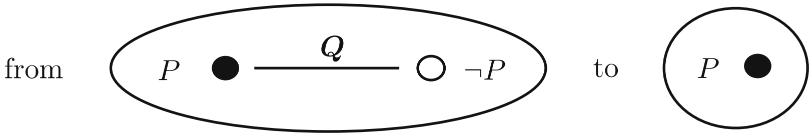
3. Public Announcement: Epistemic Logic Dynamified

The basic paradigm of an information-changing action is saying something in public. In the simplest case, someone says something she knows, and this occurs in public, well-understood by all members of the relevant group. The resulting epistemic dynamics has been studied since around 1990. It is a surprisingly rich pilot case for the true realities of everyday communication, where people say things on weaker grounds, with a certain amount of hiding and secrecy. In this section, we give basic definitions and known results, and then discuss a number of interesting questions that have remained open till to-day. For further background in dynamic logic of programs, cf. [**62**]. The specific open questions discussed here continue an earlier broader survey in [**22**].

3.1. World elimination: the system *PAL*

An answer "Yes" to a question "P?" is a public announcement of the proposition P in the group $\{\boldsymbol{Q}, \boldsymbol{A}\}$. Such an announcement changes the initial model.

Example 3 (***answering a question***)**.** With the model of Example 2, announcing P would just leave the black world:



To the right, $C_{\{\boldsymbol{Q},\boldsymbol{A}\}}P$ holds. □

Thus, public announcement involves world elimination:

Definition 4. For any model $\boldsymbol{M}$, world s, and formula P true at s, $(\boldsymbol{M} \mid P, s)$ ("$\boldsymbol{M}$ relativized to P at s") is the submodel of $\boldsymbol{M}$ whose domain is $\{t \in \boldsymbol{M} \mid \boldsymbol{M}, t \models P\}$. □

This simple process describes many epistemic puzzles and scenarios in the literature. We have a universe of information states (static epistemic models) related by possible transitions: viz. announcement actions $P!$ taking models $\boldsymbol{M}$ to submodels $\boldsymbol{M} \mid A$.

Definition 5 (*PAL language and semantics*). The language of *public announcement logic PAL* is the above epistemic language, with added action expressions

Formulas	P:	$p \mid \neg\varphi \mid \varphi \vee \psi \mid K_i\varphi \mid C_G\varphi \mid [A]\varphi$
Action expressions	A:	$P!$

The semantic clause for the dynamic modality is as follows:

$$\boldsymbol{M}, s \models [P!]\varphi \quad \text{iff} \quad \textit{if } \boldsymbol{M}, s \models P, \textit{ then } \boldsymbol{M} \mid P, s \models \varphi. \qquad \square$$

Actually, there are two natural kinds of model here. One is the universe of all epistemic models. But smaller natural models can also be "conversation spaces" with just some initial model plus all its updates by successive "admissible assertions."

Many notions from standard epistemic model theory apply in this dynamic setting. For example, public announcement respects bisimulation in the sense of [**14**]:

> If $\boldsymbol{M}, s$ has a bisimulation with $\boldsymbol{N}, t$, then, for any epistemic assertion P, $\boldsymbol{M} \mid P, s$ has a bisimulation with $\boldsymbol{N} \mid P, t$.

Restricting the given bisimulation to these two submodels is a bisimulation between the updated models. Thus, the language of PAL is invariant for epistemic bisimulations in the earlier sense.

Statements like $[P!]K_i\varphi$ or $[P!]\, C_G\varphi$ state what agents know after announcements. In this manner, the complete logic for PAL is an exact mathematical calculus of communication.

Theorem 2. *PAL without common knowledge is axiomatized completely by the usual axioms for epistemic logic plus the following reduction axioms*:

$$[P!]q \quad \leftrightarrow \quad P \to q \quad \textit{for atomic facts } q$$

$$\begin{array}{lcl}
[P!]\neg\varphi & \leftrightarrow & P \rightarrow \neg[P!]\varphi \\[2ex]
[P!]\varphi \wedge \psi & \leftrightarrow & [P!]\varphi \wedge [P!]\psi \\[2ex]
[P!]K_i\varphi & \leftrightarrow & P \rightarrow K_i[P!]\varphi
\end{array}$$

These axioms provide an obvious reduction procedure taking any dynamic PAL formula to an equivalent one in static epistemic logic. It follows that

Corollary 1. *PAL is decidable.*

As it stands this translation is exponential, but [**65**] proposes a more efficient one for SAT purposes, showing that satisfiability for PAL remains ***Pspace***-complete. Incidentally, model checking for PAL is in $\boldsymbol{P}$ – as observed by several people.

The full language of PAL with common knowledge raises some complications. There is no obvious reduction axiom for assertions $[P!]\, C_G\varphi$! This issue has been resolved only in [**61**]. First, an extension is needed of the basic epistemic language, with a new modality of *relativized common knowledge*

> $\boldsymbol{M}, s \models C_G(\varphi, \psi)$ iff ψ holds after every finite sequence of accessibility steps for agents going through φ-worlds only.

Then we do have a valid equivalence

$$[P!]\, C_G\varphi \leftrightarrow C_G(P, [P!]\varphi).$$

Relativized common knowledge is not definable in the basic epistemic language – but it is bisimulation-invariant, and existing completeness proofs are easily generalized. On this extended base, we have the following valid general reduction axiom in PAL:

Theorem 3. *PAL with relativized common knowledge is axiomatized completely by adding the reduction law*

$$[P!]\, C_G(\varphi, \psi) \leftrightarrow C_G(P \wedge [P!]\varphi, [P!]\psi).$$

Finally, the dynamic character of *PAL* can be taken further. Conversation involves sequences of assertions, governed by regular *program constructions* as in dynamic logic. These include

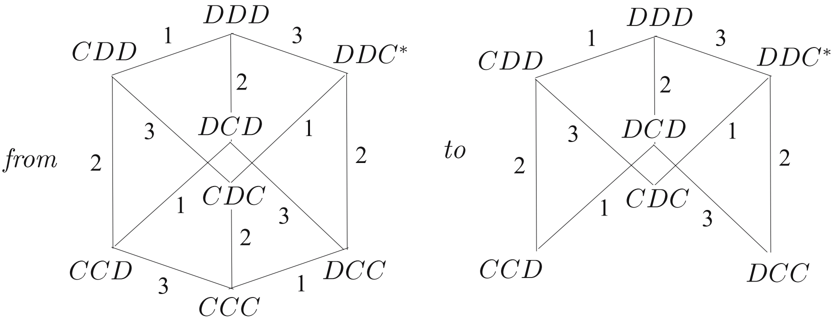
(a) sequential composition ;
(b) guarded choice *IF*... *THEN* ... *ELSE*...
(c) guarded iterations *WHILE*... *DO*...

Example 4 (*the puzzle of the Muddy Children*). Here is a simple story that occurs in many variants in the literature:

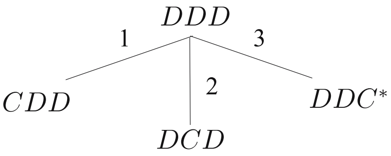
> *After playing outside, two of three children have mud on their foreheads. They all see the others, but not themselves, so they do not know their own status. Now their Father comes along and says: "At least one of you is dirty." He then asks: "Does anyone know if he is dirty?" The children answer truthfully. As this question–answer episode repeats, what will happen?*

Nobody knows in the first round. But then, the muddy children both know their status, as each of them can argue as follows. "If I were clean, the one dirty child I see would have seen only clean children around her, and so she would have known that she was dirty at once. But she did not. So I must be dirty, too!" This is symmetric for both muddy children – so both know in the second round. The third child knows it is clean one round later, after they have announced that. The puzzle easily extends to more clean and dirty children. Here is the update sequence for this particular case:

Updates start with the Father's public announcement that at least one child is dirty. This is about the simplest communicative action, and it merely eliminates those worlds from the initial model where the stated proposition is false, i.e., ***CCC*** disappears:



When no one knows his status, the bottom worlds disappear:



The final update is then to

$$DDC^*$$

Clearly, the conversation instructions in this puzzle involve all three regular program operations. □

It is easy to add such operations to create compound assertions for more complex conversational instructions. PAL must then be extended with the usual axioms for PDL. But the combination is surprising, as has been shown in [**68**]:

Theorem 4. *PAL with PDL operations is undecidable.*

Indeed, Muddy Children involves a form of parallel program composition as well, since children speak simultaneously in each round. No canonical *PAL* system has been proposed yet for dealing with this further complexity of communication.

This ends our lightning tour of *PAL*. It might seem that we now know all there is to this system, but this is far from true! Here are four excursions into the unknown.

3.2. What are the real update laws?

The earlier axiom system leaves many questions unanswered.

Generalized models

Consider restricted families of epistemic models with assertions P only admissible as long as the new model $\boldsymbol{M} \mid P$ is still inside the family – the above "conversation models." Let some assertion P be forbidden in such a model. Then the principle $P \rightarrow \langle P!\rangle\top$ fails, though it follows from the above *PAL* axioms.

What is the complete PAL logic of all generalized models?

There maybe a simple relativization trick at work here. For example, consider the reduction axiom for knowledge, in its valid existential form on standard models:

$$\langle P!\rangle\langle i\rangle\varphi \leftrightarrow P \wedge \langle i\rangle\langle P!\rangle\varphi.$$

For example, from left to right, this is invalid on our general models, since the announcement action $P!$ available in the current situation $(\boldsymbol{M}, s)$ need not be available at some shifted situation $(\boldsymbol{M}, t)$ with $s \sim_i t$. But we can put in safeguards, reformulating the axiom to a valid version which assumes that the relevant actions are present:

$$\langle P!\rangle\langle i\rangle\varphi \wedge \langle i\rangle\langle P!\rangle\top \leftrightarrow \langle P!\rangle\top \wedge \langle i\rangle\langle P!\rangle\varphi.$$

We put $\langle P!\rangle\top$ for the availability of the announcement action $P!$, not just the earlier P. We still have precondition $\langle P!\rangle\top \to P$ – but as we saw, not its converse.

Structural rules

Here is a more dynamic notion of inference in models with update (cf. [**87**] and [**14**]). Conclusion φ *follows dynamically* from $P_1, \ldots, P_k$ if, after public announcements of the successive premises, all worlds in the new information state satisfy φ. In terms of PAL, the following implication must be valid:

$$[P_1!]\,[\ldots]\,[P_k!]\,C_G\varphi.$$

This notion behaves differently from standard logic in its structural rules. Permutation of premises, contraction of the same premise, or monotonicity adding premises all fail.

Theorem 5. *The structural properties of dynamic inference are axiomatized by*:

Left Monotonicity	$X \Rightarrow A$ implies $B, X \Rightarrow A$
Cautious Monotonicity	$X \Rightarrow A$ and $X, Y \Rightarrow B$ imply $X, A, Y \Rightarrow B$
Left Cut	$X \Rightarrow A$ and $X, A, Y \Rightarrow B$ imply $X, Y \Rightarrow B$

This completeness extends to a modal language of all structural properties of announcements over the universe of all epistemic models (cf. [**26**]). But it remains to be determined how this substructural logic of announcement actions relates to substructural calculi motivated by resource management (cf. [**74**]).

Schematic validities

Unlike most standard logical calculi, PAL update logic is not *substitution-closed.* Most conspicuously, its basic axiom

$$[P!]q \leftrightarrow (P \to q)$$

for atoms q fails when we substitute arbitrary formulas φ for q. After all, the point of update is that truth values of complex assertions like “I do not know if ψ” can change.

Definition 6 (*substitution core*). The *substitution core* of update logic consists of those schemata in the language of PAL all of whose substitution instances are valid formulas. □

There are interesting principles valid in this sense, which did not surface in the earlier “formula by formula” axiom system for PAL. An example is iterated announcement:

Fact 2. *The equivalence* $[A!]\,[B!]\varphi \leftrightarrow [([A!]B)!]\varphi$ *is a valid schematic principle of* PAL.

Here is a total list of schematically valid principles for public announcement which covers all cases we have found so far.

$$\begin{array}{lcl}
[P!]T & & \\
[P!]\perp & \leftrightarrow & \neg P \\
[P!]\neg\varphi & \leftrightarrow & P \to \neg[P!]\varphi \\
[P!]\,(\varphi \wedge \psi) & \leftrightarrow & [P!]\varphi \wedge [P!]\psi \\
[P!]K_i\varphi & \leftrightarrow & P \to K_i[P!]\varphi \\
[P!]\,C_G(\varphi, \psi) & \leftrightarrow & C_G(P \wedge [P!]\varphi, [P!]\psi) \\
[A![[B!]\varphi & \leftrightarrow & [A!][B!]\varphi
\end{array}$$

Question 1. Is the substitution core of PAL decidable; or at least axiomatizable?

3.3. Model theory of learning

Understanding the effects of communicative actions can be highly nontrivial. For example, it might seem "obvious" that any public announcement of a proposition P in a group results in common knowledge of P. But there is no axiom $[P!]\, C_G P$ in the complete list for PAL. And indeed, the intuition fails. Here is a much-cited counter-example.

Example 5 (*self-refuting assertions*). Let p be the case, but I do not know this. If you know both these facts, and announce truly "You do not know that p, but it is true" ($\varphi = \neg K_{\text{you}} p \wedge p$), the current model gets updated to one with p true throughout, and p becomes common knowledge between us. But then the given statement φ becomes false by its very announcement. □

Other examples are the ignorance assertions in the puzzle of the "Muddy Children" whose repeated public announcement eventually led to common knowledge in the group. This observation raises an issue of model-theoretic preservation.

Question 2. Which syntactic shapes of epistemic formulas P are "self-fulfilling," i.e.: they guarantee common knowledge of P after their announcement?

Van Benthem [**22**] notes that self-fulfillment holds for the epistemic equivalent of "universal formulas," true in any submodel of models where they hold:

Fact 3. *All shapes generated by the following grammar are self-fulfilling:* $(\neg)p, (\neg)q, \ldots \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid K_i\varphi \mid C_G\varphi$.

Here the final clause allows arbitrary common knowledge formulas. But there are other self-fulfilling cases, at least when the accessibility relations in our models are equivalence relations. An example guaranteeing common knowledge is then $\neg K_i p$.

Remark 2 (*most general postconditions*). Van Benthem [**22**] relates the difficulty of the above preservation question to the impossibility of defining "most general postconditions," in the sense of computer science, for assertions inside PAL. Stating a fact $p!$ has a most general postcondition $C_G p$, but the argument is ad-hoc. In general, the generic description of the strongest effect of $P!$ is backward-looking: "there *was* a model from which the current one arose by announcing P." This requires *temporal* past operators beyond PAL. □

Self-fulfilling propositions arise in many ways.

Example 6 (*verificationism and learnability*). Van Benthem [**28**] discusses the Verificationist Thesis in epistemology, which holds that "all true assertions may be known." This calls for assertions which make some given truth common knowledge. There are several relevant versions of self-fulfillment then, ranging from more local to more global. Validity of $[\varphi!]\,C_G\varphi$ means that a true statement can always be learnt by announcing φ itself. But a statement φ is learnable in a weaker sense if there is some assertion ψ such that $\varphi \to [\psi!]\,C_G\varphi$ is valid. And it is learnable in a still weaker sense if that trigger ψ may depend on the model where φ is true. These notions form a strict hierarchy in models for multi-$\boldsymbol{S5}$, where they are all decidable. It is an open question whether this decidability of learnability notions extends to other model classes for epistemic logic, say $\boldsymbol{S4}$. □

More locally, in a given model, [**22**] shows that

Fact 4. *In finite models, any announcement with a proposition has an update which can be generated equivalently by a proposition which becomes common knowledge after its announcement.*

No version is known for this local result on infinite models.

3.4. Communication and planning

PAL is not just a formalism for analyzing given statements or longer conversations. It can also be used for planning assertions, just as dynamic logics of programs can be used for both analysis and design. Here is one obvious issue that arises then.

Maximal communication

Here is what a group can achieve by maximal public announcement. Epistemic agents in a model $(\boldsymbol{M}, s)$ can tell each other things they know, thereby cutting down the model to smaller sizes, until nothing changes.

Theorem 6. *Each model* $(\boldsymbol{M}, s)$ *has a unique minimal submodel reachable by maximal communication of known propositions among all agents. Up to bisimulation, its domain is the set*

$$COM(\boldsymbol{M}, s) = \Big\{ t \mid s \bigcap_{i \in G} \sim_i t \Big\}.$$

An agent j can even reach $COM(\boldsymbol{M}, s)$ by speaking just once, if she takes bisimulation contractions of updates all along the way. For then, the strongest proposition known to j corresponds to the set $\{t \mid s \sim_j t\}$. As this is definable, she can state that definition. $COM(\boldsymbol{M}, s)$ is related to the earlier notion of distributed knowledge $D_G\varphi$. Still, there remains a difference between evaluating the assertion φ in the whole model $\boldsymbol{M}$, or just within the submodel $COM(\boldsymbol{M}, s)$. The latter notion seems a better candidate for the intuitive notion of "implicit group knowledge," but it has not yet been axiomatized.

The importance of the operation of relation intersection here suggests another open problem.

Problem 1. *Find a natural extended notion of bisimulation that is characteristic for epistemic logic enriched with* $D_G\varphi$.

Planning and reachability

More delicate planning issues include announcing facts publicly between some agents while leaving some others in the dark.

Example 7 (*"Moscow Puzzle"*). 7 cards are distributed among A, B, C. A gets 3, B gets 3, C gets 1. How should A, B communicate publicly, in hearing of C, to find out the real distribution of the cards while C does not? Solutions depend on the numbers of cards (cf. [**43**]). □

The logic of hiding in public view has some initial observations so far, but no general theory. Of course, "public privacy" goes only so far, and we will look at real hiding phenomena in communication in Section 4. Generalizing from this example, one can look at any model $(\boldsymbol{M}, s)$ (perhaps inside some "conversation model"), and ask which models $(\boldsymbol{N}, t)$ satisfying certain desiderata are *reachable* from $(\boldsymbol{M}, s)$ by means of finite sequences of available assertions. Here is one technical notion related to this.

Definition 7 (*learnability*). The *learnability modality* is defined by a quantification over possible assertions:

$$\boldsymbol{M}, s \models \langle learn \rangle \varphi \quad \text{iff} \quad \begin{array}{l} \text{there is some assertion } P \\ \text{with } \boldsymbol{M}, s \models \langle P! \rangle \varphi \end{array} \qquad \square$$

Question 3. Is PAL plus the operator $\langle learn \rangle$ still decidable?

3.5. Group knowledge

Epistemic logic is mostly about individual agents and their interaction. But plural agents can also perform actions, like winning a soccer match or electing a president – and likewise, it makes sense to ascribe knowledge to plural subjects, such as groups or organizations. Common knowledge was one step in this direction, although its definition is essentially still a reduction to knowledge for individual group members. Less reductive was the notion of implicit group knowledge, which arises only if individuals cooperate and share information. One obvious way of doing this is by imagining plural epistemic subjects with internal communication channels, so that subgroups can exchange information (cf. [**22**]):

Problem 2. Develop a version of dynamic-epistemic logic of groups with communication channels as primitive entities.

In natural language, we often switch between talk of "individual" and "collective" agents – with some predicates applying only to the former, and others only to the latter. Some group predicates are straightforward lifts of individual behavior for all members ("the boys had the flu"), whereas others are not ("the prisoners liberated each other"). Thus, there are no total reductions between the two levels, and both seem essential in our way of describing the world. The same duality might apply to epistemic agents, and hence, a richer epistemic logic of collective agents and group-forming operations seems of interest.

4. Dynamic Epistemic Logic

Many forms of communication have private aspects, inadvertently, or with deliberate hiding. People read cards in public view without showing them to others, they whisper in crowded lecture theatres, and they send each other secret messages along a channel they believe to be safe. Civilized social life is full of procedures where information flows in restricted ways. Such forms of communication are high-lighted in parlour games, which have been designed to manipulate information flow. Cf. [**42**] for a complete analysis of informational moves in the popular game *Cluedo*. Such moves can be explained to a large extent in *dynamic-epistemic logic*, which provides a principled account of update of information models by "event models." The main source for this calculus is [**6**] and a forthcoming textbook is [**45**].

4.1. Information from arbitrary events: product update

The static information models provided by epistemic logic have a natural dynamic companion:

Definition 8 (*event models*). The set of relevant events A in a communication scenario forms an *event model*

$$\boldsymbol{A} = (A, \{\sim_i \mid i \in G\}, \{PRE_a \mid a \in A\}),$$

with agents' uncertainty relations $\sim_i$. The latter encode which events agents cannot distinguish. For example, when I read my card, and you know it must be either red or black, you cannot distinguish the two events of "my reading red" and "my reading black." But you can distinguish either from "my reading orange" or "Mount Etna erupting." Finally, actions a always have *preconditions* PRE_a for their being executable. We will assume that preconditions are formulated within our language. For example, "my reading red" presupposes that I hold a red card – or, as in an earlier example, my asking a question may presuppose that I do not know the answer, but think that you do. □

Here is our general way of computing a new information state.

Definition 9 (*product update*). Let models $(\boldsymbol{M}, s)$ and $(\boldsymbol{A}, a)$ be given. The product model $(\boldsymbol{MxA}, (s, a))$ has domain

$$\{(s, a) \mid s \text{ a world in } \boldsymbol{M},\ a \text{ an action in } \boldsymbol{A},\ (\boldsymbol{M}, s) \models PRE_a\},$$

and the new uncertainties satisfy

$$(s, a) \sim_i (t, b) \quad \text{iff} \quad \text{both } s \sim_i t \text{ and } a \sim_i b.$$

Thus, new uncertainty can only come from existing uncertainty via indistinguishable events. Finally, the valuation for atoms p at (s, a) is copied from that at s in $\boldsymbol{M}$. The new actual world is the pair (s, a) of the old actual world and the actual event. □

This mechanism was proposed in [**6**], building on earlier work by Gerbrandy, Groeneveld, and Plaza. It can model a wide range of phenomena, including games (cf. [**42**] and [**19**]). In particular, it can deal with misleading actions as well as truthful ones – though this requires leaving the realm of epistemic models with equivalence relations. As with public announcement, truth values of propositions can change drastically under product update.

Definition 10 (*update language and semantics*). The dynamic-epistemic language for this new setting is:

$$p \,|\, \neg\varphi \,|\, \varphi \vee \psi \,|\, K_i\varphi \,|\, C_G\varphi \,|\, [\boldsymbol{A}, a]\varphi,$$

where $(\boldsymbol{A}\,, a)$ is any finite event model with actual event a. Semantic interpretation takes place as in the above, with key clause

$$\boldsymbol{M}, s \models [\boldsymbol{A}, a]\varphi \quad \text{iff} \quad \boldsymbol{MxA}, (s, a) \models \varphi. \qquad \Box$$

Theorem 7. *Dynamic epistemic logic is effectively axiomatizable and decidable.*

Here is the key reduction axiom extending the earlier one for public announcement:

$$PRE_a \to \&\{K_i(PRE_b \to [\boldsymbol{A}, b]\varphi) \,|\, b \sim_i a \textit{ in } \boldsymbol{A}\}.$$

As before, a reduction axiom for common knowledge in subgroups requires a language extension. Van Benthem, van Eijck, and Kooi [**31**] do this in a somewhat baroque (but natural) extension of epistemic logic, which allows any PDL-style program with tests and regular operations inside the epistemic language. There are also quite different reformulations of dynamic-epistemic logic. For example, [**5**] provides a very general *co-algebraic* version, relying heavily on the bisimulation invariance of the above language. But in this section, we concentrate on other issues, having to do more with new features of the product update framework.

4.2. Update evolution

Unlike updates for public announcement, product update can blow up the size of the initial input model $\boldsymbol{M}$. Here is an illustration.

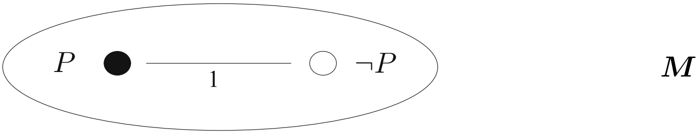
Example 8 (*blow-up of information models*). Suppose a public announcement of the true fact P takes place in a group $\{1, 2\}$, but 2 is not sure whether it was an announcement of P, or just some identity event Id which could happen anywhere. In that case, a two-world model with worlds for P and $\neg P$ turns into a three-world model with states (p, "$P!''$), (p, Id), and $(\neg P, Id)$. □

A typical example of blow-up occurs in *games*. Players start from an initial situation $\boldsymbol{M}$, say a deal of cards, and the event model $\boldsymbol{A}$ contains all possible moves that they have – with preconditions restricting when these are available at players' turns. Then the game tree consists of all possible evolutions through the nodes in the following tree model:

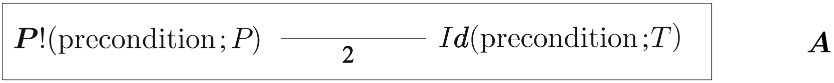
Definition 11 (*update evolution models*). Let an initial epistemic model $\boldsymbol{M}$ be given, and an event model $\boldsymbol{A}$. Then $Tree(\boldsymbol{M}, \boldsymbol{A})$ is the infinite epistemic model consisting of disjoint copies of all successive product update layers $\boldsymbol{MxA}, (\boldsymbol{MxA})\boldsymbol{xA}, \ldots$. □

But this infinity can be spurious – as happens in many games, where complexity of information can grow, but then decrease again toward the end game.

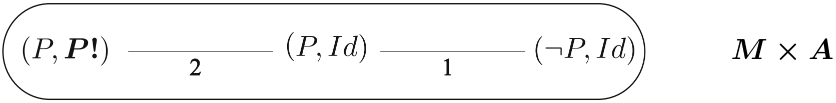
Example 9 (*stabilization under bisimulation*). Consider a model with two worlds P, $\neg P$, between which agent 1 is uncertain, though 2 is not. The actual world has P:

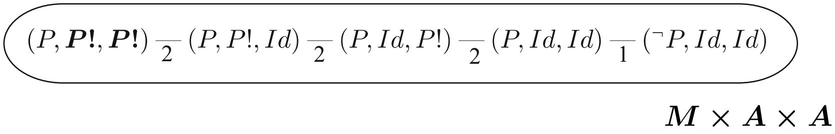


Now a true announcement of P takes place, and agent 1 hears this. But agent 2 thinks the announcement might just be a statement "True" which could hold anywhere. The event model for this scenario looks like this:


$\boldsymbol{P}!$(precondition ; P)
2
$\boldsymbol{Id}$(precondition ;T)
$\boldsymbol{A}$

The next two levels of $Tree(\boldsymbol{M}, \boldsymbol{A})$ then become as follows:




$(P, \boldsymbol{P!}, \boldsymbol{P!})$
2
$(P, P!, Id)$
2
$(P, Id, P!)$
2
(P, Id, Id)
1
$(\neg P, Id, Id)$
$\boldsymbol{M} \times \boldsymbol{A} \times \boldsymbol{A}$

But note that there is an epistemic bisimulation between these two levels, connecting the lower three worlds to the left with the single world $(P, \boldsymbol{P}!)$ in $\boldsymbol{MxA}$. Thus, $\boldsymbol{MxAxA}$ is bisimilar with $\boldsymbol{MxA}$, and the iteration remains finite modulo bisimulation. □

Van Benthem [**19**] determines the exact conditions under which an extensive game with imperfect information can be represented in this tree format. Moreover, [**22**] stated a "Finite Evolution Conjecture" saying that, starting from a given finite $\boldsymbol{M}$ and $\boldsymbol{A}$, the model $Tree(\boldsymbol{M}, \boldsymbol{A})$ always remains finite modulo bisimulation. This would imply that some horizontal levels $\boldsymbol{MxA}^k$ and $\boldsymbol{MxA}^l$ in that tree must be bisimilar, with $k < l$.

The Finite Evolution Conjecture was refuted in [**81**], which views $Tree(\boldsymbol{M}, \boldsymbol{A})$ as a dynamical system – and then shows that

(a) The Conjecture holds in single-agent $\boldsymbol{S5}$-models.

(b) The Conjecture fails in some models with two $\boldsymbol{S5}$-agents. But it holds for many special cases of such models: for example, when the epistemic accessibility relations for all agents in the model $\boldsymbol{A}$ are linearly ordered by inclusion.

Sadzik [**81**] uses finite pebble games over $\boldsymbol{M}$ and $\boldsymbol{A}$ to determine when $Tree(\boldsymbol{M}, \boldsymbol{A})$ is finite modulo bisimulation. This is the case if an only if the "responding player" in the game has a winning strategy. The computational complexity of this game is still unknown. Also, the above analysis still leaves open the possibility that large classes of scenarios fall into the Finite Evolution class – for example, those corresponding to most parlour games.

Remark 3 (*action emulation*). Other new questions about product update focus on the behavior of action models. Van Eijck, Ruan, and Sadzik [**47**] introduce a relation of *action emulation* between action models $\boldsymbol{A}$, $\boldsymbol{B}$ which holds if and only if $\boldsymbol{A}$, $\boldsymbol{B}$ produce bisimilar results on all bisimilar inputs $\boldsymbol{M}$, $\boldsymbol{N}$. This turns out to be different from a simple notion of bisimulation between action models, and it has an interesting independent characterization. □

4.3. Questions of language design

It is still somewhat of an open question which language has the best expressive power for dynamic-epistemic logic. Here we just state some directions which have not yet been fully explored.

Baltag, Moss, and Solecki [**6**] use a standard epistemic language with a common knowledge modality. But this fails to generate intuitive *reduction axioms* for formulas $[\boldsymbol{A}, a]\, C_G\varphi$ which are crucial to understanding group communication. As mentioned before, van Benthem, van Eijck, and Kooi [**31**] do find such axioms in a *PDL*-style extension of epistemic logic. But are there natural languages in between "*DEL*" and their "*E-PDL*" that do the job? Or conversely, going to even richer languages, does the same "reductive equilibrium" also hold for the full epistemic μ-calculus?

Another potential extension has to do with *concurrency* in communication, such as the simultaneous assertion by the Muddy Children of their status. This might involve the same sort of *process-algebraic* calculi for concurrency as those developed in the 1980s in computer science. In fact, product update itself is a typical graph operation of the sort studied in that area, which respects modal bisimulation. What is the connection between dynamic-epistemic logic and process algebra? Cf. [**44**] for a first approach to parallel update actions.

Finally, the system still has a glaring asymmetry. Information models $\boldsymbol{M}$ interpret a *language*, but event models $\boldsymbol{A}$ do not! In particular, preconditions PRE are not formulas *true* at a in $\boldsymbol{A}$: they refer to what is to be true in $\boldsymbol{M}$. But it is easy to introduce a second epistemic language describing properties of actions or events. It has atomic properties, Boolean combinations, and epistemic modalities such as "for some action indistinguishable from the current one." Van Benthem [**16**] and ten Cate [**39**] show how this simplifies product update in a joint language for information and action models. Van Benthem [**30**] relates this to modal languages for more general product operations, and reductions of truth in a product to truth in its components. But a convincing and usable language of event models per se still has not emerged.

4.4. Extensions of empirical coverage

These were all questions about dynamic epistemic logic and product update as they stand. But in studying real communication, many further issues will come to the fore that might lead to extensions of the framework. Here are a few examples where no mathematical theory worth reporting has been developed as yet.

(a) In Section 7, we will look at temporal settings which look at the past of some current update process, as well as its future.

(b) Van Benthem and Liu [**32**] show that different update rules hold for *different types of agents*, including those which are not logically omniscient, because of memory limitations. They also suggest a general study of heterogeneous groups, whose members do not all update in the same way. This is like "bounded agents" in game theory.

(c) Bleeker and van Eijck [**36**] study *security* and *cryptographic protocols* in a product update model which includes world-changing actions.

(d) Castelfranchi [**38**] points out that belief dynamics goes hand in hand with dynamics of *changing goals and intentions.* The latter dimension of update has been left out completely in the logicians' systems so far.

Communication is a vast area, with many "thresholds" of complexity, for example, when moving from public to private actions, or from plain speaking to misleading or lying. The eventual hope would be that dynamic epistemic logics, of whatever enriched sort, help in locating and understanding these practices.

5. Background in Standard Logics

Epistemic logic has usually been considered as an applied system, rather than a vehicle for mathematical research as such. And indeed, for technical purposes, it is often easy to look at related systems in which it can be embedded. There are well-known direct translations from epistemic logics into modal or dynamic logics, but also into first-order logic, and into fixed-point logics. We mention some open problems in those areas that seem relevant to our concerns in the preceding sections.

5.1. Modal logic

Model-changing operators

$[P!]\varphi$ is really an instance of a sort of modal operator that occurs more often in the recent literature. Its evaluation *shifts the current model.* Other examples of such operators are the "deletion modality" $\langle - \rangle\varphi$ of [**20**], which states truth in some submodel with one accessibility link deleted:

$$\begin{aligned} \boldsymbol{M}, s \models \langle - \rangle\varphi \quad \text{iff} \quad & \text{for some } (s,t) \in R_M, \\ & (W_M, R_M - \{(s,t)\}, V_M), s \models \varphi. \end{aligned}$$

This notion seems simple, but [**64**] shows that this modal logic has a ***Pspace***-complete model checking problem, while satisfiability is undecidable. Another example are the "bisimulation quantifiers" $\langle bis\text{-}p\rangle\varphi$ of [**57**]:

> $\boldsymbol{M}, s \models \langle bis\text{-}p\rangle\varphi$ iff for some model $(\boldsymbol{N}, t)$ with a bisimulation for the language of φ minus p between $\boldsymbol{M}$, $\boldsymbol{N}$ linking s to t, $\boldsymbol{N}$, $t \models \varphi$.

This evaluates φ in some model bisimilar to the current one, disregarding truth values for the atom p. As an extension of PDL, this has equal expressive power with the modal μ-calculus, and uses of this formalism are still increasing. The modalities $\langle \boldsymbol{A}, a\rangle\varphi$ of dynamic epistemic logic belong to the same family, and hence their expressive power and complexity behavior are of interest.

Generalized consequence

Model-jumping also occurs inside modal logic. In many completeness proofs, one has a modal formula φ true in some model $(\boldsymbol{M}, s)$, and then finds another model $(\boldsymbol{N}, t) \models \varphi$ with nicer structural properties, bisimilar to $(\boldsymbol{M}, s)$. This suggests two new notions of modal consequence:

$\varphi \Rightarrow [bis]\psi$ ψ holds in all models bisimilar to a model for φ

$\varphi \Rightarrow \langle bis\rangle\psi$ every model for φ is bisimilar to one for ψ

As was shown in [**15**], the former notion is recursively axiomatizable and even decidable. For the second, we have equivalence to conservativity of ψ over φ w.r.t. existential consequences:

for all existential modal formulas α, if $\psi \models \alpha$, then $\varphi \models \alpha$.

Question 4. Is the second, existential notion of bisimulation consequence decidable, too?

5.2. First-order logic

Relativization

Announcing A amounts to the well-known first-order operation of *relativizing* a model $\boldsymbol{M}, s$ to a definable submodel $\boldsymbol{M} \mid A, s$. This may also be described via syntactic relativization of formulas φ by the update assertion A:

$$\boldsymbol{M} \mid A, s \models \varphi \quad \text{iff} \quad \boldsymbol{M}, s \models (\varphi)^A.$$

The reduction axioms of PAL are just the usual inductive definition of relativization. In this perspective, the point of our relativized common knowledge was that the epistemic language with just $C_G\varphi$ is not closed under relativization, whereas the basic dynamic logic PDL is. Even so, PAL is a modal axiomatization of model-theoretic relativization.

Problem 3. Give a complete logic of relativizations $(\varphi)^A$ in *first-order logic.*

One new valid principle at this level is

$$((A)^B)^C \text{ is logically equivalent to } {}_A((B)^C) \qquad \textit{Associativity}$$

This is the first-order counterpart of the earlier valid principle

$$[A!][B!]\varphi \leftrightarrow [([A!]B)!]\varphi$$

in the substitution-closed schematic version of PAL.

Relative interpretability

Relativization often occurs together with other operations on models, such as *translation* of predicates and formation of new objects as *pairs*, – for example, in the notion of "relative interpretation" of one theory into another. An example is the way in which the theory of the rationals is embedded into that of the natural numbers. Rationals may be viewed as pairs of natural numbers, and one then looks at a definable subset of NxN, with some newly defined predicates. Product update involves exactly the same features, and hence dynamic epistemic laws may also be viewed as an axiomatization of such further model-forming operations. Thus, DEL might point the way toward a systematic modal meta-theory of model-theoretic operations.

Interpolation

Update is also related to *interpolation*. Barwise and van Benthem [**9**] define a general notion of "entailment along a relation" R:

> φ *entails* ψ *along* R if, for all models $\boldsymbol{M}$, $\boldsymbol{N}$ with $\boldsymbol{MRN}$,
> if $\boldsymbol{M} \models \varphi$, then $\boldsymbol{N} \models \psi$.

Reasoning often involves using what we know about one situation to infer properties of *another*. Standard logical consequence is entailment along the identity relation. Of particular interest in a modal setting is entailment along modal bisimulation. Now we get interesting combined preservation-interpolation theorems:

Theorem 8. *The following are equivalent for all first-order formulas* φ, ψ:

(a) φ *entails* ψ *along bisimulation,*

(b) *there exists a modal formula* α *with* $\varphi \models \alpha \models \psi$.

The modal interpolant α is the "bridge" which allows the jump from models $\boldsymbol{M}$ where φ holds to bisimilar models $\boldsymbol{N}$ where ψ holds. In the special case where φ is invariant for bisimulation (i.e., φ entails itself along bisimulation), this gives the usual result that

φ must be equivalent to some modal formula. Similar interpolation results might make sense for dynamic-epistemic logic. Here is a simple illustration (recall Example 6).

Example 10 (*interpolation by self-fulfilling statements?*). Consider a valid PAL statement $\varphi \to [P!]\psi$ about the effects of a P-announcement. Can we always use self-learning statements as "interpolants" to explain this, in the sense that $\varphi \to [P!]\psi$ is valid iff there exists some self-fulfilling assertion α such that

(i) $\varphi \models \alpha$, (ii) $\alpha \models \psi$, (iii) $\models [\alpha!]\, C_G\alpha$?

The answer seems negative. □

5.3. Fixed-point logics

Fixed-point extensions exist for both modal and first-order languages (cf. [**62**], [**85**] on the μ-calculus, and [**46**] on $LFP(FO)$). These languages contain all predicates which are definable as smallest or greatest fixed-points of monotone set operations. In particular, with epistemic logic, common knowledge may be viewed as a greatest fixed-point

$$\nu p \bullet \varphi \wedge \&_{i\in G} K_i p.$$

Many features of epistemic logic with common knowledge become clearer against this background. For example, the two characteristic principles for the usual complete axiomatization express the two aspects of this definition: an axiom stating the fixed-point equation by itself, plus an induction rule stating that this is a *greatest* fixed-point.

But there is also a less attractive side to this reduction. For, the explicit programs of propositional dynamic logic disappear, and the language becomes "static" again, expressing properties of worlds. Nevertheless, programs can still be extracted under certain circumstances. The basic idea of PDL is finite *reachability*: each program describes a regular set of traces consisting of basic action steps and test for formulas. Here is a relevant fragment of the *μ-calculus* which contains *PDL*, in which all approximation sequences for smallest or greatest fixed-points stop after ω steps.

Definition 12 (*the ω-μ-calculus*). The *ω-μ-calculus* only allows smallest fixed-point operators in the following existential format, whose approximation sequences always stabilize by stage ω:

$\mu p \;\bullet\; \varphi(p)$ with φ constructed according to the syntax
$p \,|\, p$-free formulas $\;|\; \wedge \;|\; \vee \;|$ existential modalities.

□

The reason for guaranteed approximation by stage ω is the special syntax of $\varphi(p)$. Van Benthem [**14**] proves a preservation theorem showing the equivalence, for first-order formulas, of this format with the key semantic property of "Finite Distributivity" for the approximation maps. The ω-μ-calculus is still too strong, though, since even a simple ω-μ-formula like

$$\mu p \;\bullet\; q \vee (\langle 1\rangle p \wedge \langle 2\rangle p)$$

is not definable in *PDL*. Still, *PDL* is closed under simultaneous fixed-points of a yet more special syntactic type of recursion, with only disjunctions of existential formulas $\langle\pi\rangle p$ where the propositional recursion variables p occurs only in the end position. We omit details (cf. the Appendix to [**31**]) – but this provides one more format for reduction axioms in dynamic-epistemic logic. The interesting issue remains which natural fragments of the μ-calculus suffice for communication events.

Of the many further questions raised by the connection with fixed-point logics, we point out just one. Evidently, we would like to have analogues of classical meta-properties like interpolation or preservation for our static or dynamic epistemic logics including common knowledge. But it is a bit of a scandal that we do not know if these hold! For example, interpolation for *PDL* has been open for some 30 years now, with several ship-wrecked proofs in the prestigious published literature. The problem is that the usual model-theoretic arguments based on compactness fail for fixed-point languages. And so do their analogues for infinitary languages (cf. [**7**] and [**9**]) – as fixed-point languages strike out from first-order logic in a different way, allowing for an explicit definition of well-foundedness.

Question 5. Do *PAL* and *DEL* with common knowledge inherit nice model-theoretic properties of their finitary modal base logics, such as interpolation, Los-Tarski, or Lyndon theorems?

We do not know. The languages seem so simple that a positive answer might be expected, but few proofs of this kind exist. One exception is the propositional μ-calculus, where d'Agostino and Hollenberg [**1**] established uniform interpolation by *automata-theoretic* methods. These methods might also work here.

6. From Information Update to Belief Revision

6.1. From knowledge to belief

Dynamic-epistemic logic as presented so far seems mainly concerned with knowledge – but this is an artefact of our presentation. One could just as well formulate everything so far in terms of agents' *beliefs* $B_i\varphi$, interpreted over models for a minimal modal logic without special requirements on the now *directed* accessibility relations. The only technical modification worth pointing out is the issue of "common belief." The fixed-point operator for common knowledge enforced veridicality:

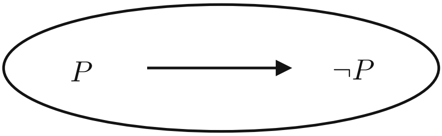
$$C_G\varphi \leftrightarrow \varphi \wedge \&_{i\in G}K_iC_G\varphi.$$

But the modification is simply this:

$$CB_G\varphi \leftrightarrow \&_{i\in G}(B_i\varphi \wedge B_iC_G\varphi).$$

Still, there remains a desideratum, since product update does not do genuine belief revision. Here is a simple illustration.

Example 11 (*updating with conflicting information*). Suppose that p is true, but an agent believes that $\neg p$:



Now an announcement p! occurs. The product update rule will eliminate the $\neg p$-world, leading to a one-point p-world with an *empty* accessibility relation. Here, the agent believes a contradiction, or indeed any formula. □

Now, there is an easy fix in the preceding case: just make the accessibility relation reflexive in the updated model, and let the agent believe that p. But the point is that there is no *principled* way of doing this in the current framework. And it seems hard to find a way of modifying accessibility relations dealing with all more delicate cases that arise with as "strong prior preference for $\neg p$" versus "strong new evidence for p."

6.2. Dynamic doxastic logic

An extension of product update to deal with belief revision has been proposed in [**4**]. The basic idea is as follows. One first introduces a more refined language for belief, with "graded operators" – an idea due to [**83**]:

$B_i^\alpha \varphi$ agent i believes up to plausibility level α.

This requires enriched epistemic models $\boldsymbol{M}$, retaining the indistinguishability relations $\sim_i$, but expanded with maps $\varkappa_i(s)$ assigning doxastic plausibility values $\varkappa_i(s)$ for agents to worlds s:

$$\boldsymbol{M}, s \models B_i^\alpha \varphi \quad \text{iff} \quad \textit{for all } t \sim_i s \textit{ with } \varkappa_i(t) \leqslant \alpha \colon \boldsymbol{M}, t \models \varphi.$$

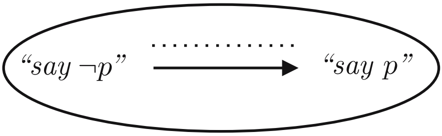
This could even be generalized to having world-dependent plausibility values, but there seems to be no need for this in the most basic cases of belief – if we assume that agents know their own beliefs and plausibilities. It is easy to axiomatize the combined epistemic doxastic logic of these models, especially with a trick from [**63**]. Just add proposition letters $\mathit{plaus}_i(\alpha)$ to the language, and interpret these as true at just those worlds whose plausibility for agent i is at most α. Then $B_i^\alpha \varphi$ may be defined explicitly as

$$K_i(\mathit{plaus}_i(\alpha) \to \varphi)$$

and its special properties follow automatically, such as the two introspection laws

$$B_i^\alpha \varphi \to K_i B_i^\alpha \varphi \quad \text{and} \quad B_i^\alpha (B_i^\alpha \varphi \to \varphi).$$

Next, we lift the same way of thinking to *event models.* These, too, can be viewed as having plausibility structure. For example, I may hear you say something, and believe that it is most likely to be "p," although it might also be "$\neg p$," as a negation got lost in the wind. Say, in fact, you said $\neg p$:" This would give rise to this event model, where both actions are epistemic possibilities, and one is doxastically preferred:



The preconditions may still encode agents' common knowledge about these actions, which would work exactly as before. One could also make them reflect agents' private beliefs about these actions. For example, if you think the speaker is a liar, then you would believe that "*say p*" tends to happen when $\neg p$ is in fact the case. The latter sort of modified precondition has not yet been studied – and we continue with the former, simpler case. Quite sophisticated scenarios can occur, even in this idealized format.

In what follows, we omit some complications having to do with the finite range of plausibility values used in [**4**], for technical reasons that are irrelevant here.

Definition 13 (*product update for belief*). Product models $(\boldsymbol{M}, s)$ $\boldsymbol{x}$ $(\boldsymbol{A}, a)$ are defined as before for their purely epistemic part. The crucial additional rule updates plausibility values:

$$\varkappa_i((s,a)) = \varkappa_i(s) + \varkappa_i(a) - MIN(\varkappa_i(t) \,|\, \boldsymbol{M}, t \models PRE_a\}.$$

The correction factor $MIN(\varkappa_i(t) \,|\, M, t \models PRE_a\}$ subtracts the lowest value for worlds in the original model $\boldsymbol{M}$ satisfying the precondition of the action just performed. □

Some drawing of diagrams for simple scenarios will show how this stipulation works in practice.

Example 12 (*updating beliefs*).

(a) Start from uncertainty about p while believing that $\neg p$, with values 1 for the p-world, and 0 for the $\neg p$-one. Now listen to a true announcement of p, as in the earlier problematic example of belief collapse after public announcement. Then the plausibility value of the only remaining world $(p, \text{"}\textit{say } p\text{"})$ is computed as

$$\begin{aligned}
&\varkappa_i((p, \text{"say } p\text{"})) \\
&= \varkappa_i(p) + \varkappa_i(\text{"}\textit{say } p\text{"})) - MIN(\varkappa_i(t) \,|\, \boldsymbol{M}, t \models p\} \\
&= 1 + 0 - 1 = 0.
\end{aligned}$$

This is as it should be: after the update the agent believes that p.

(b) Next, let the agent hear a statement which she believes to be a true announcement of "p" (plausibility *value* 0 in the event model), though it might also be one of "$\neg p$" (plausibility *value* 1). Then two worlds remain, and their plausibility values become:

$$\begin{aligned}&\varkappa_i((p, \text{"}say\ p\text{"}))\\ &= \varkappa_i(p) + \varkappa_i(\text{"}say\ p\text{"})) - MIN(\varkappa_i(t) \mid \boldsymbol{M}, t \models p\}\\ &= 1 + 0 - 1 = 0,\end{aligned}$$

$$\begin{aligned}&\varkappa_i((\neg p, \text{"}say\ \neg p\text{"}))\\ &= \varkappa_i(\neg p) + \varkappa_i(\text{"}say\ \neg p\text{"})) - MIN(\varkappa_i(t) \mid \boldsymbol{M}, t \models \neg p)\\ &= 0 + 1 - 0 = 1.\end{aligned}$$

□

Still, these very precise numerical calculations of plausibility values may be more specific than what is intuitively supported by our understanding of belief revision.

Problem 4. Find a more qualitative version of belief revision.

Given this update mechanism for beliefs, it is easy to find complete axiom systems for dynamic doxastic logic in the earlier dynamic-epistemic style. In particular, the product update rule for $\varkappa_i$-values turns into a reduction axiom for graded beliefs recording the same numerical convention. Still, intuitive questions remain:

The net effect of the product update rule is "radical:" the last observed event largely determines the new beliefs. This is clear in case (b) of the preceding example, where the belief about the statement just heard wipes out prior beliefs, whatever plausibility one had before for $\neg p$. This is not all we want, however – since our logic must account for the undeniable phenomenon of *different policies for belief revision*: more radical, or more conservative. Liu [**63**] studies parametrized variants for different sorts of agent, giving different weights to the factors in the update rule:

$$\lambda \bullet \varkappa_i(s) + \mu \bullet \varkappa_i(a).$$

Problem 5. Axiomatize dynamic belief logic with parametrized update rules for revision policies.

6.3. Better-known theories of belief revision

Belief revision theory is mostly associated with the "*AGM*" paradigm (cf. [**49**]). The core of this is a system of axiomatic postulates on three operations for changing a current theory T after some new fact A comes into focus:

update $T + A$,
revision $T * A$,
contraction $T - A$.

The third of these is different from what we have looked at so far, giving a theory as much like T as possible except for leaving out A. We will continue with the first two. A semantic account of belief revision was given in [**52**]. It has close analogies with Lewis-style models for conditional logic, which sets

$\boldsymbol{M}, s \models A \Rightarrow B$ iff B is true in all A-worlds "closest" to s in some given similarity relation between worlds.

Basically, an update with A moves a current theory T, viewed as a set of worlds $\|T\|$, to the set of most plausible worlds, as seen from T's standpoint, which satisfy A. This connection between belief revision and conditional logic, suitably dynamified, has been a persistent intuition in the area – with many different more precise formulations. The paper [**80**] is a sophisticated recent example.

Remark 4 (*updates for real change*). Modern belief revision theories include both belief revision concerning one fixed situation, and *world update* in the sense of [**58**], where incoming assertions can also report real changes in the world. The latter moves beyond update and revision in the stricter sense of this paper. Nevertheless, it is quite easy to incorporate these notions in the framework of this paper – for example, by allowing events in action models to change truth values of proposition letters. Cf. [**31**] for one particular implementation. □

The setting of AGM is not straightforwardly comparable with those for dynamic–doxastic update, since it only considers non-iterated theory change for single agents, and that only on the basis of non-doxastic factual statements. For instance, AGM starts with an apparently intuitive "Success Postulate" $A \in T^{*}A$. But this only makes sense for update with non-doxastic factual statements, witness the earlier discussion of the Learning Problem with public announcement. Moreover, the repertoire of three operations is much smaller than that of the above plausibility event models, which can model infinitely many different action scenarios. And finally, the AGM postulates do not provide systematic reduction axioms for beliefs after the update.

But there is an alternative modal-logic based reconstruction of AGM-style belief revision which does provide some features closer to this paper. Segerberg [**82**] (following the abstract "dynamic modal logic" of update in [**12**], [**75**]) proposes a so-called "dynamic-doxastic logic" DDL with PDL-style operators

$$[+A]\varphi, \quad [^{*}A]\varphi$$

for update and revision with non-modal statements. These modal operators satisfy a quite standard modal set of axioms expressing, amongst others, the partial functionality of these operations, and the fact that factual assertions lead to belief. One semantics for this system uses *hyper-theories*, which are like the models for "premise semantics" of conditionals in [**86**] (found also with Kratzer and Lewis). Hyper-theories $\boldsymbol{H}$ are families of sets of worlds, pre-encoding the potential revisions an agent is willing to make. They satisfy some technical mathematical assumptions which we omit here. Here is just an illustration of how they work.

Example 13 (***update and revision of hyper-theories***)**.**

$+$ Update with A takes a hyper-theory $\boldsymbol{H}$ to the union of

(i) $\boldsymbol{H}$, (ii) $\bigcap \boldsymbol{H} \bigcap \|A\|$,

$*$ Revision with A takes $\boldsymbol{H}$ to the union of (i) *cons* $(\boldsymbol{H}, A)$, i.e., all sets in $\boldsymbol{H}$ with a nonempty intersection with $\|A\|$, and (ii) $\bigcap(\textit{cons}\,(\boldsymbol{H}, A)) \bigcap \|A\|$. □

With these stipulations, DDL supports valid reduction axioms in the DEL style like

$$[+A]B_i\varphi \leftrightarrow B_i(A \to \varphi), \quad [{}^*A]B_i\varphi \leftrightarrow (A \Rightarrow_i \varphi),$$

where $A \Rightarrow_i \varphi$ is the earlier conditional assertion, interpreted in the Lewis style, but now with "closeness" as "relative plausibility" for the relevant agent i. This is much like the reduction axioms in [**23**] for belief after public announcement. Girard [**51**] explores further analogies between our brand of dynamic doxastic logic and that of the Segerberg systems. But much remains to be clarified.

6.4. Probabilistic update

Dynamic-doxastic logic also raises another issue, relating to the other broad tradition dealing with information update. Belief update is well-known in the much more established *Bayesian probabilistic* format, where prior probability distributions over propositions are modified to new ones by giving various weights to the priors and the new observation. The first systematic connections with dynamic-epistemic logic have been made in [**60**]. A full-fledged product update system for probabilities is found in [**24**]. Still, this is only the first merge, and this bridge between epistemic logic and probability could be broadened considerably.

Problem 6. Make a systematic comparison of DEL-style and Bayesian probabilistic update.

A first critical attempt at this is [**76**].

Moreover, the connection between plausibility approaches like that of Spohn–Aucher, and probability approaches to belief needs to be understood. For example, one difference is that plausibility approaches freely combine beliefs, because of the validity of

$$B_i^\alpha \varphi \wedge B_i^\alpha \psi \to B_i^\alpha(\varphi \wedge \psi).$$

But this principle is typically invalid in probabilistic update, as probabilities of conjunctions can fall below those of the conjuncts.

7. Temporal Epistemic Logic

7.1. Broader temporal perspectives on update

Product update in DEL modifies knowledge about the present situation. In particular, all uncertainty relations for agents are "horizontal" inside the current model. But there is a much broader temporal setting, from past to future. For instance, many epistemic puzzles contain dialogues like: "I do not know if P." "Yes, I *knew* that already," where the latter past tense refers to the initial state, not to the one updated by the first assertion. A technical motivation pointing the same way was the need for temporal past operators when defining strongest *postconditions*, of communicative actions: cf. Remark 2 above.

Other motivations for looking back at the past of some update process are when we find out that someone has been lying [**89**]. This seems to call for some sort of *backward temporal update* – which remains to be defined. Of course, all this requires maintaining a record of previous updates. Such a perspective is natural from many viewpoints. For example, consider the earlier idea of public announcement as *relativization.* So far, we discarded old information states. But now, we can keep the old state, and perform "virtual update" via relativized assertions. Thus, the initial state already contains all possible future communicative developments. This is more in line with the above update evolution models $Tree(\boldsymbol{M}, \boldsymbol{A})$ which contain all possible conversational trajectories. Such models obviously support a richer temporal language.

A final motivation for taking a broader temporal perspective comes from more global information that we may have about some communication process. For example, we might know that, however history unfolds, every question will always be answered. Such "fairness" or "liveness" properties can be formulated in standard temporal logics over $Tree(\boldsymbol{M}, \boldsymbol{A})$, now viewed as a branching-time structure. Such systematic informative restrictions on the possible runs of a system are often called *protocols.*

7.2. Knowledge and ignorance over time

Epistemic-temporal frameworks have existed since the 1980s. One famous example is the run-based model for distributed systems [**48**], or the epistemic branching temporal logic framework of [**70**]. Consider models for branching time with nodes s and histories h representing runs of a game. As a process – say, a game – proceeds, agents are in a node on some actual history whose past they know, but whose future is yet to be fully revealed. We will think of this setting as a *tree of finite sequences of events*, just as happens in temporal logics in computer science. Sometimes, a selection is made among all possible branches in the tree, leaving just the "legal runs" obeying the relevant protocol.

One convenient basic epistemic-temporal language has proposition letters for local properties of nodes, Boolean operations, as well as temporal and modal operators.

Definition 14 (*branching time semantics*). Formulas are interpreted at nodes s on histories h, in a format with clauses

(a) $\boldsymbol{M}, h, s \models F_a\varphi$ iff $s^\frown\langle a\rangle$ lies on h and $\boldsymbol{M}, h, s^\frown\langle a\rangle \models \varphi$.
The standard operator F (''at some point in the future") is the transitive closure of this one-step modality, taken over all possible events a.

(b) $\boldsymbol{M}, h, s \models P_a\varphi$ iff $s = s'^\frown\langle a\rangle$, and $\boldsymbol{M}, h, s' \models \varphi$.
Again, P ("at some point in the past") is the transitive closure.

(c) $\boldsymbol{M}, h, s \models \Diamond\varphi$ iff $\boldsymbol{M}, h', s \models \varphi$ for some history h' which coincides with h up to stage s.

□

Remark 5 (*simultaneity*). Language extensions also make sense. In particular, enriching this temporal-modal language helps describe event trees more explicitly. For example, a sideways modality for "simultaneity" would refer to truth at the end of sequences of the same length. □

One usually reads the modal operator $\Diamond$ as an absolute historical possibility. But one can also view it as some sort of epistemic possibility for agents, referring to future continuations which they think possible. In the latter case, we should enrich the above models and allow different sets of continuations for different agents.

Zanardo [**90**] has an extensive discussion of this temporal setting. In particular, protocols generalize models to a more manageable format allowing for an axiomatization in a Henkin "general model" style. The book [**11**] is a more philosophical-logical analysis of knowledge, belief and conditionals in branching time, focusing on the action logic *STIT* ("see to it that").

Problem 7. Give a systematic comparison of *STIT* with *DEL* and *DDL*.

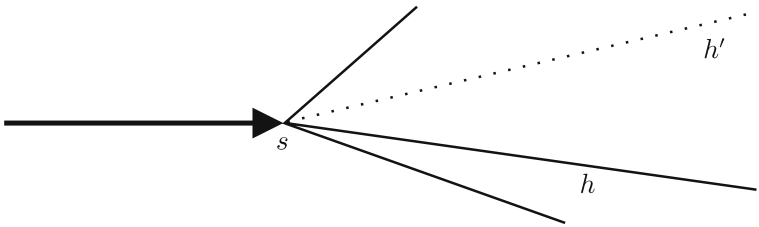
If players i also have beliefs about the course of the game, or some more general process, then we add binary relations $\leqslant_i$ to the model of *relative plausibility*, and we add a doxastic modality.

Definition 14 (continued).

(d) $\boldsymbol{M}, h, s \models B_i\varphi$ iff $\boldsymbol{M}, h', s \models \varphi$ for all histories h' which coincide with h up to stage s and are most plausible for i according to the given relation $\leqslant_i$.

Here is the existential dual version of this:

(d)′ $\boldsymbol{M}, h, s \models \langle B, i\rangle\varphi$ iff $\boldsymbol{M}, h', s \models \varphi$ for some history h' coinciding with h up to stage s and most plausible for i according to the given relation $\leqslant_i$:



This models different views of players about the future. □

Problem 8. Find a most suitable extension of the *DEL*-language that works well computationally on these event trees.

7.3. Representation of update logics

The above branching-time models specialize naturally to the event-tree semantics that comes with product update. Van Benthem [**19**], van Benthem and Liu [**32**] observe that product update will produce very special uncertainty patterns in such trees.

Theorem 9. *An event tree is isomorphic to* $Tree(\boldsymbol{M}, \boldsymbol{A})$ *for finite models* $\boldsymbol{M}$, $\boldsymbol{A}$ *iff it satisfies suitable properties of "Perfect Recall" and "Uniform No Learning."*

Similar representation theorems are provable for agents which have bounded memory to various degrees. Even so, a general update logic of different sorts of agents, and what happens when they interact, is missing so far. This would be a sort of update counterpart for the many attempts at giving up "logical omniscience" in static epistemic logic.

The current 2005 version of [**23**] reformulates dynamic-epistemic update logics in this broader setting, replacing the earlier reduction axioms for update and revision by more standard temporal-doxastic versions. Cf. also [**67**] on better-behaved regular subcases of temporal run models.

Problem 9. Relate *DEL* to the more general semantics of messages in the branching-time models of [**70**].

Problems 8, 9 are part of a more general task – which also came up in discussions at the "First Indian Congress on Logic and its Relationship with Other Disciplines," Mumbai 2005. Epistemic-temporal logics are used widely in analyzing and designing computational processes, as well as games in general (cf. [**67**], [**73**], and [**88**]). One would like to merge the various approaches mentioned here into one framework, and then explore the fine-structure of specific types of agent, message, and general action. *DEL* describes the simplest setting, while in the general framework 'anything goes'. What lies in between?

Interestingly, the latter paper also connects our qualitative sort of message update with quantitative information in the sense of mathematical Information Theory. Van Rooy [**77**] makes a similar move in the semantics of natural language.

Once connections like these have been made in some appropriate fashion, it will also be clear how to increase the scope of current update logics to include the earlier-mentioned phenomena of Section 7.1, such as more complex temporal preconditions for actions – instead of just atemporal epistemic ones,– announcements of temporal assertions, and actions that change truth values of atomic propositions. Each such step represents an upward move in expressiveness toward the total system of the two mentioned references. Another interesting topic in this setting are the earlier-mentioned *protocols* constraining general runs of a communication sequence. Protocol information is absent from *DEL* (or *DDL*) as such. But, through the use of preconditions in event models, DEL does rule out certain runs of subsequent events "locally," so to speak. The question is which types of protocol are needed: these might be classified by syntactic types of definition in the epistemic-temporal language.

7.4. Connections with other parts of mathematics

The above discussion suggests links between the dynamic logics of information update over time in this paper and a number of established areas from mathematics. These connections are still to be explored in depth. Here are a few examples.

(a) Protocols might be just any set of branches in event trees. But the latter are like Baire spaces with their *natural topology*. Thus, well-behaved sets of branches lie at various levels of the topological Borel Hierarchy over our trees. Still, pure tree topology may not suffice, as the epistemic structure of indistinguishability between tree nodes matters. Then we seem to need generalized topology, as natural closure conditions w.r.t. indistinguishability relations might become relevant.

(b) We already mentioned *probability theory* as a more quantitative account of belief and update, and likewise, *information theory* as a quantitative measure of channel transmission. Unifying such viewpoints seems a major undertaking at this stage.

(c) Event trees are also the natural universe for *learning theory* [**59**], whose connections with modal and dynamic approaches seem obvious – though just a few initial links have been made so far (cf. [**55**]).

(d) Van Benthem and Liu [**32**] suggest connections between the study of different epistemic agents and *automata theory*, with Nerode representations for trees with added uncertainty relations. This is also connected to the use of finite automata for agents with "bounded rationality" in game theory. A final mathematical connection here is the study of Turing machines with ignorance for players about what is observed on input and output tapes, as in Ann Condon's pioneering work on such devices in computational *complexity theory.*

(e) And finally, our long-term temporal perspective suggests a step from dynamic logic to mathematical *dynamical systems theory.* This move has been made in other settings, too – as one studies long-term bulk behavior of logical inference.

8. Game Logics and Game Theory

As explained in the Introduction to this paper, communication between agents naturally leads to the topic of strategic interaction generally. The best current model for such interactions are *games.* Thus, we get into connections between logic and *game theory.* "Logic and Games" is a fast-developing interface by itself these days, with many different sub-themes, including strong influences from computer science. In particular, there are two strands of research to be distinguished: using ideas from game theory in logic, and using ideas from logic in game theory in logic.

We are currently preparing a separate companion paper [**29**] with a similar list of open questions on Logic and Games, along both of these lines. Here is a preliminary table of contents, whose headings form a natural continuation of the topics addressed for update and communication in this paper:

(a) Dynamic logics and strategy calculus in extensive games

(b) Dynamic logics of players' powers for determining outcomes

(c) Operations that form new games, and game algebra

(d) Logics of preferences and rational action

(e) Logics with imperfect information

(f) Infinite games and evolution in linear and temporal logic

(g) Fine-structure of game theory: justifying Nash equilibria in strategic and extensive games

(h) Logic games with preferences?

(i) Probability, expectations and mixed strategies

(j) Evolutionary games and dynamical systems

Surveys of relevant topics, as well as some concrete results, are in [**17**], [**19**], [**21**], [**25**], and [**27**]. Recent dissertations showing the interest of the interface are [**71**], [**37**], [**41**], [**54**]. The final paper will contain a much more extensive bibliography.

9. Conclusion

This paper is a survey of new themes and questions in the dynamic logic of communication. In line with the purpose of the present Volume, most of its open problems concern mathematical *system issues* about dynamic epistemic logics – and *technical junctions* to be made between different areas of logic and mathematics.

Still, it is worth repeating a point from our Introduction about other sorts of issues that are of equal importance. In particular, on the descriptive side, there are valid concerns about the adequacy of the formalizations proposed here. For example, descriptive frameworks for rational agents in computer science or cognitive science tend to view information update in a broader setting of *beliefs*, *desires* and *intentions* (the "*BDI*" paradigm), with a concomitant dynamics of preferences, goals, and plans. Similar broad models have been around in computational linguistics since the 1980s. Sticking with plain update, or even belief revision, may be a beneficial mathematical idealization – but it may also suffer from the "Inventor's Paradox" of attempting too little.

Another missing aspect is *cognitive reality.* Update, communication, and conversation seem typical topics for controlled experiments, but no systematic collaboration between logicians and cognitive scientists has developed yet along these lines.

Finally, staying with logic itself, putting communication and many-agent interaction at center stage might have far-reaching repercussions for the agenda of the field – and even the things we want to know about its key systems. For example, which metatheorems (old, or new) for first-order logic are most relevant to its communicative, rather than its inferential use? We leave these larger issues to the philosophers of logic.

References

1. G. d'Agostino and M. Hollenberg, *Logical questions concerning the μ-calculus: Interpolation,* Lyndon & Los-Tarski. J. Symb. Logic **65** (2000), 310–332.
2. S. Artemov, *Logic of proofs*, Ann. Pure Appl. Logic **67** (1994), 29–35.
3. S. Artemov, *Evidence-Based Common Knowledge*, New York, CUNY Graduate Center, 2005.
4. G. Aucher, *A Joint System of Update Logic and Belief Revision*, Master's Thesis, ILLC, Univ. Amsterdam, 2003
5. A. Baltag, B. Coecke, and M. Sadrzadeh, *Algebra and sequent calculus for epistemic actions*, In: ENTCS Proc. Logic and Communication in Multi-Agent Systems, Workshop, ESSLLI 2004, Nancy, France, 2004.
6. A. Baltag, L. Moss, and S. Solecki, *The logic of Public announcements, common knowledge and private suspicions*, In: Proc. TARK 1998, Los Altos, Morgan Kaufmann Publishers, 1998, pp. 43–56.
7. J. Barwise, *Admissible Sets and Structures*, Berlin, Springer, 1975.
8. J. Barwise, *Three views of common knowledge*, In: Proc. TARK 1988, Los Altos, Morgan Kaufmann Publishers, 1988, pp. 365–397.
9. J. Barwise and J. van Benthem, *Interpolation, preservation, and pebble games*, J. Symb. Logic **64** (1999) no. 2, 881–903.

10. J. Barwise and L. Moss, *Vicious Circles*, Stanford, CSLI Publications, 1997.

11. N. Belnap, M. Perloff, and M. Xu, *Facing the Future*, Oxford, Oxford Univ. Press, 2001.

12. J. van Benthem, *Semantic parallels in natural language and computation*, In: H-D Ebbinghaus (ed.) *et al.*, Logic Colloquium. Granada 1987, Amsterdam, North-Holland, 1989, pp. 331–375.

13. J. van Benthem, *Reflections on epistemic logic*, Logique Anal. **133–134** (1993), 5–14.

14. J. van Benthem, *Exploring Logical Dynamics*, Stanford, CSLI Publications, 1996.

15. J. van Benthem, *Dynamic Bits and Pieces'*, Report LP-97-01, ILLC, Univ. Amsterdam.

16. J. van Benthem, *Radical epistemic dynamic logic*, note for course "Logic in Games," ILLC, Univ. Amsterdam, 1999.

17. J. van Benthem, *Logic and Games*, Electr. Lect. Notes, http://staff.science.uva.nl/~johan/, Amsterdam and Stanford, 1999–2004 [under construction].

18. J. van Benthem, *Logics for information update*, In: Proc. TARK VIII, Los Altos, Morgan Kaufmann, 2000, pp. 51–88.

19. J. van Benthem, *Games in dynamic epistemic logic*, Bull. Economic Research **53** (2001), no. 4, 219–248.

20. J. van Benthem, *An essay on sabotage and obstruction*, In: D. Hutter (ed.), Festschrift for J.rg Siekmann, Springer-Verlag, 2002.

21. J. van Benthem, *Extensive games as process models*, J. Logic Lang. Inf. **11** (2002), 289–313.

22. J. van Benthem, *One is a Lonely Number: on the Logic of Communication*, Techn. Report no. PP-2002-27, ILLC, Univ. Amsterdam, In: P. Koepke (ed.) *et al.*, Colloquium Logicum, Providence, Am. Math. Soc. Publications [to appear].

23. J. van Benthem, *Belief over Time*, Manuscript, ILLC, Univ. Amsterdam, 2003; Current version: *Update and Revision in Games*, to be presented at APA-ASL Symposium on Games, San Francisco, March 2005.

24. J. van Benthem, *Conditional probability meets update logic*, J. Logic Lang. Inf. **12** (2003), no. 4, 409–421.

25. J. van Benthem, *Rational dynamics and epistemic logic in games*, In: S. Vannucci (ed.), Logic, Game Theory and Social Choice III, Dept. Political Economy, Univ. Siena, 2003, pp. 19–23.

26. J. van Benthem, *Structural properties of dynamic reasoning*, In: J. Peregrin (ed.), Meaning: the Dynamic Turn, Amsterdam, Elsevier, 2003, pp. 15-31.

27. J. van Benthem, *A mini-guide to Logic in Action*, Phil. Researches, Suppl., 21-30, Beijing, Chinese Acad. Sci.

28. J. van Benthem, *What one may come to know*, Analysis **64 (282)** (2004), 95–105.

29. J. van Benthem, *Open Problems in Game Logics*, ILLC, Univ. Amsterdam, On public web-page "Games, Logic, and Computation": http://www.illc.uva.nl/lgc/ [to appear].

30. J. van Benthem, *Two logical concepts of information*, In: L. Moss (ed.), Memorial Volume for Jon Barwise, Dept. Comput. Sci., Bloomington, Indiana [to appear].

31. J. van Benthem, J. van Eijck,and B. Kooi, *Logics for Communication and Change*, ILLC, Univ. Amsterdam, CWI Amsterdam, Phil. Inst., and Univ. Groningen, 2004; To appear in Proc. TARK Singapore, 2005.

32. J. van Benthem and F. Liu, *Diversity of Logical Agents in Games*, Report PP-2004-13, ILLC, Univ. Amsterdam, In: Phil. Sci. **8** (2004), no. 2, 163–178.

33. J. van Benthem and D. Sarenac, *The Geometry of Knowledge*, Techn. Report no. PP-2004-20, ILLC, Univ. Amsterdam, 2004.

34. P. Blackburn, J. van Benthem, and F. Wolter (eds.), *Handbook of Modal Logic*, Amsterdam, Elsevier [to appear].

35. P. Blackburn, M. de Rijke, and Y. Venema, *Modal Logic*, Cambridge, Cambridge Univ. Press, 2001.

36. A. Bleeker and J. van Eijck, *The Epistemics of Encryption*, CWI Report no. INS-R0019, Amsterdam, 2000.

37. B. de Bruin, *Explaining Games*, Disser. no. 2004-03, ILLC, ILLC, Univ. Amsterdam, 2004.

38. Ch. Castelfranchi, *Reasons to Believe: Cognitive Models of Belief Change*, ISTC-CNR, Roma – Workshop Changing Minds, ILLC, Univ. Amsterdam, 2004.

39. B. ten Cate, *Internalizing epistemic actions*, In: M. Martinez (ed.), Proc. of the NASSLLI-1 Student Session, Stanford Univ., 2002, pp. 109–123.

40. B. ten Cate, *Model Theory for Extended Modal Languages*, Disser., ILLC, Univ. Amsterdam, 2005.

41. F. Dechesne, *Game, Set, Maths*, Ph.D. Disser., Phil. Inst., Katholieke Univ. Brabant, Tilburg, 2005.

42. H. van Ditmarsch, *Knowledge Games*, Disser. no. DS-2000-06, ILLC, Univ. Amsterdam and Dept. Informatics, Univ. Groningen, 2000.

43. H. van Ditmarsch, *Keeping Secrets with Public Communication*, Dept. Comput. Sci., Univ. Otago, Dunedin, 2002

44. H. van Ditmarsch, W. van der Hoek, and B. Kooi, *Concurrent dynamic epistemic logic*, In: V.F. Hendricks, K.F. Jorgensen and S.A. Pederson (eds.), Knowledge Contributors, Kluwer Academic Press, 2003, pp. 105–143.

45. H. van Ditmarsch, W. van der Hoek, and B. Kooi, *Dynamic Epistemic Logic*, Dordrecht, Kluwer-Springer Academic Publishers [to appear].

46. H-D Ebbinghaus and J. Flum, *Finite Model Theory* Berlin, Springer, 1995.

47. J. van Eijck, J. Ruan, and T. Sadzik, *Action Emulation*, CWI and ILLC, Univ. Amsterdam, and Dept. Economics, Stanford Univ., 2004.

48. R. Fagin, J. Halpern, Y. Moses, and M. Vardi, *Reasoning about Knowledge*, Cambridge (Mass.), MIT Press, 1995.

49. P. Gärdenfors and H. Rott, *Belief revision*, In: D. M. Gabbay, C. J. Hogger, and J. A. Robinson (eds.), Handbook of Logic in Artificial Intelligence and Logic Programming 4, Oxford, Oxford Univ. Press, 1995.

50. J. Gerbrandy, *Bisimulations on Planet Kripke*, Disser. no. DS-1999-01, ILLC, Univ. Amsterdam, 1999.

51. P. Girard, *DDL versus DEL*, Dept. Phil., Stanford Univ., 2004.

52. A. Grove, *Two modelings for theory change* J. Phil. Logic **17** (1988), 157–170.

53. J. Halpern and M. Vardi, *The complexity of reasoning about knowledgeand time*, J. Comput. Systems Sci. **38** (1989), no. 1, 195–237.

54. P. Harrenstein, *Logic in Conflict*, Disser. no. SIKS 2004.14, Dept. Comput. Sci., Univ. Utrecht, 2004.

55. V. Hendricks, *Active Agents*, PHILOG Newsletter, Roskilde. In: J. van Benthem and R. van Rooy (eds.), special issue on Information Theories, J. Logic Lang. Inf. **12** (2002), no. 4, 469–495.

56. W. van der Hoek and J-J. Meijer, *Epistemic Logic for AI and Computer Science*, Cambridge, Cambridge Univ. Press, 1995.

57. M. Hollenberg, *Logic and Bisimulation*, Disser., Publications Zeno Inst. Phil., **14**, Univ. Utrecht., 1998.

58. H. Katsuno and A. Mendelzon, *On the difference between updating a knowledge base and revising it*, In: P. Gärdenfors (ed.), Belief Revision, Cambridge, Cambridge Univ. Press, 1992, pp. 183–203.

59. K. Kelly, *The Logic of Reliable Inquiry*, Oxford, Oxford Univ. Press, 1996.

60. B. Kooi, *Knowledge, Chance, and Change*, Disser. no. DS-2003-01, ILLC, Univ. Amsterdam, and Dept. Inf., Univ. Groningen, 2003.

61. B. Kooi and J. van Benthem, *Reduction axioms for epistemic actions*, In: R. Schmidt, I. Pratt-Hartmann, M. Reynolds, and H. Wansing (eds.), Proc. Advances in Modal Logic 2004, Dept. Comput. Sci., Univ. Manchester. Report UMCS-04 9-1, 2004, pp. 197–211.

62. D. Kozen, D. Harel and J. Tiuryn, *Dynamic Logic*, Cambridge (Mass.), MIT Press, 2000.

63. F. Liu, *Diversity of Logical Agents*, Master's Thesis, ILLC, Univ. Amsterdam, 2004

64. Ch. Loedding and Ph. Rohde, *Solving the Sabotage Game is PSPACE-hard*, Techn. Report no. AIB-05-2003, RWTH, Aachen, 2003.

65. C. Lutz, 2004 *Expressiveness and Complexity of the Logic of Public Announcements, Informatics Institute*, manuscript, Techn. Univ., Dresden, 2004.

66. R. van der Meyden, *Common knowledge and update in finite environments*, Inf. Comput. **140** (1998), no. 2, 115–157.

67. R. van der Meyden, *Model checking the logic of knowledge*, In: Tutorial, First Indian Congress on Logic and its Relationship with Other Disciplines [to appear].

68. J. Miller and L. Moss, *The Undecidability of Iterated Modal Relativization*, Techn. Report, Indiana Univ., 2003.

69. R. Parikh, *Social software*, Synthese **132** (2002), 187–211.

70. R. Parikh and R. Ramanujam, *A knowledge based semantics of messages*, CUNY New York & Chennai, India. In: J. van Benthem and R. van Rooy (eds.), special issue on Information Theories, J. Logic Lang. Inf. **12** (2003), no. 4, 453–467.

71. M. Pauly, *Logic for Social Software*, Disser. no. DS-2001-10, ILLC, Univ. Amsterdam, 2001.

72. J. Plaza, *Logics of public announcements*, In: Proc. 4th International Symposium on Methodologies for Intelligent Systems, 1989.

73. R. Ramanujam, *Closing remarks*, In: First Indian Congress on Logic and its Relationship with Other Disciplines [to appear].

74. G. Restall, *An Introduction to Substructural Logics*, London, Routledge, 2000.

75. M. de Rijke, *Extending Modal Logic*, Disser., ILLC, Univ. Amsterdam, 1992.

76. J.-W. Romeyn, *Meaning Shifts, Epistemic Actions, and Diachronic Dutch Books*, Dept. Psychology, Univ. Amsteredam, 2005.

77. R. van Rooy, *Quality and quantity of information exchange* In: J. van Benthem and R. van Rooy (eds.), special issue on Information Theories, J. Logic Lang. Inf. **12** (2003), no. 4, 423–451.

78. H. Rott, *Change, Choice, and Inference*, Oxford, Clarendon Press, 2001.

79. J. Ruan, *Exploring the Update Universe*, Master's Thesis, ILLC, Univ. Amsterdam, 2004.

80. M. Ryan and P-Y Schobbens, *Counterfactuals and updates as inverse modalities*, J. Logic Lang. Inf. **6** (1997), 123–146.

81. T. Sadzik, *Epistemic Update as a Dynamical System*, Dept. Economics, Stanford Univ., 2004.

82. K. Segerberg, *Belief revision from the point of view of doxastic logic*, Bull. IGPL **3** (1995), no. 4 535–553.

83. W. Spohn, *Ordinal conditional functions: A dynamic theory of epistemic states*, In: W. L. Harper (ed.) *et al.*, Causation in Decision, Belief Change and Statistics II, Dordrecht, Kluwer, 105–134.

84. R. Stalnaker, *Extensive and strategic form: games and models for games*, Research Economics **53** (1999), no. 2, 93–291.

85. C. Stirling, *Bisimulation, modal logic, and model checking games*, In: A. Montanari and Y. Venema (eds.), Special issue on Temporal Logic, Logic J. IGPL **7** (1999), no.1, 103–124.

86. F. Veltman, *Logics for Conditionals*, Disser., Phil. Inst., Univ. Amsterdam, 1985.

87. F. Veltman, *Defaults in update semantics*, J. Phil. Logic **25** (1996), 221–261.

88. G. Venkatesh, *Temporal logic with preferences and reasoning about games*, IIM Banglore and Sasken, In: J. van Benthem, A. Gupta, R. Parikh, and R. Ramanujam (eds.), First Indian Congress on Logic and its Relationship with Other Disciplines, IIT Mumbai 2005. [to appear].

89. A. Yap, *Finding out Who is Lying and Cheating in Games*, Dept. Phil., Stanford Univ., 2004.

90. A. Zanardo, *First-order and Second-order Aspects of Branching-time Semantics*, In: Proc. Second International Workshop on the History and Philosophy of Logic, Mathematics, and Computation, San Sebastian (Spain), November 7-9, 2002 [to appear].

Computability and Emergence

S Barry Cooper [†]

University of Leeds
Leeds, UK

There has been much interest in recent years in the way in which global relations on structures emerge. The mathematics underlying such emergence has intimate connections with iterations of algorithms and complexity related to simple computer programs. Examples of the phenomena involved range from the emergence of natural laws in the physical universe, to patterns governing turbulent environments, to the well-known examples of fractal formation.

We look at the extent to which mathematical definability over appropriate structures can provide a key insight into what is happening. In particular, we examine the extent to which Turing's

[†] The author would like to acknowledge the support of E.P.S.R.C. Research Grant No. EP/C001389/1, *Extensions of Embeddings in the Turing Universe.*

Mathematical Problems from Applied Logics. New Logics for the XXIst Century.
Edited by Dov M Gabbay *et al.*/ International Mathematical Series, Springer, 2005

approach to real-world computability is still relevant today, and point to some fundamental questions facing those with a research interest in computability theory.

1. An Emergent World around Us

There have always been two very surprising things about the world we live in. The first is the observed high level of *regularity* and *form*, on which the success of the scientific enterprise depends. The surface of the ocean, whatever mysteries it hides, shows us easily discernible patterns which reassure and lull us with their familiar motions. Even the complications of human relationships are navigable working within the social rules and conventions established over time.

But — secondly — we are constantly confronted with the unpredictability, the sheer complexity, with which this regularity and form appears to be awash. It used to be said that only in mathematics could one have absolute certainty, but even here one increasingly has to deal with truth as an emergent phenomenon.

Now there is a third mystery to fathom: *There seems to be an intimate relationship between these first two.* Getting a better understanding of that relationship has already grown to be a massive project extending over many areas of scientific and human activity. The logician's talent for bringing out underlying principles and universalities promises to be an essential ingredient in this.

For the most part, the emergence of *Emergence* as something about which everyone has something to say, has generated more questions than answers, and more excitement than clarity. There are numerous popular books on the topic — one of the earliest being John Holland's [**26**] *Emergence – From Chaos to Order* — which tend to add to (and recycle) the store of striking examples of emergence, and expand the range of carefully constrained situations amenable to some level of analysis, or computer simulation. Steve Strogatz's recent *Sync: The Emerging Science of Spontaneous Order* [**44**] is just one example of how fascinating the topic can be, even when the focus is very specific. But this leaves us a long way from an understanding of the emergence of life on earth, of the formation of extra-galactic structures, of the origins of the laws of nature, of the relationship between the quantum and classical worlds, of the evolution of species, of the nature of consciousness as an emergent phenomenon, and of a whole host of more modest examples.

As Holland [**26**] points out:

> *"Despite its ubiquity and importance, emergence is an enigmatic, recondite topic, more wondered at than analyzed."*

This article is an attempt to take an overview of what is happening, and point to some important mathematical tasks which follow directly from this.

2. Descriptions, Algorithms, and the Breakdown of Inductive Structure

What one is typically confronted with is some particular physical system whose constituents are governed by perfectly well-understood basic rules. These rules are usually *algorithmic*, in that they can be described in terms of functions simulatable on a computer, and their simplest consequences are mathematically predictable. But although the global behavior of the system is *determined* by this algorithmic content, it may not itself be recognizably algorithmic. We certainly encounter this in the mathematics, which may be *nonlinear* and not yield the exact solutions needed to retain predictive control of the system. We may be able to come up with a perfectly precise *description* of the system's development which does not have the predictive — or algorithmic — ramifications the atomic rules would lead us to expect.

If one is just looking for a broad understanding of the system, or for a prediction of selected characteristics, the description may be sufficient. Otherwise, one is faced with the practical problem of extracting some hidden algorithmic content, perhaps via useful approximations, special cases, or computer simulations. Geroch and Hartle [**20**] discuss this problem in their 1986 paper, in which they suggest that "quantum gravity does seem to be a serious candidate for a physical theory for whose application there is no algorithm." (Interestingly, Georg Kreisel — see below — is one of those thanked by the authors for their "helpful advice on a preliminary version of this paper.")

For the logician, this is a familiar scenario, for whom something describable in a structure is said to be *definable*. The difference between computability and definability is well-known. For example, if you go to any basic computability text (e.g., Cooper [**5**]) you will find in the *arithmetical hierarchy* a usable metaphor for what is happening here. What the arithmetical hierarchy encapsulates is the smallness of the computable world in relation to what we can describe. And Post's Theorem [**38**] shows us how language can be used to progressively describe increasingly incomputable objects and phenomena within computable structures. An analysis of lower levels of the hierarchy even gives us a clue to the formal role of computable approximations in constraining objects computably beyond our reach.

Of course, there is rather more to it than this extremely schematic picture. Later, we will see how a more detailed analysis of the system resting on some corresponding infinite mathematical structure, as is common in the real world, may lead to a relevant *model*.

Metaphor or model, we would first like to know in general terms what relevance the distinction between definability and computability has for the real world. To do this, we need to look more closely at what it is, in real situations, gives rise to descriptions whose information content is so intrinsically global in character. In the next section I will make more explicit the link between emergence and definability.

The key ingredients of any chaotic environment displaying the sort of emergent relations we are talking about are firstly parallelism — involving three or more component participants — and secondly, interactivity between those participants. Georg Kreisel was brave enough, back in 1970, to propose ([**30**, p. 143]) the possibility of a collision problem related to the 3-body problem which might give "an analog computation of a non-recursive function (by repeating collision experiments sufficiently often)." Turbulence of any kind clearly exhibits these ingredients, echoed by the non-linearity of any mathematical description.

In the biological context, here is how Francisco Varela comments on the significance of his notion of autopoiesis — or self-organization — in Chapter 12 of John Brockman's [**2**] *The Third Culture*:

> *"Regarding the subject of biological identity, the main point is that there is an explicit transition from local interactions to the emergence of the 'global' property that is, the virtual self of the cellular whole, in the case of autopoiesis. It's clear that molecules interact in very specific ways, giving rise to a unity that is the initiation of the self. There is also the transition from nonlife to life. The nervous system operates in a similar way. Neurons have specific interactions through a loop of sensory surfaces and motor surfaces. This dynamic network is the defining state of a cognitive perception domain. I claim that one could apply the same epistemology to thinking about cognitive phenomena and about the immune system and the body: an underlying circular process gives rise to an emergent coherence, and this emergent coherence is what constitutes*

> *the self at that level. In my epistemology, the virtual self is evident because it provides a surface for interaction, but it's not evident if you try to locate it. It's completely delocalized."*

The search for new computational paradigms can help us understand such phenomena. Peter Wegner has particularly focused on the apparent non-algorithmic nature of computations involving these key ingredients — see, for instance, his recent paper [**22**] with Dina Goldin, or his forthcoming edited book [**21**].

It is not at all obvious, however, even in the presence of parallelism and interactivity, that we have something new, something not simulatable by a linear computation. Martin Davis [**13**] has effectively defended the classical model against a number of recently proposed new paradigms (for example, [**8**], [**11**], [**29**]). But new and increasingly convincing tests for the classical model continue to accumulate (for example, in a relativistic context, [**16**]). We obviously need the implied infinities in the underlying mathematical model, otherwise there can be no talk of incomputability. But more than that, it seems we really do need it to be like real science — using real numbers. The model has to be, in some essential way, indiscrete. A well-known feature of the emergence of attractors is their sensitivity to small changes in initial conditions. In fact, it is this feature — that of being far from equilibrium — which becomes itself, for Fritjof Capra in his book [**3**] *The Web of Life*, the third criterion for something new and emergent:

> *"This point is called a 'bifurcation point.' It is a point of instability at which new forms of order may emerge spontaneously, resulting in development and evolution.*
>
> *Mathematically a bifurcation point represents a dramatic change of the system's trajectory in phase space. A new attractor [fixed point, periodic or strange] may suddenly appear, so that the system's behavior as a whole 'bifurcates,' or branches off, in a new direction. Prigogine's*

> *detailed studies of these bifurcation points have revealed some fascinating properties of dissipative structures ...”*

Taking a computability-theoretic perspective, Odifreddi ([**34**, p. 110]) discusses incomputability arising from discrete systems, and paraphrases Kreisel from 1965:

> ***“Thesis P (for ‘probabilistic’) (Kreisel [1965])*** *Any possible behavior of a discrete physical system (according to present day physical theory) is recursive.”*

As Odifreddi comments, the evidence for or against Thesis P is inconclusive. It may well be that as we become better at modelling and analysing interactive computation, building a repertoire of informative theoretical constructs, and hence narrow the gap between what we observe in nature and what we build in computability theory, Kreisel’s thesis will eventually need to be hedged around with qualifications — qualifications which essentially express a historical view of what comprises a ‘discrete physical system’ (tacit in the statement of Thesis P, even).

What is clear from the mathematics is the simplicity of how incomputability arises from computability. We just take an overview of a sufficiently complex computable function, and what we observe (the range of the function) happens to be not computable. Incomputability is what lies ‘at the edge of computability’ as much as emergence lies ‘at the edge of chaos’ — where the high degree of interactivity in the physical situation corresponds naturally to the global observation in the mathematical setting. In both cases, the new level of information content is achieved via a breakdown of inductive structure, a kind of phase transition. To pursue this analogy further one needs to mathematically model the way in which emergent forms feed back into the system, becoming part of the process of ‘bootstrapping’ remarked on by Varela and others.

A special interest in emergence dates back to the beginnings of computability theory. Alan Turing, with his characteristic knack of fixing on scientific questions with hidden significance, was one of the first to try to say something mathematical about the emergence of form in nature, and wrote seminal papers on the morphogenesis — for example [**49**]. According to Odifreddi, '[Gerald] Edelman quotes Turing as a precursor of his work on morphogenesis'.

Turing also had an interest in emergence in the mind, where the emergent forms far outstrip our ability to describe them mathematically. In 1939 he published a paper, little understood at the time, using the constructive ordinals $\mathcal{O}$ of Church and Kleene to inductively extend theories via Gödel-like unprovable sentences. On these opaque technicalities he was able to base some interesting speculations regarding the non-algorithmic nature of intuition. Here is what Turing ([**47**, p. 134–135]) says about the underlying meaning of his paper:

> *"Mathematical reasoning may be regarded . . . as the exercise of a combination of . . . intuition and ingenuity. . . . In pre-Gödel times it was thought by some that all the intuitive judgements of mathematics could be replaced by a finite number of . . . rules. The necessity for intuition would then be entirely eliminated. In our discussions, however, we have gone to the opposite extreme and eliminated not intuition but ingenuity . . . "*

Here he is addressing the familiar mystery of how we often arrive at a mathematical result via what seems like a very unmechanical process, but then promptly retrieve from this a proof which is quite standard and communicable to other mathematicians. Poincaré was also interested in the role of intuition in the mathematician's thinking. A few years after Turing wrote the above passage, Jacques Hadamard [**24**] recounts how Poincaré got stuck on a problem (the content of which is not important):

> *"At first Poincaré attacked [a problem] vainly for a fortnight, attempting to prove there could not be any such function . . . [quoting Poincaré:]*
>
> *Having reached Coutances, we entered an omnibus to go some place or other. At the moment when I put my foot on the step, the idea came to me, without anything in my former thoughts seeming to have paved the way for it . . . I did not verify the idea . . . I went on with a conversation already commenced, but I felt a perfect certainty. On my return to Caen, for conscience sake, I verified the result at my leisure."*

Just as we have been, Turing and Poincaré are talking about the apparent breakdown in the algorithmic glue holding our world together. And what they are pointing to is how this is reflected in the way we come by the descriptions of what is happening. As Holland [**26**, p. 9] comments:

> *". . . developing a theoretical construct in science . . . is not a matter of deduction. The standard deductive presentation of theoretical constructs in science hides the earlier metaphor-driven models that lead to the constructs."*

To summarize — what we have noticed so far is the ubiquity of emergent phenomena in nature, and the distance between mathematically describing them, and between extracting predictions from those descriptions we can find. We have examined the parallel gap between mathematical definability and computability, and linked the physical situation to well-known hierarchies of incomputable objects. This parallel was reinforced when we noticed the role played by globality — on the one hand via quantification, on the other as attractors arising from local, but highly interactive, complexity — in the emergence of incomputability and of new physical relations. If we understand more about definability in particular structures, the hope is, we may find general and unifying principles governing the way the world works. Mathematical logic (remembering the seminal [**41**] of chaos theory innovator Robert Shaw) may even have something to say about something as mundane as a dripping tap!

3. Ontology and Mathematical Structure

We have noticed some basic things about how we extract mathematical descriptions of emergent phenomena. More dramatic in some ways is the observation that descriptions arising from physical structures have themselves a physical reality, whether or not our senses readily confirm that. It is hard to formalize any criterion for distinguishing one particular contingent description as being any more *real* than another. This is nothing to do with old philosophical questions about the intrinsic reality of mathematics. I will keep to areas where a philosophically naïve mathematician can say something useful, and the philosopher can benefit from a mathematical perspective.

We see around us a level of existence, inhabited by us with a reasonable level of success. We are all too aware of other levels, distanced from us by the limitations of reductive science. Our own activities are complex enough to throw up emergent patterns which constrain our lives, but of which we can be imperfectly aware. Here is Hermann Broch, around 1930, on the collective madness of the First World War, and how uncertain a grip on it individual rationality provided (from *Schlafwandler*, translated [**1**] by Willa and Edwin Muir, p.374):

> *"Our common destiny is the sum of our single lives, and each of these single lives is developing quite normally, in accordance, as it were, with its private logicality. We feel the totality to be insane, but for each single life we can easily discover logical guiding motives. Are we, then, insane because we have not gone mad?"*

On the other hand, our everyday physical world which we see around us we now know to be a less-than-solid crust on an ocean of subatomic particles. It is clear that the emergent shapes we live amongst would be as elusive to an observer of subatomic proportions, inhabiting quantum reality, as are the patterns we seek to detect in human history and civilization.

Looking from above, we do believe classical reality to be firmly based on the underlying quantum level, but know better than to try and reduce our everyday problems to this substratum. Analysis at one level is in terms of the relations appropriate to that level, and depends on the algorithmic content of these. We can recognize entities and laws of behavior at the quantum level. But whatever descriptions we can identify relevant to the way the different levels relate, we cannot depend on their predictive content.

Looking from below, even entities and laws are hard to grasp. We certainly cannot "see" them in the way we see our own world. Any school student who has to write an essay on the causes of the First World War is made all too aware of this! The relations by which higher forms are connected are not of our world, and observation of them must be indirect. But they would be entirely real to an inhabitant of that higher world, which is no more surprising than is the sureness with which we move around above our hypothetical subatomic observer. As Holland [**26**] describes it, p. 7:

> *"Persistent patterns at one level of observation can become building blocks for persistent patterns at still more complex levels."*

We now see emergence not just as a producer of unexpected patterns, but as the midwife of different levels of physical reality. And the resulting ontology can be translated into *any* context in which our everyday conceptual framework is stretched or actually fails us. What happens when the familiar laws of nature become invalid, such as near a singularity of a standard model of some aspect of the Universe? How can we say something about the nature of existence itself? If we want mathematical models relevant to such questions, we need to think in very basic terms about existence and its emergent structure.

4. Where Does It All Start?

It is hard to say anything new, unless it comes out of some relatively new piece of knowledge — as in this article, some mathematical notions which have not been widely thought about. Let us begin with the *principle of sufficient reason*, which in the words of Gottfried Wilhelm Leibniz (see [**32**, Secs. 31, 32]) says that:

> *"... there can be found no fact that is true or existent, or any true proposition, without there being a sufficient reason for its being so and not otherwise, although we cannot know these reasons in most cases."*

The mystery of why anything exists at all is beyond us, of course. This is the ultimate failure of reductionism, something which finds its echo in so many aspects of science and human knowledge in general. Hermann Broch [**1**] describes the role of God here in terms of the non-Euclidean point of intersection of two parallel lines, reducing our changing view to presentational 'style' (p. 426):

> *"... the First Cause has been moved beyond the 'finite' infinity of a God that still remained anthropomorphic, into a real infinity of abstraction; the lines of inquiry no longer converge on this idea of God ..., cosmogony no longer bases itself on God but on the eternal continuance of inquiry, on the consciousness that there is no point at which*

> *one can stop, that questions can forever be advanced, that there is neither a First Substance or a First Cause discoverable, that behind every system of logic there is still a meta-logic, that every solution is merely a temporary solution, and that nothing remains but the act of questioning itself . . . "*

But existence itself — of facts or other entities — must take a form inductively determined by the mathematics of its emerging specifics. It must materialize according to sufficient reason, and that mathematics we know tells us that mathematical entities can exist which are not uniquely determined by their context. From which observation, we might expect to encounter realities which are not fully determined, and so materialize according to a range of possibilities. What happens when a particle's existence is guaranteed by 'sufficient reason', but its history is not? Clearly, if there is no pressing reason for a unique history, but there is a pressing reason for history, then it must be non-unique history for which there is sufficient reason. And then, from this perspective, there is nothing mysterious in itself about the parallelism encountered at the quantum level, for instance — so long as we have a convincing mathematical model within which such failures of 'sufficient reason' are expected to occur.

So what do we mean in mathematical terms by 'sufficient reason'? How does a structure determine facts or entities which are not obviously part of the basic knowledge we have of that structure. Clearly, if we can uniquely describe some object or relation in the structure, then, according to what we said in the previous section, it has an existence. And again from what we said before, the corresponding mathematical concept is that of *definability* in an appropriate mathematical model. However, in mathematics language is known to have limited descriptive powers, and our experience of everyday life suggests a similar situation there. In mathematics one can expand ones language, but in human communication there are obvious limitations on the usefulness of this.

This does not stop us seeking out extended notions of 'description' to help us make sense of the origins of perceived reality.

There is another way of making mathematically precise what we mean by an aspect of a structure being determined by sufficient reason. Let us first consider a simple example of an organization in which the members each fill their own individual roles. It may be possible to reorganize the membership in such a way that the organization continues to function just as it did before the reorganization - a number of people now do different jobs from previously, but observers of the workings of the organization only notice that certain operatives have different names. It may be though that however one reorganizes, certain people necessarily have to retain their original allocated positions, due to particular personal qualities or expertise, or because of *the relationships of these to the other members of the organization.* These people are *invariant* under any such reorganization, so we can reasonably say that that constitutes sufficient reason for them having their designated jobs. The mathematician will instantly recognize in such a reorganization of a structure an *automorphism* — that is, a one-to-one mapping of the structure onto itself which retains all the basic relations between members of the structure. And the notion of invariance need not only be applied to the individuals of the structure, but to any relation on it. For instance, there may be an invariant *set* of members of the structure, which is not moved by any automorphism (although there may be movement within the set). Of course, it may be that a structure has no automorphisms which move anything (all the automorphisms are *trivial*), in which case the structure is said to be *rigid*. Back in the real world, a rigid Universe would be one in which there was no quantum ambiguity, in which every history was uniquely determined by sufficient reason of things necessarily being that way. One can see from this that if we succeed in finding an appropriate mathematical model of the real world, then an understanding of its automorphism group might clarify many mysteries facing scientists.

What we have so far is a mathematical framework within which to set a basic model, but no model as yet. Given a model, the framework promises to go some way towards helping it play a fundamental role in explaining emergence and the familiar fragmentation of scientific knowledge. To get any further, we need to think constructively.

How do we start out from nothing, and end up with a whole Universe? We cannot escape from the mystery of existence, which is not in keeping with our broadly interpreted principle of sufficient reason, which underlies most of our thinking so far. We have to fit our intuitive faith in causality to some assumption about what is given to us without reason. When we say 'without reason' here, we are thinking about immanence, and talking about phenomena originating not from within the known bounds of our universe. And when we say 'given to us without reason', we must also include here rules for actions which violate the principle of sufficient reason.

The pre-scientific scenario was that we were given the world roughly as we see it now, by some divine intervention. This takes many different forms, for instance that of Genesis making some concessions to the modern conception of form created out of formlessness. What at first seems a very different view, that of the quantum theorist who asks us to accept randomness as a given, actually turns out to have a lot in common with this picture. The difference is that in Genesis the formlessness derives information content, and is turned into our familiar classical reality, by a god beyond human comprehension: whereas in the latter scenario the classical world emerges according to various unverified speculations from a quantum world featuring randomness, and an existing high information content. Of course, what is well-known is that randomness is as much a symptom of high (if hidden) information content as is richness of form. And there is no such thing as absolute randomness. Mathematically, randomness is defined relative to the level of information content of those forms it avoids.

Since there is no ultimate information content, there is no absolute randomness, but the more randomness displayed, the greater the accompanying information content. You must be very clever to avoid what people expect of you! This makes randomness as a given a vague and unsatisfying assumption, and certainly one which violates our principle of sufficient reason much as Genesis and other creationist conceptions do.

No, if we are to start with little, we must have regularity, predictability, algorithmic content. And as we have seen, it can be a short step from such simple beginnings to complexity and emergent new regularities. And such beginnings are not just capable of providing a basis for real-world-like complexity — they are in keeping with our basic quest for manifest and sufficient reason in all things. Even knowing how far we must be from understanding, or even observing, any 'First Cause', this does not stop us observing the levels of human experience we do have access to, and trying to find unifying principles behind them.

To summarize again: We saw in Secs. 2 and 3 how one could get a better understanding of our experience of the real-world by just looking at the logical framework governing how it is described. To an extent, such things as emergence and the fragmentation of science are aspects of the basic structure of information content. In this section we moved on to attempt to mathematically *build* a universe with some relationship to our own by applying sufficient reason to the germs of an instantiation of such structure.

Of course, underlying this must be some assumptions about the way in which information content is presented (what is information content?) and its relationship to the perceived reality it seeks to capture. Information is what we extract from our experience of the world around us. For the scientist it is framed in terms of real numbers. And it has become increasingly standard practice to view information and the material universe as essentially interchangeable. This is not to say that energy and matter *are* just information — but we find them to be neatly captured in informational packages, which as far as we are concerned, correspond with their physical presences. In mathematical terms, we are talking about the familiar, but far from simple, notion of a *presentation* of a structure. This identification of information and the material universe is a viewpoint that has delivered many valuable insights in both directions, both in information theory and science. The possibility that a true picture of the Universe depends on something that is not describable as 'information', that cannot form part of a model based on information content, cannot be excluded, of course. And this could well be relevant, just as admitting many-worlds can be used to construct a (to some people) satisfying narrative. Or, for that matter, just as the creationist scenario does. But is it *necessary* to admit such a possibility? Let us see if we can run with a mathematical model of an immanently formative Universe which enables us to avoid this level of metaphysics. And let us continue to wield Occham's Razor with gusto, hoping that no damage is done — and that it will make more obvious what sort of structure is most appropriate to describe a germinal version of our universe, and beyond.

5. Towards a Model Based on Algorithmic Content

We want a model that is truly fundamental, that is not already built upon specifics of our universe which we aim to understand better. We want it to capture the essence of how information content is structured in the real world, without losing its wide applicability.

Now, it is common in mathematics to de-emphasize the distinction between object and process. This is seen, for instance, in the construction of certain programming languages, such as LISP, Haskell and Erlang, derived from Church's lambda calculus. There are parallel fluidities in nature, such as between energy and matter, or between waves and particles. However, as participants in the world, the distinction between object and process has an immediacy and qualitative difference which it is not useful to abstract away from. Our reason for maintaining a distinction is based on a need to retain some sense of type structure, where matter is observed, actions need to be predicted. We objectify matter more easily than actions. Our aim is to get a model closer to our experience.

For us, objects present observable, sometimes very complex, information (this is what makes them objects for us), whereas processes are received as relations between entities. Objects we are used to having *high information content*. They are the subjects of events (such as being observed). We try to *predict* processes, to reduce them to simpler components, and are more disconcerted by a lack of algorithmic content. One may say, there is no strict dividing line here, and the differences are ones of degree, but this does not detract from the usefulness of the distiction. Scientifically, the dichotomy is an essential one, and can be made precise. What the working scientist tries to achieve is firstly a presentation of some physical configuration as a real number, and then an algorithm for computing its new value displaced (such as in time or space) by some other appropriate real. In essence, the scientist aims to make predictions in the form of algorithmic relations or functions between reals. If successful, he or she arrives at a relationship between reals which can be computably *approximated* — in that, a close rational approximation to the input to the algorithm yields a correspondingly close approximation to the output. Mathematically, we want our algorithmically described process to be *continuous*. Of course, many processes in nature cannot be presented in such a continuous way. But everything we know (with some notable exceptions at the quantum level, which we will return to later) tells us that these involve non-linear phenomena describable in terms of more basic processes which *are* continuous. What lies behind such phenomena is the infinitary interactivity discussed earlier, and what comes out of our model is a presentation of emergence in terms information-theoretic structure.

All we are trying to do here is to justify a model which gives different roles to information content and relations over it. And which deals with those relations in the first instance in terms of their algorithmic content. One could say more regarding the feasibility and naturalness of doing this. But for now, it suffices to notice that for particular applications the discussion becomes vacuous, and in very general contexts the usefulness of the model motivates a necessary search for greater clarity. In any case, it is useful to have an analysis of the algorithmic content of computationally complex environments, for which one needs to objectify that which is not algorithmic. Then the usefulness of the analysis grows with our conviction of the naturalness of its basis.

We owe to Alan Turing [**47**], who thought a lot about interactive computing, the precise mathematical ideas on which is based the standard model of computationally complex environments. Having described in 1936 [**46**] a mathematical model of mechanical computability, which is still the basis of much of computability and complexity theory, his 1939 paper sought to relate computability and certain describable relations over the natural numbers. In doing so, he allowed his computing machines to interact with an 'oracle', providing an exterior source of information, which may or may not be computable information. These oracle machines, in computing finite pieces of information from finite pieces of information, can be used to compute from reals to reals via suitable corresponding rational approximations. The associated functions are algorithmic, and continuous over the reals. They exactly correspond to how our working scientist aims to capture the algorithmic content of the universe.

Mathematically, Turing's oracle machines give the real numbers an algorithmic infrastructure, which comprises the *Turing universe*. Emil Post [**39**] gathered together the computably equivalent reals of this structure, and called the resulting ordering the *degrees of unsolvability* — later called the *Turing degrees* — and this has become the mathematical context for the study of the Turing universe.

It is not surprising that attention has turned to Turing's universe of computably related reals as providing a model for scientific descriptions of a computationally complex real universe (see [**4**], [**7**], [**8**], etc.) What is surprising is that it has taken so long to happen — see [**6**] for some comments on why this should have been so. This new interest in the Turing universe is based on a growing appreciation of how algorithmic content brings with it implicit infinities, and, as we have already mentioned, a science — increasingly coming to terms with chaotic and non-local phenomena — necessarily framed in terms of reals rather than within some discrete or even finite mathematical model. However, most of the research activity concerned with the computational significance of evolutionary and emergent form, and emergence in more specific contexts, has inevitably been ad hoc in nature. The potential for drawing out unities and universalities here is as yet almost untapped. Turing's work on emergence of form in nature, and his seminal papers on the topic — for example [**49**] — rather fit this pattern. After 1939, Turing the inventor of oracle computing machines seems to have had no direct impact on his later work on interactive computing and morphogenesis.

Of course, the relevance of such a model in a particular situation depends on the relative importance of specific properties of the algorithmic content present and those common to a wide range of algorithmic structures. It may be that in certain closely constrained situations, the general analysis does little more than provide a conceptual context for results arrived at by non-computability theoretic means.

There are other considerations. For instance, do we want a model which tells us about the local escalation of information content into the incomputable, or one which tells us something global about already computationally complex environments? There is a strand of thought — the hypercomputational, as Jack Copeland and others term it — concerned with contriving incomputability via explicitly physical versions of the Turing universe. What is common to both the computability-theoretic and hypercomputational strands is that both the emergence of incomputability, and the emergence of new relations in a universe which admits incomputability, are based on a better understanding of how the local and the global interact. Whatever the context, the key mathematical parallel here is that of definability or invariance, even if within rather different corresponding frameworks. This is not very explicit in building hypercomputational models, which enables Martin Davis [**13**] and others to trivialise what is happening as being the use of oracles to shuffle around existing incomputability.

It is not surprising that the human mind points us in the direction of particular hypercomputational models.

Speculations regarding the potential of new connectionist theories to transcend the classical McCulloch and Pitts [**33**] artificial neuron formalism have been around for some time — for instance, in 1988 Smolensky [**42**, p. 3] observed:

> *"There is a reasonable chance that connectionist models will lead to the development of new somewhat-general-purpose self-programming, massively parallel analog computers, and a new theory of analog parallel computation: they may possibly even challenge the strong construal of Church's Thesis as the claim that the class of well-defined computations is exhausted by those of Turing machines."*

Turing himself anticipated the importance now given to connectionist models of computation — see his discussion of "unorganized machines" in [**48**], and Jack Copeland and Diane Proudfoot's article [**10**] "On Alan Turing's Anticipation of Connectionism." What we are seeing now for the first time is the adoption of Turing's own oracle model of interactive computation in a real world setting, and the enlisting of the explanatory power of the mathematical theory of the Turing universe based on that model.

The fact that the hypercomputational case has brought us little but confusion so far does not mean that it is a bad project. What it may mean is that on the one hand the mathematical tools for analysing hypercomputational proposals are just too crude, that those who have to evaluate the results of their application are either lacking mathematical sophistication in some part, or have less expertise in thinking about the real world than they have in the rigours of recursion theory — and most importantly, that there is still thinking to be done before vague intuitions can be convincingly communicated. The picture we carry forward from our earlier discussions is of processes operating over some raised level of information content, sufficient to give us a stucture in which new relations can be described. This picture is hierarchical, information defined within structures which can only be fully contained within new structures. The Turing model may be appropriate for clarifying some big scientific mysteries, but maybe needs refining for a better understanding of hypercomputation, perhaps incorporating little-world constraints on time and space. And at the level of human affairs, which can never be successfully captured by our scientist working over the reals, maybe arguments from analogy need to be carried forward to a more detailed consideration of definability over structures based on even higher information content.

6. Levels of Reality

But let us now return to what the Turing model can do. Let us try to be more clear about how, from very simple beginnings, we can get from the basic fact of existence to what is for us an even greater puzzle — because we have to take what is happening under the umbrella of sufficient reason — the quite amazing emergence of individual entities. From this point of view, it is not quantum ambiguity which is surprising, but the existence of the well-defined world of our everyday experience.

More generally, we have the problem that even though we have natural laws to help us understand much of what happens in the universe, we have no idea where those laws themselves come from. Their apparent arbitrariness lies at the root of the present day confusion of speculative science, verging on the metaphysical.

For Alan Guth [**23**], the problem is:

> *"If the creation of the universe can be described as a quantum process, we would be left with one deep mystery of existence: What is it that determined the laws of physics?"*

While Roger Penrose [**35**] asks for a *strong determinism*, according to which (pp. 106–107):

> *". . . all the complication, variety and apparent randomness that we see all about us, as well as the precise physical laws, are all exact and unambiguous consequences of one single coherent mathematical structure."*

Science has come a long way since David Hume first set out to 'enquire how we arrive at the knowledge of cause and effect', and insisted, in *An Enquiry Concerning Human Understanding*, that:

> *"I shall venture to affirm, as a general proposition, which admits of no exception, that the knowledge of this relation is not, in any instance, attained by reasonings a priori, but arises entirely from experience, when we find that any particular objects are constantly conjoined with each other. Let an object be presented to a man of ever so strong natural reason and abilities; if that object be entirely new to him, he will not be able, by the most accurate examination of its sensible qualities, to discover any of its causes or effects. Adam, though his rational faculties be supposed, at the very first, entirely perfect, could not have*

> *inferred from the fluidity and transparency of water that it would suffocate him, or from the light and warmth of fire that it would consume him. No object ever discovers, by the qualities which appear to the senses, either from the causes which produced it, or the effects which will arise from it; nor can our reason, unassisted by experience, ever draw any inference concerning real existence and matter of fact."*

Hume may still be right, but the match between mathematics and experience has become more all-embracing, with string theory perhaps the most ambitious of the attempts to unify the two. The Turing model may be as yet very far from clarifying the specific details of relativity or quantum theory, but it does promise a release from the arbitrariness to which all less basic theories — superstring theory, M-theory, inflation, decoherence, the pilot wave, gauge theory, etc. — are subject, and is based almost entirely upon experience.

What is specially relevant here is that, far from Hume's comments causing problems for us, they can be used to clarify not just how we 'draw any inference concerning real existence and matter of fact', but, further, how in general entities and relations derive existence from their global context. Reading Hume, we find a graphic description of how we derive predictive patterns from observations of events. We recognize a parallel between how we know things — a process of *definition* by accumulated experience, of establishing an *invariance* emerging from various possibilities — and in the way the Universe can 'know' itself, and immanently establish its own structure and relations. Although what the Turing model primarily tells us about is not an emergence of particular events from events, but of natural laws from the structure of information content.

What does the Turing model suggest regarding the basic structure of matter and the laws governing it? Let us review some of the ground covered in more detail in [**7**].

What we know of the Turing universe is consistent with the possibility that the information content or level of interactivity of a given entity may be insufficient to guarantee it a unique relationship to the global structure. This is what one might expect to apply at an early stage in the development of the universe, or at levels where there is not a sufficiently density of interactions to give information a global role. A number of classic experiments on subatomic particles confirm such a prediction. On the other hand, mathematically entangling such low level information content, perhaps with content at levels of the Turing universe at which rigidity sets in, will inevitably produce new content corresponding to a Turing invariant real. The prediction is that there is a level of material existence which does not display such ambiguity as seen at the quantum level, and whose interactions with the quantum level have the effect of removing such ambiguity — confirmed by our everyday experience of a classical level of reality, and by the familiar 'collapse of the wave function' associated with observation of quantum phenomena. Since there is no obvious mathematical reason why quantum ambiguity should remain locally constrained, there may be an apparent non-locality attached to the collapse. Such a non-locality was first suggested by the well-known Einstein-Podolsky-Rosen thought experiment, and, again, has been confirmed by observation. The way in which definability asserts itself in the Turing universe is not known to be computable, which would explain the difficulties in predicting exactly how such a collapse might materialize in practice, and the apparent randomness involved.

One might hope that in the course of time the theory of Turing definability might explain aspects of subatomic structure. A conjecture is that when one observes atomic structure, one is looking at *relations* defined on some lower level of matter lacking any sort of observable form, out of which arise peaks of definability observed by us as subatomic particles. This may even lead to a theoretical explanation of 'dark matter'. Until such matter is organized into relations, of which particles are the instantiations, we have no structure capable of being interacted with. It would be as alien to the world of particle physics as that world is to our classical level of human existence.

As we have already mentioned, the Turing model may have implications for how the laws of nature immanently arise. And also how they collapse near the big bang 'singularity', and the occurrence or otherwise of such a singularity. What we have in the Turing universe are not just invariant individuals, but a rich infrastructure of more general Turing definable relations. These relations grow out of the structure, and constrain it, in much the same sort of organic way observable in familiar emergent contexts. These relations operate at a universal level. The prediction is that a Universe *with sufficiently developed information content* to replicate the defining content of the Turing universe will manifest corresponding material relations. The existence of such relations one would expect to be susceptible to observation, these observations in turn suggesting regularities capable of mathematical description. And this is what the history of science confirms. The conjecture is that there is a corresponding parallel between natural laws and relations which are definable in an appropriate fragment of the Turing universe.

The early Universe one would not expect to replicate such a fragment. The homogenization and randomization of information content consequent on the extreme interconnectivity of matter would militate against higher order structure. The manifest fragment of the Turing universe, based on random reals, might still contain high information content, but content dispersed and made largely inaccessible to the sort of Turing definitions predicted by the theory. Projected singularities, such as within black holes or associated with boundary states of the Universe, depend on a constancy of the known laws of physics. But immanently originating laws must be of global extraction. This means that their detailed manifestations may vary with global change, and disappear even.

Notice the difference here between what we are saying, and what the upholders of the various versions of Everett's many worlds scenario are. On the one hand, we have an application of the principle of sufficient reason to the world as we know it, which gives a plausible explanation of quantum ambiguity, the dichotomy between quantum and classical reality, and promises some sort of reconciliation between science, the humanities, and our post-modern everyday world. On the other we have something more like metaphysics.

The Turing model, and its connections with emergence, also lead us to expect the familiar fragmentation of science, and human knowledge in general. As we know from computability theory, a Turing definition of a given relation does not necessarily yield a computable relationship with the defining information content. But working within the relations at a given level, there may well be computable relationships emerging, which may become the basis for a new area of scientific investigation. For instance research concerning the cells of a living organism may not be usefully reduced to atomic physics, but deals with a higher level of directly observed regularities. Sociologically, one studies the interactions governing groups of people with only an indirect reference to psychological or biological factors. Entire relations upon cells (humans) defined in some imperfectly understood way by the evolutionary process provide the raw material underlying the new discipline, which seeks to identify a further level of algorithmic content. This algorithmic content may not be directly expressed in terms of numbers. But inasmuch as the area in question does have basic notions, corresponding to the new emergent relations, shared by workers in the field, and descriptions of entities and regularities are formulated in a shared language, the algorithmic content is not dissimilar in kind to that at lower levels.

In [**4**] we mentioned a number of areas in which one can observe qualitatively similar problems, all connected with parallel issues of definability and nonrigidity. One example is that of the origin of life on Earth. Another concerns the exact nature of evolution — as Stuart Kauffman [**28**] observes (p. 644):

> *"Evolution is not just 'chance caught on the wing.' It is not just a tinkering of the ad hoc, of bricolage, of contraption. It is emergent order honored and honed by selection."*

There is the mysterious emergence of large scale structure in the Universe. Also in the 1999 paper is a section on epistemological relativism. There is a basic intuition that an analysis of the epistemology derived from our Universe is potentially just as complex as that of the Universe itself. So it should not be surprising that emergence and the mathematics of definability should be relevant here. And there is the whole question of the nature of human thought processes, touched on earlier.

There are questions about the range of possibilities embodied in such things as quantum ambiguity: Going from the uniqueness of a defined phenomenon to — what? Are there any overall constraints apart from those imposed by the mathematics specific to the emergent structures? There seems to be one unavoidable rule — obvious when it is pointed out — which is that each superimposed alternative must be viable by itself. Which, in addition to the specifics, demands that the information content develops within the rules experience and the computability theory lead us to expect. In particular, there can be at most countably many such alternatives. It is known that there exist at most countably many Turing automorphisms.

What may be most important of all, though, is the way we get a new model, replacing the one Laplace's predictive demon gave us [**31**] around 200 years ago:

> *"Given for one instant an intelligence which could comprehend all the forces by which nature is animated and the respective situations of the beings who compose it — an intelligence sufficiently vast to submit these data to analysis — it would embrace in the same formula the movements of the greatest bodies and those of*

> *the lightest atom; for it, nothing would be uncertain and the future, as the past, would be present to its eyes."*

We now see not only the bottomless mystery in Broch's point at infinity where foundations fail, but a hierarchy of lesser mysteries rising through the entire structure. The idea of a controlling god makes no sense in any context, but, taking god to be the embodiment of an unpredictable creative principle in all things, that of a ubiquitous god does. And, whether we are religious or not, man is indeed made in the image of this god, a microcosm of the wider universe, part of and contributor to its emergent creativity. The all-understanding humanity of enlightenment science may be dead, but the vitality of our participation in the world takes on a new life.

7. Algorithmic Content Revisited

The reader may still be left with some basic objections, not just to what we have been saying, but to the very advocacy of computability theory as applied science. These basic doubts are not going to be dissipated by specific examples of the usefulness of the techniques of computability, such as Robert Soare's beautifully presented paper [**43**] on *Computability theory and differential geometry*. One can address particular objectives — in this case a rebuttal of *Simpson's Thesis*, clarified by S. GṠimpson in a *Foundations of Mathematics (FOM) Network* e-mail communication on Aug. 4, 1999:

> "The concise statement of Simpson's Thesis is:
>
> *Priority methods are almost completely absent from applied recursion theory.*"

One just ends up like the boy trying to seal the breach in the dyke with his finger. Paradigm changes depend on a number of ingredients being in place, including the unifying concept behind the individual pieces of evidence.

For those with a finitist view of the Universe, almost everything said, right from the beginning, will be at best irrelevant, including any argument about the role of priority methods. It is hard to dislodge an outlook traceable back to the beginnings of science. Here is Archimedes in the introduction to *The Sand Reckoner*:

> *"Many people believe, King Gelon, that the grains of sand are infinite in multitude; and I mean by the sand not only that which exists around Syracuse and the rest of Sicily, but also that which is found in every region, whether inhabited or unhabited. Others think that although their number is not without limit, no number can ever be named which will be greater than the number of grains of sand. But I shall try to prove to you that among the numbers which I have named there are those which exceed the number of grains in a heap of sand the size not only of the earth, but even of the universe."*

But — despite the fact there are probably less than 10^{87} particles in the universe — for most of us, a finite model of the universe will not do, as the sort of things said in Secs. 2 and 3 should hopefully persuade all but the most incorrigible finitists. But will not, of course, in view of the already extensive literature on the topic!

It is worth trying to be a bit more explicit, though. We argued that presentations of aspects of the universe lead us to particular mathematical models. And that if the model fits closely enough, things describable in terms of that model can be expected to be aspects of the original physical situation, maybe not visible to us, but a very real element in us developing an understanding of that system. How can it be that a structure which masquerades as being finite, on closer inspection necessarily needs an infinite structure to explain it? The key ingredient is algorithmic content, and this derives from our basic principle of sufficient reason. Despite Humean caution, we can go beyond a purely fortuitous link between experience and form.

How does one envisage a germinal Universe, involving minimal information content, in which something recognizable as ‘events’ occur — which, we may suppose, also have the most basic information content imaginable. In such an impoverished (but very strange!) environment, there cannot be ‘sufficient reason’ for diversity within (or for unique manifestation of) particular modes of event. In other words, we already find it hard to avoid (there is just not enough information content) to make particular kinds of development non-uniform. But our only constraint on the actuality of this nascent structure is its mathematics, a mathematics which has a general applicability to similar structures, and to this structure at similar stages of its development. And the mathematics of such structures with such a uniformity of infrastructure is what we can only characterize as *algorithmic*. Of course, the mathematics may not actually give us well-defined events. But even the ambiguity must be uniformly instantiated.

So the close relationship between the mathematics and its real-world avatar entails a Universe which is not just hard to understand apart from its algorithmic content, but which actually *embodies* algorithmic content. As Hume would have us know, the exact nature of that algorithmic content may be beyond reason (despite the advances in mathematics and science since his time), but his vision of how we do know things presages a mathematical model of how entities develop in more general contexts. By so closely following his analysis, we come up with a mathematics he would find hard reject the relevance of.

Why do the laws of physics appear so uniform throughout the Universe? Why do they appear to be algorithmic in effect? The more interesting question is: How could they be otherwise?

So the finite model is not just impractical, it fails to describe what is happening. Neither, we suspect, can science live with its close relation, the discrete model. Even if Richard Feynman did suggest [**17**], after a scientific lifetime working with mathematics over the reals, the following radical resolution of the uneasy relationship between reality and its discrete representations:

> *"It is really true, somehow, that the physical world is representable in a discretized way, and ... we are going to have to change the laws of physics."*

But even if Feynman were right applied computability theory is not affected, at least until we gather more convincing evidence for Kreisel's Thesis P.

8. What Is to Be Done?

The theory of Turing definability is a notoriously difficult and dangerous area of research. It is the mathematical equivalent climbing Everest's Kangshung face or K2's Magic Line (and you can fall off). So far, we have only achieved a glimpse of the richness of structure hidden there, a fitting counterpart to that of the real world. As we observed in [**6**], the complexity of the Turing model was not always seen in this light. In the late twentieth century, relative computability became area of research known for its mathematical unlovelyness and forbidding pathology. Its relevance was not at all clear during the recursion theoretic years. The difficulty of the area may have surrounded researchers of the 1960s — pre-eminently Gerald Sacks — with a vaguely heroic, even machismatic, aura. But as time went on this had become a double-edged weapon, and by the 1990s almost no one was impressed by the length and incomprehensibility of groundbreaking new proofs. "Touching the Void" — and having accidents — was all very well for mountaineers but, as the new century approached, mathematics was very much about deliverables. At times the very value of research into relative computability was questioned. Images of such dissent stick in the memory: Sacks himself, lecturing at Odifreddi's CIME summer school in Bressanone, Italy in 1979, illustrating his view of 'Ordinary Recursion Theory' with a slide of the Chinese masses in cultural revolution turmoil — his metaphor for an activity obsessive, formless, pointless; or, ten years later, Robin Gandy's contribution to a discussion on the future of logic, at a conference in Varna, Bulgaria — communicating an impression of the structure of the Turing degrees via exaggeratedly desperate scribbles on a blackboard.

Things have changed, and we have what we described in [**6**] as a 'Turing renaissance'. What is currently so exciting is that the sorts of questions which preoccupied Turing, and the very basic extra-disciplinary thinking which he brought to the area, are being revisited and renewed by researchers from quite diverse backgrounds. What we are seeing is an emergent coming together of logicians, computer scientists, theoretical physicists, people from the life sciences, and the humanities and beyond, around an intellectually coherent set of computability-related problems. The recurring and closely linked themes here are the relationship between the local and the global, the nature of the physical world, and within that the human mind, as a computing instrument, and our expanding concept of what may be practically computable.

The specific form in which these themes become manifest are quite varied. For some there is a direct interest in incomputability in Nature, such as that coming out of the n-body problem or quantum phenomena. For others it is through addressing problems computing with reals and with scientific computing. The possibility of computations 'beyond the Turing barrier' leads to the study of analog computers, while theoretical models of hypercomputation figure in heated cross-disciplinary controversies. There is also intensive research going on into a number of practical models of natural computing, which present new paradigms of computing whose exact content is as yet not fully understood. In many scientific areas it is the emergence of form which is deeply puzzling, and, as we have described above, there is a key role here for the sort of mathematical models we have been discussing.

However, what we have described so far has been largely twentieth-century mathematics, even if many of the ingredients only appeared in the final decade of the last century. Many of the key ideas were described in the 1999 paper [**4**], while the joint paper of Cooper and Odifreddi [**7**] was largely concerned with clarification and elaboration, and with reining in the ambition of that earlier paper. This article is to some extent a further step in that direction. We have dwelled on some basic issues concerning the link between definability and emergence, but for the big picture, [**4**] is still indispensible, in that it draws together so many strands in contemporary science.

We will finish with something very new. Not by any means a retreat, but a making more explicit of some of the limitations of the classical theory of the Turing universe as a model for emergence. And a pointing to other areas in which computability theory can adapt to once again address basic scientific issues. We have pointed to important questions regarding computability theoretic structure and emergence, but not all these relate to standard structures. There appears to be little alternative to the Turing model in relation to ontology and other fundamental questions regarding the origins of the material universe and the emergence of natural laws. There seems to be a specially fundamental role for this analysis in throwing new light on basic puzzles concerning the exact role of entropy, and other areas where thinking is unsatisfyingly ad hoc and bound by scientific cliché. And it is certainly true that at one level, substructures of the Turing model can provide an instantiation of many emergent phenomena. But this is not a useful model for prediction of detail, any more than classical computability addresses practical computational questions in a direct way. What we get is a context, a conceptual resource, a formative influence on the scientific culture, the big picture in the way we expect from logic and philosophy, deep and essential insights — and theoretical foundations on which important practical developments can be based. Of course, Alan Turing's 1936 paper [**46**] did all that.

In [**6**], we made some comments on how in the years following Turing's paper, computability theory became dominated by mathematical concerns which, while stimulating necessary technical developments, took the area away from its real-world context. One early decision [**39**] was to view reducibilities in terms of degree structures. Mathematically, this enabled the new area to be developed in the context of familiar structures such as partial orderings, upper semi-lattices and Boolean algebras. Reducibilities which were not transitive were ruthlessly discarded, despite the fact that in real-life computation, transitivity commonly fails. In fact, part of the puzzling non-locality posed by the EPR thought experiment of Einstein, Podolsky and Rosen [**15**] comes from just such a non-transitivity in relation to events which can be connected in real time and space. The problem is that if one were to take seriously non-transitive reducibilities which correspond to what we meet in the real world, we would have to develop not just new computability-theoretic structures, but new mathematical abstractions with no existing theory. Our orderings would be non-transitive, while our metrics would be non-symmetric — something which physicists, significantly, sometimes talk about. But the computability theory related to such structures does not yet exist. This is a major project, but potentially of great importance. One cannot even begin to imagine how definability in such structures might turn out, or what automorphisms might look like. But one can be confident that the classical theory will continue to play an important role, and to technically underpin new developments. And one can expect to get an even closer relationship between the structures of computability and complexity theory, and their real-life avatars.

References

1. H. Broch, *The Sleepwalkers* (trans. W. and E. Muir). New York, Vintage Intern., 1996.
2. J. Brockman (ed.), *The Third Culture: Beyond the Scientific Revolution*, New York–London, Simon and Schuster, 1995.
3. F. Capra, *The Web of Life : A New Scientific Understanding of Living Systems*, London, Harper Collins, 1996.
4. S. B. Cooper, *Clockwork or Turing U/universe? – remarks on causal determinism and computability*, In: S. B. Cooper and J. K. Truss (eds.), *Models and Computability* London Math. Soc. Lect. Note Ser. **259** (1999), pp. 63–116.
5. S. B. Cooper, *Computability Theory*, New York–London, Chapman & Hall/ CRC Press, 2004.
6. S. B. Cooper, *The incomputable Alan Turing*, In: Proceedings of *Turing 2004: A celebration of his life and achievements* [To be electronically published by the British Computer Society, 2005].
7. S. B. Cooper, and P. Odifreddi, *Incomputability in nature*, In: S. B. Cooper and S. S. Goncharov (eds.), *Computability and Models: Perspectives East and West*, New York etc., Kluwer Academic/ Plenum Publishers, 2003, pp. 137–160.
8. J. Copeland, *Turing's O-machines, Penrose, Searle, and the brain*, Analysis, **58** (1998), 128–38.
9. J. Copeland, *Narrow versus wide mechanism: Including a re-examination of Turing's views on the mind-machine issue*, J. Phil. **96** (2000), 5–32.
10. J. Copeland and D. Proudfoot, *On Alan Turing's anticipation of connectionism.* Synthese, **108** (1996), 361–377, Reprinted in: R. Chrisley (ed.), *Artificial Intelligence: Critical Concepts in Cognitive Science. Vol. 2: Symbolic AI*, London, Routledge, 2000.
11. J. Copeland and D. Proudfoot, *Alan Turing's forgotten ideas in computer science.* Scientific American, **253:4** (1996), 98–103.
12. M. Davis, *The Universal Computer: The Road from Leibniz to Turing*, New York, W. W. Norton, 2000.

13. M. Davis, *The myth of hypercomputation*, In: Teuscher (2004), 195–211.

14. A. Einstein, *Out of My Later Years*. New York, Phil. Library, 1950.

15. A. Einstein, B. Podolsky, and N. Rosen, *Can quantum mechanical description of physical reality be considered complete?*. Phys. Rev. **47** (1935), 777–780.

16. G. Etesi, and I. Németi, *Non-Turing computations via Malament-Hogarth space-times*. Int. J. Theoretical Phys. **41** (2002), 341–370.

17. R. P. Feynman, *Simulating physics with computers*, Int. J. Theoretical Phys. **21** (1982), 467–488.

18. T. Franzen, *Inexhaustibility - A Non-Exhaustive Treatment*. Association for Symbolic Logic/ A K Peters, Wellesley, Mass. 2004.

19. R. O. Gandy and C. E. M. Yates (eds.), *Collected Works of A. M. Turing: Mathematical Logic*, Amsterdam etc., North-Holland, 2001.

20. R. Geroch and J. B. Hartle, *Computability and physical theories*, Foundations Phys. **16** (1986), 533–550.

21. D. Goldin, S. Smolka, and P. Wegner (eds.) *Interactive Computation: the New Paradigm*, Springer-Verlag [To appear].

22. D. Goldin and P. Wegner, *Computation Beyond Turing Machines*, Communications of the ACM, 2003.

23. A. H. Guth, *The Inflationary Universe – The Quest for a New Theory of Cosmic Origins*, New York etc., Addison-Wesley, 1997.

24. J. Hadamard, *The Psychology of Invention in the Mathematical Field*, Princeton, Princeton Univ. Press, 1945.

25. A. Hodges, *Alan Turing: The Enigma*, London etc., Vintage, 1992.

26. J. H. Holland, *Emergence – From Chaos to Order*. Oxford etc. Oxford Univ. Press, 1998.

27. D. Hume, *An Enquiry Concerning Human Understanding* (1748), (L. A. Selby-Bigge and P. H. Nidditch, Eds.). Oxford, Oxford Univ. Press, 1975 edn.

28. S. A. Kauffman, *The Origins of Order – Self-Organisation and Selection in Evolution*. Oxford–New York, Oxford Univ. Press, 1993.

29. T. Kieu, *Quantum algorithm for the Hilbert's Tenth Problem*, Int. J. Theoretical Phys. **42** (2003), 1461–1478.

30. G. Kreisel, *Church's Thesis: a kind of reducibility axiom for constructive mathematics*, In: A. Kino, J. Myhill, and R. E. Vesley (eds.), *Intuitionism and Proof Theory: Proceedings of the Summer Conference at Buffalo N.Y. 1968* Amsterdam etc., North-Holland, 1970, pp. 121–150.

31. P. S. de Laplace, *Essai philosophique sur les probabilités* (1819); English translation by F. W. Truscott and F. L. Emory, New York, Dover, 1951.

32. G. W. Leibniz, *The Monadology* (1714), In: L. E. Loemker (ed.), *Gottfried Wilhelm Leibniz: Philosophical Papers and Letters*, Dordrecht, 1969.

33. W. McCulloch and W. Pitts, *A logical calculus of the ideas immanent in nervous activity*, Bull. Math. Biophys. **5** (2003), 115–133.

34. P. Odifreddi, *Classical Recursion Theory*, Amsterdam etc., North-Holland, 1989.

35. R. Penrose, *Quantum physics and conscious thought*, In: B. J. Hiley and F. D. Peat (eds.), *Quantum Implications: Essays in Honour of David Bohm*, London–New York, Routledge & Kegan Paul, 1987, pp. 105–120.

36. R. Penrose, *Shadows of the Mind: A Search for the Missing Science of Consciousness*, Oxford, Oxford Univ. Press, 1994.

37. E. L. Post, *Absolutely unsolvable problems and relatively undecidable propositions – Account of an anticipation* (1941), In: M. Davis (ed.), Collected Works of Post, 1994, pp. 375–441.

38. E. L. Post, *Recursively enumerable sets of positive integers and their decision problems*, Bull. Am. Math. Soc. **50** (1944), 284–316.

39. E. L. Post, *Degrees of recursive unsolvability: preliminary report* (abstract), Bull. Am. Math. Soc. **54** (1948), 641–642.

40. D. G. Saari and Z. Xia (Jeff), *Off to infinity in finite time*, Notices Am. Math. Soc. **42** (1995), 538–546.

41. R. Shaw, *The Dripping Faucet as a Model Chaotic System*, Santa Cruz, CA, Aerial Press, 1984.

42. P. Smolenskyo, *On the proper treatment of connectionism*, Behavioral and Brain Sciences **11** (1988), 1–74.

43. R. I. Soare, *Computability theory and differential geometry*, Bull. Symbolic Logic **10** (2004), 457–486.

44. S. Strogatz, *Sync: The Emerging Science of Spontaneous Order*. Hyperion Press, 2003.

45. C. Teuscher (ed.), *Alan Turing: Life and Legacy of a Great Thinker*, Berlin–Heidelberg, Springer-Verlag, 2004.

46. A. M. Turing, *On computable numbers, with an application to the Entscheidungsproblem*, Proc. London Math. Soc. (2) **42** (1936–7), 230–265, Reprinted in: A. M. Turing, *Collected Works: Mathematical Logic*, pp. 18–53.

47. A. M. Turing, *Systems of logic based on ordinals*, Proc. London Math. Soc. (2) **45** (1939), 161–228, Reprinted in: A. M. Turing, *Collected Works: Mathematical Logic*, pp. 81–148.

48. A. M. Turing, *Intelligent machinery* (1948), In: B. Meltzer and D. Michie (eds.), *Machine Intelligence 5* Edinburgh, Edinburgh Univ. Press, 1969, pp. 3–23, Reprinted in: A. M. Turing, *Collected Works: Mechanical Intelligence*, Amsterdam etc., North-Holland, 1992.

49. A. M. Turing, *The chemical basis of morphogenesis*. Phil. Trans. Royal Soc. London, **237** (1952), 37–72, Reprinted in: A. M. Turing, *Collected Works: Morphogenesis*, Amsterdam etc., North-Holland, 1992.

50. A. M. Turing, *Collected Works: Mathematical Logic* Amsterdam etc., North-Holland, 2001

Samsara[†]

John N Crossley

Monash University
Clayton, Victoria, Australia

In order to answer the question of what logic may be like in this twenty-first century we examine the history of logic. One thing that we see is a recurrence of ideas but the ideas change form, sometimes quite dramatically. We also see a simple idea getting developed so much that it becomes very complicated, or at least the source of very complicated ideas – and then disappears, to re-emerge, perhaps, as a new idea.

Another aspect is the way the rôle of logic has changed. Logic was at one time the queen of the (mathematical) sciences. Now she is definitely not, but she has certainly become the handmaid of computer science, playing many roles. To assist our enquiry we address four questions.

[†] "The endless cycle of death and rebirth to which life in the material world is bound." (OED)

Mathematical Problems from Applied Logics. New Logics for the XXIst Century.
Edited by Dov M Gabbay *et al.*/ International Mathematical Series, Springer, 2005

1. What logics do we need?
2. What are logical systems and what should they be?
3. What is a proof? and briefly,
4. What foundations do we need?

We take an historical approach rather than a highly technical one.

1. Introduction

Mathematical logic[1] was unknown before about 1850 (but see the remarks on Leibniz below). It was very lively around 1900–1930 and, in the author's opinion, reached its zenith with the 1957 Cornell AMS Summer Institute on Symbolic Logic [**74**] and Paul Cohen's proof of the independence of the axiom of choice [**14, 15**] in 1963.

In the second half of the twentieth century, logic may no longer have been the queen of the (mathematical) sciences, but she has certainly become the handmaid of computer science, playing many roles. Logic has become essential in laying a theoretical foundation for computing.

1.1. The structure of the paper

We first give an example, in Section 2, of the way that a part of a discipline, that has been well studied, may go into decline but subsequently be resurrected by new, transforming ideas.

In Section 3 we consider the development of logic, particularly in the last hundred years. Then we go on, in the remaining parts of Section 3, to consider in some detail, how the move to intuitionist logic led to the extraction of programs from proofs. The author does not pretend that this is the most important part of logic. It is, of course, an area with which he is familiar. Very similar

[1] Here the phrase "Mathematical logic" refers to the pure discipline of mathematical logic.

remarks could be made about the development of another part of the author's work, that on Recursive Equivalence Types. There one can also see how the simple ideas of Dekker and Myhill [**25**] were developed dramatically by Anil Nerode [**65**] and subsequently extended to a whole set of different algebraic structures in [**20**]. Interest has now substantially waned in this area with only a very few people still publishing in the area according to Mathematical Reviews. The message we are trying to convey is that one should look at how logic has developed in the past in order to determine how it may develop in the future.

Near the end of this section, in Section 3.5, we see how the combination of techniques from computer science and logic can be used together to develop new (kinds of) logics. In Section 3.7 we compare the two standard ways of obtaining "correct" programs.[2]

In Section 4 we briefly consider some other logics, or parts of logic, that have yielded different kinds of systems in which to do mathematics. Next, in Section 5 we consider a few aspects of the notion of proof. Perhaps it is slightly ironical that we should end with foundations. However, the present author finds it hard to accept that the particular foundations of mathematics that we use would make much difference to the logic that we presently do. This is despite his having lived through a period when intuitionist logic was transformed from a rather odd and idiosyncratic logic to one which is fundamental to computer science.

Along the way we note a number of conclusions. These are not ends, but starting points for new discoveries (or inventions) or even of new modes of discovery and invention.

2. An Example of a Process

Let us first of all not consider logic, but linear algebra (as it is now called).

[2] We use the word "correct" in the sense that the program meets its specification.

In the nineteenth century the study of canonical forms of matrices became more and more complicated. Muir's book [**63**] is regarded as leading to the death of determinants. As Carl C. Cowen put it in [**17**]:

> [Kenneth O.] May [in the MAA film *Who Killed Determinants?* in the 1960s] documented how determinants flourished in the 19th century with its connections to the study of invariants and how the study of determinants developed into the linear algebra we know today, where determinants are far from central. Linear algebra did not really come to be recognized as a subject until the 1930s ... Historian Jean-Luc Dorier [**28**] regards Paul Halmos' book [**37**] *Finite Dimensional Vector Spaces*, first published in 1942, as the first book about linear algebra written for undergraduates.

The study of vector spaces did indeed transform the subject. As an undergraduate in the late 1950s, determinants and matrices were familiar to me, but Halmos's book opened up a new world. It transformed the subject from a very complicated, indeed arcane, one into one where what one wrote down was much simpler. With this dramatic increase in simplicity [3] – both typographical and conceptual – the subject became immensely more powerful. How could one have studied Hilbert spaces without vector spaces?

3. What Logics Do We Need?

Subsidiary to this question we shall also ask:

- How do we find/invent these logics?

In order to attack these questions let us ask

- How has logic developed? How will it develop?

[3] Perhaps it might be better to say "dramatic move to the basics," in the sense of the basic principles of vector space theory being more fundamental.

I believe I am following the ancient Greek philosopher Aristotle when I say that logic is the (correct) rearranging of facts to find the information that we want. Logic has two aspects: formal and informal. In a sense logic belongs to everyone although we often accuse others of being illogical. Informal logic exists whenever we have a language. In particular Indian logic has been known for a very long time (see, for example, [**62, 61**]).

Formal (often called, "mathematical") logic has its origins in ancient Greece in the West with Aristotle.[4] Mathematical logic has two sides: syntax and semantics. Syntax is how we say things; semantics is what we mean. Later I will suggest that we should perhaps add at least a third element to these two (see Section 3.6 below).

By looking at the way that we behave and the way the world behaves, Aristotle was able to elicit some basic laws. His style of categorizing logic led to the notion of the *syllogism*.

In the seventeenth century Leibniz began the axiomatization of sets (and also of real numbers, in a manuscript in the Niedersächsische Landesbibliothek, now the Gottfried Wilhelm Leibniz Bibliothek). The latter was complicated and became, as far as I can ascertain, essentially lost until the twentieth century.

In the nineteenth century Boole developed his laws of thought (see [**6**]). It was natural that he should refer to "thought" since, at that time, logic was in the domain of psychology. Then Frege developed his *Begriffsschrift* [**32**] whence came Russell and Whitehead's *Principia Mathematica* [**77**]. In tandem Dedekind's axioms for arithmetic (see [**24**]) were formalized by Peano (see, for example, [**66**]).

Frege developed a relatively small set of concepts and notations which were, apparently, [5] adequate to deal with the whole of logic – and mathematics. Russell – initially – built a seemingly simple

[4] There is an interesting, and different, account of this history in [**2**].

[5] Frege's concepts were adequate but he did not get all the axioms necessary for a complete system of first order logic.

system for dealing with all of mathematics and logic. But he wanted to reduce mathematics to logic. He did not succeed.

The system that Russell developed became more complicated because of

1) the axiom of infinity [6] and
2) the axiom of reducibility which surely must be a nonlogical axiom. [7]

Thus about the year 1900 there was just "one true logic": classical logic. But it is not clear: how do we check an infinite number of instances? What does it mean to say that there is no largest pair of twin primes, that is to say that there is an end to such pairs such as 5 and 7; 11 and 13 or even 202 289 and 202 291?

On the other hand, saying that there are infinitely many pairs of twin primes does have a clear meaning if we can show that, for every pair, there is a larger pair. In this context compare the way that Euclid IX.20 established that there are infinitely many prime numbers (although he did not phrase it like that) [**38**]. He gave a method for constructing a larger prime from a given (finite) set of prime numbers.

Because of the above style of questioning, led by Brouwer, a Dutch mathematician, indeed a topologist, "constructive logic" or "intuitionist logic" arose. [8]

For Brouwer, the problem of classical logic is that it is not evident that a mathematical proof actually gives you the way of performing the necessary construction. However, that is perhaps

[6] See also below Section 5.1.

[7] The axiom of reducibility was used by Russell to avoid the paradoxes of set theory. Its *ad hoc* nature prevented giving a good philosophical justification for it.

[8] Many people have regarded intuitionist logic as being restrictive because it eschews the use of the law of the excluded middle, $A \vee \neg A$, but this was not Brouwer's motivation. Further, intuitionist logic actually *includes* constructive logic in that there is a uniform translation (the "negative translation" whereby any formula, A, of classical logic can be translated into a formula, $A^{\neg}$, of intuitionistic logic) which is provable in intuitionist logic if, and only if, A is provable in classical logic. (See Gödel [**36**].)

the wrong way to look at it. Brouwer was concerned only with constructing mathematical objects that were claimed to exist. He did not like mathematical logic and did not consider it relevant. However, when his approach was formalized, as it was by Heyting in 1930 (see [**42**]), the details are buried inside the proof. Nowadays one might say that they are buried in the way that algorithms are buried inside computer programs.

The actual mechanism of providing proofs in the Russell and Whitehead system was cumbrous, although the techniques could be learned. For example, if one wants to prove the formula $(p \supset p)$ (read "p implies p") then a simple analysis of the axioms reveals that there are very limited possibilities for producing a proof and these are quickly exhausted. (Of course, ingenuity and practice make the task of finding such proofs easier, but Russell, in his autobiography [**72**], revealed that it had taken him a very great deal of time and effort to work out the formal proofs.)

So proving theorems in the way dictated by formal logic, that is to say, writing out strings of symbols and applying formal (mechanical) rules to obtain new formulae, became very tedious.

Here we already see Russell's system mimicking Aristotle: for the latter always had two premisses in his syllogisms. From these premisses he derived the conclusion.

What happened next? People began to say "It can be done," rather than going to the trouble of writing out the formal proof. This was nicely expressed, though not for our specific context, by Hoare [**43**, p. 578]:

> Once a powerful set of supplementary rules has been developed, a "formal proof" reduces to little more than an informal indication of how a formal proof could be constructed.

Indeed, the attitude among mathematicians, the present author included, was often that the approach, of simply showing that a proof existed, was accepted as sufficient. (But see below Section 5.1.)

However the way was already open (though it was not known immediately) for Gödel's incompleteness theorem (see [**35, 58**]). This theorem showed that the apparatus of formal first order logic was not sufficient to do all that mathematicians would wish to do.

The first major consequence of Gödel's incompleteness theorem is that Peano's axioms, when formalized in *first* order logic, are not sufficient to characterize the natural numbers.[9] But second order logic was unacceptable and has only now begun to resurface (see below Section 4.1).

Other methods for establishing that a proof exists were developed, based on semantics. Most striking amongst these are those arising from model theory (see [**3, 44, 9**]).

Thus the first concern of mathematical logic became the relation between syntax and semantics.[10] Syntax is how we say things; semantics is what we mean.

The ultimate result in the positive direction is Gödel's completeness theorem(see [**11**] or Gödel's original [**34**]) which was obtained long before model theory was invented (by Alfred Tarski).[11]

Nowadays the formulation of the completeness theorem involves models:

Theorem 1 (Gödel, Henkin). *A set of formulae* (*of first order logic*) *is consistent if, and only if, it has a model.*

So one needs the notion of model. Henkin's proof (see [**39**]) constructs such a model. This involves providing a (syntactic style) "witness" for the x whenever one wants a formula of the

[9] Of course, in second order logic, they are categorical, that is to say, there is only one model up to isomorphism.

[10] It could perhaps be argued that this had always been the first concern of logic.

[11] Tarski's great contribution was to formalize what seems obvious to most students – until they are asked to write it down formally: he gave a formal definition of "model" (see, for example, [**58**]).

form $\exists x A(x)$ to be true in a (formal) model that one is constructing.

Henkin originally came upon this idea of giving names to such witnesses when he was looking at set theory (see his [**41**]) and he also applied it to the theory of types [**40**], see below Section 4.1.

All of this had been predicated on the assumption that there was just one kind of logic: “one true logic.” The great success of the Frege/Russell initiative had been that it seemed to cover everything.

Brouwer’s abrupt departure from the classical world was ultimately to lead to many other kinds of logics. But there were other influences, coming from philosophy: modal logic has a very long history. The modalities of necessity and possibility also come from Aristotle and were introduced into formal logic by C. I. Lewis (see [**55**]) in 1932.

However it was not until the work of Saul Kripke [**51**] that connexions were established between models of modal logics and models of first order logic. Surprisingly, taking many Henkin models, with suitable relations between them, was sufficient to cover the modal case. They even covered the logic of Brouwer’s thought [**52**], although Brouwer did not approve of the formalization of his logic.

Nowadays there are many different logics that all have their value and application. These logics include various kinds of modal logics.

Many of these logics also have completeness theorems analogous to Theorem 1. The proofs of these have similarities with the proof of the completeness of intuitionist logic. Indeed, Kripke proved completeness results for modal logics first [**51**] and only subsequently used his ideas there to prove the completeness of intuitionist logic. Details of such theorems may be found in [**18**].

Recently category theory and its connexions with computer science have given a profitable way of generating useful modal systems through the connexion between coalgebras [12] and modal logic (see, for example, [**47, 53**]).

One original aim of Leibniz was to reduce argumentation to calculating with symbols, witness his famous remark: "Let us calculate" in *Ars Inveniendi* (*The Art of Discovery*) of 1685, reprinted in [**16**]. Many people would say that we are well on the way to that. An extensive discussion of this took place at the recent Royal Society Discussion Meeting on 18–19 October 2004 (see http://www.royalsoc.ac.uk/event.asp?id=1334). At this meeting some people seemed to feel that Leibniz's Golden Age had arrived, others that it was unattainable! We shall take up some of the issues below in Section 5.

However, there are certainly new styles of logic that have developed in the last fifty years. For example, description logics. As Nardi and Brachman put it in [**64**]:

> Description Logics in part arose from a need to respond to the inadequacy – the lack of a formal semantic basis – of early semantic networks and frame systems.

Description Logics (see [**1**]) may be regarded as an adaptation of (first order) logic to databases; databases themselves coming (on the logical side) from the notion of relational system or model.

I believe this is typical of the way that mathematical logic and its practitioners have responded to the needs of computing and computer science. It then follows that

Conclusion 1. *We should expect to see many more logics developing in the twenty-first century.*

3.1. Extracting constructions from proofs

What do we want from the logics we create? If we look at the work of Frege and Russell, I believe it is fair to say that they wished to

[12] Algebras are characterized by having a domain and functions on that domain. Taking the category-theoretic approach one reverses the arrows to get a costructure, in this case a coalgebra. In particular this means that coalgebras can be used to characterize the behavior of machines that change state. The arrows in the coalgebra are the state transitions.

clarify how mathematics worked or at least to make a fool-proof system for mathematics. For Aristotle, as I said above, logic is the (correct) rearranging of facts to find the information that we want. In present day computer science we are often trying to understand the logic of machines.

The first interpretation we shall consider is the logic of how computers can reliably get results. In this approach one imposes the logic first of all. One proves, for example, that for every x there is a y such that $A(x, y)$. Then one can extract a program from the proof, provided the proof is written in a suitable logic.

When Brouwer's proofs were formalized the information about the constructions he was giving became embedded in the proofs. Therefore, Brouwer's logic: intuitionist logic, is a suitable logic. We give the rules for this logic in Fig. 1. However, if we want to recover the construction we have to do some work.

It was Gentzen [**75**] in the 1940s who was the first to produce a formal system of logic where it was readily possible to see the information being moved around and, as it turned out, to make it possible to recover information on constructions.

3.2. The Lambda Calculus and the Curry–Howard correspondence

How do we extract the information from a proof in mathematical logic? Curry [**23**] started, and Bill Howard [**46**] developed, the basic idea which exploited Gentzen's achievements.

We use the lambda calculus. This was established by Church [**10**]. In ordinary mathematics if we apply the function $\lambda x.f$ to a then we get $f[a/x]$, which is read "f with a for x." In the lambda calculus however this is not the same as the (application) term $\lambda x.fa$, i.e., $\lambda x.f$ applied to a. In particular they are syntactically different. We therefore have to introduce the notion of

Assume that x, y are individual variables, and that t and t' are individual terms.

$$\frac{}{A \vdash A}\ (\text{Ass-I})$$

$$\frac{\Delta, A \vdash B}{\Delta \vdash (A \to B)}\ (\to\text{-I}) \qquad \frac{\Delta \vdash A \quad \Delta' \vdash (A \to B)}{\Delta, \Delta' \vdash B}\ (\to\text{-E})$$

$$\frac{\Delta \vdash A}{\Delta \vdash \forall x.A}\ (\forall\text{-I}) \qquad \frac{\Delta \vdash \forall x.A}{\Delta \vdash A[t/x]}\ (\forall\text{-E})$$

x is free in A, not free in Δ

$$\frac{\Delta \vdash A[t'/y]}{\Delta \vdash \exists y.A}\ (\exists\text{-I}) \qquad \frac{\Delta_1 \vdash \exists y.A \quad \Delta_2, A[x/y] \vdash C}{\Delta_1, \Delta_2 \vdash C}\ (\exists\text{-E})$$

where x is not free in C

$$\frac{\Delta \vdash A \quad \Delta' \vdash B}{\Delta, \Delta' \vdash (A \wedge B)}\ (\wedge\text{-I})$$

$$\frac{\Delta \vdash (A_1 \wedge A_2)}{\Delta \vdash A_1}\ (\wedge\text{-E}_1) \qquad \frac{\Delta \vdash (A_1 \wedge A_2)}{\Delta \vdash A_2}\ (\wedge\text{-E}_2)$$

$$\frac{\Delta \vdash A_1}{\Delta \vdash (A_1 \vee A_2)}\ (\vee\text{-I}_1) \qquad \frac{\Delta \vdash A_2}{\Delta \vdash (A_1 \vee A_2)}\ (\vee\text{-I}_2)$$

$$\frac{\Delta \vdash A \vee B \quad \Delta_1, A \vdash C \quad \Delta_2, B \vdash C}{\Delta_1, \Delta_2, \Delta \vdash C}\ (\vee\text{-E})$$

$$\frac{\Delta \vdash \bot}{\Delta \vdash A}\ (\bot\text{-E})$$

$A[a/x]$ is read "A with a for x" and denotes the formula A with a substituted for x.

FIGURE 1. The basic rules of intuitionistic logic

β-reduction[13]

$$\lambda x.fa \rhd f[a/x].$$

(Here $\rhd$ is read "reduces to.")

[13] α-Reduction refers to the simple renaming of one variable by another (without clashes).

Now note the similarities between →-introduction, the rule (→-I), and →-elimination (→-E) (in Fig. 1) on the one hand, and λ-introduction and λ-elimination on the other, the β-rule.

Next consider a proof of B from A from which we get a proof[14] of $(A \to B)$ (by the rule (→-I)):

$$\begin{array}{c} [A] \\ \vdots \\ \dfrac{B}{(A \to B)} \end{array} \tag{1}$$

and lambda abstraction (which abstracts a function from the process where $a \in A$ gives us $f(a) \in B$): that is $\lambda x.f$. Consider the figure:

$$\begin{array}{c} a \\ \vdots \\ \dfrac{f[a/x]}{\lambda x.f} \end{array}$$

What is the connexion?

The most obvious thing, I hope, is that the *shapes* are the same! If this is difficult to see then replace the bottom line by $x \to f(x)$.

Conclusion 2. *Patterns will arise in formal studies that reflect each other.*

3.3. Proofs as types

Russell, in his Appendix B to [**70**] introduced the theory of types in an informal setting, and later in a formal setting in [**71**]. The theory of types was invented by Russell (see [**77**]) to resolve the

[14] The square brackets indicate that A can be discharged, i.e., is not needed for the proof of B, though it is for the proof of B, of course.

difficulties causes by Russell's paradox.[15] The *typed* lambda calculus that we shall consider, that is to say lambda calculus with each term having a *type* assigned to it, can be regarded as the amalgam of two systems: logic, or more precisely, systems of predicate calculus, and the lambda calculus.

Originally types were built up from basic types by one simple operation. The original idea was that they formed a classification of sets. Sets at a "higher" type contained (in a sense), or reflected, sets of lower types. In Howard's system the types were identified with formulae of (propositional) logic.

A special kind of typed lambda calculus involves taking formulae of logic as the types. Now this is a strange idea to accept but it is easier to work with if one thinks of a type (i.e., formula) as the *set of proofs* of that formula. Instead, therefore, of variables, we use typed variables of the form $a : A$ where A is the type. Later one can simply treat the types as labels (see footnote 21 below).

The rule of *modus ponens* (→-I) of (1) then becomes:

$$\frac{a : A \; f : (A \to B)}{(fa) : B}. \tag{2}$$

If we had a proof of B from A then we would get an expression $\lambda x : A.f : B$ by the rule of (→-I) and this has type $(A \to B)$. If the f in the expression (2) is actually of the form $(\lambda x : A.g : B) : (A \to B)$, then we get

$$\frac{a : A \; (\lambda x : a.f : B) : (A \to B)}{((\lambda x : A.f : B) : (A \to B))a : A) : B}$$

which is somewhat hard to read. However the bottom line has the formula B as its type, and the expression reduces to

$$f[a/x] : B[a : A/x : A] \tag{3}$$

[15] The set of sets which are not members of themselves yields a contradiction.

where the substitution of $a : A$ for $x : A$ takes place throughout the term $f : B$.

If we translate this back into proofs it means that the corresponding proofs look as follows. On the one hand we have the complicated proof:

$$\dfrac{\begin{array}{c}\vdots\\ A\end{array} \qquad \dfrac{\begin{array}{c}\underline{[A]}\\ \vdots\\ B\end{array}}{(A \rightarrow B)}}{B} \tag{4}$$

and on the other hand, by putting the proof of A from the left on top of the proof of B (from the hypothesis A), and not introducing the $\rightarrow$, we no longer need the hypothesis $[A]$ in the proof on the right in order to get a proof of B.

That is to say, we reduce the proof in (4) to a simple proof of B of the form

$$\begin{array}{c}\vdots\\ A\\ \vdots\\ B\end{array}$$

This corresponds in the lambda calculus to the reduction[16] that resulted in (3). So we have a direct correspondence between proofs and terms of our typed lambda calculus. This is called the *Curry–Howard correspondence.*[17]

[16] This process of reduction is also called *cut elimination.*

[17] Some people use the term isomorphism but there are technical difficulties involved in making the correspondence one to one, so I prefer the weaker terminology.

3.4. Strong normalization and program extraction

Now it is obvious that a long and complicated formal proof has an even longer typed lambda calculus expression associated with it. If, however, all the possible reductions are carried out it may become considerably simpler. Indeed, in the cases with which we are concerned we can usually omit all the types. (They will have served their purpose of ensuring that we get a result of the correct type when the proof is complete. This is related to the use of types in computer programming languages.)

The maximum benefit is when we have a *Strong Normalization Theorem* for the system. Such a theorem says that, whatever the order of the reductions – and there may be many possible different reductions for a long lambda term – the process always stops. (One reason the process might be expected not to stop is clear when you look at substituting $x + x$ for x: the number of xs goes up at each substitution and the expression gets longer!)

The Curry–Howard correspondence can be extended to the other logical connectives by modifying the lambda calculus. Surprisingly, in addition to the above operations involving lambdas, we only need the formation of ordered pairs and the projections onto the first and second elements of those pairs in order to capture all first order logic.[18] We give only a few examples; the full details can be found in [**22**]. The Curry–Howard term for a conjunction $(A_1 \wedge A_2)$ obtained by the rule of $\wedge$-introduction is the ordered pair $(p : A_1, q : A_2)$ of type $(A_1 \wedge A_2)$ where $p : A_1$ is the Curry–Howard term for the proof of A_1, and similarly $q : A_2$ is the Curry–Howard term for the proof of A_2. Conversely we use the projections fst and snd for the rules $(\wedge\text{-}E_1)$ and $(\wedge\text{-}E_2)$. For the rule ($\exists$-I) we get the term $(t', p : A[t'/y])$ where the premise has the Curry–Howard term $p : A[t'/y]$. Thus the Curry–Howard term contains the term t' that had earlier been proved to exist.

[18] The process can also be extended to higher order logic.

The major consequence of the Strong Normalization theorem is then that, if we prove a formula of the form $\exists x A(x)$, we can actually extract, from the normalized proof (i.e., the lambda, or Curry–Howard, term in which no more reductions are possible), an x such that $A(x)$. Further, if we can prove $\forall x \exists y A(x, y)$ then we can actually get a program such that, given an x, it will compute a corresponding y. Moreover, we have a proof of $A(x, y)$ for this x and y so the program is "correct" in the sense that it meets its specification.[19]

Curry–Howard terms are, in general, a generalization of the idea known variously as formulae-as-types or, better, as proofs-as-types: the terms code up a whole proof by successively encoding the applications of the logical rules in a proof.

Not surprisingly, not all rules of logic allow us to prove a strong normalization theorem. One major obstacle is the law of double negation: From $\neg\neg A$ infer A. If we had a rule that would allow us to prove $\exists x A(x)$ from $\neg\neg\exists x A(x)$, how do we obtain such an x? There is no clear way. So we generally restrict ourselves to constructive logic and all is well.

Changing to other systems, for example, arithmetic, may bring in other axioms. Here the most dramatic is the rule of induction. Fortunately the induction axiom

$$\frac{A(0) \quad \forall x (A(x) \to A(x+1))}{\forall x A(x)}$$

gives rise to a reduction *exactly* corresponding to the recursion

$$f(\overline{a}, 0) = g(\overline{a}), \tag{5}$$

$$f(\overline{a}, x+1) = h(\overline{a}, x, f(\overline{a}, x)). \tag{6}$$

[19] Intuitively speaking, the specification is the statement about the result of the program. See also below Section 3.5.1.

Happily we can prove a strong normalization theorem for arithmetic (see [**22**]). We can therefore extract programs from these proofs.

Conclusion 3. *Analogies and similarities, especially geometric similarities, can lead to new discoveries and unexpected parallels.*[20]

3.5. Beyond traditional logic in program extraction

3.5.1. *Algebraic specifications*

We now turn to an application of the above ideas to software engineering. Producing programs that satisfy their specifications is a primary goal of software engineering. We start with an algebraic specification and then construct a program. What is an algebraic specification? It is a description in formal logic of a structure, for example, the natural numbers.

As an example we use the *Common Algebraic Specification Language* (*CASL*, see [**13**]) but the technique could be employed in other specification languages, indeed originally we ourselves used a different language.

Structured specifications in *CASL* are built from *basic* (or *flat*) specifications by means of *translation* (or *renaming*), written **with**, taking *unions* of specifications, written **and**, *hiding* signatures, written **hide** and the *extension of specifications*, written **then**. A typical example of a flat specification, this one is for natural numbers, is given in Fig. 2.

When we change a specification, then what is true changes – even if simply because we use new names, for example, "car" instead of"auto," "boot" instead of "trunk," etc., but we may also

[20] On other occasions they will be at best suggestive, but possibly even misleading, as analogies have been. See [**45**] for examples of the overuse of analogy.

add new predicates (relations). We have developed logical systems to reflect the interaction between such changes and the logic statements.

$$
\begin{array}{ll}
\textbf{spec}\ \mathrm{Nat} = & \\
\textbf{sorts} & \\
\quad Nat & \\
\textbf{ops}\ 0 : Nat;\ s : Nat\ \rightarrow\ Nat;\ + : Nat \times Nat \rightarrow Nat & \\
\textbf{preds} & \\
\quad \geqslant: Nat \times Nat & \\
\textbf{axioms}\quad \forall x : Nat \bullet x + 0 = x & \%(Nat_1)\% \\
\quad \forall x; y : Nat \bullet x + s(y) = s(x + y) & \%(Nat_2)\% \\
\quad \forall x : Nat \bullet x \geqslant 0 & \%(Nat_3)\% \\
\quad \forall x; y : Nat \bullet x + y = y + x & \%(Nat_4)\% \\
\quad \forall x : Nat \bullet s(x) \geqslant x & \%(Nat_5)\% \\
\quad \forall x; y; v; w : Nat \bullet x \geqslant v \wedge y \geqslant w \rightarrow x + y \geqslant v + w & \%(Nat_6)\% \\
\textbf{end} &
\end{array}
$$

FIGURE 2. The specification Nat

Originally Martin Wirsing studied a logical calculus for structured specifications (see [**78**]). This was subsequently extended by Wirsing and his student Peterreins. The system at that stage was quite complicated. However, by reflecting on the way that logical rules are developed Wirsing and the present author were able to reformulate the rules in a way that looked almost traditional in [**79**]. Next Wirsing and the present author extended the idea to algebraic specifications, and then we went further with Iman Poernomo, to include even the parametrized specifications of the language *CASL*.

Abstractly speaking we have an annotated or labelled deductive system.[21] The basic form of a rule in such a logic can be written in the form

[21] The logical system that we then have is therefore related to the labelled deduction systems of Gabbay [**33**].

$$\frac{p : A \qquad q : B}{s(p, q) : \sigma(A, B)}$$

It is convenient to use "contexts" also. That is to say, the actual hypotheses with which we are working. These will be written in the standard logical style using the "turnstile" symbol $\vdash$. Thus one writes $\Gamma \vdash A$ to indicate that A is provable in the context Γ (or equivalently, from the hypotheses Γ).

The annotations we use also involve Curry–Howard terms, specification names and the logical connectives. We have two kinds of rules: those for the logical connectives, *logical rules*; and those for the structural changes in the specifications, *structural rules.* Even with the purely logical rules, the specification of the conclusion depends on those in the premises. For the structural rules, the change in the structure is reflected in the specification of the conclusion.

The logical rules for our system *Structured Specification Logic* are very similar to the standard rules of intuitionist logic. The complete set of rules, including the structural rules, that we have for *CASL*, with their Curry–Howard terms, may be found in [**21**] or [**67**].

When we wish to extract programs from proofs which are derived from algebraic specifications the Curry–Howard terms that we use are now more complicated for two reasons. In addition to the information from, for example, the logical rule being used, the Curry–Howard term also has to "remember" the specification. We have a similar situation for the structural rules. However, the message is as before: the Curry–Howard term carries all the information as to how we have constructed the proof so far.

In this situation we are again able to prove strong normalization. From this strong normalization theorem we are then able to give an *extraction map*, that is to say, we give a formal process which, given a Curry–Howard term for a proof of $\forall x \exists y A(x, y)$ from a given specification, the extraction map returns a suitable y for a given x. Indeed it gives a program in the programming language *Standard ML*. The extraction map works recursively and, in particular, the cases for →-introduction and elimination correspond directly to the procedures we have outlined above.

3.5.2. *Imperative programming*

My recent PhD student, Iman Poernomo, has developed a protocol for integrating ordinary computer programs into the kind of deductive system we have been discussing. This protocol he calls the *Curry–Howard* protocol. The logical system for such a situation includes the state of the system (i.e., the contents of registers in the machine) and accounts for the changes that take place when a program is run. Despite the complications this produces it is still possible to produce a constructive version of a Hoare logic (cf. [**43**]), for reasoning about imperative programs, to which the Curry–Howard isomorphism may be adapted.

However we are also concerned to use programs already in the programming language that we regard as "reliable." We do not use the word "correct" here, reserving that word for programs that have been formally proved to meet their specifications. Here we simply mean that we have programs that we are satisfied will give the correct answers. Such programs include very simple ones such as programs for the multiplication of natural numbers. This achieves a significant saving in the length of the programs extracted. Otherwise we would have to prove a formula in formal arithmetic that allows us to extract a program, for example, for the multiplication function. The proof would be inordinately long, involving several applications of induction and its corresponding program would then involve the same number of recursions. This is obviously very uneconomical because we know it is possible to write a relatively simple program for multiplication (if one is not built into the computer already).

Imperative computer programs have *side-effects*: they change the *state* of the machine and, in particular, the values in various registers. The presence of side-effects is a principal feature that distinguishes the imperative programming paradigm from the functional one. However, side-effect-free functions are also important in imperative programs because they enable access to data, obtaining views of state and producing return values. Imperative programs involve both side-effects and side-effect-free return values. Consider, for instance, a program that triples the number in the register s and returns a value that is twice the value in s. In *Standard ML* the program is

$$\mathsf{s} := !\,\mathsf{s} * 3;\ !\,\mathsf{s} * 2.$$

It has a side-effect producing assignment statement, $\mathsf{s} := !\,\mathsf{s} * 3$, followed by the return value $!\ \mathsf{s} * 2$. In many popular imperative languages such as *Standard ML* (or *LISP*) such return values are potentially complex, involving higher order functional aspects that are difficult to program correctly.

Our goal is to specify, reason about and synthesize both aspects of imperative programs – side-effects and functional return values. Our approach is as follows. We use a version of Hoare logic to synthesize the side-effect producing aspect of a program, specified in terms of pre- and post-conditions. Hoare logic [**43**] involves considering triples of the form

$$\{\textit{pre-condition}\}\mathsf{program\ step}\{\textit{post-condition}\}$$

The *pre-condition* is true before the program step commences and the *post-condition* is true after the step.

The formula

$$s_f > s_i$$

specifies a side-effect where the final value of state s, denoted by s_f, is greater than the initial value, denoted by s_i. We can use Hoare logic to synthesize a *Standard ML* program that satisfies this specification, by producing, for example, a theorem of the form

$$\vdash \mathsf{s} := !\,\mathsf{s} * 3 \bullet s_f > s_i$$

where the left-hand-side of the • symbol is the required *Standard ML* program (written in teletype font), and the right-hand-side is a true statement about the program.

To specify and synthesize return values of a program we adapt realizability and the extraction of programs from proofs. We have already treated the latter, so now we consider realizability.

When we extract a program we wish to demonstrate that it is "correct." This requires the notion of *realizing.* This is a different way of verifying proofs in intuitionistic logic by means of computable functions. It was first developed by Kleene. (See the last chapter of [**49**].) The basic idea is that we produce a program for a (partial) recursive function that is a *witness* to the proof of an assertion. Such witnesses can be produced recursively by going down through the proof. Such a program can be regarded as a number (for example, the binary string that encodes the program). For example, if we have partial recursive functions with programs p, q realizing A, B, respectively, then we take (p, q) as the realizer of $(A \wedge B)$. The full details may be found in Kleene [**49**] for the basic system of intuitionist logic and in our book [**67**] for the systems we discuss here.

Here is an example. Given the theorem

$$\mathsf{s} := \mathsf{s} * 3 \bullet s_f > s_i \wedge (\exists x : \mathit{int}.\mathit{Even}(x) \wedge x > s_i)$$

we can synthesize a program of the form

$$\mathsf{s} := \mathsf{s} * 3; f$$

where the function f is a side-effect-free function (such as $!\,\mathsf{s}{*}2$) that realizes the existential statement of the post-condition $(\exists x : \mathit{int}.\mathit{Even}(x) \wedge x > s_i)$, by providing a witness for the x.

When using our program extraction, users will have no need to manually code the return value, instead they can work within the

Hoare logic. There they prove a theorem from which the return value is then synthesized.

Conclusion 4. *Techniques developed in one part of a discipline may be applicable in another. One needs to use one's eyes, and one's ingenuity.*

3.6. Proofs from programs[22]

So far we have seen how to obtain programs from proofs in constructive systems of logic. Therefore we could conclude that all proofs are already programs, or at least, that every proof in (constructive) logic contains a program.[23]

What if we were to write the program first? Would we automatically have a proof? The answer is obviously "No!" if we simply write computer programs as many people do. However, a thoughtful computer programmer would wish to know that the program written would do what it was expected to do, that is to say, would meet its specification.[24] Therefore, as part of the task of writing the program, a proof should be produced at the same time.

The approach that we have presented shows how to accomplish both of these tasks at the same time. It does not require a separate investigation to produce a proof that the program will be correct.

From a practical point of view it is sometimes obvious how to write the proof. I studied a program for quicksort.[25] Then I wrote a proof corresponding to the program and extracted a program from it. The resulting program was essentially the quicksort program from which I had started. However I have not yet been able

[22] Alternatively this subsection might be labelled **Changing direction**.

[23] The restriction to constructive systems of logic is essential for us.

[24] This is a very serious issue when it comes to the control of powerful systems, in particular, the control of nuclear weapons.

[25] This was inspired by looking at work of Helmut Schwichtenberg on program extraction in [**4**].

to formalize the procedure that I used in producing the proof from the program. It would appear that one needs to know the *algorithm*, rather than the program, in order to construct the proof. This in itself indicates that one also needs to know that the program is a *correct* implementation of the algorithm. This is indeed work for the future.

In addition, perhaps we ought to add to syntax and semantics, as major concerns of logic, implementation. Consider the process of writing a computer program, even when a formal specification is given.

We have to work out the algorithm that will fulfil the specification. Then we have to implement that algorithm in a computer language.[26] So if one is going to use computer proofs, then besides the syntax of our formal language, perhaps we ought also to consider the implementation of our logic in the machine and the effects that will have.

3.7. Programs then proofs

The second interpretation of the statement at the beginning of Section 3.1 revolves around the behavior of computing machines. In this case, instead of starting with a proof in logic we start with a computer program.

We consider a specification. This may be an informal one but it seems inevitable that at some stage it will have been turned into a formal one. The specification is then built up into a program and, accompanying that, or subsequent to it, a verification is built.

We may contrast the two methods: the first of extracting programs from proofs, and the second of providing a proof for (or in the process of writing) a program, that is to say, verifying a program, in the following table.

[26] And some may add, for a particular implementation. On this problem Hoare's original paper [**43**] is still very valuable.

Extraction	Verification
Specification	Specification
Proof	Program
Program extraction	Program verification

This second is the method invented by Tony Hoare [**43**] mentioned above in Section 3.5.2. The problem with this method is that it gives a necessary condition for correctness and not a sufficient one. Nevertheless the method is widely used and very valuable.

Conclusion 5. *A logical method is only clearly useful if it is used.*

4. What Are Logical Systems and What Should They Be?[27]

We started out with the simple systems of Aristotle. Now we have progressed to systems where the form of the rules is (essentially) the same: two premisses and a conclusion. However, there is a great deal more baggage accompanying the traditional logic. There are labels which may represent a specification of a system, or a state of a machine.

Thus the techniques we have presented here are based on a variant of Gabbay's labelled deductive systems [**33**]. Our logical rules are of the form

$$\frac{\text{Logical context, State, Curry–Howard term} \ \vdash \ \text{Formula}}{\text{New Logical context, New State, New Curry–Howard term} \ \vdash \ \text{New Formula}}$$

although the actual order may vary. Further, each of the items on the lower line may depend on, that is to say, be functions of, any

[27] This section heading is inspired by Dedekind [**24**].

or all of those on the top line, and of course there may be two or more sequences on the top line.

The semantics of these rules will depend on the structures that we are using. Also the interpretation of the informal terms: Logical context, State, etc. will also vary.

What seems to be most important is that we have extended the notion of logic in two ways. First of all we now have programs or other constructions (for example, specifications) interacting with the standard logical connectives. Secondly, the context of the logic may change in the course of a proof. This certainly happens in the context of algebraic specifications. In the simplest case we may just be changing the language, say, from English to American. Thirdly, we are now discussing logics (plural) and we arrive at such a logic by an analysis of a technical setting. This seems to me to be following Aristotle's approach of looking at the real world, or a small part of it, and then abstracting the logical principles that work in that arena. But we have come a long way from the "one true logic" that I mentioned in Section 3, near the beginning!

But this has only been in a very limited number of contexts, in particular, algebraic specifications and imperative programming. The idea of building proofs and programs together has surely much more potential.

Conclusion 6. *The rules that we have in traditional logic represent the rules of rational thought. There is no reason why we should not look at the logic of other procedures or constructions. Logics can reflect the logic of systems other than human thought.*

4.1. Higher order logic

Let us now turn to more extensions of logic. It should not be surprising that the logician Dana Scott should have taken up the question of the semantics of computer programs and programming languages when he came in contact with Christopher Strachey in the 1960s. (See [**73**].) However, in this context it is very startling that the (untyped) lambda calculus, which sits at the base of the Scott–Strachey semantics, should not admit set-theoretic models (see [**68**]).

The models require equating (in some sense) a set A with the set of all functions from A to A, that is, $A \to A$ or A^A which is immediately an uncountable set if A is infinite.

An equally disturbing situation arose when Henkin was establishing his completeness proofs for the theory of types [**40**]. In this theory (even as first devised by Russell [**70, 71**]), types are built up from basic types by means of juxtaposition. Given two types σ and τ one forms the type $(\sigma\tau)$ which can be regarded as the collection of all functions from type σ to type τ. In his thesis, Leon Henkin produced two completeness proofs (see [**39, 40**]): the one for first order logic referred to in Theorem 1 above and one for the theory of types. The models Henkin constructed in his proof for first order logic gave names to all the elements of the model. In the theory of types (and the same applies to higher order logic) there are uncountably many objects and therefore one cannot give names to all of them (from a countable alphabet/language). Thus there arose models that were not standard.[28] In such models, the set of all subsets of a given set, for example, was modelled by a set which did not have the "correct" cardinality: its members were only ones that had names.

Perhaps because of this, second order logic (and other higher order logic) fell into disuse. It has recently been resurrected. This is partly because of a renewed interest in the theory of types and its applications in computer science (for example, for type-checking) and partly because it is possible to simulate second (and higher) order objects by using different sorts of first order ones. Thus one may distinguish points and lines in a graph by having predicates $P(x)$ for "x is a point" and $L(x)$ for "x is a line" when dealing

[28] These should be distinguished from the nonstandard models of Abraham Robinson [**69**].

with graph theory. To ensure these work together appropriately one requires extra axioms. An illustration may be found in [**48**]. Feferman [**30**] also notes this point.

Conclusion 7. *We use an alphabet constructed from a finite number of symbols to talk about infinite things. Countability is not an issue for the analysts, why should it be one for logicians?*

Another drive for looking at higher order logic has come from logic programming. Originally logic programming(see, for example, [**12**]) essentially dealt with only a quantifier-free fragment of first order logic. Over recent years, however, it has been developed into higher order logic and has embraced the lambda calculus. The work of Dale Miller has been central in this. (See [**60**] and more recently [**59**] and the links there.) As Miller says: "This programming language incorporates large amounts of logic." (Downloaded from http:// www.lix.polytechnique.fr/Labo/Dale.Miller/lProlog/cse 360/syllabus.html)

Conclusion 8. *Logic can be used not only to analyze but also to synthesize.*

4.2. A note on set theory

There was a dichotomy between set theory and logic existing from around 1900. Set theory had entirely different origins from logic, although the elucidations of foundations[29] did invigorate both. Cantor [**8**] did not reach his paradise from logic but from analysis. Problems abut Fourier series gave rise to sequences of operations longer than ω, the first infinite ordinal.

In the middle of last century logic reached a zenith, in the author's opinion, with Paul Cohen's work on the independence of

[29] Whatever "foundations" may mean!, cf. Feferman's book [**30**] and Bostock's logistical [**7**].

the axiom of choice [**14, 15**]. Since that time the area of set theory has become more and more complicated.[30] The latest moves in this area, however, seem to be of a different nature. Woodin's recent work on the Continuum Hypothesis [**80**], with his new idea of Strong Logics takes us to a different approach to set theory. Whether it will lead to a renaissance by simplifying the area remains to be seen.

4.3. Computation and proof

Algorithms are now today's lifeblood as functions were in the nineteenth and early twentieth centuries. Algorithms make us think of computations but, as we know to our cost as computer users, algorithms, implemented in the software that we use everyday, do not always produce the answer or behavior that they should.[31]

Along with means of computation one also needs means of proof and we have shown two approaches to supplying such proofs above. So computation and proof should be developed together. Otherwise, how does anyone see the need for a proof that the computation actually works?

5. The Nature of Proof

There is also the question of why proofs are needed in mathematics in general. This is sometimes harder to see.

Frank Harary gave a very nice exposition on this topic many years ago in Malaysia.[32] I heard it, and I still remember and heed

[30] The same can be said of parts of model theory, thanks to the spectacular work of Shelah.

[31] This is because there are two kinds of problems here. The first is the obvious one of human error. The second is that the specification for the computation, while being correct as a specification, may not be a specification for what was really desired to be computed.

[32] This was at a seminar at the University of Malaya associated with the first bi-annual meeting of the South-East Asian Mathematical Society in 1974.

his instructions. The essence was as follows. Suppose you want to present to an audience a theorem that A happens under the hypothesis H.

1. Give examples where A occurs.
2. Give counterexamples where it does not occur.
3. Prove the theorem under the hypothesis H.
4. Give examples where the hypothesis H does not hold, and A does not occur.

(The last item may be strengthened. Harary was at pains to give theorems where the hypothesis was as weak as possible.)

Such a procedure also leads to better understanding, and surely understanding should accompany doing (in particular, calculating or computing).

Proofs give understanding, or at least they should; computations give results. But what is a proof?

In the preceding sections we have restricted ourselves almost entirely to proofs in formal logical systems, though we have seen how these systems have been developed far beyond the original ones of, say, Frege.

We have also noted that mathematicians have often replaced giving a proof by instead showing that a proof exists.

5.1. The question of scale and the rôle of technology

At this point in our history we can use mechanical or, more frequently, electronic devices to perform repetitious tasks or to shortcut them. This has been the case since slide rules were introduced – some might even say, since the abacus was introduced. What is different now is that a multitude of tasks, identical or different, can be performed at, so to say, the touch of a button and very quickly. This change in quantity brings with it a change in quality.[33]

It is easy to deal well with single computations such as finding an nth root. This is because we can give a clear and explicit process for finding such a root and a proof that it is correct. On the other hand we do not seem to be able to deal as well or as adequately with long sequences of computations or of proofs. The very complexity of some computations is too much for us to grasp. Further mechanical aids may give wrong results. For example, try taking the square root of a number over and over again on (different) hand-held calculators. Usually twenty or so times will suffice to illustrate the problem.

We do not seem to be able to handle such long sequences in a way that is satisfactory enough from a formal point of view (cf. Devlin's article [**26**]).[34]

Let us also be thoughtful about when it is appropriate to use computers or calculators. We use them when we do repetitious work. When we need to do a calculation a thousand times, or even weekly or daily, it is foolish to use pen and paper: a computer or calculator is more reliable and less stressful. This is an issue involving scale, here meaning the number of times we repeat a calculation.

Note. I have not touched on the subject of complexity although I believe that it will become very much involved, especially in the context of computer proofs (see above Section 5.1). This is partly because I find the notion of complexity ill defined. Recent work on parametrized (or as they call it "parameterized") complexity by Downey and Fellows, see, for example, [**29**], has helped to make some improvement so that one can get a more realistic, that is to

[33] Compare the music of Philip Glass where repetition gives a different, distinctive, quality to his music.

[34] The way that we (attempt to) cope with this is to take a more macroscopic view and to look at the structure of the computation in a more coarse-grained way. Then we may be able to see our way through the various levels of the computation.

say, useful, approach to the concept. One of the great virtues of mathematics is that, since mathematics gives us the logic of the world, we can use it for any scale of activity. Here “scale” does not only refer to physical size. It also includes complexity. Thus in training people to use and understand mathematics we should show them how we can shift from macrocosm to microcosm or anywhere in between and still be able to use our mathematical techniques.[35] Likewise we can look at processes and at pieces of software in the same mathematical way.

To what extent can we formalize this? To what extent should we formalize proofs? One overriding advantage of formal logic as practised by Russell was that the proofs obtained were tangible and could be mechanically checked. However they differ remarkably even from proofs in the mid-twentieth century. Consider, for example, a standard modern algebra text [**5**]. The treatment starts off with integral domains (very much modelled on the natural numbers and then the integers). Induction is introduced but within a few pages the idea of induction has gone from its use *within* the domain to induction about the domain. To be precise: induction is used to prove, for example, the distributive law

$$x * (y + z) = x * y + x * z$$

yet a few pages later it is used to prove the general associative law

$$(a_1 * a_2 * \cdots * a_{n-1}) * a_n = a_1 * (a_2 * \cdots * a_n)$$

In this setting we have a proof in the metalanguage. This is not remarked on by the authors but is very striking to a logician. This is by no means an isolated example. So one becomes accustomed to “layering” as I have referred to it. (See my paper [**19**].) However, apart from acknowledging that one has moved to a metalanguage,

[35] Of course there are some physical situations, for example in atomic physics, where the scale means we have to use *different* mathematics, but we still use mathematics.

there seems to be a dearth of formal systems allowing one to make such moves.

Creating such a system would allow the contracting of proofs in a very formal way. Nevertheless, the problems of sheer size (as noted in Barendregt's [**2**]) have to be dealt with in some way and, as Alan Robinson points out (quoted by Donald MacKenzie in his paper at the same Royal Society Discussion Meeting on 18–19 October 2004, see [**57**])

> You've got to prove the theorem-proving correct. You're in a regression, aren't you?

That is to say, if we have a mechanical means of proving a theorem, we then need a proof that this prover is correct. If we have a mechanical way of proving that, then we again need a proof of correctness for this latest device, and so on. So the question of a final authority emerges.

Despite Lewis Carroll's salutary paper [**27**] of 1895, around 1900 there was a general faith that there was such a final authority and people were at pains to provide one. Russell's system of logic was intended to do that, and as we have noted, it failed. Nowadays I believe it would be foolish even to search for such a final authority. Nevertheless there is a remarkable robustness about mathematical proofs as John Shepherdson point out to me many years ago.[36] And this is despite the assaults of Imre Lakatos on the notion of proof and his beautiful illustration of the evolution of a specific proof and the mathematical definitions surrounding it in his [**54**]. Frank Ramsey's dictum,[37] presumably from the 1920s, remains a serious question:

> Suppose a contradiction were to be found in the axioms of set theory. Do you seriously believe that a bridge would fall down?

[36] In private conversation, Melbourne 1992.

[37] Quoted in [**56**].

We shall not pursue the nature of mechanical proof further here but do commend the papers at the Royal Society Discussion Meeting on 18–19 October 2004 which we hope will all eventually appear in print. We simply draw attention to other kinds of proof.

In a footnote on p. 101 in his [**50**], Kreisel pointed out that if we had, and accepted, a classical proof that there were indeed infinitely many twin primes, then we should immediately have an algorithm: just test pairs of primes bigger than the given pair until we find the next pair!

The question then arises: what is the appropriate logic?

If one takes intuitionism seriously, as did Brouwer, and espouses its philosophy, then surely only proofs constructed in an intuitionist fashion are acceptable. In particular, a proof that a theorem can be proved intuitionistically should be an intuitionistic proof. For many of us this is too much to ask. We would rather accept the approach of Kreisel's footnote. Obviously one can repeat this argument at any level.

Finally just in case the notion of proof seems purely logical, consider what I regard as the most astounding proof in all the mathematics I have ever seen or even heard of. Near the end of Dedekind's [**24**] there is a (claimed) proof that infinite sets exist.[38] It begins: "Consider the realm of my thoughts ..." It then goes on to consider not just thoughts but thoughts of thoughts. In this way an infinite set is constructed. How psychological and unmathematical this is! It is so far away from the studious way that Dedekind has built up the characterization of the natural numbers. The proof is not very well known. It should be better known to serve as an object lesson.

Conclusion 9. *We should consider not only the logic in which we do a* (*logical*) *proof, but also the logic of the system in which we do the proof.*

[38] The assertion of the theorem should be compared with Russell's axiom of infinity.

5.2. Foundations

Which of these logics, or perhaps, rather, combinations of logics, we should use, seems quite problematic.

It also brings up the question of foundations. The, to the author's mind, unsatisfactory debates of the twentieth century on logicism, formalism and intuitionism seem to have left out the human dimension and the fact that Mathematics (and with it Logic) is a human activity. Work such as MacKenzie's [**57**] must surely be taken into account.

Added to this we shall also need to consider the time that it takes to get a proof. Even if we are producing a mechanical proof the question will ultimately arise as to deciding at what stage the computation is satisfactory, cf. Alan Robinson's remark above.

Finally, there is the question of how deeply we can, or should, examine what a logical system is. The work of Yesenin-Volpin, though much neglected, has been mentioned recently in my hearing. There are at least two aspects worthy of consideration. The first is whether the (logical) universe is finite, and in particular, whether there is only a finite number of natural numbers. This is documented in [**81**]. The second item is his analysis, not recorded in that work but presented in lectures at SUNY, Buffalo about 1972, which I heard. This concerns his detailed analysis of even the symbols used in the logical expressions and their repetitions: – if they are indeed repetitions; in what sense are the symbols "the same." At the present time this seems too far-fetched to warrant further investigation, but what has seemed obvious in earlier centuries has sometimes later turned out to be quite problematical. The most outstanding example of this is perhaps that of infinitesimals which were happily used by Leibniz, then came into disrepute in the nineteenth century, and finally were resurrected and justified by Abraham Robinson in the 1960s, see [**69**]. However, I do not believe that Abraham Robinson's infinite-simals are in any way the same as those of centuries earlier. There is moreover a question as to whether Robinson actually defined a unique set of infinite-simals (even in the context of the real numbers).[39]

In a different setting, while the proof systems look very similar (one only has to give up the law of the excluded middle to go from classical to intuitionist logic) nevertheless the actual working of the systems of intuitionist logic and classical logic are dramatically different. In this case as is often the case:

Conclusion 10. *We often need to start again each time we revisit an idea. There is no guarantee that the previously used techniques will work.*

6. Final Remarks

Then conclusions that we have drawn encompass a number of different aspects of the development of logic. First there is the variety of logics that now exist and can be expected to proliferate (Conclusion 1). Then there is the “transfer of technology”: the use of the same pattern, same idea, or an analogous one in another context (Conclusions 2, 3, 4). Prejudices should be broken down: we should look beyond our own narrow horizons (Conclusions 6, 7, 9). I believe we also have a duty to our fellows: a logical method is only clearly useful if it is used (Conclusion 5). We should think of our fellows in developing logic. Then there is the fact that logic can be used not only to analyze but also to synthesize (Conclusion 8). But each time we may need to start again, almost from scratch at times, but informed by the past (Conclusion 10).

So our conclusions are not endings, but new beginnings.

Samsara.

[39] He gives a procedure for getting a model of the real numbers, including infinite-simals, but any one of a range of models will give the required results.

References

1. F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider (eds.), *The Description Logic Handbook: Theory, Implementation, and Applications,* Cambridge, Cambridge Univ. Press, 2003.
2. H. P. (Henk) Barendregt, *The challenge of computer mathematics*, Submitted to the Royal Society, http://www. cs.ru.nl/ ˜ freek/notes/RSpaper.pdf (accessed 8.03.05).
3. J. L. Bell and A. B. Slomson, *Models and Ultraproducts: an Introduction*, Amsterdam, North-Holland, 1969.
4. U. Berger and H. Schwichtenberg, *Program extraction from classical proofs*, In: D. Leivant (ed.), *Logic and Computational Complexity, International Workshop LCC 94,* Indiapolis, IN, USA, October 1994, p. 77–97.
5. G. Birkhoff and S. Maclane, *A Survey of Modern Algebra*, New York, Macmillan, 1941.
6. G. Boole, *An Investigation of the Laws of Thought on which Are Founded the Mathematical Theories of Logic and Probabilities*, New York, Dover, 1958. Originally published in 1854.
7. D. Bostock, *Logic and Arithmetic*, Oxford, Clarendon Press, 1974–79.
8. G. Cantor, *Uber die Ausdehnung eines Satzes aus der Theorie der trigonometrischen Reihen*, Math. Ann. **5** (1872), 123–132.
9. C.-C. Chang and H. J. Keisler, *Model Theory*, North-Holland, 1973; 3rd ed., 1990.
10. A. Church, *An unsolvable problem of elementary number theory*, Proc. Nat. Acad. Sci. USA **20** (1934), 584–590.
11. A. Church, *Introduction to Mathematical Logic, Vol. I*, Princeton, Princeton Univ. Press, 1956.
12. W. F. Clocksin and C. S. Mellish, *Programming in Prolog*, New York, Springer-Verlag, 3rd ed., 1987.
13. *CoFI Language Design Task Group on Language Design*, CASL,[40] Summary, 25 March 2001,

[40] CASL — The Common Algebraic Specification Language.

http://www.brics.dk/Projects/CoFI/Documents/CASL/Summary/ (accessed 3.01.2005).

14. P. J. Cohen, *The independence of the continuum hypothesis*, Proc. Nat. Acad. Sci. USA **50** (1963), 1143–1148.
15. P. J. Cohen, *The independence of the continuum hypothesis. II*, Proc. Nat. Acad. Sci. USA **51** (1964), 105–110.
16. L. Couturat, *Opuscules et fragments inédits de Leibniz*, Paris, Feliz Alcan, 1903.
17. C. C. Cowen, *On the centrality of Linear Algebra in the curriculum, remarks on receiving the Deborah and Franklin Tepper Haimo Award for Distinguished College or Univ. Teaching of Mathematics*, San Diego California, January 1997. http://www.maa.org/features/cowen.html
18. M. J. Cresswell and G. Ed. Hughes, *A New Introduction to Modal Logic*, Taylor & Francis Inc., 1996.
19. J. N. Crossley, *Structures with features*, Presented at the International Conference in Honor of Fr B.F. Nebres, Manila, February 2001, and submitted for publication in Matimyás Matematika.
20. J. N. Crossley and A. Nerode, *Combinatorial Functors*, volume 81 of Ergebnisse der Mathematik und ihrer Grenzgebiete, **81**, New York, Springer, 1979.
21. J. N. Crossley, I. Poernomo, and M. Wirsing, *Extraction of structured programs from specification proofs*, In: D. Bert, C. Choppy, and P. Mosses (eds.), Workshop on Algebraic Development Techniques, Lect. Notes Comput. Sci. **1827** (1999), pp. 419–437.
22. J. N. Crossley and J. C. Shepherdson, *Extracting programs from proofs by an extension of the Curry–Howard process*, In: J. N. Crossley, J. B. Remmel, R. A. Shore, and M. E. Sweedler (eds.), Logical Methods: In honor of Anil Nerode's Sixtieth Birthday, Boston, Birkhäuser, 1993, pp. 222–288.
23. H. B. Curry, *Functionality in combinatory logic* Am. J. Math. **58** (1936), 345–363.
24. R. Dedekind, *The nature and meaning of numbers*, In: Essays on Theory of Numbers, Dover, 1963. Originally published in 1901.

25. J. C. E. Dekker and J. Myhill, *Recursive equivalence types*, Univ. California Publ. Math. **3** (1960), 67–214.

26. K. Devlin, *Devlin's Angle: When is a proof?* In: MAA[41] Online, June 2003, http://www.maa.org/devlin/devlin0603.html (accessed 8.03.05).

27. C. L. Dodgson, *What the Tortoise said to Achilles*, Mind, **4** (1895), no. 14, 278– 280.

28. J. Dorier, *A general outline of the genesis of vector space theory*, Historia Math. **22** (1995), 227–261.

29. R. Downey, *Parameterized complexity for the skeptic*, In: 18th IEEE Annual Conference on Computational Complexity, 7–10 July 2003, Los Alamitos, California, IEEE, 2003, pp. 147–168.

30. S. Feferman, *In the Light of Logic*, Oxford, Oxford Univ. Press, 1998.

31. S. Feferman, J. W. Dawson, Jr, S. C. Kleene, G. H. Moore, R. M. Solovay, and J. van Heijenoort (eds.), *Kurt Gödel: Collected Works, Vol. I, Publications 1929–1936*, Oxford, Oxford Univ. Press, 1986.

32. G. Frege, *Begriffsschrift und andere Aufsätze*, Olms, Hildesheim, 2nd edition, 1995. Notes by E. Husserl and H. Scholz, edited by I. Angelelli.

33. D. Gabbay, *Labelled Deductive Systems*, Oxford, Oxford Univ. Press, 1996.

34. K. Gödel, *Die Vollständigkleit der Principia Mathematica und verwandter Systeme*, Monatsh. Math. Phys. **37** (1930), 349–360.

35. K. Gödel, *Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme*, Monatsh. Math. Phys. **38** (1931), 173–198. Reprinted in [**31**, pp. 144–194].

36. K. Gödel, *Zur intuitionistischen Arithmetik und Zahlentheorie*, Erg. Math. Kolloqu. **4** (1933), 34–38. Reprinted in [**31**, pp. 286–295].

37. P. R. Halmos, *Finite Dimensional Vector Spaces*, Princeton, Princeton Univ. Press, 1942.

38. T. L. Heath (ed.), *The Thirteen Books of Euclid's Elements*, Cambridge, Cambridge Univ. Press, Cambridge, 2nd edition, 1925. Reprinted, New York, Dover Books, 1956.

[41] MAA – Mathematical Association of America.

39. L. Henkin, *The completeness of the first-order functional calculus*, J. Symbolic Logic, **14** (1949), 159–166.

40. L. Henkin, *Completeness in the theory of types*, J. Symbolic Logic, **15** (1950), 81–91.

41. L. Henkin, *The discovery of my completeness proofs*, Bull. Symbolic Logic, **2** (1996), 127–158.

42. A. Heyting, *Die formalen Regeln der intuitionistischen Logik*, Sitzungsberichte Akad. Berlin, **9** (1930), 42–56.

43. C. A. R. Hoare, *An axiomatic basis for computer programming*, Commun. ACMC, **12** (1969), 576–580.

44. W. Hodges, *Model Theory*, Cambridge, Cambridge Univ. Press, 1992.

45. D. R. Hofstadter, *Gödel, Escher, Bach: an Eternal Golden Braid*, New York, Vintage Books, 1999.

46. W. Howard, *The formulae-as-types notion of construction*, In: J. R. Hindley and J. Seldin (eds.), To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus, and Formalism, Academic Press, 1969, pp. 479–490.

47. B. Jacobs and J. Rutten, *A tutorial on (Co)Algebras and (Co)Induction*, Bull. EATCS **62** (1997), 222–259.

48. J. S. Jeavons, I. Poernomo, B. Basit, and J. N. Crossley, *A layered approach to extracting programs from proofs with an application in Graph Theory*, In: Rod Downey, Ding Decheng, Tung Shih Ping, Qiu Yu Hui, and Mariko Yasugi (eds.), Proceedings of the 7th and 8th Asian Logic Conferences, Chongqing, China, 29 August -2 September 2002, Singapore, Singapore Univ. Press and World Scientific, 2003, pp. 193–222,

49. S. C. Kleene, *Introduction to Metamathematics*, Amsterdam, North-Holland, 1952.

50. G. Kreisel, *Interpretation of analysis by means of constructive functionals of finite types.* In: Arend Heyting (ed.), Constructivity in Mathematics, Proceedings of the Colloquim held at Amsterdam in 1957, Amsterdam, North-Holland, 1959, pp. 101–128.

51. S. Kripke, *A completeness theorem in modal logic*, J. Symbolic Logic, **24** (1959), 1–14.

52. S. Kripke, *Semantical analysis of intuitionistic logic I*, In: J. N. Crossley and M. A. E. Dummett (eds.), Formal Systems and Recurive Functions, Amsterdam, North-Holland, 1965.

53. A. Kurz and A. Palmigiano, *Coalgebras and modal expansions of logics*, Electron. Notes Theor. Comput. Sci. **106** (2004), 243–259.

54. I. Lakatos, *Proofs and refutations: the logic of mathematical discovery*, J. Worrall and Elie Zahar (eds.), Cambridge–New York, Cambridge Univ. Press, 1976.

55. C. I. Lewis and C. H. Langford, *Symbolic Logic*, New York, Dover, 2nd edition, 1959. Originally published in 1932.

56. Des MacHale, *Comic Sections: The book of Mathematical Jokes, Humour, Wit, and Wisdom*, Dublin, Boole Press, 1993.

57. D. MacKenzie, *Computing and the cultures of proving*, Presented at the the Royal Society Discussion Meeting on 18–19 October 2004, http://www.sps.ed.ac.uk/staff/Royal/20Society/20paper.pdf (accessed 8.03.05).

58. E. Mendelson, *Introduction to Mathematical Logic*, 4th ed., Chapman & Hall, 1997.

59. D. Miller, *λ prolog*, http://www.lix.polytechnique.fr/Labo/Dale.Miller/lProlog/index.html (accessed 8.03.05).

60. D. Miller, *A logic programming language with lambda-abstraction, function variables, and simple unification*, J. Logic Comput. **1** (1991), no.4, 497–536.

61. B. K. Motilal, *Logic, Language, and Reality: an Introduction to Indian Philosophical Studies*, Delhi, Motilal Banarsidass, 1985.

62. B. K. Motilal, *The Character of Logic in India*, J. Ganeri and H. Tiwari (eds.) Oxford, Oxford Univ. Press, 2000.

63. Sir T. Muir, *The Theory of Determinants in the Historical Order of Development*, London, Macmillan, 1906.

64. D. Nardi and R. J.Brachman, *An introduction to Description Logics*, http://www.inf.unibz.it/ franconi/dl/course/dlhb/dlhb-01.pdf (accessed 2.03.05).

65. A. Nerode, *Extensions to isols*, Ann. Math. **73** (1961), 362–403.

66. G. Peano, *Formulario Mathematico* Ed. V (Tomo V de Formulario completo), Turin, 1908.

67. I. H. Poernomo, J. N. Crossley, and M. Wirsing, *Adapting Proofs-as-Programs*, New York, Springer, 2005 [to appear].

68. J. C. Reynolds, *Polymorphism is not set-theoretic*, In: Semantics of Data Types (Valbonne, 1984), Berlin, Springer, 1984, pp. 145–156.

69. A. Robinson, *Non-Standard Analysis*, North-Holland, 1966.

70. B. A. W. Russell, *Principles of Mathematics*, Cambridge, Cambridge Univ. Press, 1903.

71. B. A. W. Russell, *Mathematical logic as based on the Theory of Types*, Am. J. Math. **30** (1908), 222–262. Reprinted in [**76**, pp. 152–182].

72. B. A. W. Russell, *The autobiography of Bertrand Russell*, 3 vols., London, Allen and Unwin, 1967–1969.

73. J. E. Stoy, *Denotational Semantics: The Scott–Strachey Approach to Programming Language Theory*, MIT Press, 1977.

74. *Summaries of talks*, Summer Institute for Symbolic Logic, Cornell Univ. Institute for Defense Analyses, Princeton, 2nd edition, 1960. First edition, 3 vols, cyclostyled, 1957.

75. M. E. Szabo (ed.), *The collected papers of Gerhard Gentzen*, Amsterdam, North-Holland, 1969.

76. J. van Heijenoort (ed.), *From Frege to Gödel*, Harvard Univ. Press, 1967.

77. A. N. Whitehead and B. A. W. Russell, *Introduction to Mathematical Philosophy*, Cambridge, Cambridge Univ. Press, 1925–1927. First published in 1910–1913.

78. M. Wirsing, *Structured specifications: Syntax, semantics and proof calculus*, In: M. Broy (ed.), Informatik und Mathematik, Festschrift für F. L. Bauer, Berlin, Springer, 1991, pp. 269–283.

79. M. Wirsing, J. N. Crossley, and H. Peterreins, *Proof normalization of structured algebraic specifications is convergent*, In: J. Fiaderio (ed.), Workshop on Algebraic Development Techniques, Proceedings of the Twelfth International Workshop on Recent Trends in Algebraic Development Techniques, Lect. Notes Comput. Sci. **1589** (1998), 326–340.

80. H. Woodin, *The Continuum Hypothesis*, 2000. Lectures at the Paris Logic Colloquium, http://math.berkeley.edu/˜woodin/talks/lc2000.1.pdf,

math.berkeley.edu/ woodin/talks/lc2000.3.pdf.

81. A. S. Ésénine-Volpine [Yesenin-Volpin], *Le programme ultraintuitionniste des fondements des mathématiques*, In: Infinitistic methods. Proceedings of the Symposium on Foundations of Mathematics, Warsaw, 2–9 September 1959, Oxford, Pergamonn Pres – Warszawa, Państwowe Wydawnictwo Naukowe, 1960, pp. 201–223.

Two Doors to Open

Wilfrid Hodges

Queen Mary, University of London
London, UK

Mathematicians do not care to speculate in print about the future of their subject. There are several good reasons for this. One is that significant advances in mathematics nearly always involve the injection of unpredictable new ideas. A few rash mathematicians have published predictions for their own fields; these predictions were generally based on the stock of ideas already in the field, and with hindsight they became steadily more embarrassing as the subject moved onwards to higher things. Nor is it wise to publish bold speculations until you have checked that they work – and then they are no longer bold speculations.

For a mathematician, problem lists are a safer bet than predictions. A description of a problem leaves it open whether one can solve the problem routinely with known methods, or whether something radically new is needed. The best problem lists have come from relatively young researchers working at or near the crest

Mathematical Problems from Applied Logics. New Logics for the XXIst Century.
Edited by Dov M Gabbay *et al.*/ International Mathematical Series, Springer, 2005

of the wave. David Hilbert was not yet forty when he delivered his famous list of problems at the 1900 Paris Congress.

With these remarks I think I have disqualified myself from writing anything about the future of my own field, model theory. I also told the editors that I had misgivings about the phrase "New Logics" in their title. The phrase suggests that logic consists of an assembly of "logics." That view of logic was widely held in the 1930s, when for example Alfred Tarski thought it was appropriate to supply each deductive theory with its own set of logical axioms and rules of inference. In some applications of logic within computer science this framework is no doubt still appropriate, for special reasons. For logic as a whole it seems a bad anachronism. At present there are no signs that model theory or set theory will advance in the foreseeable future by taking on board new logics (unless perhaps the Ω-logic of Woodin [**50**], but I doubt this is what the editors had in mind). Nevertheless the editors were kind enough to answer that they still wanted a contribution from me, so I gratefully supplied what follows.

I describe two different developments that I would like to see in logic. The first is a serious interaction between mathematical logic and cognitive science. The second is the study of the semantic ideas of medieval Arab linguists, particularly those outside the Aristotelian tradition. These two areas are very different. On the one side it seems to me that a closer cooperation between logic and cognitive science is inevitable, and the most I can hope is to nudge it along. On the other side, historical work on Arab semantics is unlikely to have any dramatic impact on present-day semantic thinking, but I am convinced that the Arabs have things of value that should be treasured; they represent an unfamiliar viewpoint, and I hope some future workers will find it a source of inspiration.

For their kindness in giving me various comments and pieces of information I thank Salman Al-Saad, Khaled El-Rouayheb, Brendan Gillon, †Jeffrey Gray, Fritz Hamm, Sally Hodges, Juliette

Kennedy, Michiel van Lambalgen, Dan Osherson, David Over, John Sowa, Keith Stenning, Sam Tarzi, Tony Street and Kees Versteegh. They are not responsible for any errors or foolishnesses below.

1. Logic and Cognitive Science

The body of mathematics, past and present, is an extraordinary monument to human powers of rationality. Cognitive scientists inevitably ask: How on earth do mathematicians manage to do the things that they do? There is already a strong literature on some aspects of this question, for example on teaching mathematics, or computer simulation of mathematical reasoning, or scans of brain activity during mathematical thinking. From the mathematician's point of view, there are some obvious limitations visible in nearly all of this literature. In an MRI scan, for example, one is limited to studying how a human being works on a problem over the course of a few minutes, typically a simple multiplication of integers. Mathematical researchers by contrast are quite used to spending five years of slow and steady grind on a problem, and the problem will be conceptually very much more complicated than multiplying integers.

From their side, mathematicians know that their chief working tool – their mathematical consciousness – is something that they barely understand. Ramanujan imagined that his mathematical results came to him from the goddess Namagiri, and the rest of us would find it hard to say anything more convincing about our own mental workings. Imagine painters who could not predict what colour and stroke their brush would make, or builders who relied on guesswork to check a vertical line. We mathematicians need to understand ourselves. The only reason we are not hammering at the door of the cognitive scientists to learn more from them is that we do not yet believe they know more about our mathematical thinking than we know ourselves. But cognitive science has advanced dramatically in the last half century, and there is bound to be a time when collaboration becomes fruitful.

When it comes, the collaboration will be between cognitive science and logic, because logic is the formal study of sound reasoning, and sound mathematical reasoning is what the cognitive scientists will be telling us about. David Mumford might well disagree; in his plenary lecture to the Beijing International Congress of Mathematicians in 2002 ([**28**, p.402]), talking about visual perception and pattern recognition, he said:

> This issue of logic vs. statistics in the modeling of thought has a long history going back to Aristotle ... I think it is fair to say that statistics won.

The history of gas dynamics shows that this is a false dichotomy. To a very close approximation, gases behave according to laws that do not mention any statistical notions at all, but we now accept that any explanation of these laws has to be statistical. How do theorem-proving humans behave? Given what we know about the physical construction of the brain, it would be very surprising if one could account for human reasoning abilities in depth without invoking statistics. But theorem-proving itself is a logical activity, not a statistical one. (I should add that the cognitive scientists will be right to pay just as much attention to reasoning in algebraic geometry as to reasoning in logic, of course.)

The raw material within mathematics for cognitive scientists to study is enormous. This material has some distinctive features.

- We did not evolve for mathematics. We did evolve for language; there are earmarked language areas in our brains. (At least there are areas for syntax; Keith Stenning warns me to be more cautious about semantics.) There are probably earmarked areas for counting small numbers and estimating larger ones. But there are certainly no areas for completing commutative diagrams or extracting cube roots.

- Mathematical reasoning delivers results of extraordinary precision. For example the number of discrete subgroups of the group of isometries of a three-dimensional vector space over the real numbers, containing three linearly independent translations, is 230. Everything in this statement is perfectly precise, given the definitions.

- Mathematical results and proofs are objective and culture-free to an extraordinary degree. If Archimedes proved a thing, we do not need to rewrite his proof except perhaps to clear up points of notation.

- Although in principle all known mathematical theorems can be stated and proved without any reference to space or the visual field, many mathematicians rely very heavily on visual and spatial intuition in their work.

- More generally, mathematicians often solve problems by translating or reducing them to a different form where the solution is easier. The translations may be quite elaborate and may involve new concepts.

In what follows I want to point to some places where reflective mathematicians themselves have illustrated these points.

1.1. Spatial intuition

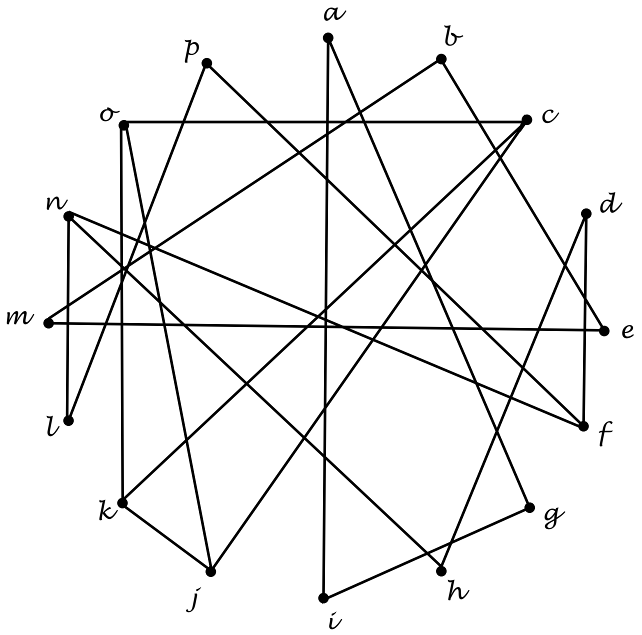
Michael Atiyah [**4**, p. 5f] remarked during the millennium celebrations:

> If I look out at the audience in this room I can see a lot; in one single second or microsecond I can take in a vast amount of information, and that is of course not an accident. Our brains have been constructed in such a way that they are extremely concerned with vision. . . . Understanding, and making sense of, the world that we see is a very important part of our evolution.

> Therefore, spatial intuition or spatial perception is an enormously powerful tool, and that is why geometry is actually such a powerful part of mathematics – not only for things that are obviously geometrical, but even for things that are not. We try to put them into geometrical form because that enables us to use our intuition. Our intuition is our most powerful tool.

There are (at least) two different things that people have in mind when they talk of spatial intuition, namely speed and self-evidence.

I guess that Atiyah is chiefly talking about speed. Here is an example, from a coursework question I set my engineering students earlier this term. The diagram shows a graph with several connected components superimposed; the problem is to separate out the connected graphs.



(1)

Programming a computer to solve this problem, we would probably supply (or program the computer to extract) the adjacency matrix of the graph (1):

$$\begin{array}{c|cccccccccccccccc}
 & a & b & c & d & e & f & g & h & i & j & k & l & m & n & o & p \\
\hline
a & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
b & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
c & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\
d & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
e & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
f & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
g & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
h & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
i & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
j & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
k & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
l & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
m & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
n & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
o & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
p & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0
\end{array} \tag{2}$$

Human students can solve the problem from the matrix (2) too, but they have to be taught an algorithm, they perform it very slowly and they tend to make errors. By contrast the separate components in (1) leap out at the eye. Somewhere below the level of consciousness, some very fast computing is going on.

Another example is finding orientations in tesselations of the plane. If a tesselation is not isomorphic to its mirror image, then it must contain some feature whose mirror image never appears in it. I sometimes ask students to find such features. Again the mathematician's eye jumps to the rescue. Most people need a few practice runs to train their eye. One interesting feature of this problem is that there is no particular geometric configuration that the eye is looking for; any configuration with the appropriate abstract property will do. The computing power involved must be gigantic.

Another example is sorting numbers. If I give you thirty numbers chosen at random between 0 and 100:

$$\begin{array}{rrrrrrrrrrrrrrr} 72 & 93 & 92 & 78 & 21 & 16 & 2 & 70 & 58 & 19 & 27 & 3 & 29 & 76 & 64 \\ 67 & 32 & 4 & 63 & 37 & 41 & 60 & 61 & 51 & 72 & 33 & 29 & 43 & 35 & 47 \end{array} \tag{3}$$

you can quickly sort them into increasing order. I gave my students a table of random numbers and asked them to time how long it took to sort k numbers for various k, plot the time against k and then write down how they thought they did the sorting. Some students produced graphs that were plausibly $O(k^2)$ or $O(k \log k)$, with algorithms to match. What I had not expected was a substantial bunch of students who came back with linear graphs. Looking at their descriptions showed the reason: they had broken up the set of numbers into blocks within which they could sort almost instantaneously just by looking. They seemed to be able to do this with blocks of size up to about 10. Next time I give this problem to a class, I will choose k large enough to frustrate this approach.

Presumably people's performance on examples like these differs according to their experience, and some people have natural aptitudes. Published research suggests that people with autism might have difficulties with the graph problem because it involves integrating several lines into a larger shape, and with the orientation problem because it involves assessing how an oriented feature sits in the pattern as a whole. (See for example Happé [**18**].) When I gave the orientation problem to a class, a student assessed with Asperger's found the problem extremely hard. Another student with an optic nerve disability did noticeably well, and I wondered if he had compensated for poor visual information by developing his visual mental powers.

1.2. Kurt Gödel and the choice of representation

No mathematician would accept intuition in Atiyah's sense as a guarantor of truth. Atiyah points to our ability to see complicated facts quickly; but we can still make mistakes and the facts need to be checked. Quite different from this is what the logician Kurt Gödel calls "mathematical intuition": this is our ability to see with the mind's eye that certain things are "obviously true." We see immediately that if A is to the left of B and B is to the left of C, then A is to the left of C and B is between A and C. There it is, right in front of our mind's eye. People interested in the epistemology of mathematics have often appealed to "intuition" in this sense as a guarantor of truth. (People who take this view have to account for the fact that in earlier generations, just such intuitions were used to demonstrate that space is euclidean, or that if A is a proper part of B then A must be smaller than B.) I doubt the wisdom of using the word "intuition" in this sense, except within strictly philosophical contexts where history has decreed that "intuition" translates the German "Anschauung." In colloquial English "intuition" too easily suggests "hunch." So instead I say "self-evidence" in what follows.

The relevance of self-evidence to cognitive science is a tricky matter. In the first place, computer simulations of human thinking are a well-established tool of the subject, and clearly we cannot attribute self-evidence to the thoughts of the computer. On the other hand, if the thinking – either human or simulated – is not guided by rules of truth and rationality, then strictly it will not be reasoning. Thad Polk did an interesting experiment with Allen Newell's cognitive simulation system SOAR: he fed into SOAR some rules for operating syllogisms, including some invalid rules, and he showed that SOAR performed on a range of syllogisms with an accuracy quite close to that recorded in human populations studied by Johnson-Laird and Bara [**29**, pp. 396–410]. These results may be helpful for understanding how humans behave when faced with syllogistic problems in psychological tests. But it is clear that human behavior in these contexts depends on more than just rationality – it often involves ill-founded and unchecked guesses made against deadlines. When the mathematicians come to ask the cognitive scientists how mathematical researchers solve problems and prove theorems, this will not be what they are asking for. They will be asking about a kind of reasoning where the reasoner's perception of truth provides a constant check on the accuracy of the workings.

In the rest of this section I discuss some of Gödel's contributions. Gödel already has close ties to cognitive science. For example he was the first person to illustrate in exact terms how an appropriate choice of representation can lead to dramatic improvement in our proving abilities: it allows us to prove more things, and faster. (This is the content of his short paper *On the lengths of proofs* [**12**].) Second, already in 1956 he came close to inventing and applying the notion of a problem with polynomial-time complexity (in a letter to von Neumann, [**16**, p. 375]). Third, he was a pioneer in the use of formal systems to describe and analyse concepts.

Gödel saw himself as a philosophically-minded mathematician, not as a cognitive scientist. My contention is that *his philosophical contributions provide a rich collection of tools and problems for the cognitive study of mathematics.* He was obsessively introspective, and above all he introspected his own mathematical thinking. Often he reported what he found in terms of self-evidence (or "mathematical intuition"). I think cognitive scientists will generally want to read past what Gödel says about the evidential value of his intuitions, and concentrate on what he says about the form of his thinking, and about how to describe it in mathematical terms.

Take for example the question how we mentally represent a problem. It is a familiar fact that we use different types of thinking for different problems. One of the central themes of Keith Stenning's recent book *Seeing Reason* [**41**] is that there are many different "systems of representation," some more appropriate for one task, some for another, and that humans differ considerably in their ability to use different systems of representation. Stenning's approach is mainly empirical, taking systems of representation as he finds them in the literature; he places them on a scale that (to oversimplify a little) has visual diagrams at one end and text at the other. Kurt Gödel thought a great deal about systems of representation for parts of mathematics, and he classified them in ways quite different from Stenning's.

In the first half of the twentieth century it was the custom for logicians to analyse a branch of mathematics by formulating it within a fully interpreted formal system. A formal system has a precisely defined formal language, each sentence of this language has a precisely specified meaning, certain sentences are claimed to be true and are labelled "axioms," and rules are given for deducing further true sentences from the axioms. Each formalism rests on a number of basic concepts, called its *primitives*, and for each primitive there is a *primitive term* expressing it in the formal language.

Gödel did not just accept this style of analysis – he demanded it, and in particular he demanded that all the expressed concepts should be absolutely precise, and that there should be no ambiguity about the syntax or the rules of inference. For example he criticised "the intuitionists" for being too vague about their concepts (cf. [**15**, p. 190]):

> . . . the primitive terms of intuitionistic logic lack the complete perspicuity and clarity which should be required for the primitive terms of an intuitionistic system.

He criticised Whitehead and Russell for being sloppy with their syntax (cf. [**14**, p. 120]):

> What is missing [from *Principia*] is a precise statement of the syntax of the formalism. Syntactical considerations are omitted even in cases where they are necessary for the cogency of the proofs.

Today logicians are usually more relaxed about these things than Gödel was, but not as relaxed as some cognitive scientists.

In fact Gödel's requirements for analysing a part of mathematics as a formal system correspond fairly closely to David Marr's second level of description of an information-processing system (cf. [**27**, p. 23]):

> The second level ... involves choosing two things: (1) a *representation* for the input and for the output of the process and (2) an *algorithm* by which the transformation may actually be accomplished. ... For addition, we might choose Arabic numerals for the representations, and for the algorithm we could follow the usual rules about adding the least significant digits first and "carrying" if the sum exceeds 9.

One can write Marr's addition representation and algorithm in a form that Gödel would have regarded (at least after Turing's work) as an acceptable formal system. Many cognitive scientists have accepted Marr's account of how to describe an information-processing system, but they do not often come close to Gödel's standards of precision. Is it reasonable to expect them to, given their subject matter? I discuss an example in Section 1.3 below.

When we have the formalism, we can use it in two ways. First, we can play what Weyl and Hilbert called a "game with formulas:" the axioms are strings of symbols and the inference rules are rules for "deriving" new strings of symbols from given ones. The game with formulas is simply to use the rules to derive new strings.

Or second, we can reason with the concepts and the necessary truths. The formalism then serves to describe possible reasonings and to check the accuracy of reasonings, but it is not strictly part of the system of concepts that we are reasoning with. Gödel wrote much (not all of it published in his lifetime) about the differences between one formalism and another in this second style of use. For him, the important cognitive differences between one act of reasoning and another lie in the concepts and introspected truths that are used. (At this stage some people find it important to distinguish a class of concepts called "logical." Gödel sees no need to do this; I doubt that the distinction has any cognitive significance.) We can contrast sets of concepts in various ways.

(a) Given the concept C of some kind of entity, we can form and use the concept C' of a set of entities of kind C. Some facts about sets are self-evident, for example that if a is a set and P a property that all members of a either have or do not have, then there is a set b consisting of the members of a that have P. Along with the new concept of a set of entities of kind C, our formalism should list some facts of this kind. We say that the new concept is of a *higher order* than C, and reasoning with it is said to use a *higher-order logic.* One can repeat this extension of the concept system and introduce a concept C'' of sets of sets of kind C', and so on.

This is the situation of Gödel's paper [**12**] mentioned above, on lengths of proofs. Gödel shows that in general, passing to a higher order not only allows us to deduce new facts in the lower level language, it also allows us to give much shorter proofs of facts that we could already prove in the lower level language.

(b) A variant of (a) is to introduce new abstract objects, normally sets, that are not necessarily sets of entities at the lower level. For example we can introduce the concept "natural number," together with the concepts of 0, 1, addition, multiplication etc. An important self-evident truth in this case is the axiom of induction up to ω:

> Suppose P is a property that all natural numbers either have or do not have, and 0 has property P, and whenever a natural number n has property P then so does $n+1$. Then every natural number has property P.

Speed-up applies here too, and more things in the original language become provable. We can improve the improvement by replacing the set ω of natural numbers by the set of all ordinals less than some given transfinite ordinal α. Gödel was interested in the types of reasoning that correspond to different choices of α.

Of course there is no way we can explicitly use the concepts of all the infinitely many natural numbers when we apply induction up to ω. But we do not need to; all we need is the concepts of the set of natural numbers, the number 0 and the operation of adding 1 (together with the concept answering to the property P). Some mathematicians have claimed that we can handle induction up to ω by spatial intuition, seeing

$$0,\ 1,\ 2,\ 3,\ 4,\ 5,\ \ldots \tag{4}$$

in our mind's eye. There is a problem here, discussed for example by Bernays [**7**] in 1922: one of the evident facts associated with the concept of adding 1 is that for every natural number there is a greater natural number. In our mind's eye we can only see a limited number of numbers; we certainly cannot see, for each of them, a larger one. So the concept "add 1" is not really given in spatial intuition. At best one could put into one's mind a marker like the " ... " in (4) above. But this " ... " is a symbol representing a concept; it belongs to a higher level of abstraction.

Incidentally the problem of justifying induction up to ω by spatial intuition is closely related to the problem of representing eternity in music, which I discussed in [**20**, pp. 95f, 105f]. Some composers (for example Haydn, Wagner) express eternity as a long but finite time; this is a different concept but the listeners are supposed to use their imagination. Other composers (Smetana, Britten) use a musical equivalent of the '. . . '

(c) A different way of increasing the reasoning power is to form concepts about concepts, or in Gödel's words [**14**, p. 272f], concepts

> which do not have as their content properties or relations of *concrete objects* (such as combinations of symbols), but rather of *thought structures* or *thought contents* (e.g., proofs, meaningful propositions, and so on), where in the proofs of propositions about these mental objects insights are needed which are not derived from a reflection upon the combinatorial (space-time) properties of the symbols representing them, bur rather from a reflection upon the *meanings* involved. [Gödel's emphases]

Gödel believed that there is a level of reasoning about concepts that is still open to the evidence of immediate mental inspection. He conjectured that this level of reasoning has a sharp upper boundary (namely that in reasoning power it corresponds to induction up to any ordinal strictly less than the first epsilon-number ε_0). His conjecture seems to have some cognitive content, relating to the kinds of infinite tree structure that we can picture to ourselves. He comments [**14**, p. 273f]:

> Whether the necessity of abstract concepts for the proof of induction [up to ε_0 or beyond] is due solely to the impossibility of grasping intuitively the complicated (though only *finitely* complicated) combinatorial relations involved, or arises for some essential reason, cannot be decided off hand. [Gödel's emphasis. A footnote refers to the "proof-theoretic characterization of concrete intuition".]

We can also extend our concept system by forming *concepts of formulas*, in other words, passing to a metalanguage. But there is a catch here. If the concepts are just concepts of particular symbols or symbol strings appearing in our formulas, they give us nothing more than we already had in the formulas; there is no speed-up except insofar as the concepts act as labels for large chunks of text. But we do get extra power by introducing metalevel concepts such as the concept of a formula being an instance of a schema. Reasoning with such concepts often uses induction up to ω, and this is the real source of the extra strength.

(d) In (a)–(c) Gödel is talking about our direct awareness of certain facts that we can hold in our heads by means appropriate concepts. He distinguishes this from a different kind of introspective knowledge, which is knowledge about *our own mental capacities* – or in his own words [**16**, p. 187] "insights about the given operations of our mind." He is thinking of the intuitionist mathematics of Brouwer. He gives no examples, and there is nothing in this area that I would recommend with any confidence as worth the attention of a cognitive scientist.

(e) One of Gödel's abiding interests was the process of "conceptual analysis" that takes informal abstract notions and turns them into tools of mathematical reasoning. In 1947 ([**14**, p. 177]) he noted that "the analysis of the phrase "how many" leads unambiguously to quite a definite meaning" for a question about transfinite cardinalities. In 1964 ([**13**, p. 369]) he said that Turing had given "a precise and unquestionably adequate definition" of the concept of a formal system, through his "analysis of the concept of 'mechanical procedure'." Obviously there is a close link between analysing a concept and finding a fruitful representation for reasoning with it. In an unpublished note of 1972 ([**14**, p. 306]) he said

> . . . we understand abstract terms more and more precisely as we go on using them, and [. . .] more and more abstract terms enter the sphere of our understanding. There may exist systematic methods of actualizing this development, which could form part of the procedure.

But Gödel was frustrated by his complete inability to describe any such systematic methods. The problem remains open.

1.3. A sample cognitive description of reasoning

Gödel speaks for himself. I could simply commend his ideas to the attention of the cognitive scientists; in any case I do not have the expertise to move far into their territory. But I think I need to say something more about the psychological cash value of Gödel's points.

Take for example the following argument from Johnson-Laird and Byrne [**22**, p. 9]:

Arthur is in Edinburgh or Betty is in Dundee, or both.
Betty is not in Dundee. (5)
If Arthur is in Edinburgh, then Carol is in Glasgow.

The authors describe a possible way of deducing logical consequences from these three sentences. I summarise it as follows;

I apologise if my summary distorts anything. We label the three propositions “Arthur is in Edinburgh,” “Betty is in Dundee” and “Carol is in Glasgow” as a, b and c respectively. Then we list “the set of possibilities” for these propositions as follows:

	a	b	c
i.	T	T	T
ii.	T	T	F
iii.	T	F	T
iv.	T	F	F
v.	F	T	T
vi.	F	T	F
vii.	F	F	T
viii.	F	F	F

(6)

The first premise in (5) rules out lines *vii* and *viii*, then the second premise rules out lines *i*, *ii*, *v* and *vi*, and finally the third premise eliminates *iv*. Only *iv* remains, and since *iv* makes c true we can conclude that Carol is in Glasgow. This concludes the argument.

I suspect Gödel would comment at once that this description could fit several different mental procedures. We have not been told what are the concepts used, or what properties of them are invoked.

In the first place, what is a “possibility” for a, b and c? In other words, what mental objects are the rows of the table (6) reporting? The simplest possibility is that each row stands for a statement using a, b and c; for example row *vii* stands for the statement

Arthur is not in Edinburgh, Betty is not in Dundee and Carol is in Glasgow.

In this case the step ruling out line *vii* is a logical inference:

> Arthur is in Edinburgh or Betty is in Dundee, or both. Therefore it is not the case that: Arthur is not in Edinburgh, Betty is not in Dundee and Carol is in Glasgow. (7)

We are not told how this inference is made. This is a pity, because the inference is not much more straightforward than the original deduction from (5).

Two ways of making this inference suggest themselves. The first is that the inference in (7) is immediate and does not involve any processing, other than the non-negligible amount of processing needed to choose mental representations for the sentences involved. In this particular case this suggestion is not terribly plausible.

A second way is that we read off the falsehood of "Arthur is in Edinburgh or Betty is in Dundee, or both" from the truth of "Arthur is not in Edinburgh, Betty is not in Dundee and Carol is in Glasgow," and then perform a contraposition. These steps have a better chance of being immediate, "self-evident" in Gödel's view.

A second understanding of "possibilities" is that they are literally possibilities, i.e., ways the world might have been. Arthur might have been in Edinburgh, Betty in Dundee and Carol in Glasgow, as line *i* records. Here a problem is that these eight possibilities are not really possible, given the premises (5); in fact the reported argument establishes just this. We can rescue the suggestion by revising it: possible *relative to* a given set of sentences. All of *i–viii* are possible relative to no premises, but only *i–vi* are possible relative to the first premise, and so on. Note that this reading of Johnson-Laird and Byrne ascribes to the reasoner a further concept, namely the concept of "possible relative to." So we have moved to a higher level in Gödel's terms, but I think without any gain in reasoning power. (For a fuller analysis see Stenning and Yule [**42**].)

Johnson-Laird and Byrne themselves describe the reported argument as "model-theoretic." This suggests a third and more set-theoretic notion of "possibility," for example "function which assigns a truth value to each of the propositions a, b, c". Myself I would be happy describing this as a model-theoretic notion, even if it is represented as in (6) by a string of T's and F's. But now the character of the argument changes. What does it mean for a premise to "rule out" such a function? Presumably by contradicting *vii* in the first sense of "possibility" discussed above, or eliminating *vii* in the second sense. But then the model theory here is completely redundant; we never use it except by translating it away immediately.

There is a fourth option, namely that the "possibilities" are nothing but rows of symbols, T or F, and the whole argument is conducted by using the truth table rules. In short, Johnson-Laird and Byrne are describing an application of a truth-table proof calculus, which is a good example of a "game with formulas." As a working logician I find I do use this kind of reasoning quite often, because I have learned it and it is efficient on small problems. I would be amazed to hear that someone hit on it with no training at all in the subject. Although this is a possible reading of Johnson-Laird and Byrne's text, I doubt very much that it is what they have in mind, because they explicitly distinguish their view from another approach that they call "proof-theoretic."

In short, the description that Johnson-Laird and Byrne give here begs almost every question that Gödel could have asked. From this description we have no idea what mental operations are being described; in particular the description as "model-theoretic" is no help at all. I leave it to the reader to check how far the rest of their book clarifies these ambiguities.

Logicians do sometimes talk of "model-theoretic arguments" or "semantic arguments." They mean arguments that use the notion of a model of a formal sentence. Models are set-theoretic objects and the notion of a model of a formal sentence is defined set-theoretically. The use of this notion is an example of Gödel's (b) in Section 1.2 above, and it can lead to dramatic improvements in the efficiency of proofs of first-order theorems. The main reason for this is that one can exploit set-theoretic principles such as induction on the ordinals. Dan Osherson called my attention to a beautiful example due to George Boolos [**8**]. Boolos gives a one-page proof of the following inference:

$$\forall n\ F(n,1) = S(1).$$

$$\forall x\ F(1, S(x)) = S(S(F(1,x))).$$

$$\forall n\,\forall x\ F(S(n), S(x)) = F(n, F(S(n), x)).$$

$$D(1).$$

$$\forall x\ (D(x) \to D(S(x))).$$

$$\vdash D(F(S(S(S(S(1)))), S(S(S(S(1)))))).$$

Boolos' short proof is in second order logic, written out informally. However, Boolos also shows (by a cut elimination argument) that any proof of this entailment in any of the standard proof calculi for first-order logic would be of astronomical length. One can easily adjust Boolos' short proof into a set-theoretic argument, or indeed a model-theoretic one.

Mike Oaksford and Nick Chater [**30**, p. 140], responding to a defence of Johnson-Laird by Alan Garnham, comment:

> Semantic methods of proof are simply an alternative way of passing from premises to conclusions.

Boolos' example makes this a rather odd statement. But still it is true in a sense. To the best of my knowledge, Johnson-Laird never shows any way in which human model-theoretic reasoning could exploit the extra strength of model theory; his model-theoretic steps are just standard first-order steps interpreted as reasoning about models. And though Boolos' short proof, in model-theoretic form, certainly does use semantic methods which correspond to nothing in natural deduction, Boolos does not propose any general strategy for logical reasoning that relies on such methods.

1.4. Frege versus Peirce: comparison of representations

Around 1880, Gottlob Frege and C. S. Peirce independently gave the earliest logical calculi that are adequate for first-order logic. There is no evidence that these two logicians exerted any influence on each other, but their aims had much in common. It was important for both of them that each written formula should make its semantic structure clear, something that natural language often fails to achieve. In their own words:

> **(Frege)** ... I called what alone mattered to me the *conceptual content* [*begrifflichen Inhalt*]. Hence this definition must always be kept in mind if one wishes to gain a proper understanding of what my formula language is. ... [My] deviations from what is traditional find their justification in the fact that logic has hitherto always followed ordinary language and grammar too closely. (Trans. [**19**, p. 6f])

> **(Peirce)** ... this syntax is truly diagrammatic, that is to say that its parts are really related to one another in forms of relation analogous to those of the assertions they represent, and ... consequently in studying this syntax we may be assured that we are studying the real relations of the parts of the assertions and reasonings; which is by no means the case with the syntax of English. [**33**].

Frege's two-dimensional notation *Begriffsschrift* appeared in his book of that name in 1879 [**10**]. Peirce called his two-dimensional notation *existential graphs*; the first order part consists of the *beta graphs*. He developed existential graphs over many years and published them in a variety of places; Lecture Three of his 1898 Cambridge Conference Lectures [**32**] is a good reference.

Here are some typical recent comments on Frege's *Begriffsschrift*

(a) *Begriffsschrift* uses up too much space.

(b) Nobody but Frege ever used *Begriffsschrift.*

(c) *Begriffsschrift* has a forbidding appearance.

By contrast here are some typical recent comments on Peirce's existential graphs:

(e) The graphs are interesting to study because Peirce was an important logician and he thought they were one of his best contributions.

(f) Inference in existential graphs is easy.

(g) Existential graphs exploit geometric intuition.

There is a growing literature on applications of Peirce's existential graphs in logic teaching, computer science and elsewhere. (For a sample, see Hammer [**17**] and Ketner [**25**]; scholar.google gives a few hundred further references.) I do not think anybody has suggested anything similar for *Begriffsschrift.* Why the difference?

At face value the distinction is very unfair. On (b) for example, hardly anybody but Peirce used existential graphs until recently. Points (e) and (g) apply equally well to Frege. As to (f), Frege designed *Begriffsschrift* so as to make modus ponens

Given p and "If p then q," infer q.

particularly straightforward.

My impression is that nobody has seriously compared the two notations. In fact people who discuss existential graphs rarely mention *Begriffsschrift* at all. Some writers have contrasted *Begriffsschrift* not with existential graphs but with Peirce's earlier one-dimensional notation, which is much closer to present-day notations for first-order logic. This is a less interesting comparison.

There are some obvious questions of cognitive science to ask here.

- Is Frege's notation really harder to learn than Peirce's?
- Is Peirce's notation really better than Frege's as a tool of reasoning?

Anecdotal evidence is useless; one needs a properly designed experiment in which a number of people are trained in both systems and then tested for speed, accuracy, comfort etc. Let me conjecture that there is no significant difference between the notations, at least in point of speed and accuracy. (There is a prior problem that Peirce's notation can be read in more than one way; see Shin [**39**].)

- If differences do come to light, then what are they?
- Do they apply equally well to all users or do they correspond to different people's cognitive styles?

Though the two notations carry the same information, their details are very different. Peirce came to his graphs through his study of Kempe's graph notation, an early contribution to the notation of chemistry. The source of Frege's notation is less clear, but since his diagrams are in fact the stemmata of a dependency grammar for his language, my guess is that they owe something to the parsing trees that were in regular use in Germany in the 19th century, as illustrated in Baum [**6**]. It is relevant that Frege's father was a professional language teacher.

2. Medieval Arabic Semantics

Medieval? Weren't we supposed to be moving logic forwards, not backwards?

The history of logic is an honourable and well-established discipline. In the last fifty years it has made great advances, both in the availability of material and in our understanding of the minds of our predecessors. (For example when I was a student in the 1960s, the Greek and Latin commentaries on Aristotle were mentioned only for information about variant readings of Aristotle's text. Today everyone can look up a mass of information on these commentators as logicians in their own right, and the Arabic commentaries are becoming available too.) Probably every logic teacher will agree that it is helpful to be able to show our students where their subject came from, and how a later generation managed to climb over the hurdles that defeated an earlier one. Some good material for teaching the history of logic is already available; more would be welcome.

On the other hand I tried to draw up a checklist of items drawn from classical or medieval logic that have been a clear help to research in logic during the last half century. I do not count free association here: anybody can get some inspiration by reading anything (just as Alban Berg, seeking ideas for the cadenza of his Violin Concerto, asked the soloist Louis Krasner to play just anything for an hour or so, "Bitte nur spielen – unbedingt!"). Perhaps the fact that there is a programming language called Occam is a testimony to this. Rather I looked for places where somebody published something from the history of logic and it directly affected research in logic.

So far only two examples have come to mind. One is Arthur Prior's publicising of the notions of de re and de dicto modality, which he took from Abelard [**35**]. Certainly these were useful and the names are still in regular use, though it is hard to be sure modal logic would have developed differently without Prior's paper. The second is Peter Geach's donkey sentence "Any man who owns a donkey beats it," which he describes as "medieval" [**11**, p. 117]. Certainly Geach's example has had a strong influence on research at the logical end of natural language semantics. Its connection with the history of logic is less clear. I am guessing Geach took the idea of the sentence from the early fourteenth century logician Walter Burley, who discusses the meaning of the sentence "Every man who has a donkey sees it” [**9**, *i.4*]. But Geach’s comments on it are his own and not Burley’s.

These are small pickings.

Part of the reason is the lack of novelty. The classics and the western medievals are our own ancestors, and the main features of the family history were always with us. Historical material is more likely to be useful to us if it meets two conditions:

(1) it addresses problems that we can recognise and find interesting, and

(2) it comes to these problems from a viewpoint markedly different from any that is familiar to us.

I think there is a serious chance that medieval Arabic semantics contains material meeting both these conditions. If so, then somebody from the West should study it. We may be unlucky this time too; but at least it will be a contribution to a seriously neglected area in the history of semantics.

Because the work is still to do, I cannot go beyond making a prima facie case. But let me try to do that.

On (1): to indicate that the medieval Arabs faced questions that interest us today, let me quote from Abu Ḥayyān al-Andalusī (1256–1345). In the West he is best known as one of the first linguists to write a textbook of one language in another language. (He wrote in Arabic a textbook of Turkish.) The 15th century encyclopedist As-Suyūṭi [**45**, p. 37] quotes the following remark from him:

> *al-^{c}ajabu mimman yujīzu tarkīban fī luġati min al-luġāti min ġayri ′an yasma^{c}a min ḏālika t-tarkībi naẓā′ir; wa-hal at-tarākību l-^{c}arabiyyati ′illā ka-l-mufradāti l-luġawiyyati? fa-kamā lā yajūzu ′iḥdāṯu lafẓin mufradin, ka-ḏālika lā yajūzu fī t-tarākībi; li-′anna jamī^{c}a ḏālika ′umūrun waḍ^{c}iyyatun, wa-l-′umūru l-waḍ^{c}iyyatu taḥtāju ′ilā samā^{c}in min ′ahli ḏālika l-lisāni, wa-l-farqu bayna ^{c}ilmi n-naḥwi wa-bayna ^{c}ilmi l-luġati ′anna ^{c}ilma n-naḥwi mawḍū^{c}uhu*

> *'umūrun kulliyyatun, wa-mawḍū^{c}u ^{c}ilmi l-luġati 'ašyā'u juz'iyyatun wa-qad ištarakā ma^{c}an fī l-waḍ^{c}i.*

> I find it astonishing that people allow a sentence construction in a language, even when they have never heard a construction like it ⟨before⟩. Are Arabic constructions different from the words in the dictionary? Just as one cannot use newly-invented single words, so one cannot use ⟨newly-invented⟩ constructions. Hence all these matters are subject to convention, and matters of convention require one to follow the practice of the speakers of the relevant language. The difference between syntax and lexicography is that syntax studies universal ⟨rules⟩, whereas lexicography studies items one at a time. These two sciences interlock in ⟨describing⟩ the conventions ⟨on which language is based⟩.

This passage needs to be put in context. Medieval linguists, both western and Arab, generally believed that words get their meaning by an imposed and arbitrary convention; the Arabs called this convention *waḍ^c*, and for the Latins it was *impositio*. If As-Suyūṭi is right (and there is every reason to think that he is), then Abu Ḥayyān is attacking the view that the lexicon of a language contains everything necessary for forming and understanding sentences of the language. As-Suyūṭi refers to earlier writers who had taken this view. One of them is the thirteenth century linguist Ibn Mālik; As-Suyūṭi quotes a passage in which Ibn Mālik argues that since the number of sentences of a language is unlimited, the meanings of sentences could not be fixed by *waḍ^c*. In the passage above, Abu Ḥayyān is responding that it is the practice (*samā^c*) of a language which determines meanings for an unlimited number of possible complex expressions. From this he infers that the conventions forming a language must include not just individual assignments of meanings to words, but also general rules (*'umūrun kulliyyatun*) that govern the meanings of complex expressions in terms of the syntactic constructions giving rise to them.

All this is familiar today, thanks to Husserl who repeated Abu Ḥayyān's observations (independently!) in part 4 of his *Logische Untersuchungen* ([**21**], for example, pp. 316–321) in 1900. One can trace the matter from Husserl into Tarski's truth definition, then into the general notion of compositional semantics; but that is another story. Husserl went on to note that the syntactic rules must be ones that can be applied recursively. Though I believe Abu Ḥayyān's full text does survive somewhere, I do not have access to it, and so I cannot report how much further he develops the theme. But I hope this quotation is enough to establish that he was interested in things that we still debate, and that his contribution still looks penetrating. This is confirms point (1) above. (My thanks to Brendan Gillon for pointing out to me that the Sanskrit linguist Patañjali, commenting on Pānini, had reached some of the same conclusions as Abu Ḥayyān some two thousand years ago.)

The next point to establish is (2), namely that there were medieval Arab semanticists who came at their subject from an angle markedly different from those that we are familiar with. Here my main witness is ᶜAbd al-Qāhir al-Jurjānī. His date of birth is unknown; he spent his whole life in Gorgān, an Iranian town south-east of the Caspian, and died around 1080. (He should not be confused with Aš-Šarīf al-Jurjānī.)

Although Arab writers tend to cite him as the high point of Arab semantics, and his two works *Dalā'il* and *Asrār* are still both readily available in Arabic, his name and works never reached the West until modern times. By contrast the western scholars of the 13th century eagerly read the writings of his slightly older contemporary Ibn Sīnā (Avicenna). The difference is easy to explain. The westerners translated from Arabic only what they could recognise as useful to them: medicine, mathematics, Aristotelian philosophy. Under the head of Aristotelian philosophy they took Al-Fārābī, Ibn Sīnā and Ibn Rushd (Averroes), all of whom wrote on logic. But Jurjānī came from an altogether different world.

In the first place, he trained as a linguist. (Versteegh [**47**] writes: "...the importance of his insights into the structure of language for the development of grammatical theory as such has only now been fully realized.") There is no evidence – or at least none known to me – that he knew Aristotelian logic in any detail. When he mentions logic, he shows no particular hostility to it, unlike some of his linguist contemporaries. But he does quote with approval a line of the poet Al-Buḥturī [**24**, p. 195]:

> *kallaftumūnā ḥudūda manṭiqikum; fī š-ši*^{c}*ri yakfī* ^{c}*an ṣidqihi kaḏbuhu.*
>
> You burden us with your logical definitions; but in poetry
> the lies make the truth superfluous.

Unlike typical Arabic Aristotelians, he does not define by genus and differentiae; his natural style is to explain a notion informally and with examples. I am not aware that he ever mentions Aristotle.

Jurjānī's fame in the Arab world rests on his two texts *Dalā'il* (in full, "Proofs of the miraculous nature of the Quran," but this is a misleading description of the contents) and *Asrār* (in full, "Secrets of eloquence").

Strangely only *Asrār* has been critically edited. Abu Deeb [**1**, p. 21ff] reports that these two books are only a small part of Jurjānī's output – one of his books runs to thirty volumes – but

> The manuscripts of his books on grammar and other subjects are dispersed in various libraries, and despite the great interest in his work nothing is being done, as far as I know, to bring them out in reliable critical editions.

We can only hope that some scholarly editor gets to these works before somebody bombs them in the name of Freedom.

In the two works *Dalā'il* and *Asrār*, Jurjānī's aim is to collect materials and concepts for a general theory of language as a means of communication. He works with a corpus consisting of the Quran and classical Arabic poetry. For him, speakers and writers communicate their thoughts, and thoughts can consist of anything from factual information to moral exhortation or an appreciation of the beauties of nature. He remarks [**24**, p. 196f] that truth must take precedence, but that we greatly limit the creative powers of language if we overlook the ways in which language appeals to the imagination.

Thoughts themselves are not his concern. Rather he wants to know what are the mechanisms that allow a thinker to use this sentence, constructed thus, to express a particular thought. He works as much as possible with concrete examples from his corpus, and he analyses them in sometimes painful detail. But at the same time a systematic theory is constantly present in the background.

For Jurjānī the basic unit of communication is the sentence. A typical remark is [**23**, p. 73]:

> *yaḥtāju fī l-jumlati ᶜilā 'an taḍaᶜahā fī n-nafsi waḍᶜan wāḥidan.*
>
> The need is great for the sentence to be formulated in the psyche as a single act of formulation. (Trans. [**1**, p. 36])

The word *waḍᶜan* (which Abu Deeb renders as "formulation") is the same word that we commented on in Abu Ḥayyān above. Jurjānī accepts that the meanings of the separate words of Arabic are given by a *waḍᶜ* that we have no control over, but he insists that each whole sentence is new in the hands of its creator. He goes on to explain, in one of his favourite analogies, that creating a sentence is like putting bricks together to make a building; the relationships between the bricks contribute more to the outcome than the material of each separate brick.

One reason why Jurjānī stops short at sentences is the limited nature of his corpus. The Quran seems to have come to Muhammad in single verses or small groups of verses, and Muslims commonly treat each verse as a separate revelation. The styles of the classical poets are quite different from the Quran, but there was a strong convention that each couplet should be self-contained; critics reprimanded poets who made grammatical links between two couplets. But Jurjānī does also have a more positive reason for concentrating on sentences: syntax operates at the level of sentences, and the interaction of syntax and semantics is crucial for him.

By concentrating on sentences he cuts himself off from considering the semantics of dialogues. I am not aware that he has much to say about anaphora between sentences, for example. But he has a broad notion of the relevant context of utterance, and it includes the shared beliefs of the speaker and hearer. (Readers of his western counterparts will recall that in them the context tends to shrink to a finger pointing at Socrates when the speaker says "a man." In matters like this Jurjānī feels like a return to the real world.)

Jurjānī suggests in one place that the meaning of an utterance may be determined in two steps: the sentence itself determines a "meaning," and then from this meaning we infer a "meaning of the meaning." In the example he is discussing, the second step makes an appeal to the context. ([**23**, p. 203]).

Jurjānī accepts that a simple word, say "horse," has a meaning independent of any context of use, namely that it applies to horses. He sometimes refers to this as "the meaning of the word," but for emphasis let me call it the *dictionary meaning* of the word. In his view this dictionary meaning represents only a very small part of what the word contributes to the meanings of sentences containing it. Here are some examples of what he has in mind.

(i) The order of words in a sentence serves to emphasise some words and de-emphasise others. What features of the meaning of "kill" come to the foreground when we emphasise "killed" in "*A* killed *B*?" (Much of what he says here may be untranslatable. As Badawi *et al.* [**5**, p. 326f] note, " . . . it is important to emphasize that the topic-comment sentence in Arabic is a basic structure and not the result of any movement, fronting or extraction . . . ")

(ii) More interesting for general theory, the meaning of a word must include the ways in which it can form grammatical links with other words. ([**23**, p. 314])

> *wa'-^{c}lam 'annī lastu 'aqūlu 'inna l-fikra lā yata^{c}allaqu bima^{c}ānī l-kalami l-mufradati 'aṣlan, wa-lākin 'aqūlu 'innahu lā yata^{c}allaqu bihā mujarradatan min ma^{c}ānī n-naḥwi.*
>
> Understand that I am certainly not saying that the understanding does not latch onto the meanings of separate words. What I am saying is that it does not latch onto the meanings of separate words detached from the meaning of (their) syntax.

Western medieval semanticists were of course well aware that a word has possibilities for combining with other words, but they tended to treat these possibilities as an added extra that is clamped onto the meaning. (For a typical example, "*ipse modus significandi aliquo modo est quid additum significationi rei*": Aegidius Romanus, quoted [**34**, p. 124].) Jurjānī's view is rather that the possible relationships are an integral part of the meaning. Today we would say that the meaning includes the argument structure and/or the semantic category. Exactly how Jurjānī's views of the matter differ from, say, those of Frege or the protagonists of Head-Driven Phrase Structure Grammar I cannot say; I hope future generations will have the chance to decide.

The passage just quoted is one of very many places where Jurjānī uses the phrase "meaning of the syntax" (*ma^{c}nā n-naḥwi*).

The word *ma^{c}nā* has broader connotations than "meaning" in English; in other contexts it translates as "function" or "intention." So the phrase *ma^{c}nā n-naḥwi* is not itself a semantic term; syntacticians could use it to talk about various syntactic functions. (I thank Kees Versteegh for alerting me to this.) But some of the most interesting and perplexing questions about the interaction of syntax and semantics are associated with argument structures, theta functions and related items that presumably come under *ma^{c}nā n-naḥwi* , so that Jurjānī's use of the phrase in a semantic context may point to deeper things.

(iii) Jurjānī has a notion of a word being coerced by its context to stand for different things from what it naturally stands for. In general he calls this *majāz*, but when there has to be some point of similarity between the old and new referents he talks of *isti^{c}āra*. His notion of *isti^{c}āra* seems to be very broad. I think it would catch at least the following, besides the examples of poetic metaphor that are his chief concern:

(a) the device in English (and I am told also in Chinese) that allows one to use a noun as a verb, as in "She cheesed the spaghetti";

(b) the similar device in some programming languages, that switches a variable from one data type to another when we apply a function that expects an argument of the second type;

(c) the ways in which temporal expressions shift the references of nouns or verbs within their scope (as in "I met a child," where the past tense of the verb allows the reference of "child" to be the set of children at some past time);

(d) the medieval theory of ampliation (which operates like (c) except that it expands references so as to include possible entities as well as actual ones).

Jurjānī insists that in *isti^{c}āra* the meaning of the relevant word remains its dictionary meaning; the reinterpretation through *isti^{c}āra* acts at sentence level, leaving the dictionary meaning untouched. (It follows that the dictionary meaning is not the reference. As is well known, Frege claimed that in at least some oblique contexts the reference of a word is replaced by its normal *Sinn*; there seems to be some room for disagreement about what – if anything – Frege supposed replaces the *Sinn* in such cases. One possible reading of Jurjānī is that for him the dictionary meaning of a word is its *Sinn*, and this *Sinn* stays the same in all contexts. But I think a closer look will show that Jurjānī's views are more articulated than this.)

Though most of Jurjānī's examples of *majāz* are from poetry, they are not all. He calls attention to *isti^{c}āra* in the Quran, and the Quran itself (*lxix* 41) claims to contain no poetry. Although both the words *majāz* and *isti^{c}āra* are possible translations of Aristotle's *metaphorá* (metaphor), Aristotle limited his account of metaphor to phenomena in poetry. We can read Jurjānī as talking about a much broader range of semantic phenomena.

Jurjānī has a good deal to say about identity of meanings of sentences. He believes that replacing a word by another word with the same dictionary meaning will not alter the meaning of the sentence, even if the word was used metaphorically; so it seems to be his view that the metaphorical uses of a word are determined by its literal uses. On the other hand he believes that replacing a metaphorical description by a literal description always alters the meaning, even in translations between languages; the existence of a metaphor is itself part of the meaning.

He also has a notion of the "form of a meaning" (*ṣūratu l-ma^{c}nā*). Though he says plenty about it in both the *Dalā'il* and the *Asrār*, I have not yet managed to extract a clear idea of what is going on. The form of a meaning is something like a criterion for identifying the meaning; so in particular two meanings with the same form are equal. In modern Arabic the word *ṣūra* allows a range of translations, including "representation" and "photograph."

The fullest account of Jurjānī available in English is Kamal Abu Deeb [**1**], which I have already cited several times above. Another useful account is Larkin [**26**]. Abu Deeb treats Jurjānī mainly as a literary critic; Larkin explores his usefulness for theology (though Jurjānī is clearly not a theologian himself). Briefer accounts that present him more as a semanticist are in Owens [**31**, pp. 249–263] and Versteegh [**48**, Ch. 9]. The influential Syrian poet Adonis reviews Jurjānī's place in Arab poetics in Chapter 2 of his [**2**].

Abu Deeb's book is lovingly written and has many quotations, but it is largely devoted to showing that Jurjānī anticipated various modern western theories. This supports my point (1) but could damage my point (2). If Jurjānī turns out to be a subset of I. A. Richards, then there is not much point in reading Jurjānī; we already have I. A. Richards. But from the relatively small amount of him that I have read, Jurjānī has the electricity of a robust original thinker with a novel viewpoint. My strong hope is that Jurjānī will teach us some new things.

After Jurjānī, semantics became recognised in the Arab world as a field of study, under the name *^{c}ilmu l-ma^{c}ānī* (science of meaning). A number of people made solid contributions. As-Sakkākī's textbook of semantics, from the early 13th century, is available in a German edition [**40**].

Quite independent of Jurjānī, the Arabs contributed a strong strand of research in semantics under the head of *iṣūlu l-fiqh* (jurisprudence). I know very little about this, so I will be brief. According to Islam, the Quran contains guidance for the proper behaviour of individuals and communities. This guidance needs to be extracted by interpretation, and interpretation of a text requires an understanding of how a text communicates its author's intentions. For example an interpreter needs to be able to define the concepts used in the text; to distinguish literal from metaphorical meanings; to distinguish stated meanings from ones that are implied directly by the writer; or implied by the fact that the writer said what he did; to identify indexical expressions and determine what they refer to; and so on. Many scholars contributed to this theory.

This work is religious in just two senses. First, the main intended application is to a religious text (though the theory is developed more generally than this). Second, the religious importance explains the diligence and commitment of a large number of scholars. Neither of these should provide any obstacle for a western secular reader of this literature. Three recent studies are Ali [**3**], Ramić [**36**] and Weiss [**49**].

What about medieval Arabic logic?

The medieval Arabs started serious work on Aristotelian logic rather earlier than their western counterparts, and this interest continued into modern times. Last year the Arabic bookshop in the Charing Cross Road was selling a recent Iranian edition of an Arabic text called *Methods of Logic*, by one Ṣa^{c}in al-Din Ibn Turka al-Iṣfahānī, who died in 1431. It seems to be in no way less sophisticated than the logic texts that were available in Britain in the first decades of the nineteenth century.

Nevertheless one has to search hard to find any western literature on Arabic logic. Nicholas Rescher wrote a number of articles and books (among them [**37, 38**]), and more recently Tony Street [**43, 44**] has begun work on the logic of Al-Fārābī and Ibn Sīnā Versions of Ibn Sīnā's modal logic came to dominate the Arab scene; they have some differences from Aristotle's. Paul Thom's recent book [**46**] makes useful comparisons between the Arabic and the western medieval syllogistic systems.

At the moment it seems unclear that the study of Arabic modal logic has anything to contribute to modern modal logic, and vice versa. But this work of the Arabs clearly deserves to be studied as a significant chapter in the history of logic.

References

1. Kamal Abu Deeb, *Al-Jurjānī's Theory of Poetic Imagery*, Warminster, Aris & Phillips, 1979.
2. Adonis, *An Introduction to Arab Poetics*, London, Saqi Books, 2003.
3. M. M. Y. Ali, *Medieval Islamic Pragmatics: Sunni Legal Theorists' Models of Textual Communication*, Richmond, Curzon, 2000.
4. M. Atiyah, *Mathematics in the 20th Century*, Bull. London Math. Soc. **34** (2002) 1–15.
5. E. Badawi, M. G. Carter, and A. Gully, *Modern Written Arabic: A Comprehensive Grammar*, London, Routledge, 2004.
6. R. Baum, *Dependenzgrammatik*, Tübingen, Max Niemeyer Verlag, 1976.
7. P. Bernays, *Über Hilberts Gedanken zur Grundlegung der Arithmetik*, Jahresb. Deutsch. Mat.-Ver. **31** (1922) 10–19; English transl.: *On Hilbert's thoughts concerning the grounding of arithmetic'*, In: P. Mancosu, *From Brouwer to Hilbert*, Oxford–New York, Oxford Univ. Press, 1988, pp. 215–222.
8. G. Boolos, *A curious inference*, J. Phil. Logic **16** (1987) 1–12; Reprinted in: G. Boolos, *Logic, Logic, and Logic*, Cambridge Mass., Harvard Univ. Press, 1998, pp. 376–382.
9. W. Burleigh, *De Puritate Artis Logicae*, Tractatus Longior, ed. Ph. Boehner, Franc. Inst., St. Bonaventure, New York, 1955.
10. G. Frege, *Begriffsschrift*, Halle, Nebert, 1879.
11. P. Geach, *Reference and Generality: An Examination of some Medieval and Modern Theories*, Cornell Univ. Press, 1962.
12. K. Gödel, *Über die Länge von Beweisen*, Ergebnisse eines mathematischen Kolloquiums **7** (1936) 6; English transl.: *On the length of proofs* in [**13**] pp. 396–399.

13. K. Gödel, *Collected Works I, Publications 1929–1936*, S. Feferman (ed.) et al., Oxford–New York, Oxford Univ. Press, 1986.

14. K. Gödel, *Collected Works II, Publications 1938–1974*, S. Feferman (ed.) et al., Oxford–New York, Oxford Univ. Press, 1990.

15. K. Gödel, *Collected Works III, Unpublished Essays and Lectures*, S. Feferman (ed.) et al., Oxford–New York, Oxford Univ. Press, 1995.

16. K. Gödel, *Collected Works V, Correspondence H–Z,* S. Feferman (ed.) et al., Oxford–New York, Oxford Univ. Press, 2003.

17. E. M. Hammer, *Logic and Visual Information*, Stanford, CSLI Publications, 1995.

18. F. Happé, *Parts and wholes, meaning and minds: central coherence and its relation to theory of mind*, In: S.Baron-Cohen, H. Tager-Flusberg, and D. J. Cohen, Understanding Other Minds: Perspectives from Developmental Cognitive Neuroscience, 2nd edition, Oxford, Oxford Univ. Press, 2000, pp. 203–221.

19. J. van Heijenoort (ed.), *From Frege to Gödel*, Cambridge Mass., Harvard Univ. Press, 1967.

20. W. Hodges, *The geometry of music*, In: J. Fauvel, R. Flood, and R. Wilson (eds.), Music and Mathematics, Oxford, Oxford Univ. Press, 2003, pp. 90–111.

21. E. Husserl, *Logische Untersuchungen II/1*, Tübingen, Max Niemeyer Verlag, 1993.

22. P. N. Johnson-Laird and R. M. J. Byrne, *Deduction*, Hove, Lawrence Erlbaum Associates, 1991.

23. ᶜ Abd al-Qāhir Al-Jurjānī, *Dalā'il al-'I ᶜjāz*, (ed.) Rashīd Riḍā, Cairo, 1946.

24. ᶜAbd al-Qāhir Al-Jurjānī, *Asrār al-Balāġa*, (ed.) H. Ritter, Istanbul, 1954.

25. K. L. Ketner, *Elements of Logic: An Introduction to Peirce's Existential Graphs,* Lubbock, Texas Techn. Univ. Press, 1990.

26. M. Larkin, *The Theology of Meaning: ᶜAbd al-Qāhir al-Jurjānī's Theory of Discourse*, New Haven, Am. Oriental Soc., 1995.

27. D. Marr, *Vision*, New York, W. H. Freeman, 1982.

28. D. Mumford, *Pattern Theory: The Mathematics of Perception*, In: Proc. International Congress of Mathematicians, Beijing 2002, Vol. 1, Beijing, 2002, pp. 401–422.

29. A. Newell, *Unified Theories of Cognition*, Cambridge Mass., Harvard Univ. Press, 1990.

30. M. Oaksford and N. Chater, *Rationality in an Uncertain World: Essays on the Cognitive Science of Human Reasoning*, Hove, Psych. Press, 1998.

31. J. Owens, *The Foundations of Grammar: An Introduction to Medieval Arabic Grammatical Theory*, Amsterdam, John Benjamins, 1988.

32. Ch. S. Peirce, *Reasoning and the Logic of Things: The Cambridge Conference Lectures of 1898*, Harvard Univ. Press, 1992.

33. Ch. S. Peirce, *Existential graphs*, unpublished manuscript from 1909, stored as MS 514 at the Houghton Library, Harvard Univ.; available on the web at www.jfsowa.com/peirce/ms514.htm.

34. J. Pinborg, *Die Entwicklung der Sprachtheorie im Mittelalter*, Münster Westfalen, Aschendorffsche Verlagungsbuchhandlung, 1967.

35. A. Prior, *Modality de dicto and modality de re*, Theoria **18** (1952) 174–180.

36. Š. H. Ramić, *Language and the Interpretation of Islamic Law*, Cambridge, Islamic Texts Society, 2003.

37. N. Rescher, *Studies in the History of Arabic Logic*, Pittsburgh PA, Univ. Pittsburgh Press, 1963.

38. N. Rescher, *Temporal Modalities in Arabic Logic*, Dordrecht, Reidel, 1967.

39. S.-J. Shin, *Reviving the iconicity of beta graphs*, In: M. Anderson et al., Theory and Application of Diagrams, Lect. Notes Artif. Intell. **1889**, Belin, Springer, 2000, pp. 58–73.

40. U. G. Simon, *Mittelalterliche arabische Sprachbetrachtung zwischen Grammatik und Rhetorik: ^{c}ilmu l-ma^{c}ānī bei as-Sakkākī*, Heidelberger Orientverlag, 1993.

41. K. Stenning, *Seeing Reason*, Oxford, Oxford Univ. Press, 2003.

42. K. Stenning and P. Yule, *Image and language in human reasoning: a syllogistic illustration*, Cogn. Psych. **34** (1997) 109–159.

43. T. Street, *An outline of Avicenna's syllogistic*, Arch. Gesch. Phil. **84** (2002) 129–160.

44. T. Street, *Logic*, In: P. Adamson and R. C. Taylor (eds.), The Cambridge Companion to Arabic Philosophy, Cambridge, Cambridge Univ. Press, 2005, pp. 247–265.

45. Jalal ad-Dın as-Suyuṭı, *al-Muzhir fi^c Ulum al-Lugah wa-Anwa^c iha*, Dar al-Kotob al-Ilmiyah, Beirut 1998 (original c. 1500).

46. P. Thom, *Medieval Modal Systems*, Aldershot, Ashgate, 2003.

47. K. Versteegh, *Grammar and Rhetoric, Ǧurǧānī on the verbs of admiration*, Jerusalem Stud. Arabic Islam **15** (1992). 113–133.

48. K. Versteegh, *Landmarks in Linguistic Thought III: The Arabic Linguistic Tradition*, London, Routledge, 1997.

49. B. G. Weiss, *The Spirit of Islamic Law*, Athens Georgia, Univ. Georgia Press, 1998.

50. W. H. Woodin, *The Axiom of Determinacy, Forcing Axioms, and the Nonstationary Ideal*, Berlin, De Gruyter, 1999.

Applied Logic:
A Manifesto

Lawrence S. Moss

Indiana University
Bloomington, USA

This paper presents applied logic as a general research area, situating it in the broader intellectual world. It proposes a characterization of applied logic and attempts to say why the subject is interesting. It discusses the relation of applied logic with other trends in logic, computer science, and mathematics. Rather than present any technical results, it aims for a "big picture"' view of this emerging subject.

1. What is Applied Logic?

My main purposes in this essay are to introduce applied logic as a research area, to situate it in a broader context, to make the

Mathematical Problems from Applied Logics. New Logics for the XXIst Century.
Edited by Dov M Gabbay *et al.*/ International Mathematical Series, Springer, 2005

case that it is a significant and worthwhile enterprise, and to detail some of its research areas. These are my main overt purposes. But my "covert" purpose is to write something that might open new doors for young students with interests in logic. In my younger days I remember the excitement I felt from subjects at the borders of mathematics, computer science, and linguistics. It was not until many years later that I began to think more explicitly about the "politics" of what I and many others have been doing. I think it would have helped me to ponder a manifesto or two along the way, even in my high school or college days. So my hope about this article is that somewhere, sometime, somebody will pick it up and . . ..

While many people know something about logic, I take it that the idea of applied logic will be unfamiliar to all readers. This is because it does not exist in the same institutional sense as other fields, and only a few books have the words " applied logic" in their title. There really is no consensus on what "applied logic" means, so in effect, I am making a proposal here. To explain matters, I'll need to go into a fair amount of detail about logic, and also about mathematics, computer science, philosophy, and other fields. But to keep things short, most of my discussion of these matters will be offered with only minimal support.

The main points

The reader can find most of the main points in the section headings and **boldface** lead phrases. Many of these are slogans th at I present in a deliberately provocative way.

First things first, we should say what the subject is about. It is always difficult to define fields, but I take applied logic to be defined and characterized in the following ways:

(1) It is the application of logical and mathematical methods to foundational matters that go beyond the traditional areas of mathematical logic. The central domain of application at the

present time is computer science, but it also has significant applications in other fields.

(2) It also is extension of the boundaries of logic to include *change*, *uncertainty*, *fallibility*, and *community*. Far from being the study of matters which are absolutely black or white and never change, and which exist in the mind of a single person, applied logic aims to study communication via brushstrokes of gray. In this way, it is a reconstitution of the study of *foundations*.

(3) Its ultimate interest is a concern with human reasoning, so it will ultimately lead to a rapprochement with psychology, artificial intelligence, and cognitive science. But even before this happens, the development of *tractable logical systems* are the most conspicuous *applications* of logic in many fields.

(4) Applied logic is an interdisciplinary field, and this has its own set of difficulties and opportunities.

Most of this essay is a discussion of these points. But in the spirit of a manifesto, I do want to make some grandiose claims: applied logic is the most vigorous branch of logic. And if one is interested in current research on the topics that motivate and animate logic in the first place – the concepts of formal reasoning, truth, meaning, paradox, proof, and computation – then applied logic is the place to look.

2. Mathematics and Logic, but Different from Mathematical Logic

> Logic is the study of reasoning; and mathematical logic is the study of the type of reasoning done by mathematicians.
>
> *Joseph R. Shoenfield*
>
> ***Mathematical Logic***, 1967 (cf. [**1**, p. 1])

I know that most readers will recognize the split in logic between "mathematical logic" and "philosophical logic." At the same time, they may be surprised to hear that the difference is not mainly about whether the subject is "mathematical:" philosophical logic has technical sides that use and inspire mathematics. The difference has to do more with what the targets of the studies are. Mathematical logic is a much-better-defined area, and so it makes sense to discuss it first. The quote above is the first sentence from one of the standard textbooks on mathematical logic.

I did not re-read Shoenfield's entire book, but I doubt very much that the word "reasoning" appears many times after the first sentence. Going further, I do not think we can take mathematical logic seriously as a study of mathematical reasoning. There are numerous reasons, and they all echo broader philosophical claims. First of all, one would think that in studying "mathematical reasoning," one would be interested in "reasoning" in other areas. That is, one would expect some sort of engagement with psychology. Yet ever since Frege if not earlier, mathematical logic has rejected the idea of an engagement with psychology at any level. Even putting this aside, in examining the considerable body of work in mathematical logic, we find nothing of key aspects of "flesh-and-blood" mathematical reasoning, such as the use of symbols, diagrams, formulas; hunches, nothing about evidence, and mistakes; nothing about why some types of mathematical reasoning are more interesting than other types; nothing about different mathematical fields and what they have or do not have in common; nothing about how the faculty of mathematical reasoning is acquired; and again, nothing at all about how it is related to any other human faculty. One would think that a subject devoted to mathematical reasoning would in part be interested in *all* of these issues.

2.1. Mathematical Logic and Mathematics

If mathematical logic is not the study of mathematical reasoning, what is it? Mathematical logic has three aspects:

(1) It is a *foundational* discipline which studies *an idealization of* mathematical reasoning, the reasoning done by perfect beings with no resource limitations who reason in a way captured by the axiomatic method. It is mainly concerned with idealizations of the concepts of *truth*, *proof*, *computation*, and *infinity.* It is traditionally divided into four areas that correspond to these: model theory, proof theory, recursion theory, and set theory.

(2) It also deals with a host of theory-internal questions. Each area of mathematical logic is now an active branch of mathematics, and like any branch of mathematics, there are many questions whose primary interest will be to those inside.

(3) Finally, it is concerned with applications to other areas of mathematics.

I think there are reasons not to be happy with the received view of mathematical logic as “the foundations” of mathematics, but I will not go into that here. I think it is fair to say that this foundational contribution of mathematical logic is what is usually meant by speaking of mathematical logic as a foundation of mathematics, and that despite my (and many others’) quibbles on whether it is *the ultimate* foundation, the foundational achievements of logic are of permanent interest.

For many years, the theory-internal questions of mathematical logic were the most important parts of the subject. There are whole fields of active study, such as the theory of recursively enumerable degrees, large cardinals, the fine structure of the constructible universe, and infinitary logic, which are mathematically interesting but really quite far in motivation from any foundational matters or from the study of mathematical reasoning. These questions are the real aim of Shoenfield’s book, for example. Theory-internal questions constitute most of any active field, and indeed most of the articles in the *Journal of Symbolic Logic* would fall in this category.

The dream of many logicians has been to apply logic to settle serious questions of mathematics. For many years, this dream went mostly unrealized. There were some exceptions, such as Tarski's work on real closed fields, in some of the celebrated results of set theory that had implications for the foundational questions of analysis; and indeed in the whole field of non-standard analysis. And in very recent times we see significant applications of mathematical logic in mathematics, so much so that perhaps mathematicians outside will get interested in one or another of the branches of mathematical logic. Two areas where this happens are model-theoretic algebra, with its connections to areas like arithmetic algebraic geometry; and descriptive set theory, with its connections to topology, analysis, functional analysis, and other areas.

It seems fair to say that the main thrust of mathematical logic is not the foundational contribution in point (1): this is saved for textbooks and introductory courses, or it comes up only when justifying the subject. The most valued work is in (2) answering hard technical questions about areas that have arisen because of the subject itself, and (3) contributions to more central areas of mathematics. Looking at conference programs and invited lectures, this latter thrust seems on the rise and destined to become the most important one for mathematical logic. Certainly when I talk to young people interested in the mathematical logic, I encourage them to aim for area (3).

2.2. Where applied logic differs

I think that all of the contributions in (1)–(3) above are interesting and good. But I do not think that they are the only interesting things to do in logic. My main purpose in this essay is to present an alternative agenda. It will extend the foundational impulse which motivate d mathematical logic in the first place, it will have its own internal questions, and it will be a field with applications

– indeed, as the name suggests, the applications are going to be at the center of the subject.

Once again, the closest direction for applied logic is the one that mathematical logic seems to be finished with, the foundational contribution clarifying some of the central aspects of reasoning in mathematical and formal contexts. But this idealization is a two-edged sword. Usually it is presented as an advantage, because *negative results* about the idealization imply negative results about the real thing. For example, it was shown early on that there are natural things that one might want to write a computer program to do which simply cannot be done by an idealized computer. (One example: write a program which looks at other programs and tells for sure whether or not the input program will go into an infinite loop.) Since this cannot be done on even an idealized computer, it follows that it cannot be done on a real computer either. This illustrates how when considering negative results, it might be fine to work with idealizations.

However, the other side of the sword is that sometimes the idealization might be so questionable as to make the inference from the idealization problematic. An example here concerns the most celebrated result of mathematical logic, Gödel's Incompleteness Theorem. There are many people who believe that this result implies that human beings are not computers. This may or may not be the case, but I think it is a mistake to think that the Incompleteness Theorem gives us conclusive evidence, or even suggestive evidence. The inference from the technical result to the philosophical point is questionable precisely because there is no reason to take an understanding of the Incompleteness Theorem, to be a good idealization of intelligence or what it means to be a person.

Returning to my topic, questions about *real* reasoning, or about aspects of it that we can model mathematically, are going to be important for applied logic. In this sense, applied logic is carrying forward the program of applying mathematics to the human world. This is the crux of the difference. For applied logic, mathematics will be a tool to use. And although I think that blends of applied logic and cognitive science will ultimately tell us more about mathematics than mathematical logic has as yet told us, I do not take this to be the one and only goal of applied logic.

2.3. Applied mathematics is good mathematics

Applied mathematics is bad mathematics.

Paul Halmos
In: ***Mathematics Tomorrow***, 1981 (cf. [**2**])

This enterprise of applied logic builds on and uses all the results of mathematical logic, but it is not aimed back at mathematics the way mathematical logic is. My argument in this section is that applied logic should be recognized as an area of applied mathematics.

Halmos' quote above is the title of his paper on the subject of the relation of pure and applied mathematics, one of the few papers devoted exclusively to that topic. He writes, that the concept of *motion* "plays the central role in the classical conception of what applied mathematics is all about." And in a passage comparing pure and applied mathematics, he states:

> The motivation of the applied mathematician is to understand the world and perhaps to change it; the requisite attitude (or, in any event, a customary one) is sharp focus (keep your eye on the problem); the techniques are chosen for and judged by their effectiveness (the end is what's important); and the satisfaction comes from the way the answers checks against reality and can be used to make predictions. The motivation of the pure mathematician is frequently just curiosity; the attitude is more that of a wide-angle lens than a telescopic one (is there a more interesting and perhaps deeper question nearby?); the choice of technique is dictated at

> least in part by its harmony with the context (half the fun is getting there); and the satisfaction comes from the way the answer illuminates unsuspected connections between ideas that had once seemed to be far apart.
>
> *P. Halmos* [**2**]

Halmos makes it clear that he values applied mathematics, but as his overall title indicates, he is partial to pure mathematics above all else.

I think that applied logic could well be considered as applied mathematics. It is not based on the concept of motion, but as I mentioned above regarding *change*, some evidently related concepts are at the heart of it. Applied logic is about understanding the world, and to a very limited extent, changing it. On the other hand, it is more like a wide-angle lens than a telescope, so the analogy is not perfect. But overall, based on what Halmos writes (as quoted above and elsewhere in the article), I think it is fair to say that applied logic is closer in spirit to applied mathematics than pure mathematics.

I must add that usually applied mathematics is not taken to include discrete subjects. I say "usually" here; sometimes discrete math topics are included in applied mathematics. But one need only look at departments of Applied Mathematics in the USA to see my point. (Incidentally, it seems clear that *theoretical computer science* fits Halmos' criteria for applied mathematics rather well.) And most applied mathematicians would be surprised, I think, to consider any branch of logic in the same category. Conversely, logic is rarely seen as an applied subject. So my entire discussion is intended to make a point that is controversial.

2.4. Applied logic is applied mathematics

> Throughout the centuries the great themes of pure mathematics, which were conceived without thought of usefulness, have been transformed to essential tools for scientific

> understanding. ... this transformation is now happening to mathematical logic, and ... a subject of applied logic is emerging akin in its range and power to classical applied mathematics.
>
> *Anil Nerode*
> In: ***The Merging of Disciplines: New Directions in Pure, Applied and Computational Mathematics***, 1986.
> (cf. [**3**])

I believe that Nerode is right: applied topics in logic are in the process of coalescing around a set of questions and research agendas that will constitute a coherent subject matter. I would like to think of this as applied mathematics in the same kind of way that other areas are applied mathematics: it certainly involves doing new mathematics, and doing interesting mathematics at that; but the choice of problems and viewpoints is driven primarily by modeling phenomena which exist out in the big world.

Nerode's article is the one of the few I can point to that makes the case for applied logic. His paper is mostly a compendium of examples and does not attempt to systematically present applied logic. He is most interested in applications to computer science, and I'll have more to say about this in Section 4 below. But applied logic is a very interdisciplinary study, with additional contributions and applications from artificial intelligence, cognitive science, economics, and linguistics, and with fundamental interactions with computer science, mathematics and philosophy. Before we get to that, it would be good to contrast applied logic with its much better-known cousin, mathematical logic.

3. Applied Philosophical Logic

Traditionally, logic has been divided into "mathematical logic" and "philosophical logic." At most institutions in the US which feature significant activity in logic, this division is a useful one. Few people bridge the gap.

I take philosophical logic to be the continuing foundational study that I mentioned above in connection with mathematical logic. In addition, I take it to be the home of formal, mathematical studies of all of the important concepts which somehow did not make it into the purview of mathematical logic. E. J. Lowe[1] holds that the subject's main areas are

(1) theories of reference,
(2) theories of truth,
(3) problems of logical analysis (for example, the problems of analyzing conditional and existential statements),
(4) problems of modality (that is, problems concerned with necessity, possibility and related notions), and
(5) problems of rational argument.

I contrast philosophical logic with *philosophy of logic*, and by this I have in mind more the relation of logic to more central branches of philosophy such as epistemology and metaphysics. All of my remarks in this essay are about philosophical logic rather than philosophy of logic.

My feeling is that the pure/applied continuum and the mathematical/philosophical continuum are somewhat orthogonal. Specifically, there are many applied subjects that are applications of topics originating in philosophical logic. These are mainly in Lowe's areas (4) and (5) above. I'll have more to say about one such topic from (4), epistemic logic, in Section 5.2 below. Overall, I think that philosophical logic is the source of many problems and research connections with applied logic. This is mainly because whole areas of philosophical logic have been given new life by connections to computer science. I'll return to this point after discussing the relation of computer science with applied logic.

[1] Lowe's survey article is available at `http://www.dur.ac.uk/~dfl0www/modules/logic/PHILLOG.HTM`.

I also have an overall feeling about the foundational problems that motivated logic in the last century. To be blunt, we're were a different world in 2000 than we were in 1900. Many of the questions that seemed so pressing back then have lost some of their appeal. Very few people today want to fight the old fights about the Axiom of Choice, or about predicativity, or a host of other is sues. Instead, we have a host of new questions, and new areas that at this point are in need of mathematical insight: what would models of computation look like which are appropriate to the brain as we know it? What is information? What are the best ways to represent the fallible, uncertain, and sometimes-incorrect knowledge that we all have? What are the most efficient ways for a computer to manage large amounts of changing information? What are we to make of the failure of logic to be a "magic bullet" in artificial intelligence?

3.1. Applied philosophical logic = theoretical AI

The slogan here is perhaps a bit of an overstatement, but the point is that work on the theoretical questions in artificial intelligence often looks back at earlier discussions in philosophical logic. One area where this happens is in the study of knowledge; I'll say more on this below. Another is the study of *context*: how is it to be represented, and what role does it play in reasoning? If one wants to build a robot and make it *rational*, then the hard problem of deciding what rationality means will lead back to the parallel philosophical literature.

4. What Does Computer Science Have to Do with It?

Applied logic is the most vibrant and relevant form of contemporary logic. It is primarily the study of logic that is relevant to, and in symbiosis with, computer science. So it is worthwhile at this point to go into detail about the relation of applied logic to computer science.

4.1. Logic is the Calculus of computer science

Logic is a surprisingly prevalent tool in Computer Science. NSF's Directorate for Computer and Information Science and Engineering (CISE) had a workshop[2] two years ago called "The Unreasonable Effectiveness of Logic in Computer Science." The point is that some areas of logic get used again and again in formulating the main notions of computer science. Relational databases are close to first-order relational structures, and model theory is therefore an appropriate tool. Programming language "types" are best understood with the help of much older tools from logic like typed lambda calculi. The study of abstract data types is quite close to universal algebra, so equational logic is prominent there. Verifying that a computer program or a piece of hardware does what it was designed to do requires formalization, and this formalization inevitably uses the tools of logic. Interestingly enough, the tools in verification often come originally from areas of logic where time and change are studied, so they ultimately derive from philosophical logic. Turning to artificial intelligence (AI), there was a time when AI was taken to be one big application of logic. The celebrated P=NP problem, the problem which has been called "computer science's gift to mathematics," is often cast as a problem in logic: is there a polynomial time algorithm to determine whether a boolean formula is satisfiable or not? And all the other main problems of complexity theory have logical versions.

The widespread use of logic in computer science goes back about twenty or thirty years. Although logic is not seen as a specific area of computer science the way it is in mathematics and philosophy, there are those who believe that logic is even more important in computer science than it is in mathematics, that large parts of computer science are applications of two parent disciplines: electrical engineering and logic. The slogan here is:

[2] There is also a nice survey article by essentially the same people as the presenters of the CISE workshop: Joseph Y. Halpern, Robert Harper, Neil Immerman, Phokion G. Kolaitis, Moshe Y. Vardi, and Victor Vianu (cf., [**4**]).

Logic is the Calculus of computer science

4.2. Computer science motivates logic

Just as physics was a great spur to the development of applied mathematics, so computer science will be a motivating field for applied logic. It is not surprising that nearly all of the applications mentioned above were not applications of existing theory. For the most part, the applications called out new theory, new mathematics. And this new theory is developing at a rapid pace. Here are the examples again, with mention of the new work that has come up: Database theory has given us *finite model theory* and connections of logic and probability theory. The issue of types in programming languages is now of keen interest in category theory, and to follow current developments one really needs a good background in that subject. Programming language semantics has also been a motivating force in current developments in proof theory such as linear logic. The universal algebra/computer science border has given many new questions: for example, the classical questions of universal algebra usually are asked without reference to complexity. Verification has given us numerous flavors of dynamic logic, and I will return to this in Section 5.2 below. It also has revitalized higher-order logic. AI has lead to non-monotonic logic, to blends of logic and probability, to automated theorem proving and knowledge-based programming. And complexity theory has lead to descriptive complexity theory and to learning theory.

Why has computer science been so powerful of a driving motor for logical applications and for developments inside of logic? Here are two related reasons: First, the whole tenor of computer science is toward applications that actually run. Traditional systems of logic are the right place to look to find the appropriate theory, at least at first glance. But at the same time, the main body of technical work in logic is based on idealizations: complexity does not matter, mistakes are unimportant, everything is relevant to everything else, the world may be modeled as an unchanging totality of facts, etc. Each of these has to be abandoned or at least seriously modified to make real progress in the fields I have mentioned above. The closer we get to the human world, the more we need to re-think things. And it is this reconsideration of old idealizations which has lead to many new developments in logic.

4.3. Going beyond the traditional boundaries of logic

As I mentioned above, there was a time when the logical paradigm in AI was the leading one. This is no longer the case. In AI itself, even those who do believe logic has a key role often resort to new varieties of logic, such as default logic and non-monotonic logic. These differ from standard logical systems because one can "take back" parts of arguments, or jump to conclusions (in some sense). Going further, *probabilistic methods* are now recognized as critical, not only in AI but also in fields of interest for applied logic, such as cognitive science and computational linguistics. There is a recognition that uncertainty and randomness are not flaws; they are design tools. So connectionist modeling is now widespread in cognitive science. Dealing with uncertainty is a major research area in AI. Statistical methods in natural language processing are widely believed to outperform deterministic methods.

My point here is to suggest that a coming key area for applied logic is going to be some sort of rapprochement with all of the mathematical areas that turn out to be important in modeling on the same set of phenomena. This is especially important for cognitive science, and I am encouraged by some very recent developments that relate connectionist models to non-monotonic logics.

A postscript: one of the interesting developments in recent years is the degree to which the "declarative" and "stochastic" sides *do* turn out to talk to each other. Perhaps this is because both are getting something right. In any case, I take the project for applied logic *not* to be the one of defending declarative frameworks or making improvements on them, but rather the one of accepting the points of the "other" side and working towards a stronger synthesis.

5. Other Case Studies

I have mentioned that the primary application area of applied logic is to computer science. The applications there are so well-developed that people who work on them might not even be interested in the foundational questions that I take to be an important part of applied logic. In this section, I present case studies and research questions in applied logic whose main motivation is areas outside of computer science.

5.1. Neural networks and non-monotonic logic

People investigating learning, categorization, memory, and other areas of cognitive science often use *neural networks.* There are many different kinds of neural nets, but they share the features of processing information numerically, and of doing so in a distributed way. This contrasts with serial, symbolic processing that is more natural for the computational models like Turing machines. Those kinds of computational models seem better-suited to model activities like logical reasoning. The name "neural" comes from the view that the brain, too, is a neural network. As it happens, it is much easier to use neural network models to "learn" than to give an account of learning in general. It is easier to use the models than it is to understand what they are doing.

The need for some synthesis between the symbolic/serial and numerical/parallel models has been felt by researchers for quite a while. Two people who have done work on this include Peter

Gärdenfors and Reinhard Blutner. Their overall idea is view symbolic computation as a higher-level description of what is going on in connectionist models. In other words, we would like to explain emergence of abstract symbols from subsymbolic data such as weights in a trained neural network. The logical tool employed is *non-monotonic logic*, the same subject I mentioned above in connection with logic in AI.

5.2. Dynamic epistemic logic

Modal logic is the study of logical systems which involve some qualification of the concept of truth. For examples, one studies the differences between "true," "possibly true," and "necessarily true." Epistemic logic is a branch of modal logic that deals particularly with concepts having to do with *knowledge* and *belief*. Other branches of modal logic study concepts like *before* and *after* (temporal logic), or *obligation* and *permission* (deontic logic). One can sense that the all of these areas are going to be of interest not only in philosophy but also in cognitive science and artificial intelligence.

Incidentally, modal logic in general and epistemic logic in particular are subject conspicuously missing from mathematical logic. I think this is all unfortunate, because modal logic is one of the most applicable fields of logic, and also because it has connection with many areas of mathematics. One can find to papers where modal logic is mixed with general topology, dynamical systems, universal algebra, and boolean algebra.

It probably would have surprised the early workers in the subject that their ideas would be useful in computer science, but this has indeed happened. Epistemic logic overall is one of the areas of philosophical logic that has benefited a lot from computer science, and I will go into this below. Just the same, it might have surprised the computer scientists of a few years back that economists have become interested in the subject. Overall, it is today a richer field than ever before.

The dynamic turn

I mentioned in connection with applied mathematics Halmos' point that *motion* "plays the central role in the classical conception of what applied mathematics is all about." Interestingly enough, a parallel to motion is playing a central role in many areas of applied logic. This parallel notion is *change*. Models incorporating change are prevalent in computer science, since a computational process is one in which values (of variables, or registers) change. So the logics of computer science are generally logics of change. These themselves are modal logics; temporal logics are the most prevalent kind in applied work, but others are as well. As it happens, ideas from dynamic logic have made their way back into epistemic logic. This happens primarily in connection with the issue of *common knowledge*, a matter which I would not discuss in detail.

Some work in natural language semantics now uses game theory as an overall mathematical background. Then one constructs models for use in pragmatics, for example, essentially following the slogan going back to Wittgenstein that linguistic activities are moves in a game. Tools and ideas epistemic logic might therefore turn out to be useful in modeling and studying a wide range of phenomena: conversations, buying and selling, and security protocols.

There are now areas of epistemic logic where one is concerned with the modeling of epistemic actions and the change of knowledge that comes from them. One proposes and studies models for notions like *public announcements*, *cheating in games*, *wire-tapping*, etc. The models use ideas from both epistemic and dynamic logic, and their study involves a lot of non-trivial mathematics. The overall idea of dynamic epistemic logic is to get a good mathematical treatment of all of the notions I mentioned above. Later, one would like to see how the work plays out in areas like the modeling of conversation between people, or in models for computer security.

Connections with economics/game theory

Another area where logic might shed light on matters in the social world is in economics and game theory. Here there are basic questions concerned with how agents interact, what rationality and belief come to, and how communication and action change the world. As it happens, the tools of modal logic that I mentioned above play a role in this work. A further connection to computer science comes in when one looks at *auction theory* or *mechanism design*. One interesting source to look at "Logic for Mechanism Design – A Manifesto" by Marc Pauly and Michael Woolridge.[3] Their proposal is to use a logic called *alternating-time temporal logic* as a language to formally define social procedures such as voting systems, auctions, and algorithms for fair division. The parallel is with the uses of logic in defining (specifying) computer programs; there is also a definite sense in which social algorithms are a "many-agent" version of computer programs. Perhaps the fullest expression of work in this direction is Rohit Parikh's program of *social software*. What all of this suggests to me is that applied logic is poised to dramatically enlarge the traditional scope of logic, and that in some sense it is already doing just that.

5.3. Linguistics, logic, and mathematics

> Precisely constructed models for linguistic structure can play an important role, both positive and negative, in the process of discovery itself. By pushing a precise but inadequate formulation to an unacceptable conclusion, we can often expose the exact source of this inadequacy and, consequently, gain a deeper understanding of the linguistic data. More positively, a formalized theory may automatically provide solutions for many problems other than those for which it was explicitly designed. I think that some of those linguists who have questioned the value of precise and technical development of linguistic theory have

[3] Cf. `www.csc.liv.ac.uk/ mjw/pubs/gtdt2003.pdf`

> failed to recognize the productive potential in the method of rigorously stating a proposed theory and applying it strictly to linguistic material with no attempt to avoid unacceptable conclusions by *ad hoc* adjustments or loose formulation.
>
> *Noam Chomsky*
>
> ***Syntactic Structures***, 1957 (cf. [**5**, p. 5])

For some, mathematics is a fortress that should remain on guard against contact with the world. For others, it is part of the light of the mind, the light by which we understand the world. In this section, I am concerned with the relation of linguistics and mathematics. The fields of linguistics that I have most contact with are *syntax* (the study of phrase structure and sentence structure) and *semantics*; these are also the branches of linguistics that are closest to logic and theoretical computer science.

The most important linguist in modern times is Noam Chomsky. The quote above makes the case why mathematics has something to say to linguistics. This point is immediately appealing to people like me who value the use of mathematics in the social sciences, and I remember being inspired in this direction many years ago.

Unfortunately, the story does not end here. For various reasons, Chomsky in later work has *reversed* his position. To this day, he is not in mathematical results concerning the grammars he proposes; and the majority of syntacticians follow suit. The set of people interested in the mathematical study of the syntactic formalisms is fairly small. It almost seems like they are a small remnant of candle-holders in a crowd that would rather walk in the dark. However, I am optimistic about the long-term prospects of formal work in linguistics. This is partly because I am optimistic about the use of mathematics in all fields, and partly because I think the results we already have are interesting enough to pursue further. In any case, the history of the interaction of mathematics and linguistics is an interesting one, and so I will discuss a few aspects of it. As we will see, there are echoes of the pure math/applied math split here, too.

Remarks on syntax and semantics in computer science and linguistics

Overall, I think that work on syntax and semantic in linguistics is harder than parallel work in computer science. The main reason for this is that computer languages are human creations, and so we know what they are supposed to be like and what things are supposed to mean. Syntactic problems concerning computer languages are mostly non-existent. (A partial exception: in typed languages, users often make type errors that are difficult to trace. A few people, including a recent IU Ph.D. student, have considered the problem of getting usable systems to help with type errors, systems that are based entirely on the syntax.) Semantic problems do exist and are highly non-trivial. But looking at natural language, everything is harder. We have no direct evidence for the existence of any of the traditional syntactic categories (such as noun phrase, verb phrase, etc.) These are all theoretical constructs that differ from framework to framework. In a sense, when we look at sentences in a natural language, the structures we posit are our own; they are not self-evident, or found in a manual, or in any way obvious. With semantics, things are even harder. Psycholinguistic evidence is hard to come by, and even if we had more of it I am not sure it would always be useful in semantic theory.

One sees many instances of concepts from theoretical computer science filtering back to linguistics. The main reason for this is that there is so much more activity in computer science than linguistics, and so much more sophisticated mathematics. I think it is fair to say that nearly many of the technical innovations in linguistics (especially syntax) in the past 25 years are borrowings from theoretical computer science. (The main exception here is that the main concepts of formal language theory were formulated first by Chomsky for linguistics (where they are today largely forgotten), and these quickly became a “classical” area of theoretical computer science.) This includes all of the interaction of applied logic and linguistics.

Logic in computational linguistics

Computational linguistics is concerned with all matters related to the project of getting a computer to process language. This includes speech recognition, parsing, syntax, semantics, pragmatics. I think computational linguistics bears the same relation to mainstream linguistics that applied math bears to pure math. In fact, going back to the quote from Halmos on page 324, it seems that the description of applied math fits computational linguistics even better theoretical computer science, and for that matter even better than applied math itself!

It is no surprise to find logical aspects of computational linguistics: as we have seen, this is bound to happen with all serious work with computers that requires a theoretical approach. Here are some of the many ways logic enters in: various proposals for syntax use systems like *linear logic* (an outgrowth of proof theory) or *logic programming* (coming from Horn clause logic, and now the basis of the language Prolog). An area common to logic and linguistics is *categorial grammar*, based on work by Joachim Lambek from the 1950’s and 60’s. Yet another concerns precisely the latest work in the Chomskyan tradition, the minimalist program. Here, despite the fact that Chomsky did not want to consider formalization, people have done exactly that. They’ve characterized the languages which are generable by “MG grammars” in terms of classical formal language theory. This interesting result came via a lot of other work, some of it involved formal language theory, and some of it involved proof-theoretic grammars related to Lambek’s categorial grammars.

A potentially far-reaching application of logic in the study of the syntactic formalisms is the program of *model-theoretic syntax* initiated by James Rogers in the 1980's. Rogers applied seminal work in decidability coming from mathematical logic to the syntactic formalisms. (So we see again that applied logic builds on core areas of mathematical logic.) In more detail, the basic idea here is to translate syntactic formalisms which are used by linguists into a single language. That language happens to be a version of *monadic second-order logic*. It so happens that Michael Rabin had shown the decidability of the monadic second order logic on trees, and others had made the connection between definability in that logical language and notions coming from formal language theory. So Rogers' proposal applies one of the most traditional measures of complexity coming from logic, that of asking for a given notion and a given language whether the notion is definable in the language.

Semantics

Semantics as a formal discipline owes a tremendous amount to logic. Richard Montague in the 1960's and 70's pioneered the adaptation of semantical methods from logic to fragments of natural language. This was the first, and therefore the most decisive development, in the field. Logic is at the heart of the fairly new discipline of *computational semantics*, too.

Beyond logic

As it happens, even the best ideas from logic and classical linguistics are not in practice as accurate or as fast for natural language processing as some very simple heuristics coming from statistics. One gets even better results with more sophisticated models, and perhaps the best models to date come from Markov branching processes and random fields. These models work in the sense of being, say, 94.54% correct for their tasks. But they work on the basis of correlations rather than principled explanations. (So again we have Halmos' wide-angle and telescopic lenses.) Going further, it is quite an interesting matter to *mix* declarative (logical) and statistical methods in linguistics (and in vision and planning as well). This relates to my point earlier that applied logic will need to go beyond the traditional borders of logic. In this case, what is needed is a principled blend of logic and statistics.

5.4. But is it dead?

My proposal calls for the extension of logic to incorporate a number of aspects that traditionally are missing: uncertainty, social aspects, context, and so on. One reaction to this is that doing so would lead to the *end of logic*. I am reminded of Keith Devlin's book **Goodbye, Descartes: The End of Logic and the Search for a New Cosmology of Mind**, 1997 (cf. [**6**]).

Devlin's book is a popular account of the history of logic and the related areas that I am dealing with in this essay. His book closes with the discussion pertinent to the subtitle, that what we are seeing is an end of logic – especially logic conceived of as requiring the duality, the divorce of mind and body. It also makes the case for "soft mathematics" and has proposals on what this means. Much of the points in the book are consonant with my message here.

But my feeling is that everything that would lead one to declare a death of logic could just as easily be seen the opposite way: rather than dead, the subject is rather only in its infancy (strange as that sounds for one of the most classical subjects). One would like to think that in a hundred years, or a thousand, that trends in applied logic will give rise to a subject of permanent human importance, a revitalization of the classical subject of logic that takes into account the many things missing from the subject as we now know it.

6. Being as catholic as Possible

> Main Entry: catholic
> Etymology: . . . from Greek katholikos universal, general, from katholou in general,
> . . . 2 : COMPREHENSIVE, UNIVERSAL; especially: broad in sympathies, tastes, or interests
>
> ***Webster's Collegiate Dictionary***

It should be clear that applied logic is multidisciplinary: since it is outward looking, it thrives on interactions with many other fields. There are institutional challenges for applied logic, as there are with any interdisciplinary endeavor. What I want to do here is to make a few comments on those challenges, and what can be done to help.

The Pigeonhole principle is for the birds

When mathematicians speak of the Pigeonhole Principle, they have in mind a fundamental fact: if you have more pigeons than pigeonholes, then when you put the birds in the holes, at least two are going to end up in the same place. This is not the principle that I object to. Instead, I feel hampered by the principle that *people* should be pigeonholed according to what they study, or by their academic departments. For applied logic as I am thinking of it, departments are really not of great value. We should try to see beyond the boundaries of disciplines.

One of the success stories in the field has been the European Summer School in Logic, Language, and Information. This annual school is *the* showcase for many areas of work, including those that I am calling applied logic. Its interdisciplinary spirit is inspiring, and it would seem to be important to emulate and strengthen it.

But even with modest successes, we have problematic points. The biggest paradox with an interdisciplinary field concerns exactly the phrase *inter disciplinary field.* One must value, on the one hand, contributions to outside areas, and on the other hand, solutions to problems that come internally. And the other challenge with interdisciplinary work concerns the matter of making connections between fields. In many cases, doing this is not appreciated, and at other times it even feels like a poor use of time and effort. My feeling here is that these hard "political" problems will be with us for some time.

Towards a new logic curriculum

Another challenge comes in the matter of training people to do applied logic. Here I am more conservative than in other sections of this essay. I do not think there is any substitute for thorough grounding in more established fields. In fact, to be able to do serious work in applied logic one would probably have to have a good grounding in at least two fields. Where I think it makes sense to think about change is in the logic curriculum itself, specifically in the *second* (and further) courses in the subject. In the undergraduate curriculum, these second courses are frequently ones aimed at set theory or recursion theory. I see no reason why this has to be the case: after all, subjects can be presented with different emphases, and so why should we not present a serious, mathematically engaging treatment of logic that goes in the direction of applied logic? There are already texts on logic for computer science that do this. In other directions, one could easily imagine a second course in logic that emphasized both the challenging mathematics and the stimulating interest coming from modal logic. Another alternative would be to teach some of the areas of interaction of logic and linguistics, and in this way introduce model theory and proof theory. Surely doing this is not only The graduate logic curriculum, too, can be reworked. Here I think that all of the traditional areas of mathematical logic can be presented in ways that emphasize the applied side. A course in model theory, for example, could turn into the study of logical systems: students would then learn quite a bit more about completeness theorems for many logical systems (maybe even non-standard ones) than in the traditional classes. One can imagine other applied logic graduate classes, and surely those of us who are doing this should be talking to each other.

The price one would pay for a non-traditional curriculum is that students trained like this would not be trained to do research in traditional areas. But given that applied logic is likely to be interesting to a wide range of students, and given that applied logic stands ready to blossom into an important field, this price is not too steep.

References

1. J. R. Shoenfield, *Mathematical Logic*, Reading, Mass.-Menlo Park, Cal.-London-Don Mills, Ontario: Addison-Wesley Publish. Co., 1967.
2. P. Halmos, *Applied mathematics is bad mathematics*, In: The Forest of Symbols [in Finnish], Helsinki: Art House, 1992, pp. 37–54.
3. A. Nerode, *Applied logic*, In: The Merging of Disciplines: New Directions in Pure, Applied, and Computational Mathematics, Proc. Symp. Honor G. S. Young, Laramie/Wyo. 1985, R.E. Ewing et al (eds.), New York etc.: Springer-Verlag, 1986.
4. J. Y. Halpern, R. Harper, N. Immerman, Ph. G. Kolaitis, M. Y. Vardi, and V. Vianu, *On the unusual effectiveness of logic in computer science*, Bull. Symb. Log., **7** (2001), no. 2, 213–236.
5. N. Chomsky, *Syntactic Structures*, Mouton, 1957.
6. K. Devlin, *Goodbye, Descartes: The End of Logic and the Search for a New Cosmology of Mind*, Johy Wiley & Sons, 1997.

Index