Completeness Theorems for Behavioural Distances and Equivalences

Wojciech Krzysztof Różowski

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of

University College London.

Department of Computer Science

University College London

July 31, 2025

I, Wojciech Krzysztof Różowski, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

In theoretical computer science it is customary to provide expression languages for representing the behaviour of transition systems and to study formal systems for reasoning about equivalence or similarity of behaviours represented by expressions of the interest. The key example of this approach are Kleene's regular expressions, a specification language for deterministic finite automata, as well as complete axiomatisations of language equivalence of regular expressions due to Salomaa and Kozen.

The first part of this thesis studies axiomatisations of behavioural distances. Originally considered for probabilistic and stochastic systems, behavioural distances provide a quantitative measure of the dissimilarity of behaviours that can be defined meaningfully for a variety of transition systems. As a first contribution, we consider deterministic automata and we provide a sound and complete quantitative inference system for reasoning about a shortest-distinguishing-word distance between languages represented by regular expressions. Then, we move on to more complicated case of behavioural distance of Milner's charts, which provide a compelling setting for studying behavioural distances because they shift the focus from language equivalence to bisimilarity. As a syntax of choice, we rely on string diagrams, which provide a rigorous formalism that enables compositional reasoning by supporting a variable-free representation where recursion naturally decomposes into simpler components.

The second part focuses on generative probabilistic transition systems and presents a sound and complete axiomatisation of language equivalence of behaviours specified through the syntax of probabilistic regular expressions (PRE), a probabilistic analogue of regular expressions denoting probabilistic languages in which every word is assigned a probability of being generated. The completeness proof makes use of technical tools from the recently developed theory of proper functors and convex algebra, arising from the rich structure of probabilistic languages.

Contents

List of Figures

Chapter 1

Introduction

One of the motivations for the mathematical study of the models of computation stems from the desire for precise and formal reasoning about the correctness of computer systems. These theoretical foundations enable formal verification experts to prove that systems deployed in safety-critical areas, such as avionics or health-care, behave as expected. In theoretical computer science it is customary to model computations as state transition systems, which are discrete models where a set of states is equipped with a notion of one-step observable behaviour, describing how the system evolves. Typical examples include finite automata, Kripke frames, and Markov chains among many others.

The central topic of this thesis are axiomatisations of behaviour of transition systems. By this we mean providing expression languages for representing the behaviour of transition systems and the study of formal systems for reasoning about equivalence or similarity of behaviours represented by expressions of the interest.

The interest in axiomatising behaviour of transition systems originates from the seminal work of Kleene on regular expressions [Kle51]. In his influential paper from 1951, Kleene introduced deterministic finite automata (DFAs), which are the fundamental model of sequential deterministic computations. Each state of a DFA can be associated with a formal language, a collection of strings that are accepted starting from a given state. This characterises an important class of formal languages, known as regular languages. Classically, two states are equivalent if they recognise the same language. In the same paper, Kleene proposed regular expressions, which are an algebraic specification language for DFAs and proved that both formalisms are equally expressive through a result known nowadays as Kleene's theorem. As an open problem, he left a completeness question: are there a finite number of rules that enable reasoning about language equivalence of regular expressions?

Shortly after Kleene's paper, Redko [Red64] demonstrated that one cannot use a finite number of equational axioms to axiomatise the language equivalence. But the search for axiomatisation made of more expressive rules continued. The first answer came in 1966 from Salomaa [Sal66], who presented two axiom systems. One was infinitary, and the other used finite equations along with an implicational rule encoding Arden's lemma [Ard61] for formal languages. While Salomaa's implicational axiomatisation later became a blueprint for inference systems for reasoning about semantic equivalence or similarity of transition systems, this formal system was not algebraic. Essentially, the implicational rule relied on the productivity side-condition called empty word property (EWP) that caused the resulting axiomatisation to be unsound under substitution of letters by arbitrary expressions. This problem has motivated several researchers including Conway [Con12], Krob [Kro90] and Boffa [Bof90] to pursue the problem of obtaining algebraic axiomatisation of language equivalence of DFAs, eventually leading to the celebrated completeness result of Kozen [Koz94]. The inference system of Kozen is known nowadays under the name Kleene Algebra (KA) and it forms a basis of several formal systems for equational reasoning about imperative programs [KS96], packet-passing software defined networks [And+14], and concurrent programs [Kap+18; Wag+19] among many others.

Besides DFAs, automata theorists have studied many variants of automata, including nondeterministic [RS59], weighted [Sch61] and probabilistic [Rab63] ones, usually focusing on the notion of language equivalence or inclusion. At the advent of process algebra in the 1980s, Milner and Park brought the concept of bisimilarity [Par81], a notion of equivalence finer than language equivalence, that was motivated by the needs of the study of concurrency theory and models such as labelled transition systems (LTSs). Essentially, language equivalence is a linear-time notion, as it hides the precise moment of resolving nondeterministic choice from the external observer. At the same time, bisimilarity allows for a more fine-grained comparison of behaviours by looking at the exact moment of resolving the nondeterministic choice.

Milner [Mil84] considered a variant of LTSs that he called charts and studied the associated problem of axiomatising the bisimilarity of charts. Interestingly, while the syntax of regular expressions can be used to specify behaviours of charts, it is not expressive, that is there exist behaviours that cannot be specified using Kleene's syntax. Instead, Milner proposed a more general language called the algebra of regular behaviours (ARB) featuring binders, action prefixing, and a recursion operator. The paper introducing ARB also provided a suitable generalisation of Salomaa's non-algebraic axiomatisation and demonstrated its soundness and completeness with respect to the bisimilarity of charts.

The completeness results of Salomaa, Kozen, and Milner mentioned above are prototypical instances of the vast strain of research that has been of particular interest to theoretical computer scientists for decades. Given a transition system model and an associated notion of semantic equivalence, having a complete axiomatisation allows one to reason about model behaviour through the syntactic manipulation of terms of the specification language, which is well-suited for implementation, automation, and formal reasoning. Each time when the needs of modelling computer systems result in a new transition system model or an associated notion of semantic equivalence, it is natural to ask about the complete axiomatisation.

This thesis provides contributions to the above outlined field of axiomatisations of behaviours of transition systems in two orthogonal directions.

1. The first part of the thesis is concerned with the study of formal systems for quantitative reasoning about behavioural distances, that replace conventional notions of behavioural equivalence with a quantitative measure of how close the behaviour of two states of transition systems is.

2. The second part focuses on probabilistic transition systems and presents a sound and complete axiomatisation of language equivalence of behaviours

specified through the syntax of probabilistic regular expressions.

We now provide a brief outline of each of these directions.

1.1 Behavioural Distances

In many contexts, especially when dealing with probabilistic or quantitative models, focusing on exact equivalence of behaviours such as language equivalence or bisimilarity is too restrictive. A tiny perturbation in observed probability or weights of transition can deem two states inequivalent. Instead, it is often more meaningful to measure how far apart the behaviours of two states are.

This has motivated the development of behavioural distances, which endow the state spaces of transition systems with (pseudo)metric spaces quantifying the dissimilarity of states. In such a setting, states at distance zero are not necessarily the same, but rather equivalent with respect to some classical notion of behavioural equivalence. In a nutshell, equipping transition systems with such a notion of distance crucially relies on the possibility of lifting the distance between the states to the distance on the one-step observable behaviour of the transition system.

Behavioural distances first appeared in the context of probabilistic transition systems [Des+04; BW01], where one-step observable behaviour forms a probability distribution. In such a setting, in order to lift distances from the state space to one-step observable behaviour, one can rely on the classic Kantorovich lifting from transportation theory [Vil09].

More generally, behavioural distances are not limited to probabilistic or weighted systems; instead, they can be defined meaningfully for a variety of transition systems [Bal+18]. One of the simplest instances is deterministic finite automata, which can be equipped with a shortest-distinguishing-word distance [BKP18], where the longer the smallest word that can witness inequivalence of two states is, the closer the behaviour of compared states is. To illustrate that, given an alphabet $A = \{a\}$, we have that a state recognising the language $\{a, aa, aaa\}$ is closer to the one recognising $\{a, aa, aaa, aaa\}$ rather than $\{a\}$.

The study of axiomatisations of behavioural distances have mainly focused on concrete probabilistic cases [Bac+18a; Bac+18b; Bac+18c]. Axiomatisations of other important instances of behavioural distances are still underexplored. The main goal of the first part of this thesis is to initiate the study of axiomatisations and completeness problems for behavioural distances beyond the concrete probabilistic instances. Our starting point is the work of Bacci, Bacci, Larsen and Mardare [Bac+18a], who gave a sound and complete axiomatisation of branching-time behavioural distance of terms of a probabilistic process calculus.

1.2 Probabilistic Language Equivalence

In 1963, Rabin introduced probabilistic automata [Rab63]. This model captures the simple notion of randomised computation and acts as an acceptor for probabilistic languages. Under such semantics, each word over some fixed alphabet is associated with a weight from the unit interval capturing how likely the word is to be accepted. Throughout the years, Probabilistic Automata were deeply studied from an algorithmic point of view [Kie+11] that eventually enabled the development of practical verification tools for randomised programs [Kie+12].

In the process algebra community, Larsen and Skou [LS91] devised a notion of probabilistic bisimilarity, while Stark and Smolka [SS00] provided a probabilistic process calculus featuring binders and a recursion operator and gave a sound and complete axiomatisation of probabilistic bisimilarity of terms of their calculus. The later work of Silva and Sokolova [SS11] showed that one can extend Stark and Smolka's system with additional axioms characterising probabilistic language equivalence to obtain a complete axiomatisation of language equivalence.

While the result of Silva and Sokolova enables the use of the process algebraic syntax of Stark and Smolka for reasoning about probabilistic language equivalence, it is natural to ask if one could devise a simpler, binder-free specification language in the style of Kleene's Regular Expressions and provide a more streamlined axiomatisation in the style of Salomaa.

This problem is the central motivation for the second part of this thesis. One of the main inspirations for that comes from the probabilistic pattern matching community, where researchers already considered regular expression-like operations to specify probabilistic languages [Ros00]. They did so by replacing the union of languages and Kleene's star from the usual Regular Expression with their probabilistic counterparts, which respectively can be seen as a convex combination and a form of the Bernoulli process. At the same time, the precise connection of such syntaxes to the transition systems model was under-explored [Bee17] and the topic of axiomatisation was not tackled at all.

1.3 Coalgebra

Both behavioural distances and probabilistic language equivalence can be studied abstractly through the unifying framework of the universal coalgebra [Gum00; Rut00]. Coalgebras provide an abstract and uniform treatment of transition systems through the language of category theory. Generally speaking, transition systems can be seen as pairs consisting of a set of states and a transition function, mapping each state to its one-step behaviour. The coalgebraic outlook allows abstracting away the features of the one-step behaviour of the transition system, such as inputs, labels, nondeterminism, probability, and the like through the notion of a type, formally modelled as an endofunctor on the category of sets and functions. Given a type functor, one can uniformly instantiate abstract results concerning the transition systems of the interest.

In particular, each type of functor canonically determines a notion of behavioural equivalence of states. Under mild set-theoretic size constraints on the type functor, one can construct a final coalgebra, which provides a universal domain of behaviours of transition systems of interest. For example, the final coalgebra for the functor describing deterministic automata is isomorphic to the set of all formal languages over some alphabet [Rut00]. Concrete instances of coalgebraic behavioural equivalence usually capture variants of bisimilarity and coincide with the notions known for the literature such as bisimilarity of LTSs or probabilistic bisimilarity of Larsen and Skou [VR99].

Modelling finer notions of semantic equivalence can be phrased by changing the base category over which the type functor is defined to a more structured setting than sets and functions. For example, one of the ways to model probabilistic language equivalence in the language of coalgebra is to work with coalgebras for an appropriate type functor over the category of positive convex algebras [Sil+10; SS11]. In this category, the final coalgebra is precisely carried by the set of all probabilistic languages over some alphabet.

At the same time, the recent work on coalgebraic behavioural distances [Bal+18] provided a categorical generalisation of Kantorovich lifting to lifting endofunctors over the category of sets to the category of pseudometric spaces and nonexpansive maps between them. Final coalgebras for type functors obtained through such liftings come equipped with a pseudometric between behaviours. Such a coalgebraic outlook enables generalising the notions of behavioural distances beyond probabilistic transition systems and is extensively used in the first part of the thesis.

Using the theory of universal coalgebra for axiomatisation problems allows abstracting away the generic steps of completeness theorems and instantiating abstract categorical results to obtain concrete properties of transition systems of interest. For example, the thesis of Silva [Sil+10] follows the pattern introduced by Jacobs [Jac06] and casts completeness results as establishing appropriate universal properties in the categories of coalgebras. Similarly, the recently developed theory of rational fixpoints [Mil10] for coalgebras for proper functors [Mil18] provides a useful generalisation of the notion of regular languages to coalgebraic generality. One of the concrete instances of such a theory enables characterising the analogue of regular languages in the case of probabilistic languages [SW18] and underpins key results in the second part of the thesis.

1.4 Overview of the thesis

Having outlined the scope and the main aims of this thesis, we summarise below the content of each chapter and provide references to the main technical results. The description of each content chapter contains a table providing a high-level overview of the studied axiomatisation problem.

Chapter 2 presents a sound and complete axiomatisation of the *shortest-distinguishing-word* distance between formal languages represented by regular expressions. The axiomatisation relies on a recently developed quantitative analogue of equational logic [MPP16], allowing manipulation of rational-indexed judgements of the form $e \equiv_r f$ meaning the distance between terms e and f is less or equal to r. The technical core of the chapter is dedicated to the completeness argument that draws techniques from order theory and Banach spaces to simplify the calculation of the behavioural distance to the point it can be then mimicked by axiomatic reasoning.

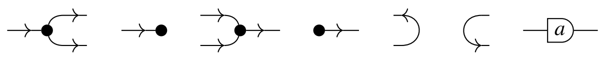
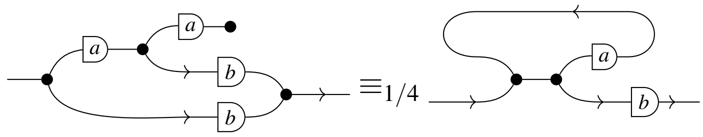
Summary of **??**	
Model	deterministic finite automata (DFA)
Syntax	$e, f \in \mathsf{RExp} ::= 0 \mid 1 \mid a \in A \mid e + f \mid e \,; f \mid e^*$
Semantics	shortest-distinguishing-word distance of languages
Example fact	$a^* \equiv_{1/4} a + 1$
Soundness	**??**
Completeness	**??**

This chapter incorporates results from the following paper:

> Wojciech Różowski. "A Complete Quantitative Axiomatisation of Behavioural Distance of Regular Expressions". In: *51st International Colloquium on Automata, Languages, and Programming (ICALP 2024)*. Ed. by Karl Bringmann, Martin Grohe, Gabriele Puppis, and Ola Svensson. Vol. 297. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024, 149:1–149:20. ISBN: 978-3-95977-322-5

Chapter 3 describes a sound and complete axiomatisation of a behavioural metric for nondeterministic processes using Milner's charts [Mil84]—a model that generalises finite-state automata by incorporating variable outputs. Charts provide a compelling setting for studying behavioural distances because they shift the focus from language equivalence to bisimilarity.

To formalise this approach, we adopt string diagrams [Sel10; PZ23b] as our syntax of choice. String diagrams closely mirror the graphical structure of charts, while providing a rigorous formalism that supports inductive reasoning and compositional semantics. Unlike traditional algebraic syntaxes, which require additional mechanisms such as binders and substitution, string diagrams offer a variable-free representation where recursion naturally decomposes into simpler components. This makes them well-suited for reasoning about behavioural distances and aligns with broader efforts to axiomatise automata-theoretic equivalences through a unified diagrammatic framework [PZ23a; Ant+25].

Summary of **??**	
Model	Milner's charts [Mil84]
Syntax	 $(a \in A)$
Semantics	bisimulation distance of regular behaviours
Example fact	
Soundness	**??**
Completeness	**??**

The findings presented in this chapter are the content of the following paper:

> Wojciech Różowski, Robin Piedeleu, Alexandra Silva, and Fabio Zanasi. "A Diagrammatic Axiomatisation of Behavioural Distance of Nondeterministic Processes". Under review. 2025

Chapter 4 introduces probabilistic regular expressions (PRE), a probabilistic analogue of regular expressions denoting probabilistic languages in which every word is assigned a probability of being generated. PRE are formed through constants from an alphabet and regular operations of probabilistic choice, sequential composition, probabilistic Kleene star, identity and emptiness. We present and prove the completeness of an inference system for reasoning about probabilistic language equivalence of PRE based on Salomaa's axiomatisation of language equivalence of regular expressions. The technical core of the chapter is devoted to the completeness proof, which relies on technical tools from the theory of convex algebra [SW18], arising from the rich structure of probabilistic languages.

Summary of Chapter 2	
Model	generative probabilistic transition systems [GSS95]
Syntax	$e, f \in \mathsf{PExp} ::= 0 \mid 1 \mid a \in A \mid e \oplus_p f \mid e \,;\, f \mid e^{[p]}$
Semantics	probabilistic language equivalence
Example fact	$a \,;\, a^{[1/4]} \equiv a \oplus_{3/4} \left(a \,;\, a^{[1/4]} \,;\, a\right)$
Soundness	Theorem 2.4.20
Completeness	Theorem 2.5.27

The results described in this chapter were published in the paper referenced below:

> Wojciech Różowski and Alexandra Silva. "A Completeness Theorem for Probabilistic Regular Expressions". In: *Proceedings of the 39th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2024, Tallinn, Estonia, July 8-11, 2024*. Ed. by Pawel Sobocinski, Ugo Dal Lago, and Javier Esparza. ACM, 2024, 66:1–66:14

Chapter 5 sketches directions for the future work and concludes this thesis.

Part I

Behavioural Distances

Chapter 2

Probabilistic Regular Expressions

Kleene [Kle51] introduced regular expressions and proved that these denote exactly the languages accepted by deterministic finite automata. In his seminal paper, Kleene left open a completeness question: are there a finite number of rules that enable reasoning about language equivalence of regular expressions? Since then, the pursuit of inference systems for equational reasoning about the equivalence of regular expressions has been subject of extensive study [Sal66; Kro90; Bof90; Koz94]. The first proposal is due to Salomaa [Sal66], who introduced a non-algebraic axiomatisation of regular expressions and proved its completeness.

Deterministic automata are a particular type of transition system: simply put, an automaton is an object with a finite set of states and a *deterministic* transition function that assigns every state and every action of the input alphabet *exactly* one next state. By varying the type of transition function one gets different systems: e.g. if the function assigns to every state and every action of the input alphabet *a set* of the next states, the resulting system is said to be non-deterministic; if the transition function assigns the next state based on any sort of *probability distribution* then the system is said to be probabilistic. Probabilistic systems appear in a range of applications, including modelling randomised algorithms, cryptographic protocols, and probabilistic programs. In this chapter, we focus on generative probabilistic transition systems (GPTS) with explicit termination [GSS95], and study the questions that Kleene and Salomaa answered for deterministic automata.

Our motivation to look at probabilistic extensions of regular expressions and axiomatic reasoning is two-fold: first, regular expressions and extensions thereof have been used in the verification of uninterpreted imperative programs, including network policies [KP00; KK05; And+14]; second, reasoning about exact behaviour of probabilistic imperative programs is subtle [Che+22], in particular in the presence of loops. By studying the semantics and axiomatisations of regular expressions featuring probabilistic primitives, we want to enable axiomatic reasoning for randomised programs and provide a basis to develop further verification techniques.

We start by introducing the syntax of Probabilistic Regular Expressions (PRE), inspired by work from the probabilistic pattern matching literature [Ros00]. PRE are formed through constants from an alphabet and *regular* operations of probabilistic choice, sequential composition, probabilistic Kleene star, identity and emptiness. We define the probabilistic analogue of *language semantics* of PRE as exactly the behaviours of GPTS. We achieve this by endowing PRE with operational semantics in the form of GPTS via a construction reminiscent of Antimirov derivatives of regular expressions [Ant96]. We also give a converse construction, allowing us to describe languages accepted by finite-state GPTS in terms of PRE, thus establishing an analogue of Kleene's theorem.

The main contribution of this chapter is presenting an inference system for reasoning about probabilistic language equivalence of PRE and proving its completeness. The technical core of this chapter is devoted to the completeness proof, which relies on technical tools convex algebra, arising from the rich structure of probabilistic languages. While being in the spirit of classic results from automata theory, our development relies on the more abstract approach enabled via the theory of universal coalgebra [Rut00]. As much as a concrete completeness proof ought to be possible, our choice to use the coalgebraic approach was fueled by wanting to reuse recent abstract results on the algebraic structure of probabilistic languages. A concrete proof would have to deal with fixpoints of probabilistic languages and would therefore require highly combinatorial and syntactic proofs about these. Instead, we reuse a range of hard results on convex algebras and fixpoints that Milius [Mil18], Sokolova and Woracek [SW15; SW18] proved in the last 10 years. In particular, we rely on the theory of rational fixpoints (for proper functors [Mil18]), which can be seen as a categorical generalisation of regular languages.

Our completeness proof provides further evidence that the use of coalgebras over proper functors provides a good abstraction for completeness theorems, where general steps can be abstracted away leaving as a domain-specific task to achieve completeness a construction to syntactically build solutions to systems of equations. Proving the uniqueness of such solutions is ultimately the most challenging step in the proof. By leveraging the theory of proper functors, our proof of completeness, which depends on establishing an abstract universal property, boils down to an argument that can be viewed as a natural extension of the work by Salomaa [Sal66] and Brzozowski [Brz64] from the 1960s.

The remainder of this chapter is organised as follows.

In Section 2.1, we introduce Probabilistic Regular Expressions (PRE), an analogue of Kleene's regular expressions denoting probabilistic languages and propose an inference system for reasoning about language equivalence of PRE.

Then, in Section 2.2, we elaborate on the main theoretical preliminaries for the technical development of this chapter. The coalgebraic approach to language semantics of GPTS is described in Section 2.3. In the remainder of that section, we provide a small-step semantics of PRE through an analogue of Antimirov derivatives endowing expressions with a structure of Generative Probabilistic Transition Systems (GPTS).

The technical core of this chapter is located in Section 2.4 and Section 2.5, where we obtain soundness and completeness results for our axiomatisation. Due to our use of proper functors, the proof boils down to a generalisation of a known proof of Salomaa for regular expressions [Sal66] exposing the connection to a classical result. We also obtain an analogue of Kleene's theorem allowing the conversion of finite-state GPTS to expressions through an analogue of Brzozowski's method [Brz64].

We conclude the chapter in Section 2.6, where we survey related work and sketch some areas for future work.

2.1 Overview

In this section, we will introduce the syntax and the language semantics of probabilistic regular expressions (PRE), as well as a candidate inference system to reason about the equivalence of PRE.

2.1.1 Syntax

Given a finite alphabet A, the syntax of PRE is given by:

$$e, f \in \mathsf{PExp} ::= 0 \mid 1 \mid a \in A \mid e \oplus_p f \mid e \,;\, f \mid e^{[p]} \qquad p \in [0,1]$$

We denote the expressions that immediately abort and successfully terminate by 0 and 1 respectively. For every letter $a \in A$ in the alphabet, there is a corresponding expression representing an atomic action. Given two expressions $e, f \in \mathsf{PExp}$ and $p \in [0,1]$, probabilistic choice $e \oplus_p f$ denotes an expression that performs e with probability p and performs f with probability $1-p$. One can think of $\oplus_p$ as the probabilistic analogue of the plus operator $(e+f)$ in Kleene's regular expressions. $e\,;\,f$ represents sequential composition, while $e^{[p]}$ is a probabilistic analogue of Kleene star: it successfully terminates with probability $1-p$ or with probability p performs e and then iterates $e^{[p]}$ again. In terms of the notational conventions, the probabilistic loop operator $(-)^{[p]}$ has the highest binding precedence, followed by sequential composition (;), with probabilistic choice $(\oplus_p)$ having the lowest precedence.

Example 2.1.1. The expression $a\,;a^{[\frac{1}{4}]}$ first performs action a with probability 1 and then enters a loop which successfully terminates with probability $\frac{3}{4}$ or performs action a with probability $\frac{1}{4}$ and then repeats the loop again. Intuitively, if we think of the action a as observable, the expression above denotes a probability associated with a non-empty sequence of a's. For example, the sequence aaa would be observed with probability $1 \cdot (1/4)^2 \cdot 3/4 = 3/64$.

2.1.2 Language semantics

PRE denote probabilistic languages $A^* \to [0,1]$. For instance, the expression 0 denotes a function that assigns 0 to every word, whereas 1 and a respectively assign probability 1 to the empty word and the word containing a single letter a from the alphabet. The probabilistic choice $e \oplus_p f$ denotes a language in which the probability of each word is the total sum of its probability in e scaled by p and its probability in f scaled by $1-p$. Describing the semantics of sequential composition and loops inductively is more involved. In particular, the semantics of loops would require a fixpoint calculation, which does not have as clear and straightforward (closed-form) formula, as the asterate of regular languages. Instead, we take an *operational approach*, and we formally define the language semantics of PRE in Section 2.3 through a small-step operational semantics, using a specific type of probabilistic transition system, which we introduce next.

2.1.3 Generative probabilistic transition systems

A GPTS consists of a set of states Q and a transition function that maps each state $q \in Q$ to finitely many distinct outgoing arrows of the form:

- *successful termination* with probability t (denoted $q \stackrel{t}{\Rightarrow} \checkmark$), or
- to another state r, via an *a-labelled transition*, with probability $s \in [0,1]$ (denoted $q \xrightarrow{a|s} r$).

We require that, for each state, the total sum of probabilities appearing on outgoing arrows sums up to less than or equal to one. The remaining probability mass is used to model unsuccessful termination, hence the state with no outgoing arrows can be thought of as exposing deadlock behaviour. These requirements reflect the intuition that a state *generates* transitions (potentially with different labels), rather than treating labels as input symbols, as in the case of DFAs.

Given a word $w \in A^*$ the probability of it being generated by a state $q \in Q$ (denoted $\mathsf{Lang}(q)(w) \in [0,1]$) is defined inductively:

$$\mathsf{Lang}(q)(\varepsilon) = t \quad \text{if } q \stackrel{t}{\Rightarrow} \checkmark \qquad\qquad \mathsf{Lang}(q)(av) = \sum_{q \xrightarrow{a|s} r} s \cdot \mathsf{Lang}(r)(v) \tag{2.1}$$

We say that two states q and q' are *language equivalent* if for all words $w \in A^*$, we have that $\mathsf{Lang}(q)(w) = \mathsf{Lang}(q')(w)$.

Example 2.1.2. Consider the following GPTS:

$$
\begin{tikzcd}
q_0 \arrow[r, dashed, "a \mid 1"] & q_1 \arrow[loop above, "a \mid \frac{1}{4}"] \arrow[r, Rightarrow, "\frac{3}{4}"] & \checkmark & q_2 \arrow[rr, bend left, "a \mid \frac{3}{4}"] \arrow[r, dashed, "a \mid \frac{1}{4}"'] & q_3 \arrow[loop below, "a \mid \frac{1}{4}"] \arrow[r, dashed, "a \mid \frac{3}{4}"'] & q_4 \arrow[r, Rightarrow, "1"] & \checkmark
\end{tikzcd}
$$

States q_0 and q_2 both assign probability 0 to the empty word ε and each word a^{n+1} is mapped to the probability $\left(\frac{1}{4}\right)^n \cdot \frac{3}{4}$. Later, we show that the languages generated by states q_0 and q_2 can be specified using expressions $a\,;a^{\left[\frac{1}{4}\right]}$ and $a \oplus_{\frac{3}{4}} \left(a\,;\left(a^{\left[\frac{1}{4}\right]}\,;a\right)\right)$ respectively.

In Section 2.3, we will associate to each PRE e an operational semantics or, more precisely, a state q_e in a GPTS. The language semantics $[\![e]\!]$ of e will then be the language $\mathsf{Lang}(q_e)\colon A^* \to [0,1]$ generated by q_e. Two PRE e and f are language equivalent if $\mathsf{Lang}(q_e) = \mathsf{Lang}(q_f)$. One of our main goals is to present a complete inference system to reason about language equivalence. In a nutshell, we want to present a system of (quasi-)equations of the form $e \equiv f$ such that:

$$e \equiv f \Leftrightarrow [\![e]\!] = [\![g]\!] \Leftrightarrow \mathsf{Lang}(e) = \mathsf{Lang}(f)$$

Such an inference system will have to contain rules to reason about all constructs of PRE, including probabilistic choice and loops. We describe next the system, with some intuition for the inclusion of each group of rules.

2.1.4 Axiomatisation of language equivalence of PRE

We define $\equiv\, \subseteq \mathsf{PExp} \times \mathsf{PExp}$ to be the least congruence relation closed under the axioms shown on Figure 2.1. We will show in Section 2.5 that these axioms are complete with respect to language semantics.

The first group of axioms capture properties of the probabilistic choice operator $\oplus_p$ (C1-C4) and its interaction with sequential composition (D1-D2). Intuitively, (C1-C4) are the analogue of the semilattice axioms governing the behaviour of $+$ in regular expressions. These four axioms correspond exactly to the axioms of barycentric algebras [Sto49], excluding the cancellation axiom (referred to as the fifth

Probabilistic choice

$$
\begin{array}{ll}
(\mathsf{C1}) & e \oplus_p e \equiv e \\
(\mathsf{C2}) & e \oplus_1 f \equiv e \\
(\mathsf{C3}) & e \oplus_p f \equiv f \oplus_{1-p} e \\
(\mathsf{C4}) & (e \oplus_p f) \oplus_q g \equiv e \oplus_{pq} \left(f \oplus_{\frac{(1-p)q}{1-pq}} g \right)
\end{array}
$$

Sequential composition

$$
\begin{array}{ll}
(\mathsf{1S}) & 1 ; e \equiv e, \\
(\mathsf{S}) & e ; (f ; g) \equiv (e ; f) ; g, \\
(\mathsf{S1}) & e ; 1 \equiv e, \\
(\mathsf{0S}) & 0 ; e \equiv 0, \\
(\mathsf{S0}) & e ; 0 \equiv 0, \\
(\mathsf{D1}) & (e \oplus_p f) ; g \equiv e ; g \oplus_p f ; g \\
(\mathsf{D2}) & e ; (f \oplus_p g) \equiv e ; f \oplus_p e ; g
\end{array}
$$

Loops

$$
\begin{array}{ll}
(\mathsf{Unroll}) & e^{[p]} \equiv e ; e^{[p]} \oplus_p 1 \\
(\mathsf{Tight}) & (e \oplus_p 1)^{[q]} \equiv e^{\left[\frac{pq}{1-(1-p)q}\right]} \\
(\mathsf{Div}) & 1^{[1]} \equiv 0 \\
(\mathsf{Unique}) & \dfrac{g \equiv e ; g \oplus_p f \qquad \boxed{E(e) = 0}}{g \equiv e^{[p]} ; f}
\end{array}
$$

Termination condition: $E : \mathsf{PExp} \to [0, 1]$

$$
\boxed{
\begin{array}{c}
E(1) = 1 \qquad E(0) = E(a) = 0 \qquad E(e \oplus_p f) = pE(e) + (1-p)E(f) \\[1em]
E(e ; f) = E(e)E(f) \qquad E\left(e^{[p]}\right) = \begin{cases} 0 & E(e) = 1 \wedge p = 1 \\ \frac{1-p}{1-pE(e)} & \text{otherwise} \end{cases}
\end{array}
}
$$

Figure 2.1: Axioms for language equivalence of PRE. The rules involving the division of probabilities are defined only when the denominator is non-zero. The function $E(-)$ provides a termination side condition to the (Unique) fixpoint axiom.

postulate in Stone's original paper). (D1) and (D2) are *right and left distributivity* rules of $\oplus$ over ;. The sequencing axioms (1S), (S1), (S) state PRE have the structure of a monoid (with neutral element 1 with absorbent element 0 – see axioms (0S), (S0). The loop axioms contain respectively *unrolling*, *tightening*, and *divergency* axioms plus a *unique fixpoint* rule. The (Unroll) axiom associates loops with their intuitive behaviour of choosing, at each step, probabilistically between successful termination and executing the loop body once. (Tight) and (Div) are the probabilistic analogues of the identity $(e+1)^* \equiv e^*$ from regular expressions. In the case of PRE, we need two axioms: (Tight) states that the probabilistic loop whose body might instantly terminate, causing the next loop iteration to be executed immediately is provably equivalent to a different loop, whose body does not contain immediate termination; (Div) takes care of the edge case of a no-exit loop and identifies it with failure. Finally, the unique fixpoint rule is a re-adaptation of the analogous axiom from Salomaa's axiomatisation and provides a partial converse to the loop unrolling axiom, given the loop body is productive – i.e. cannot immediately terminate. This productivity property is formally written using the side condition $E(e) = 0$, which can be thought of as the probabilistic analogue of empty word property from Salomaa's axiomatisation. Consider an expression $a^{[\frac{1}{2}]} ; (b \oplus_{\frac{1}{2}} 1)$. The only way it can accept the empty word is to leave the loop with the probability of $\frac{1}{2}$ and then perform 1, which also can happen with probability $\frac{1}{2}$. In other words, $[\![a^{[\frac{1}{2}]} ; (b \oplus_{\frac{1}{2}} 1)]\!](\varepsilon) = \frac{1}{4}$. A simple calculation allows to verify that $E(a^{[\frac{1}{2}]} ; (b \oplus_{\frac{1}{2}} 1)) = \frac{1}{4}$.

Example 2.1.3. We revisit the expressions from Example 2.1.2 and show their equivalence via axiomatic reasoning.

$$
\begin{aligned}
a ; a^{[\frac{1}{4}]} &\equiv a ; \left(a^{[\frac{1}{4}]} ; a \oplus_{\frac{1}{4}} 1 \right) && (\dagger) \\
&\equiv a ; \left(a^{[\frac{1}{4}]} ; a \right) \oplus_{\frac{1}{4}} a ; 1 && (\mathsf{D2}) \\
&\equiv a ; \left(a^{[\frac{1}{4}]} ; a \right) \oplus_{\frac{1}{4}} a && (\mathsf{S1}) \\
&\equiv a \oplus_{\frac{3}{4}} a ; \left(a^{[\frac{1}{4}]} ; a \right) && (\mathsf{C3})
\end{aligned}
$$

The $\dagger$ step of the proof above relies on the equivalence $e^{[p]} ; e \oplus_p 1 \equiv e^{[p]}$ derivable from other axioms under the assumption $E(e) = 0$ through a following line of reasoning:

$$
\begin{aligned}
e^{[p]} ; e \oplus_p 1 &\equiv (e ; e^{[p]} \oplus_p 1) ; e \oplus_p 1 && (\mathsf{Unroll}) \\
&\equiv (e ; (e^{[p]} ; e)) \oplus_p 1 ; e) \oplus_p 1 && (\mathsf{D1}) \\
&\equiv (e ; (e^{[p]} ; e) \oplus_p e) \oplus_p 1 && (\mathsf{1S}) \\
&\equiv (e ; (e^{[p]} ; e) \oplus_p e ; 1) \oplus_p 1 && (\mathsf{S1})
\end{aligned}
$$

$$\equiv e\,;(e^{[p]}\,;e \oplus_p 1) \oplus_p 1 \tag{D2}$$

Since $E(e) = 0$, we then have: $e^{[p]}\,;e \oplus_p 1 \overset{\text{(Unique)}}{\equiv} e^{[p]}\,;1 \overset{\text{(S1)}}{\equiv} e^{[p]}$.

2.2 Preliminaries

In this section, we review the main preliminaries for the technical development outlined in the subsequent sections.

2.2.1 Locally finitely presentable categories

In this chapter, we will rely on notions associated with the theory of locally finitely presentable categories [AR94], that allows to generalise the notion of *finiteness* to more structured categories than just Set.

$\mathcal{D}$ is a filtered category, if every finite subcategory $\mathcal{D}_0 \hookrightarrow \mathcal{D}$ has a cocone in $\mathcal{D}$. A filtered colimit is a colimit of the diagram $\mathcal{D} \to \mathcal{C}$, where $\mathcal{D}$ is a filtered category. A directed colimit is a colimit of the diagram $\mathcal{D} \to \mathcal{C}$, where $\mathcal{D}$ is a directed poset. We call a functor *finitary* if it preserves filtered colimits. An object C is *finitely presentable (fp)* if the representable functor $\mathcal{C}(C,-)\colon \mathcal{C} \to \mathsf{Set}$ preserves filtered colimits. Similarly, an object C is *finitely generated (fg)* if the representable functor $\mathcal{C}(C,-)\colon \mathcal{C} \to \mathsf{Set}$ preserves directed colimits of monomorphisms. Importantly, every finitely presentable object is finitely generated, but the converse does not hold in general.

Definition 2.2.1. A category $\mathcal{C}$ is *locally finitely presentable (lfp)* if it is cocomplete and there exists a set of finitely presentable objects, such that every object of $\mathcal{C}$ is a filtered colimit of objects from that set.

Set is the prototypical example of a locally finitely presentable category, where finitely presentable objects are precisely finite sets.

2.2.2 Monads and their algebras

A monad (over the category Set) is a triple $\mathbf{T} = (T, \mu, \eta)$ consisting of a functor $T\colon \mathsf{Set} \to \mathsf{Set}$ and two natural transformations: a unit $\eta\colon \mathsf{Id} \Rightarrow T$ and multiplication $\mu\colon T^2 \Rightarrow T$ satisfying $\mu \circ \eta_T = \mathsf{id}_T = \mu \circ T\eta$ and $\mu \circ \mu_T = \mu \circ T\mu$ A $\mathbf{T}$-algebra (also called an Eilenberg-Moore algebra) for a monad T is a pair (X,h) consisting of a set $X \in \mathcal{O}(\mathcal{C})$, called carrier, and a function $h\colon TX \to X$ such that $h \circ \mu_X = h \circ Th$ and $h \circ \eta_X = \mathsf{id}_X$. A $\mathbf{T}$-algebra homomorphism between two T-algebras (X,h) and (Y,k) is a function $f\colon X \to Y$ satisfying $k \circ Tf = f \circ h$.

$\mathbf{T}$-algebras and $\mathbf{T}$-homomorphisms form a category $\mathsf{Set}^{\mathbf{T}}$. There is a canonical forgetful functor $\mathcal{U}\colon \mathsf{Set}^{\mathbf{T}} \to \mathsf{Set}$ that takes each $\mathbf{T}$-algebra to its carrier. This functor has a left adjoint $X \mapsto (TX, \mu_X\colon T^2X \to T)$, mapping each set to its free $\mathbf{T}$-algebra. If X is finite, then we call (TX, μ_X) free finitely generated.

Given a function $f\colon X \to Y$, where Y is a carrier of a $\mathbf{T}$-algebra (Y,h), there is a unique homomorphism $f^\sharp\colon (TX, \mu_X) \to (Y,h)$ satisfying $f^\sharp \circ \eta_X = f$ that is explicitly given by $f^\sharp = h \circ Tf$.

2.2.3 Generalised determinisation

Language acceptance of nondeterministic automata (NDA) can be captured via determinisation. NDA can be viewed as coalgebras for the functor $N = 2 \times \mathcal{P}_\omega{}^A$, where $\mathcal{P}_\omega$ is the finite powerset monad. Determinisation converts a NDA $(X, \beta\colon X \to 2 \times \mathcal{P}_\omega X^A)$ into a deterministic automaton $\left(\mathcal{P}_\omega X, \beta^\sharp\colon \mathcal{P}_\omega X \to 2 \times (\mathcal{P}_\omega X)^A\right)$, where for $A \subseteq X$, we define $\beta^\sharp(A) = \bigcup_{x \in A} \beta(a)$. Additionally, $\beta^\sharp$ satisfies $\beta^\sharp(\{x\}) = \beta(x)$ for all $x \in X$. A language of the state $x \in X$ of NDA, is given by the language accepted by the state $\{x\}$ in the determinised automaton $(\mathcal{P}_\omega X, \beta^\sharp)$.

This construction can be generalised [Sil+10] to HT-coalgebras, where $T\colon \mathsf{Set} \to \mathsf{Set}$ is an underlying functor of finitary monad $\mathbf{T}$ and $H\colon \mathsf{Set} \to \mathsf{Set}$ an endofunctor that admits a final coalgebra that can be lifted to the functor $\overline{H}\colon \mathsf{Set}^{\mathbf{T}} \to \mathsf{Set}^{\mathbf{T}}$. Liftings of functors $H\colon \mathsf{Set} \to \mathsf{Set}$ to $\overline{H}\colon \mathsf{Set}^{\mathbf{T}} \to \mathsf{Set}^{\mathbf{T}}$, are in one-to-one correspondence with distributive laws of the monad T over the functor H [JSS15], which are natural transformations $\rho\colon TH \Rightarrow HT$ satisfying $H\eta_X = \rho_X \circ \eta_{HX}$ and $H\mu_X \circ \rho_{TX} \circ T\rho_X = \rho_X \circ \mu_{HX}$. In particular, given a $\mathbf{T}$-algebra $(X, k\colon TX \to X)$, we can equip HX, with a $\mathbf{T}$-algebra structure, given by the following composition of maps:

$$THX \xrightarrow{\quad \rho_X \quad} HTX \xrightarrow{\quad Hk \quad} HX$$

Generalised determinisation turns HT-coalgebras $(X, \beta\colon X \to HTX)$ into H-coalgebras $(TX, \beta^\sharp\colon TX \to HTX)$, where $\beta^\sharp$ is the unique extension arising from the free-forgetful adjunction between Set and $\mathrm{Set}^{\mathbf{T}}$. The language of a state $x \in X$ is given by $\mathsf{beh}_{\beta^\sharp} \circ \eta_X\colon X \to \nu H$, where η is the unit of the monad $\mathbf{T}$. Since $\beta^\sharp\colon TX \to HTX$ can be seen as a $\mathbf{T}$-algebra homomorphism $(TX, \mu_X) \to \overline{H}(TX, \mu_X)$, each determinisation $(TX, \beta^\sharp)$ can be viewed as an $\overline{H}$-coalgebra $((TX, \mu_X), \beta^\sharp)$. The carrier of the final H-coalgebra can be canonically equipped with $\mathbf{T}$-algebra structure, yielding the final $\overline{H}$-coalgebra. In such a case, the unique final homomorphism from any determinisation (viewed as an H-coalgebra) is precisely an underlying function of the final $\overline{H}$-coalgebra homomorphism.

2.2.4 Subdistribution monad

A function $\nu\colon X \to [0,1]$ is called a subprobability distribution or subdistribution, if it satisfies $\sum_{x\in X} \nu(x) \leq 1$. A subdistribution ν is *finitely supported* if the set $\mathrm{supp}(\nu) = \{x \in X \mid \nu(x) > 0\}$ is finite. We use $\mathcal{D}X$ to denote the set of finitely supported subprobability distributions on X. The weight of a subdistribution $\nu\colon X \to [0,1]$ is a total probability of its support:

$$|\nu| = \sum_{x\in X} \nu(x)$$

Given $\nu \in \mathcal{D}X$ and $Y \subseteq X$, we will write $\nu[Y] = \sum_{x\in Y} \nu(x)$. This sum is well-defined as only finitely many summands have non-zero probability.

Given $x \in X$, its *Dirac* is a subdistribution δ_x which is given by $\delta_x(y) = 1$ only if $x = y$, and 0 otherwise. We will moreover write $\mathbb{0} \in \mathcal{D}X$ for a subdistribution with an empty support. It is defined as $\mathbb{0}(x) = 0$ for all $x \in X$. When $\nu_1, \nu_2\colon X \to [0,1]$ are subprobability distributions and $p \in [0,1]$, we write $p\nu_1 + (1-p)\nu_2$ for the convex combination of ν_1 and ν_2, which is the probability distribution given by

$$(p\nu_1 + (1-p)\nu_2)(x) = p\nu_1(x) + (1-p)\nu_2(x)$$

for all $x \in X$. Note that this operation preserves finite support.

$\mathcal{D}$ is in fact a functor on the category Set, which maps each set X to $\mathcal{D}X$ and maps each arrow $f\colon X \to Y$ to the function $\mathcal{D}f\colon \mathcal{D}X \to \mathcal{D}Y$ given by

$$\mathcal{D}f(\nu)(x) = \sum_{y \in f^{-1}(x)} \nu(y)$$

Moreover, $\mathcal{D}$ also carries a monad structure with unit $\eta_X(x) = \delta_x$ and multiplication $\mu_X(\Phi)(x) = \sum_{\varphi \in \mathcal{D}X} \Phi(\varphi)\varphi(x)$ for $\Phi \in \mathcal{D}^2 X$. Using the free-forgetful adjunction between Set and category of $\mathcal{D}$-algebras, given $f\colon X \to \mathcal{D}Y$, there exists a unique map $f^\sharp\colon \mathcal{D}X \to \mathcal{D}Y$ satisfying $f = f^\sharp \circ \delta$ called the *convex extension of* f, and explicitly given by $f^\sharp(\nu)(y) = \sum_{x \in X} \nu(x) f(x)(y)$.

2.2.5 Positive convex algebras

By Σ_{PCA} we denote a signature given by

$$\Sigma_{\mathsf{PCA}} = \left\{ \boxplus_{i \in I} p_i \cdot (-)_i \mid I \text{ finite}, \forall i \in I.\, p_i \in [0,1], \sum_{i \in I} p_i \leq 1 \right\}$$

A positive convex algebra is a an algebra for the signature Σ_{PCA}, that is a pair $\mathcal{A} = \left(X, \Sigma^{\mathcal{A}}_{\mathsf{PCA}}\right)$, where X is the carrier set and $\Sigma^{\mathcal{A}}_{\mathsf{PCA}}$ is a set of interpretation functions $\boxplus_{i \in I} p_i \cdot (-)_i \colon X^{|I|} \to X$ satisfying the axioms:

1. (Projection) $\boxplus_{i \in I} p_i \cdot x_i = x_j$ if $p_j = 1$
2. (Barycenter) $\boxplus_{i \in I} p_i \cdot \left(\boxplus_{j \in J} q_{i,j} \cdot x_j\right) = \boxplus_{j \in J} \left(\sum_{i \in I} p_i q_{i,j}\right) \cdot x_j$

In terms of notation, we denote the unary sum by $p_0 \cdot x_0$. Throughout this chapter we will we abuse the notation by writing

$$\left(\boxplus_{i \in I} p_i \cdot e_i\right) \boxplus \left(\boxplus_{i \in J} q_j \cdot f_j\right)$$

for a single sum $\boxplus_{k \in I+J} r_k \cdot g_k$, where $r_k = p_k$ and $g_k = e_k$ for $k \in I$ and similarly $r_k = q_k$ and $g_k = f_k$ for $k \in J$. Note that this is well-defined only if $\sum_{i \in I} p_i + \sum_{j \in J} r_j \leq 1$.

The signature of positive convex algebras can be alternatively presented as a family of binary operations, in the following way:

Proposition 2.2.2 ([BSS17, Proposition 7])**.** *If X is a set equipped with a binary operation $\boxplus_p : X \times X \to X$ for each $p \in [0,1]$ and a constant $0_\boxplus \in X$ satisfying for all $x, y, z \in X$ (when defined) the following:*

$$x \boxplus_p x = x \qquad x \boxplus_1 y = x \qquad x \boxplus_p y = y \boxplus_{1-p} x$$
$$(x \boxplus_p y) \boxplus_q z = x \boxplus_{pq} \left(y \boxplus_{\frac{(1-p)q}{1-pq}} z \right)$$

then X carries the structure of a positive convex algebra. The interpretation of $\boxplus_{i \in I} p_i \cdot (-)_i$ is defined inductively by the following

$$\boxplus_{i \in I} p_i \cdot x_i = \begin{cases} 0_\boxplus & \text{if } I = \emptyset \\ x_0 & \text{if } p_0 = 1 \\ x_n \boxplus_{p_k} \left(\boxplus_{i \in I \setminus \{k\}} \frac{p_i}{1-p_k} \cdot x_i \right) & \text{otherwise, for some } k \in I \end{cases}$$

Below we state several properties of positive convex algebras, that we will use throughout this chapter.

Proposition 2.2.3. *Let I be a finite indexed set, and let $\{p_i\}_{i \in I}$ and $\{x_i\}_{i \in I}$ be indexed collections of elements of $[0,1]$ and X respectively. Then, in any positive convex algebra, the following statements hold:*

1. $$\boxplus_{i \in I} p_i \cdot x_i = \boxplus_{x \in \bigcup_{i \in I} \{x_i\}} \left(\sum_{x_i = x} p_i \right) \cdot x$$

2. *Let $=_R \subseteq X \times X$ be a congruence relation, with $[-]_R \colon X \to X/{=_R}$ being its canonical quotient map. Then,*

$$\boxplus_{i \in I} p_i \cdot x_i =_R \boxplus_{[x]_R \in \bigcup_{i \in I} \{[x_i]_R\}} \left(\sum_{x_i =_R x} p_i \right) \cdot x$$

3. *All terms $\boxplus_{i \in I} 0 \cdot x_i$ coincide and are all provably equivalent to the empty convex sum.*

4. *If $J \subseteq I$ and $\{i \in I \mid p_i \neq 0\} \subseteq J$, then*

$$\boxplus_{i\in I} p_i \cdot x_i = \boxplus_{j\in J} p_j \cdot x_j$$

5. *Let $\sigma \colon I \to I$ be a permutation of the index set I. Then, we have that*

$$\boxplus_{i\in I} p_i \cdot x_i = \boxplus_{i\in I} p_{\sigma(i)} \cdot x_{\sigma(i)}$$

Proof. We write $[\Phi]$ to denote Iverson bracket, which is defined to be 1 if Φ is true and 0 otherwise.

For 1 we have that

$$\begin{aligned}
\boxplus_{i\in I} p_i \cdot x_i &= \boxplus_{i\in I} p_i \cdot \left(\boxplus_{x\in\cup_{i\in I}\{x_i\}} [x_i = x] \cdot x \right) && \text{(Projection axiom)} \\
&= \boxplus_{x\in\cup_{i\in I}\{x_i\}} \left(\sum_{i\in I} p_i [x_i = x] \right) \cdot x && \text{(Barycenter axiom)} \\
&= \boxplus_{x\in\cup_{i\in I}\{x_i\}} \left(\sum_{x_i = x} p_i \right) \cdot x
\end{aligned}$$

2 can be shown by picking a representative for each equivalence class and then using 1. For 3, by [SW15, Lemma 3.4] we know that all terms $\boxplus_{i\in I} 0 \cdot x_i$ coincide. To see that they are provably equivalent to the empty convex sum, observe that

$$\begin{aligned}
\boxplus_{i\in I} 0 \cdot x_i &= \boxplus_{i\in I} 0 \cdot \left(\boxplus_{j\in\emptyset} p_j \cdot y_j \right) && \text{([SW15, Lemma 3.4])} \\
&= \boxplus_{j\in\emptyset} 0 \cdot y_j && \text{(Barycenter axiom)}
\end{aligned}$$

Finally, 4 follows from [SW15, Lemma 3.4], while 5 was proved in [Dob08, Proposition 3.1]. □

Lemma 2.2.4. *Let I, J be finite index sets, $\{p_i\}_{i\in I}$, $\{q_{i,j}\}_{(i,j)\in I\times J}$ and $\{x_{i,j}\}_{(i,j)\in I\times J}$*

indexed collections such that for all $i \in I$ and $j \in J$, $p_i, q_{i,j} \in [0,1]$ and $x_{i,j} \in X$. If X carries PCA *structure, then:*

$$\boxplus_{i \in I} p_i \cdot \left(\boxplus_{j \in J} q_{i,j} \cdot x_{i,j} \right) = \boxplus_{(i,j) \in I \times J} p_i q_{i,j} \cdot x_{i,j}$$

Proof.

$$\begin{aligned}
& \boxplus_{i \in I} p_i \cdot \left(\boxplus_{j \in J} q_{i,j} \cdot x_{i,j} \right) \\
&= \boxplus_{i \in I} p_i \cdot \left(\boxplus_{(k,j) \in \{i\} \times J} q_{k,j} \cdot x_{k,j} \right) \\
&= \boxplus_{i \in I} p_i \cdot \left(\boxplus_{(k,j) \in I \times J} [k = i] q_{k,j} \cdot x_{k,j} \right) && \text{(Proposition 2.2.3)} \\
&= \boxplus_{(k,j) \in I \times J} \left(\sum_{i \in I} p_i [k = i] q_{k,j} \right) \cdot x_{k,j} && \text{(Barycenter axiom)} \\
&= \boxplus_{(k,j) \in I \times J} p_k q_{k,j} \cdot x_{k,j} \\
&= \boxplus_{(i,j) \in I \times J} p_i q_{i,j} \cdot x_{i,j}
\end{aligned}$$

□

Lemma 2.2.5. *Let I be a finite index set, $\{p_i\}_{i \in I}$ and $\{q_i\}_{i \in I}$ indexed collections such that $p_i, q_i \in [0,1]$ for all $i \in I$, $\sum_{i \in I} p_i + \sum_{i \in I} q_i \leq 1$ and let $\{x_i\}_{i \in I}$ and $\{y_i\}_{i \in I}$ indexed collection such that $x_i, y_i \in X$ for all $i \in I$. If X carries* PCA *structure, then:*

$$\left(\boxplus_{i \in I} p_i \cdot x_i \right) \boxplus \left(\boxplus_{i \in I} q_i \cdot y_i \right) = \boxplus_{i \in I} (p_i + q_i) \cdot \left(\frac{p_i}{p_i + q_i} \cdot x_i \boxplus \frac{q_i}{p_i + q_i} \cdot y_i \right)$$

Proof. Let $J = \{0,1\}$. Define indexed collections $\{r_{i,j}\}_{(i,j) \in I \times J}$ and $\{z_{i,j}\}_{(i,j) \in I \times J}$, such that $r_{i,0} = \frac{p_i}{p_i + q_i}$ and $z_{i,0} = x_i$ and $r_{i,1} = \frac{q_i}{p_i + q_i}$ and $z_{i,1} = x_i$. We now reason:

$$\boxplus_{i \in I} (p_i + q_i) \cdot \left(\frac{p_i}{p_i + q_i} \cdot x_i \boxplus \frac{q_i}{p_i + q_i} \cdot y_i \right)$$

$$
\begin{aligned}
&= \boxplus_{i \in I} (p_i + q_i) \cdot \left(\boxplus_{j \in J} r_{i,j} \cdot z_{i,j} \right) \\
&= \boxplus_{(i,j) \in I \times J} (p_i + q_i) r_{i_j} \cdot z_{i,j} && \text{(Lemma 2.2.4)} \\
&= \left(\boxplus_{i \in I} p_i \cdot x_i \right) \boxplus \left(\boxplus_{i \in I} q_i \cdot y_i \right) && \square
\end{aligned}
$$

Speaking more abstractly, positive convex algebras and their homomorphisms (in the sense of homomorphisms of algebras for the signature from universal algebra) form a category, that we will call PCA. This category can be seen as a concrete presentation of an Eilenberg-Moore algebra for the subdistribution monad.

Theorem 2.2.6. *There is an isomorphism of categories between* PCA *and* $\mathsf{Set}^{\mathcal{D}}$. *Given a set* X *equipped with a positive convex algebra structure, we can define a map* $h\colon \mathcal{D}X \to X$*, given by*

$$h(\nu) = \boxplus_{x \in \mathrm{supp}(\nu)} \nu(x) \cdot x$$

for all $\nu \in \mathcal{D}X$*, making* (X, h) *into an algebra for the monad* $\mathcal{D}$*. Equivalently, given a* $\mathcal{D}$*-algebra* (X, h)*, one can define*

$$\boxplus_{i \in I} p_i \cdot x_i = h\left(\sum_{i \in I} p_i \cdot \delta_{x_i} \right)$$

for all finite I *and indexed collections* $\{p_i\}_{i \in I}$*,* $\{x_i\}_{i \in I}$*, such that* $\sum_{i \in I} p_i \leq 1$ *and for all* $i \in I$*,* $x_i \in X$*. This equips the set* X *with a positive convex algebra structure.*

Proof. See [Jac10, Theorem 4] or [Dob08, Proposition 5.3]. $\square$

Moreover, PCA as a category enjoys the following property:

Theorem 2.2.7 ([SW15])**.** *In* PCA *finitely presented and finitely generated objects coincide.*

2.2.6 Rational fixpoint

The completeness claim presented in this chapter will rely on the universal property of the rational fixpoint [AMV06; Mil10], which provides a convenient notion of a domain representing *finite* behaviours of structured transition systems, by relying on the theory of locally finitely presentable categories.

Let $\mathcal{B} \colon \mathcal{C} \to \mathcal{C}$ be a finitary functor. We will write $\mathsf{Coalg}_{\mathsf{fp}}\,\mathcal{B}$ for the subcategory of $\mathsf{Coalg}\,\mathcal{B}$ consisting only of $\mathcal{B}$-coalgebras with finitely presentable carrier. The *rational fixpoint* is defined as

$$(\rho\mathcal{B}, r) = \operatorname{colim}(\mathsf{Coalg}_{\mathsf{fp}}\,\mathcal{B} \hookrightarrow \mathsf{Coalg}\,\mathcal{B})$$

In other words, $(\rho\mathcal{B}, r)$ is colimit of the inclusion functor from the subcategory of coalgebras with finitely presentable carriers. We call it a fixpoint, as the map $r \colon \rho\mathcal{B} \to \mathcal{B}(\rho\mathcal{B})$ is an isomorphism [AMV06].

Under some restrictions on underlying category and the type functor of coalgebras, we have the following result:

Theorem 2.2.8 ([MPW20, Corollary 3.10, Theorem 3.12])**.** *If finitely presentable and finitely generated objects coincide in $\mathcal{C}$ and $\mathcal{B} \colon \mathcal{C} \to \mathcal{C}$ is a finitary endofunctor preserving non-empty monomorphisms, then rational fixpoint is fully abstract, that is, $(\rho B, r)$ is a subcoalgebra of the final coalgebra $(\mathcal{B}, t)$.*

The requirement of preserving non-empty monomorphisms is quite weak and is satisfied by any lifting of a Set endofunctor to the category of Eilenberg-Moore algebras.

Lemma 2.2.9. *Let $H \colon \mathsf{Set} \to \mathsf{Set}$ be an endofunctor and let $\mathbf{T}$ be a finitary monad on Set. Then, the lifting $\overline{H} \colon \mathsf{Set}^{\mathbf{T}} \to \mathsf{Set}^{\mathbf{T}}$ preserves non-empty monomorphisms.*

Proof. Follows from [Gum00, Corollary 3.16] and [MPW20, Lemma 2.4]. $\square$

2.3 Operational semantics

In this section, we begin by describing a coalgebraic approach to modelling the probabilistic language semantics of GPTS. Building on this, we introduce an operational semantics for PRE, drawing inspiration from Antimirov's partial derivatives for NFAs [Ant96].

2.3.1 Language semantics of GPTS

Let $\mathcal{F}\colon \mathsf{Set} \to \mathsf{Set}$ be an endofunctor given by $\mathcal{F} = \{\checkmark\} + A \times (-)$. GPTS are precisely $\mathcal{DF}$-coalgebras, that is pairs (X, β), where X is a set of states and $\beta\colon X \to \mathcal{D}(\{\checkmark\} + A \times X)$ is a transition structure. Because of this, we will interchangeably use terms "$\mathcal{DF}$-coalgebra" and "GPTS".

The functor $\mathcal{DF}$ admits a final coalgebra, but unfortunately it is not carried by the set of probabilistic languages, that is $[0,1]^{A^*}$, because the canonical semantics of $\mathcal{DF}$-coalgebras happens to correspond to the more restrictive notion of probabilistic bisimilarity (also known as Larsen-Skou bisimilarity [LS91]). Probabilistic bisimilarity is a branching-time notion of equivalence, requiring observable behaviour of compared states to be equivalent at every step, while probabilistic language equivalence is a more liberal notion comparing sequences of observable behaviour. In general, if two states are bisimilar, then they are language equivalent, but the converse does not hold.

Example 2.3.1. Consider the following GPTS:

$$q_0 \xrightarrow{a \mid \frac{2}{3}} q_1 \xrightarrow{b \mid \frac{1}{2}} q_2 \overset{1}{\Longrightarrow} \checkmark \qquad\qquad q_3 \xrightarrow{a \mid \frac{1}{2}} q_4 \xrightarrow{b \mid \frac{2}{3}} q_5 \overset{1}{\Longrightarrow} \checkmark$$

States q_0 and q_3 are language equivalent because they both accept the string *ab* with the probability $\frac{1}{3}$, but are not bisimilar, because the state q_0 can make a transition with the probability $\frac{2}{3}$, while q_3 can perform an a transition with probability $\frac{1}{2}$.

A similar situation happens when looking at nondeterministic automata through the lenses of universal coalgebra, where again the canonical notion of equivalence is the one of bisimilarity. A known remedy is the powerset construction from classic automata theory, which converts a nondeterministic automaton to a deterministic automaton, whose states are sets of states of the original nondeterministic automaton we have started from. In such a case, the nondeterministic branching structure is factored into the state space of the determinised automaton. The language of an arbitrary state of the nondeterministic automaton corresponds to the language of the singleton set containing that state in the determinised automaton.

As we have discussed in Section 2.2.3, generalised determinisation extends the above idea to HT-coalgebras, where $T\colon \mathsf{Set} \to \mathsf{Set}$ is an underlying functor of a finitary monad $\mathbf{T}$ and $H\colon \mathsf{Set} \to \mathsf{Set}$ is a functor that can be lifted to category $\mathsf{Set}^{\mathbf{T}}$ of $\mathbf{T}$-algebras. Generalised determinisation provides a uniform treatment of language semantics of variety of transition systems, where the final H-coalgebra provides a notion of language. Unfortunately, $\mathcal{DF}$-coalgebras do not fit immediately to this picture. Luckily, each such $\mathcal{DF}$-coalgebra can be seen as a special case of a more general kind of transition system, known as reactive probabilistic transition systems (RPTS) [GSS95] or Rabin probabilistic automata [Rab63].

RPTS can be intuitively viewed as a probabilistic counterpart of nondeterministic automata and they can be determinised to obtain probabilistic language semantics. In an RPTS, each state x is mapped to a pair $\langle o_x, n_x\rangle$, where $o \in [0,1]$ is the acceptance probability of state x and $n_x\colon A \to \mathcal{D}(X)$ is the next-state function, which takes a letter $a \in A$ and returns the subprobability distribution over successor states. Formally speaking, let $\mathcal{G}\colon \mathsf{Set} \to \mathsf{Set}$ be an endofunctor $\mathcal{G} = [0,1] \times (-)^A$. RPTS are precisely $\mathcal{GD}$-coalgebras, that is pairs (X,β), where X is a set of states and $\beta\colon X \to [0,1] \times \mathcal{D}(X)^A$ is a transition function. Following the convention outlined before, we will use terms "$\mathcal{GD}$-coalgebras" and "RPTS" interchangeably.

$\mathcal{GD}$-coalgebras fit into framework of generalised determinisation [SS11]. In particular, there exists a distributive law $\rho\colon \mathcal{DG} \Rightarrow \mathcal{GD}$ of the monad $\mathcal{D}$ over the functor $\mathcal{G}$ that allows to lift $\mathcal{G}\colon \mathsf{Set} \to \mathsf{Set}$ to $\overline{\mathcal{G}}\colon \mathsf{PCA} \to \mathsf{PCA}$. Speaking in concrete terms, if the set X is equipped with a convex sum operation $\boxplus_{i\in I} p_i \cdot (-)$, then so is $[0,1] \times X^A$. Let $\{\langle o_i, t_i\rangle\}_{i\in I}$ be an indexed collection of elements of $[0,1] \times X^A$. Then, we can define

$$\boxplus_{i\in I} p_i \cdot \langle o_i, t_i\rangle = \left\langle \sum_{i\in I} p_i \cdot o_i, \lambda a. \boxplus_{i\in I} p_i \cdot t_i(a) \right\rangle \tag{2.2}$$

The final coalgebra for the functor $\mathcal{G}$ is precisely carried by the set $[0,1]$ of probabilistic languages.

In order to talk about language semantics of $\mathcal{DF}$-coalgebras, we first provide an informal intuition that each $\mathcal{DF}$-coalgebra can be seen as a special case of a $\mathcal{GD}$-coalgebra.

Example 2.3.2. Fragment of a GPTS (on the left) and of the corresponding RPTS (on the right).

$$\begin{tikzcd}
\checkmark & q_0 \arrow[l, Rightarrow, "\frac{1}{4}"'] \arrow[dl, "a \mid \frac{1}{4}"'] \arrow[dr, "a \mid \frac{1}{2}"] & \\
q_1 & & q_2
\end{tikzcd}
\qquad
\begin{tikzcd}
\frac{1}{4} & q_0 \arrow[l, Rightarrow] \arrow[r, "a"] & \circ \arrow[dl, dashed, "\frac{1}{4}"'] \arrow[dr, dashed, "\frac{1}{2}"] & \\
& q_1 & & q_2
\end{tikzcd}$$

In the corresponding RPTS state q_0 accepts with probability $\frac{1}{4}$ and given input a it transitions to subprobability distribution that has $\frac{1}{4}$ probability of going to to q_1 and $\frac{1}{2}$ probability of going to q_2.

We can make the above intuition formal. Let X be a set, and let $\zeta \in \mathcal{DF}X$. Define a function $\gamma_X \colon \mathcal{DF}X \to \mathcal{GD}X$, given by

$$\gamma_X(\zeta) = \langle \zeta(\checkmark), \lambda a.\, \lambda x.\, \zeta(a,x) \rangle$$

Such functions define components of the natural transformation.

Proposition 2.3.3. *[SS11] $\gamma \colon \mathcal{DF} \Rightarrow \mathcal{GD}$ is a natural transformation with injective components.*

We now have all ingredients to specify language semantics of $\mathcal{DF}$-coalgebras. Given a $\mathcal{DF}$-coalgebra (X,β), one can use the natural transformation γ and obtain $\mathcal{GD}$-coalgebra $(X, \gamma_X \circ \beta)$. Since $\mathcal{G}$ can be lifted to PCA, we can obtain $\mathcal{G}$-coalgebra $\big(\mathcal{D}X, (\gamma_X \circ \beta)^\sharp\big)$. Note that this coalgebra carries an additional algebra structure and its transition map is a PCA homomorphism, thus making $\big((\mathcal{D}X, \mu_X), (\gamma_X \circ \beta)^\sharp\big)$ into a $\overline{\mathcal{G}}$-coalgebra. The resulting language semantics of (X,β) are given by the map

$\mathsf{Lang}_{(X,\beta)} \colon X \to [0,1]^A$ explicitly given by

$$\mathsf{Lang}_{(X,\beta)} = \mathsf{beh}_{(\gamma_X \circ \beta)^\sharp} \circ \eta_X$$

where $\eta \colon X \to \mathcal{D}X$ is a unit of the monad $\mathcal{D}$ taking each state $x \in X$ to its Dirac $\delta_x \in \mathcal{D}X$.

This can be summarised by the following commutative diagram:

$$\begin{tikzcd}
X \arrow[r, "\eta_X"] \arrow[d, "\beta"'] & \mathcal{D}X \arrow[r, dashed, "\mathsf{beh}_{(\gamma_X\circ\beta)^\sharp}"] \arrow[dd, dashed, "(\gamma_X\circ\beta)^\sharp"] & {[0,1]^A} \arrow[dd, "t"] \\
\mathcal{D}\mathcal{F}X \arrow[d, "\gamma_X"'] & & \\
\mathcal{G}\mathcal{D}X \arrow[rr, dashed, "\mathcal{G}\mathsf{beh}_{(\gamma_X\circ\beta)^\sharp}"] & & \mathcal{G}\left([0,1]^A\right)
\end{tikzcd}$$

The language semantics defined above coincide with the explicit definition of Lang we gave in Equation (2.1) (this is a consequence of a result in [SS11, Section 5.2]).

Moreover, the natural transformation $\gamma \colon \mathcal{D}\mathcal{F} \Rightarrow \mathcal{G}\mathcal{D}$ interacts well with the distributive law $\rho \colon \mathcal{D}\mathcal{G} \Rightarrow \mathcal{G}\mathcal{D}$, making the following diagram commute:

$$\begin{tikzcd}
\mathcal{D}^2\mathcal{F} \arrow[rr, "\mu_{\mathcal{G}}"] \arrow[d, "\mathcal{D}\gamma"'] & & \mathcal{D}\mathcal{F} \arrow[d, "\gamma"] \\
\mathcal{D}\mathcal{G}\mathcal{D} \arrow[r, "\rho_{\mathcal{D}}"'] & \mathcal{G}\mathcal{D}^2 \arrow[r, "\mathcal{G}\mu"'] & \mathcal{G}\mathcal{D}
\end{tikzcd}$$

The above is a consequence of $\gamma \colon \mathcal{D}\mathcal{F} \Rightarrow \mathcal{G}\mathcal{D}$ being a so-called extension law – for more details, see [JSS15, Section 7.2]. This observation yields an alternative characterisation of the transition structure induced by the generalised determinisation, which will be employed in one of the steps of the completeness proof, as stated below.

Proposition 2.3.4. *For any $\mathcal{DF}$-coalgebra (X, β), the following diagram commutes:*

$$\begin{tikzcd}
\mathcal{D}X \arrow[r, "\mathcal{D}\beta"'] \arrow[rrr, bend left, "(\gamma_X \circ \beta)^\sharp"] & \mathcal{D}^2\mathcal{F}X \arrow[r, "\mu_{\mathcal{F}}"'] & \mathcal{DF}X \arrow[r, "\gamma_X"'] & \mathcal{GD}X
\end{tikzcd}$$

Proof. We first argue that $(\gamma_X \circ \beta)^\sharp \circ \eta_X = \gamma_X \circ \mu_{\mathcal{F}X} \circ \mathcal{D}\beta \circ \eta_X$

$$\begin{aligned}
\eta_X \circ \mathcal{D}\beta \circ \mu_{\mathcal{F}X} \circ \gamma_X &= \beta \circ \eta_{\mathcal{DF}X} \circ \mu_{\mathcal{F}X} \circ \gamma_X && (\eta \text{ is natural}) \\
&= \beta \circ \gamma_X && (\text{Monad laws}) \\
&= (\beta \circ \gamma_X)^\sharp \circ \eta_X && (\text{Kleisli extension})
\end{aligned}$$

Then, we argue that $\gamma_X \circ \mu_{\mathcal{F}X} \circ \mathcal{D}\beta$ is a PCA homomorphism from the free PCA (X, μ_X) to $\overline{\mathcal{G}}(X, \mu_X)$, by checking the commutativity of the diagram below.

$$\begin{tikzcd}
\mathcal{D}^2X \arrow[r, "\mathcal{D}^2\beta"] \arrow[dd, "\mu_X"'] & \mathcal{D}^3\mathcal{F}X \arrow[r, "\mathcal{D}\mu_{\mathcal{F}X}"] \arrow[dd, "\mu_{\mathcal{DF}X}"] & \mathcal{D}^2\mathcal{F}X \arrow[r, "\mathcal{D}\gamma_X"] \arrow[dd, "\mu_{\mathcal{DF}X}"'] & \mathcal{DFD}X \arrow[d, "\rho_{\mathcal{D}X}"] \\
& & & \mathcal{G}^2\mathcal{D}X \arrow[d, "\mathcal{G}\mu_X"] \\
\mathcal{D}X \arrow[r, "\mathcal{D}\beta"'] & \mathcal{D}^2\mathcal{F}X \arrow[r, "\mu_{\mathcal{F}X}"] & \mathcal{DF}X \arrow[r, "\gamma_X"] & \mathcal{GD}X
\end{tikzcd}$$

The left diagram commutes because μ is natural, while the middle one commutes because of μ being a multiplication map of the monad. Finally, the commutativity of the rightmost subdiagram is guaranteed by γ being an extension law (see discussion above).

Since, $\mu_{\mathcal{F}X} \circ \mathcal{D}\beta \circ \eta_X$ is a PCA homomorphism that factorises through η in the same way as $(\gamma_X \circ \beta)^\sharp$, we have that $(\gamma_X \circ \beta)^\sharp = \gamma_X \circ \mu_{\mathcal{F}X} \circ \mathcal{D}\beta$. □

2.3.2 Antimirov derivatives

We now equip PExp with a $\mathcal{DF}$-coalgebra structure, that is, we define a function $\partial : \mathsf{PExp} \to \mathcal{D}(1 + A \times \mathsf{PExp})$. We refer to ∂ as the *Antimirov derivative*, as it is reminiscent of the analogous construction for regular expressions and nondeterministic automata [Ant96]. First, we define a helper function that will eventually be used to define sequential composition:

Definition 2.3.5. Let $f \in \mathsf{PExp}$. We define $(- \lhd f) : \mathcal{D}\mathcal{F}\mathsf{PExp} \to \mathcal{D}\mathcal{F}\mathsf{PExp}$ to be given by $(- \lhd f) = {c_f}^{\sharp}$, that is a convex extension of the map $c_f : 1 + A \times \mathsf{PExp} \to \mathcal{D}(1 + A \times \mathsf{PExp})$ given by the following:

$$c_f(x) = \begin{cases} \partial(f) & x = \checkmark \\ \delta_{(a,e';f)} & x = (a,e') \end{cases}$$

Using such a helper function, we can state the following definition:

Definition 2.3.6 (Antimirov derivatives for PRE)**.** Given $a \in A$, $e, f \in \mathsf{PExp}$ and $p \in [0,1]$ we define:

$$\partial(0) = \mathbb{0} \qquad \partial(1) = \delta_{\checkmark} \qquad \partial(a) = \delta_{(a,1)}$$
$$\partial(e \oplus_p f) = p\partial(e) + (1-p)\partial(f) \qquad \partial(e\,;f) = \partial(e) \lhd f$$
$$\partial\left(e^{[p]}\right)(x) = \begin{cases} \frac{1-p}{1-p\partial(e)(\checkmark)} & x = \checkmark \\ \frac{p\partial(e)(a,e')}{1-p\partial(e)(\checkmark)} & x = (a,(e'\,;e^{[p]})) \\ 0 & \text{otherwise} \end{cases}$$

In the definition above, the expression 0 is mapped to the empty subdistribution, intuitively representing a deadlock. On the other hand, the expression 1 represents immediate acceptance, that is, it transitions to $\checkmark$ with probability 1. For any letter $a \in A$ in the alphabet, the expression a performs a-labelled transition to 1 with probability 1. The outgoing transitions of the probabilistic choice $e \oplus_p f$ consist of the outgoing transitions of e with probabilities scaled by p and the outgoing transitions of f scaled by $1-p$. In the definition of $\partial(e;f)$, we use the helper function $(- \lhd f)$ defined above (see Definition 2.3.5). Intuitively speaking, we need to factor in the possibility that e may accept with some probability t, in which case the outgoing transitions of f contribute to the outgoing transitions of $e\,;f$. In such a case, $(- \lhd f)$ reroutes the transitions coming out of $\partial(e)$: acceptance (the first case of c_f from Definition 2.3.5) is replaced by the behaviour of f, and the probability mass of transitioning to e' (the second case of c_f from Definition 2.3.5) is reassigned to $e' \,;f$.

Example 2.3.7. Below, we give a pictorial representation of the effect on the derivatives of $e\,;f$. Here, we assume that $\partial(e)$ can perform a a-transition to e' with probability s; we will make the same assumption in the informal descriptions of derivatives for the loops, later on.

$$\begin{tikzcd}
\partial(f)\ \cancel{\checkmark} & e\,;f \arrow[l, Leftarrow, "t"'] \arrow[d, "a\,|\,s"'] \\
 & e'\,;f
\end{tikzcd}$$

The definition of loops is slightly more involved. This stems from the requirement that $\partial\left(e^{[p]}\right)$ is the least subdistribution satisfying the following equation:

$$\partial\left(e^{[p]}\right) = p\partial(e) \lhd e^{[p]} + (1-p)\partial(\checkmark)$$

In the case when $\partial(e)(\checkmark) \neq 0$, the above becomes a fixpoint equation (as in such a case, the unrolling of the definition of $(- \lhd e^{[p]})$ involves $\partial\left(e[p]\right)$). It turns out, we can give the definition $\partial\left(e^{[p]}\right)$ in a closed form, but we need to consider two cases. If $\partial(e)(\checkmark) = 1$ and $p = 1$, then the loop body is constantly executed, but the inner expression e does not perform any labelled transitions. We identify such divergent loops with deadlock behaviour and hence $\partial(e^{[p]})(x) = 0$. Otherwise, we look at $\partial(e)$ to build $\partial\left(e^{[p]}\right)$.

First, we make sure that the loop may be skipped with probability $1-p$. Next, we modify the branches that perform labelled transitions by adding $e^{[p]}$ to be executed next. The remaining mass is $p\partial(e)(\checkmark)$, the probability that we will enter the loop and immediately exit it without performing any labelled transitions. We discard this possibility and redistribute it among the remaining branches.

Example 2.3.8. As before, we provide an informal visual depiction of the probabilistic loop semantics below, using the same conventions as before. The crossed-out checkmark along with the dashed lines denotes the redistribution of probability mass described above.

$$\begin{tikzcd}
\not\checkmark \arrow[dr, dashed, bend right] & e^{[p]} \arrow[l, Rightarrow, "pt"'] \arrow[rr, "a \mid ps/(1-pt)"] \arrow[d, Rightarrow, "\frac{1-p}{1-pt}"'] & & e';e^{[p]} \\
& \checkmark & &
\end{tikzcd}$$

Having defined the Antimirov transition system, one can observe that the termination operator $E(-)\colon \mathsf{PExp} \to [0,1]$ precisely captures the probability of an expression transitioning to $\checkmark$ (successful termination) when viewed as a state in the Antimirov GPTS.

Lemma 2.3.9. *For all $e \in \mathsf{PExp}$ it holds that $E(e) = \partial(e)(\checkmark)$.*

Proof. By structural induction. The base cases $E(0) = 0 = \partial(0)(\checkmark)$, $E(1) = 1 = \partial(1)(\checkmark)$ and $E(a) = 0 = \partial(a)(\checkmark)$ hold immediately. For the inductive steps, we have the following:

$\boxed{\text{Probabilistic choice}}$

$$\begin{aligned}
E(e \oplus_p f) &= pE(e) + (1-p)E(f) \\
&= p\partial(e)(\checkmark) + (1-p)\partial(f)(\checkmark) \\
&= \partial(e \oplus_p f)(\checkmark)
\end{aligned}$$

$\boxed{\text{Sequential compositon}}$

$$\begin{aligned}
E(e\,;f) &= E(e)E(f) \\
&= \partial(e)(\checkmark)\partial(f)(\checkmark) \\
&= (\partial(e) \lhd f)(\checkmark) \\
&= \partial(e\,;f)(\checkmark)
\end{aligned}$$

$\boxed{\text{Loops}}$ First, we consider the case when $\partial(e)(\checkmark) = 1$ and the loop probability is 1. By induction hypothesis, also $E(e) = 1$ and hence $E\left(e^{[1]}\right) = \partial\left(e^{[1]}\right)(\checkmark)$.

Otherwise, we have the following:

$$\begin{aligned} E(e^{[p]}) &= \frac{1-p}{1-pE(e)} \\ &= \frac{1-p}{1-p\partial(e)(\checkmark)} \\ &= \partial\left(e^{[p]}\right)(\checkmark) \end{aligned}$$

□

Given an expression $e \in \mathsf{PExp}$, we write $\langle e \rangle \subseteq \mathsf{PExp}$ for the set of states reachable from e by repeatedly applying ∂. It turns out that the operational semantics of every PRE can be always described by a finite-state GPTS given by $(\langle e \rangle, \partial)$.

Lemma 2.3.10. *For all $e \in \mathsf{PExp}$, the set $\langle e \rangle$ is finite. In fact, the number of of states is bounded above by $\#(-)\colon \mathsf{PExp} \to \mathbb{N}$, where $\#(-)$ is defined recursively by:*

$$\#(0) = \#(1) = 1 \quad \#(a) = 2 \quad \#(e \oplus_p f) = \#(e) + \#(f)$$
$$\#(e\,;f) = \#(e) + \#(f) \quad \#(e^{[p]}) = \#(e) + 1$$

Proof. We adapt the analogous proof for GKAT [Sch+21].

For any $e \in \mathsf{PExp}$, let $|\langle e \rangle|$ be the cardinality of the carrier set of the least subcoalgebra of $(\mathsf{PExp}, \partial)$ containing e. We show by induction that for all $e \in \mathsf{PExp}$ it holds that $|\langle e \rangle| \leq \#(e)$.

For the base cases, observe that for 0 and 1 the subcoalgebra has exactly one state. Hence, $\#(0) = 1 = |\langle 0 \rangle|$. Similarly, we have $\#(1) = 1 = |\langle 1 \rangle|$. For $a \in A$, we have two states; the initial state, which transitions with probability 1 on a to the state which outputs ✓ with probability 1.

For the inductive cases, assume that $|\langle e \rangle| \leq \#(e)$, $|\langle f \rangle| \leq \#(f)$ and $p \in [0,1]$.

- Every derivative of $e \oplus_p f$ is either a derivative of e or f and hence $|\langle e \oplus_p f \rangle| \leq |\langle e \rangle| + |\langle f \rangle| = \#(e) + \#(f) = \#(e \oplus_p f)$.
- In the case of $e\,;f$, every derivative of this expression is either a derivative of f or some derivative of e followed by f. Hence, $|\langle e\,;f \rangle| = |\langle e \rangle \times \{f\}| + |\langle f \rangle| \leq \#(e) + \#(f) = \#(e\,;f)$.

- For the probabilistic loop case, observe that every derivative of $e^{[p]}$ is a derivative of e followed by $e^{[p]}$ or it is the state that outputs $\checkmark$ with probability 1. It can be easily observed that $|\langle e^{[p]}\rangle| \leq |\langle e\rangle| + 1 = \#(e) = \#(e^{[p]})$. $\square$

Example 2.3.11. Operational semantics of the expression $e = a \oplus_{\frac{3}{4}} a ; a^{[\frac{1}{4}]} ; a$ correspond to the following GPTS:

$$\begin{tikzcd}
a \oplus_{\frac{3}{4}} a ; a^{[\frac{1}{4}]} ; a \arrow[rr, "a \mid \frac{3}{4}"] \arrow[dr, "a \mid \frac{1}{4}"'] & & 1 \overset{1}{\Longrightarrow} \checkmark \\
& a^{[\frac{1}{4}]} ; a \arrow[ur, "a \mid \frac{3}{4}"'] \arrow[loop right, "a \mid \frac{1}{4}"] &
\end{tikzcd}$$

One can observe that the transition system above for e is isomorphic to the one starting in q_2 in Example 2.1.2.

Given the finite-state GPTS $(\langle e\rangle, \partial)$ associated with an expression $e \in \mathsf{PExp}$ we can define the language semantics of e as the probabilistic language $[\![e]\!] \in [0,1]^{A^*}$ generated by the state e in the GPTS $(\langle e\rangle, \partial)$.

2.3.3 Roadmap to soundness and completeness

The central aim of this chapter is to show that the axioms in Figure 2.1 are sound and complete to reason about probabilistic language equivalence of PRE, that is:

$$\begin{tikzcd}
e \equiv f \arrow[r, shift right, Rightarrow, "\text{Soundness}"'] & {[\![e]\!] = [\![f]\!]} \arrow[l, shift right, Rightarrow, "\text{Completeness}"']
\end{tikzcd}$$

We now sketch the roadmap on how we will prove these two results to ease the flow into the upcoming technical sections. Perhaps not surprisingly, the completeness direction is the most involved.

The heart of both arguments will rely on arguing that the semantics map $[\![-]\!] \colon \mathsf{PExp} \to [0,1]^{A^*}$ assigning a probabilistic language to each expression can be seen as the following composition of maps:

$$\begin{tikzcd}
\mathsf{PExp} \arrow[r, "{[-]}"] \arrow[rr, bend right, "{[\![-]\!]}"'] & \mathsf{PExp}/{\equiv} \arrow[r, "\mathsf{beh}_d"] & {[0,1]^{A^*}}
\end{tikzcd}$$

The technical core of both arguments will rely on equipping $\mathsf{PExp}/\equiv$ with a structure of a $\mathcal{G}$-coalgebra, possesing additional well-behaved PCA structure making it into a $\overline{\mathcal{G}}$-coalgebra. In the picture above $[-] \colon \mathsf{PExp} \to \mathsf{PExp}/\equiv$ is a quotient map taking expressions to their equivalence class modulo the axioms of $\equiv$, while $\mathsf{beh}_d \colon \mathsf{PExp}/\equiv \to [0,1]^{A^*}$ is a final $\mathcal{G}$-coalgebra homomorphism taking each equivalence class to the corresponding probabilistic language. In such a case, soundness follows as a sequence of three steps:

$$e \equiv f \Rightarrow [e] = [f] \Rightarrow \mathsf{beh}_d([e]) = \mathsf{beh}_d([f]) \Rightarrow [\![e]\!] = [\![f]\!] \tag{2.3}$$

In general, obtaining the appropriate transition system structure on $\mathsf{PExp}/_{\equiv}$ needs a couple of intermediate steps, which then lead to soundness:

1. We first prove the soundness of a subset of the axioms of Figure 2.1: omitting (S0) and (D2) yields a sound inference system, which we call $\equiv_b$, with respect to a finer equivalence–probabilistic bisimilarity as defined by Larsen and Skou [LS91] (Lemma 2.4.1). As a consequence, there exists a deterministic transition system structure on the set $\mathcal{D}\mathsf{PExp}/\equiv_b$, such that $\mathcal{D}[-]_{\equiv_b} \colon \mathcal{D}\mathsf{PExp} \to \mathcal{D}\mathsf{PExp}/\equiv_b$ is a $\mathcal{G}$-coalgebra homomorphism (Lemma 2.4.2).

2. We then prove that the set of expressions modulo the bisimilarity axioms, that is $\mathsf{PExp}/_{\equiv_b}$, has the structure of a positive convex algebra $\alpha_{\equiv_b} \colon \mathcal{D}\mathsf{PExp}/\equiv_b \to \mathsf{PExp}/\equiv_b$ (Lemma 2.4.10). This allows us to flatten a distribution over equivalence classes into a single equivalence class. This proof makes use of a *fundamental theorem* decomposing expressions (Theorem 2.4.3). Additionally, we obtain that the coarser quotient $\mathsf{PExp}/\equiv$ also has a PCA structure (Lemma 2.4.12), that will become handy in the proof of completeness.

3. With the above result, we equip the set $\mathsf{PExp}/\equiv_b$ with a $\mathcal{G}$-coalgebra structure and show that the positive convex algebra structure map on $\mathsf{PExp}/\equiv_b$ is also a $\mathcal{G}$-coalgebra homomorphism from $\mathcal{D}\mathsf{PExp}/\equiv_b$ into it (Lemma 2.4.13).

4. Through a simple argument (this step encapsulates the key part of the soundness argument), we show that there exists a unique deterministic transition system structure on the coarser quotient, that is $\mathsf{PExp}/{\equiv}$, that makes further identification using axioms (S0) and (D2) (denoted $[-]_{\equiv} \colon \mathsf{PExp}/{\equiv_b} \to \mathsf{PExp}/{\equiv}$) into a $\mathcal{G}$-coalgebra homomorphism (Lemma 2.4.14). We compose all homomorphisms into a map of the type $\mathcal{D}\mathsf{PExp} \to \mathsf{PExp}/{\equiv}$ and show the correspondence of the probabilistic language of the state $[e]$ in the above mentioned $\mathcal{G}$-coalgebra with the one of δ_e in the determinisation of Antimirov GPTS (Lemma 2.4.19), thus establishing soundness (Theorem 2.4.20).

As much as our proof of soundness is not a straightforward inductive argument like in ordinary regular expressions, it immediately sets up the stage for the completeness argument. To obtain completeness we want to reverse all implications in Equation (2.3)–and they all are easily reversible except $[e] = [f] \Rightarrow \mathsf{beh}_d([e]) = \mathsf{beh}_d([f])$. To obtain this reverse implication we will need to show that beh_d is *injective*. We will do this, by showing that the (algebraically structured) coalgebra on $\mathsf{PExp}/{\equiv}$ satisfies a universal property of the rational fixpoint, that generalises the idea of regular languages representing finite-state deterministic automata.

As we will see in Section 2.5, determinising a finite-state GPTS can lead to an infinite state $\mathcal{G}$-coalgebra. Instead, we will rely on the theory of locally finitely presentable categories to characterise finite behaviour. It turns out, each determinisation of a finite-state GPTS carries a structure of a positive convex algebra, that is *free finitely generated*. Thanks to the work of Milius [Mil18] and Sokolova & Woracek [SW18] on *proper functors*, we will see that establishing that $\mathsf{PExp}/{\equiv}$ is isomorphic to the rational fixpoint boils down to showing uniqueness of homomorphisms from determinisations of finite-state GPTS. We will reduce this problem to converting GPTS to language equivalent expressions through the means of axiomatic manipulation using a procedure reminiscent of Brzozowski's equation solving method [Brz64] for converting DFAs to regular expressions. As a corollary, we will obtain an analogue of (one direction of) Kleene's theorem for GPTS and PRE. To sum up, the completeness result is obtained in 4 steps:

1. We show that the structure map of $\mathcal{G}$-coalgebra on $\mathsf{PExp}/\equiv$ constructed in previous step is in fact a PCA homomorphism, thus making it into a $\overline{\mathcal{G}}$-coalgebra (Lemma 2.5.1).

2. We show that determinisations of GPTS, as well as $\overline{\mathcal{G}}$-coalgebra structure on $\mathsf{PExp}/\equiv$, are precisely coalgebras for the functor $\hat{\mathcal{G}} \colon \mathsf{PCA} \to \mathsf{PCA}$ that is proper and a subfunctor of $\overline{\mathcal{G}}$.

3. Following a traditional pattern in completeness proofs [Sal66; Bac76; Mil84], we represent GPTS as *left-affine* systems of equations within the calculus and show that these systems have *unique* solutions up to provable equivalence (Theorem 2.5.19).

4. We then show that these solutions are in 1-1 correspondence with well-behaved maps from $\hat{\mathcal{G}}$-coalgebras obtained from determinising finite-state GPTS into the $\hat{\mathcal{G}}$-coalgebra on $\mathsf{PExp}/\equiv$ (Lemma 2.5.7).

5. Finally, we use this correspondence together with an abstract categorical argument to show that $\hat{\mathcal{G}}$-coalgebra structure on $\mathsf{PExp}/\equiv$ has a universal property of the rational fixpoint (Corollary 2.5.24) that eventually implies injectivity of beh_d, establishing completeness (Theorem 2.5.27).

2.4 Soundness

We are now ready to execute the roadmap to soundness described in Section 2.3.3.

2.4.1 Step 1: Soundness with respect to bisimilarity

We first check that a subset of axioms generating $\equiv$ is sound with respect to bisimilarity of $\mathcal{DF}$-coalgebras, which is a coarser notion of equivalence than probabilistic language equivalence. Let $\equiv_b \subseteq \mathsf{PExp} \times \mathsf{PExp}$ denote the least congruence relation closed under the axioms on Figure 2.1 except (S0) and (D2). We will use the following notation for the quotient maps associated with $\equiv$ and $\equiv_b$:

$$\begin{tikzcd}
\mathsf{PExp} \arrow[r, "{[-]_{\equiv_b}}"'] \arrow[rr, bend left, "{[-]}"] & \mathsf{PExp}/{\equiv_b} \arrow[r, "{[-]_{\equiv}}"'] & \mathsf{PExp}/{\equiv}
\end{tikzcd}$$

A straightforward induction on the length derivation of $\equiv_b$ allows us to show that this relation is a bisimulation equivalence on $(\mathsf{PExp}, \partial)$. As mentioned before, in the case of GPTS this notion corresponds to bisimulation equivalences in the sense of Larsen and Skou [LS91]. Due to readability concerns, the proof of that result is delegated to **??**.

Lemma 2.4.1. *The relation* $\equiv_b \subseteq \mathsf{PExp} \times \mathsf{PExp}$ *is a bisimulation equivalence.*

As a consequence of Lemma 2.4.1 and **??**, there exists a unique coalgebra structure $[\partial]_{\equiv_b} \colon \mathsf{PExp}/{\equiv_b} \to \mathcal{DF}\mathsf{PExp}/{\equiv_b}$, which makes the quotient map $[-]_{\equiv_b} \colon \mathsf{PExp} \to \mathsf{PExp}/{\equiv_b}$ into a $\mathcal{DF}$-coalgebra homomorphism from $(\mathsf{PExp}, \partial)$ to $(\mathsf{PExp}/{\equiv_b}, [\partial]_{\equiv_b})$. It turns out, that upon converting those $\mathcal{DF}$-coalgebras to $\mathcal{GD}$-coalgebras using the natural transformation $\rho \colon \mathcal{DF} \to \mathcal{GD}$ and determinising them, $\mathcal{D}[-]_{\equiv_b} \colon \mathcal{D}\mathsf{PExp} \to \mathcal{D}\mathsf{PExp}/{\equiv_b}$ becomes a homomorphism between the determinisations.

Lemma 2.4.2. $\mathcal{D}[-]_{\equiv_b} \colon \mathcal{D}\mathsf{PExp} \to \mathcal{D}\mathsf{PExp}/{\equiv_b}$ *is a* $\mathcal{G}$*-coalgebra homomorphism from* $(\mathcal{D}\mathsf{PExp}, (\rho_{\mathsf{PExp}} \circ \partial)^\sharp)$ *to* $(\mathcal{D}\mathsf{PExp}/{\equiv_b}, (\rho_{\mathsf{PExp}/\equiv_b} \circ [\partial]_{\equiv_b})^\sharp)$. *In other words, the following diagram commutes:*

$$\begin{tikzcd}
 & \mathcal{D}\mathsf{PExp} \arrow[rr, "{\mathcal{D}[-]_{\equiv_b}}"] \arrow[dddl, bend right, "{(\gamma_{\mathsf{PExp}} \circ \partial)^\sharp}"] & & \mathcal{D}\mathsf{PExp}/{\equiv_b} \arrow[dddl, bend right, "{(\gamma_{\mathsf{PExp}/\equiv_b} \circ [\partial]_{\equiv_b})^\sharp}"] \\
\mathsf{PExp} \arrow[ur, "\eta_{\mathsf{PExp}}"] \arrow[rr, "{[-]_{\equiv_b}}"'] \arrow[d, "\partial"'] & & \mathsf{PExp}/{\equiv_b} \arrow[ur, "\eta_{\mathsf{PExp}/\equiv_b}"] \arrow[d, "{[\partial]_{\equiv_b}}"'] & \\
\mathcal{DF}\mathsf{PExp} \arrow[d, "\gamma_{\mathsf{PExp}}"'] & & \mathcal{DF}\mathsf{PExp}/{\equiv_b} \arrow[d, "\gamma_{\mathsf{PExp}/\equiv_b}"'] & \\
\mathcal{GD}\mathsf{PExp} \arrow[rr, "{\mathcal{GD}[-]_{\equiv_b}}"] & & \mathcal{GD}\mathsf{PExp}/{\equiv_b} &
\end{tikzcd}$$

Proof. The top face of the diagram commutes by naturality of η. Front face of the diagram commutes because $[-]_{\equiv_b}$ is $\mathcal{DF}$-coalgebra homomorphism and because of [Rut00, Theorem 15.1]. The sides of the diagram commute because of the free-forgetful adjunction between Set and PCA. Finally, the commutativity of the square at the back of the diagram above follows from [Sil+10, Theorem 4.1]. $\square$

2.4.2 Step 2a: Fundamental theorem

We show that every PRE is provably equivalent (modulo the axioms of $\equiv_b$) to a decomposition involving sub-expressions obtained in its small-step semantics. This property, often referred to as the fundamental theorem (in analogy with the fundamental theorem of calculus) is useful in proving soundness. In order to encode elements of $\mathcal{F}\mathsf{PExp}$ using the syntax of PExp, we define a function $\exp\colon \mathcal{F}\mathsf{PExp} \to \mathsf{PExp}$, given by $\exp(\checkmark) = 1$ and $\exp(a, e') = a; e'$ for all $a \in A$ and $e' \in \mathsf{PExp}$. To be able to syntactically express finitely supported subdistributions, we will use n-ary convex sum of elements of PExp obeying the axioms of positive convex algebras, that exists because of Proposition 2.2.2 and axioms ($\mathsf{C1}$-$\mathsf{C4}$). Using it, we can state the following:

Theorem 2.4.3. *For all $e \in \mathsf{PExp}$ we have that*

$$e \equiv_b \bigoplus_{d \in \operatorname{supp}(\partial(e))} \partial(e)(d) \cdot \exp(d)$$

Before we give the proof of the result above, we start by establishing a couple of intermediate results. Firstly, we show that the binary probabilistic choice satisfies the following identities:

Lemma 2.4.4. *The following facts are derivable in $\equiv_b$*

1. $e \oplus_p (f \oplus_q g) \equiv_b \left(e \oplus_{\frac{p}{1-(1-p)(1-q)}} f\right) \oplus_{1-(1-p)(1-q)} g$
2. $(e \oplus_p f) \oplus_q (g \oplus_p h) \equiv_b (e \oplus_q g) \oplus_p (f \oplus_q h)$

Proof. For 1, let $k = \frac{p}{1-(1-p)(1-q)}$ and $l = 1-(1-p)(1-q)$. We derive the following:

$$e \oplus_p (f \oplus_q g) \equiv_b (f \oplus_q g) \oplus_{1-p} e \tag{C3}$$

$$
\begin{aligned}
&\equiv_b (g \oplus_{1-q} f) \oplus_{1-p} e && \text{(C3)}\\
&\equiv_b g \oplus_{1-l} (f \oplus_{1-k} e) && \text{(C4)}\\
&\equiv_b (f \oplus_{1-k} e) \oplus_l g && \text{(C3)}\\
&\equiv_b (e \oplus_k f) \oplus_l g && \text{(C3)}
\end{aligned}
$$

For ② we show the following:

$$
\begin{aligned}
(e \oplus_p f) \oplus_q (g \oplus_p h) &\equiv_b e \oplus_{pq} \left(f \oplus_{\frac{1-pq}{1-pq}} (g \oplus_p h) \right) && \text{(C4)}\\
&\equiv_b e \oplus_{pq} \left((g \oplus_p h) \oplus_{\frac{1-q}{1-pq}} f \right) && \text{(C3)}\\
&\equiv_b e \oplus_{pq} \left(g \oplus_{\frac{p(1-q)}{1-pq}} \left(h \oplus_{1-q} f \right) \right) && \text{(C4)}\\
&\equiv_b e \oplus_{pq} \left(g \oplus_{\frac{p(1-q)}{1-pq}} \left(f \oplus_q h \right) \right) && \text{(C3)}\\
&\equiv_b \left(e \oplus_q g \right) \oplus_p \left(f \oplus_q h \right) && \text{(①)}
\end{aligned}
$$

This completes the proof. □

Then, we argue that any non-empty n-ary convex sum can be expressed as a binary probabilistic choice and an $(n-1)$-nary convex sum.

Lemma 2.4.5. *Let $\{p_i\}_{i \in I}$ and $\{e_i\}_{i \in I}$ be non-empty collections indexed by a finite set I, such that for all $i \in I$, $p_i \in [0,1]$ and $e_i \in \mathsf{PExp}$. For any $j \in I$ it holds that:*

$$
\bigoplus_{i \in I} p_i \cdot e_i \equiv_b e_j \oplus_{p_j} \left(\bigoplus_{i \in I \setminus \{j\}} \frac{p_i}{1-p_j} \cdot e_i \right)
$$

Proof. In the edge case, when $p_j = 1$ (and therefore $I = \{j\}$) we have that $\bigoplus_{i \in I} p_i \cdot e_i \equiv_b e_j$ and therefore

$$
\begin{aligned}
e_j &\equiv_b e_j \oplus_1 0 && \text{(C2)}\\
&\equiv_b e_j \oplus_{p_j} \left(\bigoplus_{i \in \emptyset} \frac{p_i}{1-p_j} \cdot e_i \right) && \text{(Def. of empty n-ary convex sum)}
\end{aligned}
$$

$$\equiv_b e_j \oplus_{p_j} \left(\bigoplus_{i \in I \setminus \{j\}} \frac{p_i}{1 - p_j} \cdot e_i \right) \qquad (I = \{j\})$$

Note that despite the fact that $p_j = 1$, the n-ary sum on the right is well-defined as it ranges over an empty index set and thus division by zero never happens. The remaining case, when $p_j \neq 1$ holds because of Proposition 2.2.2. □

Using the above result, we can also split the normal form used in Theorem 2.4.3 into two parts; one describing acceptance and one describing labelled transitions.

Lemma 2.4.6. *For all $e \in \mathsf{PExp}$,*

$$\bigoplus_{d \in \mathrm{supp}(\partial(e))} \partial(e)(d) \cdot \exp(d) \equiv_b 1 \oplus_{\partial(e)(\checkmark)} \left(\bigoplus_{d \in \mathrm{supp}(\partial(e)) \setminus \{\checkmark\}} \frac{\partial(e)(d)}{1 - \partial(e)(\checkmark)} \cdot \exp(d) \right)$$

Proof. If $\mathrm{supp}(\partial(e)) = \emptyset$, then

$$\begin{aligned}
\bigoplus_{d \in \mathrm{supp}(\partial(e))} \partial(e)(d) \cdot \exp(d) &\equiv_b 0 && \text{(Def. of empty } n\text{-ary convex sum)} \\
&\equiv_b 0 \oplus_1 1 && (\mathsf{C2}) \\
&\equiv_b 1 \oplus_0 0 && (\mathsf{C3}) \\
&\equiv_b 1 \oplus_{\partial(e)(\checkmark)} 0 && (\partial(e)(\checkmark) = 0) \\
&\equiv_b 1 \oplus_{\partial(e)(\checkmark)} \left(\bigoplus_{d \in \emptyset} \frac{\partial(e)(d)}{1 - \partial(e)(\checkmark)} \cdot \exp(d) \right) \\
&\equiv_b 1 \oplus_{\partial(e)(\checkmark)} \left(\bigoplus_{d \in \mathrm{supp}(\partial(e)(\checkmark)) \setminus \{\checkmark\}} \frac{\partial(e)(d)}{1 - \partial(e)(\checkmark)} \cdot \exp(d) \right)
\end{aligned}$$

The remaining case when $\mathrm{supp}(\partial(e)) \neq 0$ holds by Lemma 2.4.5 and the fact that $\exp(\checkmark) = 1$. □

Finally, we generalise the axiom (D1) to n-ary convex sums.

Lemma 2.4.7. *Let $f \in \mathsf{PExp}$, I be a finite index set and let $\{p_i\}_{i\in I}$ and $\{e_i\}_{i\in I}$ indexed collections of probabilities and expressions respectively. Then,*

$$\left(\bigoplus_{i\in I} p_i \cdot e_i\right) ; f \equiv_b \bigoplus_{i\in I} p_i \cdot e_i ; f$$

Proof. By induction. If $I = \emptyset$, then using $(\mathsf{0S})$ we can show that

$$\left(\bigoplus_{i\in I} p_i \cdot e_i\right) ; f \equiv_b \mathsf{0} ; f \equiv_b \mathsf{0} \equiv_b \bigoplus_{i\in I} p_i \cdot e_i ; f$$

If there exists $j \in I$, such that $p_j = 1$, then

$$\left(\bigoplus_{i\in I} p_i \cdot e_i\right) ; f \equiv_b e_j ; f \equiv_b \left(\bigoplus_{i\in I} p_i \cdot e_i ; f\right)$$

Finally, for the induction step, we have that

$$\begin{aligned}
\left(\bigoplus_{i\in I} p_i \cdot e_i\right) ; f &\equiv_b \left(e_j \oplus_{p_j} \left(\bigoplus_{i\in I\setminus\{j\}} \frac{p_i}{1-p_j} \cdot e_i\right)\right) ; f \\
&\equiv_b e_j ; f \oplus_{p_j} \left(\bigoplus_{i\in I\setminus\{j\}} \frac{p_i}{1-p_j} \cdot e_i\right) ; f && (\mathsf{D1}) \\
&\equiv_b e_j ; f \oplus_{p_j} \left(\bigoplus_{i\in I\setminus\{j\}} \frac{p_i}{1-p_j} \cdot e_i ; f\right) && \text{(Induction hypothesis)} \\
&\equiv_b \left(\bigoplus_{i\in I} p_i \cdot e_i ; f\right)
\end{aligned}$$

□

We now have all the ingredients to show the fundamental theorem.

Proof of Theorem 2.4.3. We proceed by the structural induction on $e \in \mathsf{PExp}$. For the base cases, we have the following:

$\boxed{e = \mathsf{0}}$

$$\mathsf{0} \equiv_b \bigoplus_{d\in\emptyset} \partial(\mathsf{0})(d) \cdot \exp(d) \qquad \text{(Proposition 2.2.3)}$$

$$\equiv_b \bigoplus_{d \in \mathrm{supp}(\partial(0))} \partial(0)(d) \cdot \exp(d) \qquad (\mathrm{supp}(\partial(0)) = \emptyset)$$

$\boxed{e = 1}$

$$\begin{aligned} 1 &\equiv_b \exp(\checkmark) && \text{(Def. of exp)} \\ &\equiv_b \bigoplus_{d \in \mathrm{supp}(\partial(1))} \partial(1)(d) \cdot \exp(d) && (\partial(1) = \delta_{\checkmark}) \end{aligned}$$

$\boxed{e = a}$

$$\begin{aligned} a &\equiv_b a \,;1 && \text{(S1)} \\ &\equiv_b \exp((a, \checkmark)) && \text{(Def. of exp)} \\ &\equiv_b \bigoplus_{d \in \mathrm{supp}(\partial(a))} \partial(a)(d) \cdot \exp(d) && (\partial(a) = \delta_{(a,\checkmark)}) \end{aligned}$$

We now move on to inductive steps.

$\boxed{e = f \oplus_p g}$

$$\begin{aligned} & f \oplus_p g \\ \equiv_b & \left(\bigoplus_{d \in \mathrm{supp}(\partial(f))} \partial(f)(d) \cdot \exp(d) \right) \oplus_p \left(\bigoplus_{d \in \mathrm{supp}(\partial(g))} \partial(g)(d) \cdot \exp(g) \right) \\ & \text{(Induction hypothesis)} \\ \equiv_b & \left(\bigoplus_{d \in \mathrm{supp}(\partial(f \oplus_p g))} \partial(f)(d) \cdot \exp(d) \right) \oplus_p \left(\bigoplus_{d \in \mathrm{supp}(\partial(f \oplus_p g))} \partial(g)(d) \cdot \exp(g) \right) \\ & \text{(Proposition 2.2.3)} \\ \equiv_b & \bigoplus_{d \in \mathrm{supp}(\partial(f \oplus_p g))} (p\partial(f)(d) + (1-p)\partial(g)(d)) \cdot \exp(d) \qquad \text{(Barycenter axiom)} \\ \equiv_b & \bigoplus_{d \in \mathrm{supp}(\partial(f \oplus_p g))} \partial(f \oplus_p g)(d) \cdot \exp(d) \qquad \text{(Definition 2.3.6)} \end{aligned}$$

$\boxed{e = f\,;g}$

$$
\begin{aligned}
f\,;g &\equiv_b \left(\bigoplus_{d\in\mathrm{supp}(\partial(f))} \partial(f)(d)\cdot\exp(d)\right);g && \text{(Induction hypothesis)}\\
&\equiv_b \left(1 \oplus_{\partial(f)(\checkmark)} \left(\bigoplus_{d\in\mathrm{supp}(\partial(f))\setminus\{\checkmark\}} \frac{\partial(f)(d)}{1-\partial(f)(\checkmark)}\cdot\exp(d)\right)\right);g && \text{(Lemma 2.4.6)}\\
&\equiv_b \left(1\,;g \oplus_{\partial(f)(\checkmark)} \left(\bigoplus_{d\in\mathrm{supp}(\partial(f))\setminus\{\checkmark\}} \frac{\partial(f)(d)}{1-\partial(f)(\checkmark)}\cdot\exp(d)\right);g\right) && \text{(D1)}\\
&\equiv_b g \oplus_{\partial(f)(\checkmark)} \left(\bigoplus_{d\in\mathrm{supp}(\partial(f))\setminus\{\checkmark\}} \frac{\partial(f)(d)}{1-\partial(f)(\checkmark)}\cdot\exp(d)\,;g\right)\\
& && \text{(Lemma 2.4.7 and 1S)}\\
&\equiv_b \left(\bigoplus_{d\in\mathrm{supp}(\partial(g))} \partial(g)(d)\cdot\exp(d)\right)\\
&\qquad \oplus_{\partial(f)(\checkmark)} \left(\bigoplus_{d\in\mathrm{supp}(\partial(f))\setminus\{\checkmark\}} \frac{\partial(f)(d)}{1-\partial(f)(\checkmark)}\cdot\exp(d)\,;g\right)\\
& && \text{(Induction hypothesis)}\\
&\equiv_b \left(\bigoplus_{d\in\mathrm{supp}(\partial(f;g))} \partial(g)(d)\cdot\exp(d)\right)\\
&\qquad \oplus_{\partial(f)(\checkmark)} \left(\bigoplus_{d\in\mathrm{supp}(\partial(f))\setminus\{\checkmark\}} \frac{\partial(f)(d)}{1-\partial(f)(\checkmark)}\cdot\exp(d)\,;g\right) && \text{(Proposition 2.2.3)}
\end{aligned}
$$

Now, we simplify the subexpression on the right part of the convex sum. Define $n\colon F\mathsf{PExp} \to [0,1]$ to be:

$$
n(d) = \begin{cases} \partial(f)(a,f') & d = (a, f'\,;g) \\ 0 & \text{otherwise} \end{cases}
$$

By applying Proposition 2.2.3 and the preceding definition, it follows that:

$$
\bigoplus_{d\in\mathrm{supp}(\partial(f))\setminus\{\checkmark\}} \frac{\partial(f)(d)}{1-\partial(f)(\checkmark)}\cdot\exp(d)\,;g \equiv_b \bigoplus_{d\in\mathrm{supp}(\partial(f;g))} \frac{n(d)}{1-\partial(f)(\checkmark)}\cdot\exp(d)
$$

By combining this with the previous derivation and applying the barycenter axiom, we can conclude that

$$f\,;g \equiv_b \bigoplus_{d\in\mathrm{supp}(\partial(f;g))} (\partial(f)(\checkmark)\partial(g)(d)+n(d))\cdot\exp(d)$$

Combining it with the previous derivation, using the barycenter axiom, we can show that:

$$f\,;g \equiv_b \bigoplus_{d\in\mathrm{supp}(\partial(f;g))} (\partial(f)(\checkmark)\partial(g)(d)+n(d))\cdot\exp(d)$$

Observe that for $d=(a,f'\,;g)$, we obtain

$$\partial(f)(\checkmark)\partial(g)(d)+n(d)=\partial(f)(\checkmark)\partial(g)(a,f'\,;g)+\partial(f)(a,f')=\partial(f\,;g)(d)$$

When $d=\checkmark$, it follows that

$$\partial(f)(\checkmark)\partial(g)(d)+n(d)=\partial(f)(\checkmark)\partial(g)(d)=\partial(f\,;g)(d)$$

In all remaining cases, both functions assign the value 0 to d. Consequently, we conclude that

$$f\,;g \equiv_b \bigoplus_{d\in\mathrm{supp}(\partial(f;g))} \partial(f\,;g)(d)\cdot\exp(d)$$

which establishes the desired result for this case.

$\boxed{e=f^{[p]}}$

We begin by considering the case where $\partial(f)(\checkmark)=1$ and $p=1$.

$$\begin{aligned}
f^{[p]} &\equiv_b \left(\bigoplus_{d\in\mathrm{supp}(\partial(f))} \partial(f)(d)\cdot\exp(d)\right)^{[1]} && \text{(Induction hypothesis)}\\
&\equiv_b \mathsf{1}^{[1]} && (\partial(f)(\checkmark)=1)\\
&\equiv_b \mathsf{0} && \text{(Div)}\\
&\equiv_b \bigoplus_{d\in\mathrm{supp}\left(\partial\left(f^{[1]}\right)\right)} \partial\left(f^{[1]}\right)(d)\cdot\exp(d)
\end{aligned}$$

Otherwise, we first apply the (Tight) axiom to the loop body as follows:

$$
\begin{aligned}
f^{[p]} &\equiv_b \left(\bigoplus_{d \in \mathrm{supp}(\partial(f))} \partial(f)(d) \cdot \exp(d) \right)^{[p]} && \text{(Induction hypothesis)} \\
&\equiv_b \left(1 \oplus_{\partial(f)(\checkmark)} \left(\bigoplus_{d \in \mathrm{supp}(\partial(f)) \setminus \{\checkmark\}} \frac{\partial(f)(d)}{1 - \partial(f)(\checkmark)} \cdot \exp(d) \right) \right)^{[p]} && \text{(Lemma 2.4.6)} \\
&\equiv_b \left(\left(\bigoplus_{d \in \mathrm{supp}(\partial(f)) \setminus \{\checkmark\}} \frac{\partial(f)(d)}{1 - \partial(f)(\checkmark)} \cdot \exp(d) \right) \oplus_{1-\partial(f)(\checkmark)} 1 \right)^{[p]} && \text{(C3)} \\
&\equiv_b \left(\bigoplus_{d \in \mathrm{supp}(\partial(f)) \setminus \{\checkmark\}} \frac{\partial(f)(d)}{1 - \partial(f)(\checkmark)} \cdot \exp(d) \right)^{\left[\frac{(1-\partial(f)(\checkmark))p}{1-p\partial(f)(\checkmark)}\right]} && \text{(Tight)}
\end{aligned}
$$

For convenience, we denote by $g^{[r]}$ the expression obtained from the preceding derivation. We proceed by applying the (Unroll) axiom.

$$
\begin{aligned}
g^{[r]} &\equiv_b \left(\bigoplus_{d \in \mathrm{supp}(\partial(f)) \setminus \{\checkmark\}} \frac{\partial(f)(d)}{1 - \partial(f)(\checkmark)} \cdot \exp(d) \right) ; g^{[r]} \oplus_r 1 && \text{(Unroll)} \\
&\equiv_b \left(\bigoplus_{d \in \mathrm{supp}(\partial(f)) \setminus \{\checkmark\}} \frac{\partial(f)(d)}{1 - \partial(f)(\checkmark)} \cdot \exp(d) \right) ; f^{[p]} \oplus_r 1 && (f^{[p]} \equiv_b g^{[r]}) \\
&\equiv_b \left(\bigoplus_{d \in \mathrm{supp}(\partial(f)) \setminus \{\checkmark\}} \frac{\partial(f)(d)}{1 - \partial(f)(\checkmark)} \cdot \exp(d) ; f^{[p]} \right) \oplus_r 1 && \text{(Lemma 2.4.7)} \\
&\equiv_b \left(\bigoplus_{d \in \mathrm{supp}(\partial(f)) \setminus \{\checkmark\}} \frac{\partial(f)(d)}{1 - \partial(f)(\checkmark)} \cdot \exp(d) ; f^{[p]} \right) \\
&\qquad \oplus_r \left(\bigoplus_{d \in \mathrm{supp}\left(\partial\left(f^{[p]}\right)\right)} \delta_{\checkmark}(d) \cdot \exp(d) \right)
\end{aligned}
$$

Next, we simplify the left-hand side of the binary convex sum. Define $n\colon F\mathsf{PExp} \to [0,1]$ as follows

$$
n(d) = \begin{cases} \partial(f)(a, f') & d = \left(a, f' ; f^{[p]}\right) \\ 0 & \text{otherwise} \end{cases}
$$

By applying Proposition 2.2.3 and the definition above, we obtain

$$\bigoplus_{d \in \mathrm{supp}(\partial(f)) \setminus \{\checkmark\}} \frac{\partial(f)(d)}{1-\partial(f)(\checkmark)} \cdot \exp(d) \,;\, f^{[p]} \equiv_b \bigoplus_{d \in \mathrm{supp}\left(\partial\left(f^{[p]}\right)\right)} \frac{n(d)}{1-\partial(f)(\checkmark)} \cdot \exp(d)$$

By combining the above with the previous derivation and applying the barycenter axiom, we obtain:

$$f^{[p]} \equiv_b \bigoplus_{d \in \mathrm{supp}\left(\partial\left(f^{[p]}\right)\right)} \left(\frac{pn(d)}{1-p\partial(f)(\checkmark)} + \frac{1-p\delta_{\checkmark}(d)}{1-\partial(f)(\checkmark)p} \right) \cdot \exp(d)$$

Observe that for $d = (a, f' \,; g)$, we obtain

$$\frac{pn(d)}{1-p\partial(f)(\checkmark)} + \frac{1-p\delta_{\checkmark}(d)}{1-p\partial(f)(\checkmark)} = \frac{p\partial(f)(a,f')}{1-p\partial(f)(\checkmark)} = \partial\left(f^{[p]}\right)(d)$$

When $d = \checkmark$, it follows that

$$\frac{pn(d)}{1-p\partial(f)(\checkmark)} + \frac{1-p\delta_{\checkmark}(d)}{1-p\partial(f)(\checkmark)} = \frac{1-p}{1-p\partial(f)(\checkmark)} = \partial\left(f^{[p]}\right)(d)$$

In all remaining cases, we have that

$$\frac{pn(d)}{1-p\partial(f)(\checkmark)} + \frac{1-p\delta_{\checkmark}(d)}{1-p\partial(f)(\checkmark)} = 0 = \partial\left(f^{[p]}\right)(d)$$

Thus, we obtain the following:

$$f^{[p]} \equiv_b \bigoplus_{d \in \mathrm{supp}\left(\partial\left(f^{[p]}\right)\right)} \partial\left(f^{[p]}\right)(d) \cdot \exp(d)$$

This establishes the desired result. □

A direct corollary of the result established above is that every loop is provably equivalent to a loop whose body does not assign any probability to transitions to $\checkmark$.

Corollary 2.4.8 (Productive loop)**.** *Let* $e \in \mathsf{PExp}$ *and* $p \in [0,1]$. *We have that* $e^{[p]} \equiv_b f^{[r]}$ *for some* $f \in \mathsf{PExp}$ *and* $r \in [0,1]$, *such that* $E(f) = 0$.

Proof. If $\partial(e)(\checkmark) = 1$ and $p = 1$, then it follows that:

$$
\begin{aligned}
e^{[1]} &\equiv_b \left(\bigoplus_{d \in \mathrm{supp}(\partial(e))} \partial(e)(d) \cdot \exp(d) \right)^{[1]} && \text{(Theorem 2.4.3)} \\
&\equiv_b \mathsf{1}^{[1]} \\
&\equiv_b \mathsf{0} && \text{(Div)} \\
&\equiv_b \mathsf{0} \mathbin{;} \mathsf{0} \oplus_1 \mathsf{1} && \text{(0S and C2)} \\
&\equiv_b \mathsf{0}^{[1]} && \text{(Unique fixpoint rule and } E(\mathsf{0}) = 0)
\end{aligned}
$$

Therefore, $e^{[1]} = \mathsf{0}^{[1]}$. In this case, it follows that $E(\mathsf{0}) = 0$. In the remaining cases, we obtain the following:

$$
\begin{aligned}
e^{[p]} &\equiv_b \left(\bigoplus_{d \in \mathrm{supp}(\partial(e))} \partial(e)(d) \cdot \exp(d) \right)^{[p]} && \text{(Theorem 2.4.3)} \\
&\equiv_b \left(\mathsf{1} \oplus_{\partial(e)(\checkmark)} \left(\bigoplus_{d \in \mathrm{supp}(\partial(e)) \setminus \{\checkmark\}} \frac{\partial(e)(d)}{1 - \partial(e)(\checkmark)} \cdot \exp(d) \right) \right)^{[p]} && \text{(Lemma 2.4.6)} \\
&\equiv_b \left(\left(\bigoplus_{d \in \mathrm{supp}(\partial(e)) \setminus \{\checkmark\}} \frac{\partial(e)(d)}{1 - \partial(e)(\checkmark)} \cdot \exp(d) \right) \oplus_{1 - \partial(e)(\checkmark)} \mathsf{1} \right)^{[p]} && \text{(C3)} \\
&\equiv_b \left(\left(\bigoplus_{(a,e') \in \mathrm{supp}(\partial(e))} \frac{\partial(e)(a,e')}{1 - \partial(e)(\checkmark)} \cdot a \mathbin{;} e' \right) \oplus_{1 - \partial(e)(\checkmark)} \mathsf{1} \right)^{[p]} && \text{(Def. of exp)} \\
&\equiv_b \left(\bigoplus_{(a,e') \in \mathrm{supp}(\partial(e))} \frac{\partial(e)(a,e')}{1 - \partial(e)(\checkmark)} \cdot a \mathbin{;} e' \right)^{\left[\frac{p(1 - \partial(e)(\checkmark))}{1 - p\partial(e)(\checkmark)} \right]} && \text{(Tight)}
\end{aligned}
$$

Observe that the body of the loop above is an n-ary probabilistic sum involving terms of the form $a \mathbin{;} e'$ (where $a \in A$, $e' \in \mathsf{PExp}$), for which $E(a \mathbin{;} e') = 0$. By examining the definition of the n-ary sum (Proposition 2.2.2) and the termination operator $E(-)$ (Figure 2.1), we immediately conclude that the loop body is mapped to 0 by $E(-)$, which completes the proof. □

2.4.3 Step 2b: Algebra structure

Then, we equip the set $\mathsf{PExp}/{\equiv_b}$ with a PCA structure. To do so, we first observe that as a consequence of Theorem 2.4.3, we have that $[\partial]_{\equiv_b} \colon \mathsf{PExp}/{\equiv_b} \to \mathcal{DF}\mathsf{PExp}/{\equiv_b}$ is an isomorphism.

Corollary 2.4.9. *The structure map* $[\partial]_{\equiv_b} : \mathsf{PExp}/{\equiv_b} \to \mathcal{DF}\mathsf{PExp}/{\equiv_b}$ *of the quotient coalgebra* $(\mathsf{PExp}/{\equiv_b}, [\partial]_{\equiv_b})$ *is an isomorphism in the category* Set.

Proof. Given $\nu \in \mathcal{DF}\mathsf{PExp}/{\equiv_b}$ define a function $[\partial]^{-1}_{\equiv_b} \colon \mathcal{DF}\mathsf{PExp}/{\equiv_b} \to \mathsf{PExp}/{\equiv_b}$ as follows:

$$[\partial]^{-1}_{\equiv_b}(\nu) = \left[\nu(\checkmark)\cdot 1 \oplus \left(\bigoplus_{(a,[e']_{\equiv_b})\in \mathrm{supp}(\nu)} \nu(a,[e']_{\equiv_b})\cdot a \,;e'\right)\right]_{\equiv_b}$$

First, observe that for arbitrary $\nu \in \mathcal{DF}\mathsf{PExp}/{\equiv_b}$ we have that:

$$\begin{aligned}
&\left([\partial]_{\equiv_b}\circ[\partial]^{-1}_{\equiv_b}\right)(\nu)(\checkmark)\\
&= ([\partial]_{\equiv_b}\circ[-]_{\equiv_b})\left(\nu(\checkmark)\cdot 1 \oplus \left(\bigoplus_{(a,[e']_{\equiv_b})\in \mathrm{supp}(\nu)} \nu(a,[e']_{\equiv_b})\cdot a \,;e'\right)\right)(\checkmark)\\
&= (\mathcal{DF}[-]_{\equiv_b}\circ\partial)\left(\nu(\checkmark)\cdot 1 \oplus \left(\bigoplus_{(a,[e']_{\equiv_b})\in \mathrm{supp}(\nu)} \nu(a,[e']_{\equiv_b})\cdot a \,;e'\right)\right)(\checkmark)\\
&\hspace{20em}([-]_{\equiv} \text{ is a } \mathcal{DF}\text{-coalgebra homomorphism})\\
&= \partial\left(\nu(\checkmark)\cdot 1 \oplus \bigoplus_{(a,[e']_{\equiv_b})\in \mathrm{supp}(\nu)} \nu(a,[e']_{\equiv_b})\cdot a \,;e'\right)(\checkmark)\\
&= \nu(\checkmark) \hspace{20em} (\text{Definition 2.3.6})
\end{aligned}$$

Similarly, for any $(b,[f']_{\equiv_b}) \in \mathrm{supp}(\nu)$, it follows that:

$$\begin{aligned}
&([\partial]_{\equiv_b}\circ[\partial]^{-1}_{\equiv_b})(\nu)(b,[f']_{\equiv_b})\\
&= ([\partial]_{\equiv_b}\circ[-]_{\equiv_b})\left(\nu(\checkmark)\cdot 1 \oplus \left(\bigoplus_{(a,[e']_{\equiv_b})\in \mathrm{supp}(\nu)} \nu(a,[e']_{\equiv_b})\cdot a \,;e'\right)\right)(b,[f']_{\equiv_b})
\end{aligned}$$

$$= (\mathcal{DF}[-]_{\equiv_b} \circ \partial) \left(\nu(\checkmark) \cdot 1 \oplus \left(\bigoplus_{(a,[e']_{\equiv_b}) \in \mathrm{supp}(\nu)} \nu(a,[e']_{\equiv_b}) \cdot a\,;e' \right) \right) (b,[f']_{\equiv_b})$$

([−]$_\equiv$ is a $\mathcal{DF}$-coalgebra homomorphism)

$$= \sum_{g \equiv f'} \partial \left(\nu(\checkmark) \cdot 1 \oplus \left(\bigoplus_{(a,[e']_{\equiv_b}) \in \mathrm{supp}(\nu)} \nu(a,[e']_{\equiv_b}) \cdot a\,;e' \right) \right) (b,g)$$

$$= \nu(b,[f']_{\equiv_b}) \qquad \text{(Definition 2.3.6)}$$

For the second part of the proof, let $e \in \mathsf{PExp}$. As a consequence of Theorem 2.4.3, it follows that:

$$e \equiv_b \bigoplus_{d \in \mathrm{supp}(\partial(e))} \partial(e)(d) \cdot \exp(d) \qquad \text{(Theorem 2.4.3)}$$

$$\equiv_b \partial(e)(\checkmark) \cdot 1 \oplus \left(\bigoplus_{(a,e') \in \mathrm{supp}(\partial(e))} \partial(e)(a,e') \cdot a\,;e' \right)$$

$$\equiv_b \partial(e)(\checkmark) \cdot 1 \oplus \left(\bigoplus_{(a,[e']_{\equiv_b}) \in A \times \mathsf{PExp}/{\equiv_b}} \left(\sum_{g \equiv e'} \partial(e)(a,g) \right) \cdot a\,;e' \right) \qquad \text{(Proposition 2.2.3)}$$

Next, observe that:

$$([\partial]^{-1}_{\equiv_b} \circ [\partial]_{\equiv_b})[e]_{\equiv_b} = ([\partial]^{-1}_{\equiv_b} \circ \mathcal{DF}[-]_{\equiv_b} \circ \partial)(e)$$

([−]$_\equiv$ is a $\mathcal{DF}$-coalgebra homomorphism)

$$= \left[\partial(e)(\checkmark) \cdot 1 \oplus \left(\bigoplus_{(a,[e']_{\equiv_b}) \in \mathrm{supp}((\mathcal{DF}[-]_{\equiv_b} \circ \partial)(e))} (\mathcal{DF}[-]_{\equiv_b} \circ \partial)(e)(a,[e']_{\equiv_b})) \cdot a\,;e' \right) \right]_{\equiv_b}$$

$$= \left[\partial(e)(\checkmark) \cdot 1 \oplus \left(\bigoplus_{(a,[e']_{\equiv_b}) \in A \times \mathsf{PExp}/{\equiv_b}} (\mathcal{DF}[-]_{\equiv_b} \circ \partial)(e)(a,[e']_{\equiv_b})) \cdot a\,;e' \right) \right]_{\equiv_b}$$

(Proposition 2.2.3)

$$= \left[\partial(e)(\checkmark) \cdot 1 \oplus \left(\bigoplus_{(a,[e']_{\equiv_b}) \in A \times \mathsf{PExp}/{\equiv_b}} \left(\sum_{g \equiv e'} \partial(e)(a,g) \right) \cdot a\,;e' \right) \right]_{\equiv_b} \qquad \text{(Def. of } [\partial]_{\equiv_b})$$

$$= [e]_{\equiv_b} \qquad \text{(Derivation above)}$$

This completes the proof. □

The result above allows to define a map $\alpha_{\equiv_b} : \mathcal{D}\mathsf{PExp}/{\equiv_b} \to \mathsf{PExp}/{\equiv_b}$ as the following composition of morphisms:

$$\mathcal{D}\mathsf{PExp}/{\equiv_b} \xrightarrow{\mathcal{D}[\partial]_{\equiv_b}} \mathcal{D}\mathcal{D}\mathcal{F}\mathsf{PExp}/{\equiv_b} \xrightarrow{\mu_{\mathcal{F}\mathsf{PExp}/\equiv}} \mathcal{D}\mathcal{F}\mathsf{PExp}/{\equiv_b} \xrightarrow{{[\partial]_{\equiv_b}}^{-1}} \mathsf{PExp}/{\equiv_b}$$

In fact, this map equips the set $\mathsf{PExp}/{\equiv_b}$ with a positive convex algebra structure.

Lemma 2.4.10. $(\mathsf{PExp}/{\equiv_b}, \alpha_{\equiv_b})$ *is an Eilenberg-Moore algebra for the finitely supported subdistribution monad.*

Proof. We first verify that $\alpha_{\equiv_b} \circ \eta_{\mathsf{PExp}/\equiv_b} = \mathsf{id}_{\mathsf{PExp}/\equiv_b}$.

$$\begin{aligned}
\alpha_{\equiv_b} \circ \eta_X &= [\partial]_{\equiv_b}^{-1} \circ \mu_{\mathcal{F}\mathsf{PExp}/\equiv_b} \circ \mathcal{D}[\partial]_{\equiv_b} \circ \eta_{\mathsf{PExp}/\equiv_b} && \text{(Def. of } \alpha_{\equiv_b}\text{)}\\
&= [\partial]_{\equiv_b}^{-1} \circ \mu_{\mathcal{F}\mathsf{PExp}/\equiv_b} \circ \eta_{\mathcal{D}\mathcal{F}\mathsf{PExp}/\equiv_b} \circ [\partial]_{\equiv_b} && (\eta \text{ is natural})\\
&= [\partial]_{\equiv_b}^{-1} \circ [\partial]_{\equiv_b} && \text{(Monad laws)}\\
&= \mathsf{id}_{\mathsf{PExp}/\equiv_b} && \text{(Corollary 2.4.9)}
\end{aligned}$$

Then, we show that $\alpha_{\equiv_b} \circ \mathcal{D}\alpha_{\equiv_b} = \alpha_{\equiv_b} \circ \mu_{\mathsf{PExp}/\equiv_b}$

$$\begin{aligned}
\alpha_{\equiv_b} \circ \mathcal{D}\alpha_{\equiv_b} &= [\partial]_{\equiv_b}^{-1} \circ \mu_{\mathcal{F}\mathsf{PExp}/\equiv_b} \circ \mathcal{D}[\partial]_{\equiv_b} \circ \mathcal{D}[\partial]_{\equiv_b}^{-1} \circ \mathcal{D}\mu_{\mathcal{F}\mathsf{PExp}/\equiv_b} \circ \mathcal{D}^2[\partial]_{\equiv_b}\\
& && \text{(Def. of } \alpha_{\equiv_b}\text{)}\\
&= [\partial]_{\equiv_b}^{-1} \circ \mu_{\mathcal{F}\mathsf{PExp}/\equiv_b} \circ \mathcal{D}\mu_{\mathcal{F}\mathsf{PExp}/\equiv_b} \circ \mathcal{D}^2[\partial]_{\equiv_b} && \text{(Corollary 2.4.9)}\\
&= [\partial]_{\equiv_b}^{-1} \circ \mu_{\mathcal{F}\mathsf{PExp}/\equiv_b} \circ \mu_{\mathcal{D}\mathcal{F}\mathsf{PExp}/\equiv_b} \circ \mathcal{D}^2[\partial]_{\equiv_b} && \text{(Monad laws)}\\
&= [\partial]_{\equiv_b}^{-1} \circ \mu_{\mathcal{F}\mathsf{PExp}/\equiv_b} \circ \mathcal{D}[\partial]_{\equiv_b} \circ \mu_{\mathsf{PExp}/\equiv_b} && (\mu \text{ is natural})\\
&= \alpha_{\equiv_b} \circ \mu_{\mathsf{PExp}/\equiv_b} && \text{(Def. of } \alpha_{\equiv_b}\text{)}
\end{aligned}$$

This completes the proof. □

Moreover, using the isomorphism between PCA and $\mathsf{Set}^{\mathcal{D}}$ one can calculate the concrete formula for PCA structure on $\mathsf{PExp}/{\equiv_b}$.

Lemma 2.4.11. *The* PCA *structure on* $\mathsf{PExp}/{\equiv_b}$ *is concretely given by:*

$$\boxplus_{i\in I} p_i \cdot [e_i]_{\equiv_b} = \left[\bigoplus_{i\in I} p_i \cdot e_i\right]_{\equiv_b}$$

Proof.

$$\begin{aligned}
\boxplus_{i\in I} p_i \cdot [e_i]_{\equiv_b} &= \alpha_{\equiv_b}\left(\sum_{i\in I} p_i \delta_{[e_i]_{\equiv_b}}\right) && \text{(Theorem 2.2.6)}\\
&= [\partial]_{\equiv_b}^{-1} \circ \mu_{\mathcal{F}\mathsf{PExp}/\equiv_b} \circ \mathcal{D}[\partial]_{\equiv_b}\left(\sum_{i\in I} p_i \delta_{[e_i]_{\equiv_b}}\right) && \text{(Def. of } \alpha_{\equiv_b})\\
&= [\partial]_{\equiv_b}^{-1} \circ \mu_{\mathcal{F}\mathsf{PExp}/\equiv_b}\left(\sum_{i\in I} p_i \delta_{[\partial]_{\equiv_b}([e_i]_{\equiv_b})}\right)\\
&= [\partial]_{\equiv_b}^{-1}\left(\sum_{i\in I} p_i [\partial]_{\equiv_b}([e_i]_{\equiv_b})\right)\\
&= [\partial]_{\equiv_b}^{-1}\left(\sum_{i\in I} p_i (\mathcal{D}\mathcal{F}[-]_{\equiv_b} \circ \partial)(e_i)\right) && ([-]_{\equiv} \text{ is a } \mathcal{D}\mathcal{F}\text{-coalgebra homomorphism})\\
&= \left[\left(\sum_{i\in I} p_i \partial(e_i)(\checkmark)\right) \cdot 1 \oplus \left(\bigoplus_{(a,[e']_{\equiv_b})\in A\times \mathsf{PExp}/\equiv_b} \left(\sum_{i\in I} p_i \partial(e_i)(a,[e']_{\equiv_b})\right)\right) \cdot a \,;\, e'\right]_{\equiv_b}\\
& && \text{(Def. of } [\partial]_{\equiv_b}^{-1} \text{ and Proposition 2.2.3)}\\
&= \left[\bigoplus_{i\in I} p_i \cdot \left(\partial(e_i)(\checkmark) \cdot 1 \oplus \left(\bigoplus_{(a,[e']_{\equiv_b})\in A\times \mathsf{PExp}/\equiv_b} \partial(e_i)(a,[e']_{\equiv_b}) \cdot a \,;\, e'\right)\right)\right]_{\equiv_b}\\
& && \text{(Barycenter axiom)}\\
&= \left[\bigoplus_{i\in I} p_i \cdot \left(\partial(e_i)(\checkmark) \cdot 1 \oplus \left(\bigoplus_{(a,[e']_{\equiv_b})\in \mathrm{supp}(\partial(e_i))} \partial(e_i)(a,[e']_{\equiv_b}) \cdot a \,;\, e'\right)\right)\right]_{\equiv_b}\\
& && \text{(Proposition 2.2.3)}\\
&= \left[\bigoplus_{i\in I} p_i \cdot e_i\right]_{\equiv_b}
\end{aligned}$$

The last line of the proof above follows from Theorem 2.4.3. □

We can also equip the coarser quotient, that is $\mathsf{PExp}/{\equiv}$, with a PCA structure.

Lemma 2.4.12. *The set* $\mathsf{PExp}/{\equiv}$ *can be equipped with a positive convex algebra* *structure, given by the following:*

$$\boxplus_{i\in I} p_i \cdot [e_i] = \left[\bigoplus_{i\in I} p_i \cdot e_i\right]$$

Moreover, $[-]_{\equiv} \colon \mathsf{PExp}/{\equiv_b} \to \mathsf{PExp}/{\equiv}$ *is a* PCA *homomorphism.*

Proof. The positive convex algebra structure on $\mathsf{PExp}/{\equiv}$ is well-defined, because $\equiv$ is a congruence and the definition on n-ary probabilistic choice (Proposition 2.2.2). To show that $[-]_{\equiv}$ is a PCA homomorphism, we argue the following:

$$\begin{aligned}
\left[\boxplus_{i\in I} p_i \cdot [e_i]_{\equiv_b}\right]_{\equiv} &= \left[\left[\bigoplus_{i\in I} p_i \cdot e_i\right]_{\equiv_b}\right]_{\equiv} && \text{(Lemma 2.4.11)}\\
&= \left[\bigoplus_{i\in I} p_i \cdot e_i\right] \\
&= \boxplus_{i\in I} p_i \cdot [e_i] && \text{(Def. of PCA structure on } \mathsf{PExp}/{\equiv})\\
&= \boxplus_{i\in I}^{n} p_i \, [[e_i]_{\equiv_b}]_{\equiv} && \square
\end{aligned}$$

2.4.4 Step 3: Coalgebra structure

Having established the necessary algebraic structure, we move on to showing how we can equip the quotient $\mathsf{PExp}/{\equiv}$ with a structure of coalgebra for the functor $\mathcal{G}\colon \mathsf{Set} \to \mathsf{Set}$. First, we focus on the $\mathcal{G}$-coalgebra structure $c\colon \mathsf{PExp}/{\equiv_b} \to \mathcal{G}\mathsf{PExp}/{\equiv_b}$ on the finer quotient $\mathsf{PExp}/{\equiv_b}$, defined as the following composition of maps:

$$\begin{tikzcd}
\mathsf{PExp}/{\equiv_b} \arrow[r, "{[\partial]_{\equiv_b}}"] \arrow[rrr, bend right, "c"'] & \mathcal{D}\mathcal{F}\mathsf{PExp}/{\equiv_b} \arrow[r, "\gamma_{\mathsf{PExp}/{\equiv_b}}"] & \mathcal{G}\mathcal{D}\mathsf{PExp}/{\equiv_b} \arrow[r, "\mathcal{G}\alpha_{\equiv_b}"] & \mathcal{G}\mathsf{PExp}/{\equiv_b}
\end{tikzcd}$$

Formally, we equip a quotient $\mathsf{PExp}/{\equiv_b}$ with $\mathcal{D}\mathcal{F}$-coalgebra structure, which exists due to soundness of $\equiv_b$ with respect to bisimilarity. Then, we transform it into a $\mathcal{G}\mathcal{D}$-coalgebra using the natural transformation $\gamma\colon \mathcal{D}\mathcal{F} \Rightarrow \mathcal{G}\mathcal{D}$. Determining this coalgebra directly would lead to changing the state space to $\mathcal{D}\mathsf{PExp}/{\equiv_b}$, which we would like to avoid. Instead, we flatten each reachable subdistribution using the algebra map $\alpha_{\equiv_b}\colon \mathcal{DF}\mathsf{PExp}/{\equiv_b} \to \mathsf{PExp}/{\equiv_b}$, thereby inducing a $\mathcal{G}$-coalgebra structure. This construction is closely related to the determinisation of $\big(\mathsf{PExp}/{\equiv_b}, \gamma_{\mathsf{PExp}/\equiv_b} \circ [\partial]_{\equiv_b}\big)$. In particular, we have the following result

Lemma 2.4.13. *The* PCA *structure map* $\alpha_{\equiv_b}\colon \mathcal{D}\mathsf{PExp}/{\equiv_b} \to \mathsf{PExp}/{\equiv_b}$ *is a* $\mathcal{G}$-*coalgebra homomorphism of the following type:*

$$\alpha_{\equiv_b}\colon \left(\mathcal{D}\mathsf{PExp}/{\equiv_b}, (\gamma_{\mathsf{PExp}/\equiv_b} \circ [\partial]_{\equiv_b})^\sharp\right) \to (\mathsf{PExp}/{\equiv_b}, c)$$

Proof. We show that the following diagram commutes:

$$\begin{tikzcd}
\mathcal{D}\mathsf{PExp}/{\equiv_b} \arrow[rr, "\alpha_{\equiv_b}"] \arrow[dr, "\mathcal{D}[\partial]_{\equiv_b}"] \arrow[ddd, "(\gamma_{\mathsf{PExp}/\equiv_b}\circ[\partial]_{\equiv_b})^\sharp"'] & & \mathsf{PExp}/{\equiv_b} \arrow[d, "{[\partial]_{\equiv_b}}"'] \\
& \mathcal{DDF}\mathsf{PExp}/{\equiv_b} \arrow[r, "\mu_{\mathcal{F}\mathsf{PExp}/\equiv_b}"] & \mathcal{DF}\mathsf{PExp}/{\equiv_b} \arrow[d, "\gamma_{\mathsf{PExp}/\equiv_0}"'] \\
& & \mathcal{GD}\mathsf{PExp}/{\equiv_b} \arrow[d, "\mathcal{G}\alpha_{\equiv_b}"'] \\
\mathcal{GD}\mathsf{PExp}/{\equiv_b} \arrow[rr, "\mathcal{G}\alpha_{\equiv_b}"] & & \mathcal{G}\mathsf{PExp}/{\equiv_b}
\end{tikzcd}$$

For the top right square, we have the following:

$$\begin{aligned}
[\partial]_{\equiv_b} \circ \alpha_{\equiv_b} &= [\partial]_{\equiv_b} \circ [\partial]_{\equiv_b}^{-1} \circ \mu_{\mathcal{F}\mathsf{PExp}/\equiv_b} \circ \mathcal{D}[\partial]_{\equiv_b} && (\text{Def. of } \alpha_{\equiv_b}) \\
&= \mu_{\mathcal{F}\mathsf{PExp}/\equiv_b} \circ \mathcal{D}[\partial]_{\equiv_b} && (\text{Corollary 2.4.9})
\end{aligned}$$

The commutativity of the bottom hexagon diagram follows directly from Proposition 2.3.4. ∎

We can utilise the aformentioned $\mathcal{G}$-coalgebra structure on $\mathsf{PExp}/{\equiv_b}$ to induce a corresponding $\mathcal{G}$-coalgebra structure on the coarser quotient $\mathsf{PExp}/{\equiv}$.

Lemma 2.4.14. *There exists a unique* $\mathcal{G}$-*coalgebra structure* $d\colon \mathsf{PExp}/{\equiv} \to$

$\mathcal{G}\mathsf{PExp}/{\equiv}$ *such that the following diagram commutes:*

$$\begin{tikzcd}
\mathsf{PExp} \arrow[r, "{[-]_{\equiv_b}}"] \arrow[d, "\partial"'] & \mathsf{PExp}/{\equiv_b} \arrow[r, "{[-]_{\equiv}}"'] \arrow[d, "{[\partial]_{\equiv_b}}"'] & \mathsf{PExp}/{\equiv} \arrow[dddd, dashed, "d"] \\
\mathcal{D}\mathcal{F}\mathsf{PExp} \arrow[r, "{\mathcal{D}\mathcal{F}[-]_{\equiv_b}}"'] & \mathcal{D}\mathcal{F}\mathsf{PExp}/{\equiv_b} \arrow[d, "{\gamma_{\mathsf{PExp}/\equiv_b}}"'] & \\
& \mathcal{G}\mathcal{D}\mathsf{PExp}/{\equiv_b} \arrow[d, "{G\alpha_{\equiv_b}}"'] & \\
& \mathcal{G}\mathsf{PExp}/{\equiv_b} \arrow[rr, "{\mathcal{G}[-]_{\equiv}}"'] & & \mathcal{G}\mathsf{PExp}/{\equiv}
\end{tikzcd}$$

The above lemma encapsulates the key step of the soundness proof. To establish the existence of $\mathcal{G}$-coalgebra structure on $\mathsf{PExp}/{\equiv}$, we will rely on diagonal fill-in property [Gum00, Lemma 3.17] to show the existence of the $\mathcal{G}$-coalgebra structure on $\mathsf{PExp}/{\equiv}$. Consequently, it suffices to demonstrate for all $e, f \in \mathsf{PExp}$ the following:

$$e \equiv f \implies \mathcal{G}[-]_{\equiv} \circ c(e) = \mathcal{G}[-]_{\equiv} \circ c(f) \tag{2.4}$$

We begin by demonstrating that the map appearing in the equation on the right-hand side of the implication above can be expressed explicitly.

Lemma 2.4.15. *For all $o \in [0,1]$ and indexed collections $\{p_i\}_{i\in I}$, $\{a_i\}_{i\in I}$, and $\{e_i\}_{i\in I}$, such that $p_i \in [0,1]$, $a_i \in A$ and $e_i \in \mathsf{PExp}$ for all $i \in I$ and $\sum_{i\in I} p_i \leq 1 - o$, we have that:*

$$(\mathcal{G}[-]_{\equiv} \circ c)\left(\left[o \cdot 1 \oplus \left(\bigoplus_{i\in I} p_i \cdot a_i \,; e_i\right)\right]_{\equiv_b}\right) = \left\langle o, \lambda a. \left[\bigoplus_{a_i = a} p_i \cdot e_i\right]\right\rangle$$

Proof. We first observe the following:

$$\begin{aligned}
c &= \mathcal{G}[-]_{\equiv} \circ \mathcal{G}\alpha_{\equiv_b} \circ \gamma_{\mathsf{PExp}/\equiv_b} \circ [\partial]_{\equiv_b} \circ [-]_{\equiv_b} && \text{(Def. of } c\text{)} \\
&= \mathcal{G}\alpha_{\equiv} \circ \mathcal{G}\mathcal{D}[-]_{\equiv} \circ \gamma_{\mathsf{PExp}/\equiv_b} \circ [\partial]_{\equiv_b} \circ [-]_{\equiv_b} && \text{(Lemma 2.4.12)} \\
&= \mathcal{G}\alpha_{\equiv} \circ \mathcal{G}\mathcal{D}[-]_{\equiv} \circ \gamma_{\mathsf{PExp}/\equiv_b} \circ \mathcal{D}\mathcal{F}[-]_{\equiv_b} \circ \partial \\
& && \text{(}[-]_{\equiv_b} \text{ is a } \mathcal{D}\mathcal{F}\text{-coalgebra homomorphism)}
\end{aligned}$$

$$
\begin{aligned}
&= \mathcal{G}\alpha_{\equiv} \circ \mathcal{G}\mathcal{D}[-]_{\equiv} \circ \mathcal{G}\mathcal{D}[-]_{\equiv_b} \circ \gamma_{\mathsf{PExp}} \circ \partial && \text{(Proposition 2.3.3)} \\
&= \mathcal{G}\alpha_{\equiv} \circ \mathcal{G}\mathcal{D}[-] \circ \gamma_{\mathsf{PExp}} \circ \partial && \text{(Definition of } [-]) \\
&= \mathcal{G}\alpha_{\equiv} \circ \gamma_{\mathsf{PExp}/\equiv} \circ \mathcal{D}\mathcal{F}[-] \circ \partial && \text{(Proposition 2.3.3)}
\end{aligned}
$$

Using the above reasoning, we can obtain the following:

$$
\begin{aligned}
&\mathcal{G}\alpha_{\equiv} \circ \gamma_{\mathsf{PExp}/\equiv} \circ \mathcal{D}\mathcal{F}[-] \circ \partial \left(o \cdot 1 \oplus \left(\bigoplus_{i \in I} p_i \cdot a_i \,;\, e_i \right) \right) \\
&\quad = \mathcal{G}\alpha_{\equiv} \circ \gamma_{\mathsf{PExp}/\equiv} \circ \mathcal{D}\mathcal{F}[-] \left(o\delta_{\checkmark} + \left(\sum_{i \in I} p_i \delta_{(a_i, e_i)} \right) \right) && \text{(Definition 2.3.6)} \\
&\quad = \mathcal{G}\alpha_{\equiv} \circ \gamma_{\mathsf{PExp}/\equiv} \left(o\delta_{\checkmark} + \left(\sum_{i \in I} p_i \delta_{(a_i, [e_i])} \right) \right) \\
&\quad = \mathcal{G}\alpha_{\equiv} \left\langle o, \lambda a. \sum_{a_i = a} p_i \delta_{[e_i]} \right\rangle && \text{(Def. of } \gamma) \\
&\quad = \left\langle o, \lambda a. \left[\bigoplus_{a_i = a} p_i \cdot e \right] \right\rangle && \text{(Theorem 2.2.6)}
\end{aligned}
$$

This establishes the desired result. □

Since the proof of Lemma 2.4.14 requires passing through the quotient $\mathsf{PExp}/{\equiv_b}$, which identifies the expressions modulo the axioms of the finer relation $\equiv_b$, remains only to verify the soundness of axioms (S0) and (D2) present in the finer relation $\equiv$.

Before proceeding, we first show that (S0) and (D2) can be reformulated in a simpler yet equally expressive form, which simplifies checking their soundness. Define the relation $\doteq \subseteq \mathsf{PExp} \times \mathsf{PExp}$ as follows: Let $\doteq \subseteq \mathsf{PExp} \times \mathsf{PExp}$ be a relation defined by the following

1. If $e \equiv_b f$, then $e \doteq f$

2. $a\,;(e \oplus_p f) \doteq a\,; e \oplus_p a\,; f$

3. $a\,;0 \doteq 0$

for all $e, f \in \mathsf{PExp}$, $p \in [0,1]$ and $a \in A$. It can be easily seen that both relations are equal.

Lemma 2.4.16. *For all $e, f \in \mathsf{PExp}$, $e \equiv f$ if and only if $e \doteq f$*

Proof. We split the proof into two cases.

$\boxed{e \doteq f \implies e \equiv f}$

This implication holds immediately. If $e \equiv_b f$, then $e \equiv f$. The remaining rules stating that $a\,;(e \oplus_p f) \doteq a\,;e \oplus_p a\,;f$ and $a\,;0 \doteq 0$ are special cases of (D2) and (S0) axioms of $\equiv$ specialised to single letters of the alphabet.

$\boxed{e \equiv f \implies e \doteq f}$

Axioms of $\equiv$ are either axioms of $\equiv_b$, which are already contained in $\doteq$ or are instances of (D2)/(S0) axioms. It suffices to show that latter two are derivable in $\doteq$. First, we show by induction that for all $e \in \mathsf{PExp}$, $e\,;0 \doteq 0$.

$\boxed{e = 0}$

$$\begin{aligned} e\,;0 &\doteq 0\,;0 && (e = 0)\\ &\doteq 0 && (0\mathsf{S}) \end{aligned}$$

$\boxed{e = 1}$

$$\begin{aligned} e\,;0 &\doteq 1\,;0 && (e = 1)\\ &\doteq 0 && (1\mathsf{S}) \end{aligned}$$

$\boxed{e = a}$

$$\begin{aligned} e\,;0 &\doteq a\,;0 && (e = a)\\ &\doteq 0 && (\text{Def. of } \doteq) \end{aligned}$$

$\boxed{e = f \oplus_p g}$

$$\begin{aligned} e\,;0 &\doteqdot (f \oplus_p g)\,;0 && (e = f \oplus_p g)\\ &\doteqdot f\,;0 \oplus_p g\,;0 && \text{(D1)}\\ &\doteqdot 0 \oplus_p 0 && \text{(Induction hypothesis)}\\ &\doteqdot 0 && \text{(C1)} \end{aligned}$$

$\boxed{e = f\,;g}$

$$\begin{aligned} e\,;0 &\doteqdot (f\,;g)\,;0 && (e = f\,;g)\\ &\doteqdot f\,;(g\,;0) && \text{(S)}\\ &\doteqdot f\,;0 && \text{(Induction hypothesis)}\\ &\doteqdot 0 && \text{(Induction hypothesis)} \end{aligned}$$

$\boxed{e = f^{[p]}}$

First, by Corollary 2.4.8 we know that $f^{[p]} \doteqdot g^{[r]}$, such that $E(g) = 0$.

$$\begin{aligned} 0 &\doteqdot 0 \oplus_r 0 && \text{(C1)}\\ &\doteqdot g\,;0 \oplus_r 0 && \text{(Induction hypothesis)} \end{aligned}$$

Since $E(g) = 0$, we can use unique fixpoint axiom and obtain:

$$\begin{aligned} 0 &\doteqdot g^{[r]}\,;0 && \text{(Unique)}\\ &\doteqdot f^{[p]}\,;0 && \text{(Corollary 2.4.8)} \end{aligned}$$

Secondly, we show by induction that for all $e, f, g \in \mathsf{PExp}$ and $p \in [0,1]$ we have that $e\,;(f \oplus_p g) \doteqdot e\,;f \oplus_p e\,;g$.

$\boxed{e = 0}$

$$\begin{aligned}
e\,;(f \oplus_p g) &\doteq 0\,;(f \oplus_p g) && (e = 0)\\
&\doteq 0 && (0\mathsf{S})\\
&\doteq 0 \oplus_p 0 && (\mathsf{C1})\\
&\doteq 0\,; f \oplus_p 0\,; g && (0\mathsf{S})
\end{aligned}$$

$\boxed{e = 1}$

$$\begin{aligned}
e\,;(f \oplus_p g) &\doteq 1\,;(f \oplus_p g) && (e = 1)\\
&\doteq f \oplus_p g && (1\mathsf{S})\\
&\doteq 1\,; f \oplus_p 1\,; g && (1\mathsf{S})
\end{aligned}$$

$\boxed{e = a}$

$$\begin{aligned}
e\,;(f \oplus_p g) &\doteq a\,;(f \oplus_p g) && (e = a)\\
&\doteq a\,; f \oplus_p a\,; g && (\text{Def. of } \doteq)
\end{aligned}$$

$\boxed{e = h \oplus_r i}$

$$\begin{aligned}
e\,;(f \oplus_p g) &\doteq (g \oplus_r h)\,;(f \oplus_p g) && (e = h \oplus_r i)\\
&\doteq h\,;(f \oplus_p g) \oplus_r i\,;(f \oplus_p g) && (\mathsf{D1})\\
&\doteq (h\,; f \oplus_p h\,; g) \oplus_r (i\,; f \oplus_p i\,; g) && (\text{Induction hypothesis})\\
&\doteq (h\,; f \oplus_r i\,; f) \oplus_p (h\,; g \oplus_r i\,; g) && (\text{Lemma 2.4.4})\\
&\doteq (h \oplus_r i)\,; f \oplus_p (h \oplus_r i)\,; g && (\mathsf{D1})
\end{aligned}$$

$\boxed{e = h\,;i}$

$$\begin{aligned}
e\,;(f \oplus_p g) &\doteqdot (h\,;i)\,;(f \oplus_p g) && (e = h\,;i)\\
&\doteqdot h\,;(i\,;(f \oplus_p g)) && \text{(S)}\\
&\doteqdot h\,;(i\,;f \oplus_p i\,;g) && \text{(Induction hypothesis)}\\
&\doteqdot (h\,;(i\,;f) \oplus_p h\,;(i\,;g)) && \text{(Induction hypothesis)}\\
&\doteqdot (h\,;i)\,;f \oplus_p (h\,;i)\,;g && \text{(S)}
\end{aligned}$$

$\boxed{e = h^{[r]}}$

First, by Corollary 2.4.8 we know that $h^{[r]} \doteqdot i^{[q]}$, such that $E(i) = 0$.

Next, we derive the following:

$$\begin{aligned}
i^{[q]}\,;f \oplus_p i^{[q]}\,;g &\doteqdot (i\,;i^{[q]} \oplus_q 1)\,;f \oplus_p (i\,;i^{[q]} \oplus_q 1)\,;g && \text{(Unroll)}\\
&\doteqdot (i\,;i^{[q]}\,;f \oplus_q 1\,;f) \oplus_p (i\,;i^{[q]}\,;g \oplus_q 1\,;g) && \text{(D1)}\\
&\doteqdot (i\,;i^{[q]}\,;f \oplus_q f) \oplus_p (i\,;i^{[q]}\,;g \oplus_q g) && \text{(0S)}\\
&\doteqdot (i\,;i^{[q]}\,;f \oplus_p i\,;i^{[q]}\,;g) \oplus_q (f \oplus_p g) && \text{(Lemma 2.4.4)}\\
&\doteqdot i\,;(i^{[q]}\,;f \oplus_p i^{[q]}\,;g) \oplus_q (f \oplus_p g) && \text{(Induction hypothesis)}
\end{aligned}$$

Since $E(i) = 0$, we can use (Unique) rule to derive

$$i^{[q]}\,;f \oplus_p i^{[q]}\,;g \doteqdot i^{[q]}\,;(f \oplus_p g)$$

Since $h^{[r]} \doteqdot i^{[q]}$, we have that

$$h^{[r]}\,;f \oplus_p h^{[r]}\,;g \doteqdot h^{[r]}\,;(f \oplus_p g)$$

This completes the proof. □

We now have all the ingredients to obtain the desired result.

Proof of Lemma 2.4.14. Assume $e \equiv f$. Because of Lemma 2.4.16, we have that $e \doteqdot f$. We will argue that $(\mathcal{G}[-]_{\equiv} \circ c)([e]_{\equiv_b}) = (\mathcal{G}[-]_{\equiv} \circ c)([f]_{\equiv_b})$. Because of the definition of $\doteq$ we have only three cases to consider.

$\boxed{e \equiv_b f}$

In this case, it follows immediately that $[e]_{\equiv_b} = [f]_{\equiv_\beta}$, which implies Equation (2.4).

$\boxed{b\,;(g \oplus_p h) \doteq b\,;g \oplus_p b\,;h}$

Applying $\mathcal{G}[-]_{\equiv} \circ c$ and using Lemma 2.4.15 to both sides immediately gives

$$(\mathcal{G}[-]_{\equiv} \circ c)[b\,;(g \oplus_p h)]_{\equiv} = \langle 0, s\rangle = (\mathcal{G}[-]_{\equiv} \circ c)[b\,;g \oplus_p b\,;h]_{\equiv}$$

where $s\colon A \to \mathsf{PExp}/{\equiv}$ is a function given by

$$s(a) = \begin{cases} [g \oplus_p h] & \text{if } a = b \\ [\mathsf{0}] & \text{otherwise} \end{cases}$$

$\boxed{b\,;\mathsf{0} \doteq \mathsf{0}}$

Once again, we apply Lemma 2.4.15 and obtain the following:

$$(\mathcal{G}[-]_{\equiv} \circ c)[b\,;\mathsf{0}]_{\equiv} = \langle 0, \lambda a.\,[\mathsf{0}]\rangle = (\mathcal{G}[-]_{\equiv} \circ c)[\mathsf{0}]_{\equiv}$$

This leads to the desired result.

□

As a direct corollary of Lemma 2.4.15 and the result established above, we obtain a concrete characterisation of the map d.

Corollary 2.4.17. *For all $[o \cdot \mathsf{1} \bigoplus_{i\in I} p_i \cdot a_i\,;e_i] \in \mathsf{PExp}/{\equiv}$, we have that*

$$d\left(\left[o \cdot \mathsf{1} \bigoplus_{i \in I} p_i \cdot a_i\,;e_i\right]\right) = \left\langle o, \lambda a.\left[\bigoplus_{a_i = a} p_i \cdot e_i\right]\right\rangle$$

2.4.5 Step 4: Soundness result

Through a simple finality argument, we can show that the unique $\mathcal{G}$-coalgebra homomorphism from the determinisation of the Antimirov transition system, can be viewed as the following composition of homomorphisms, which we have obtained in the earlier steps.

Lemma 2.4.18. *The following diagram commutes:*

$$\begin{tikzcd} \mathcal{D}\mathsf{PExp} \arrow[r, "\mathcal{D}[-]_{\equiv_b}"] \arrow[rrrr, bend right, "\mathsf{beh}_{(\gamma_{\mathsf{PExp}} \circ \partial)^\sharp}"'] & \mathcal{D}\mathsf{PExp}/{\equiv_b} \arrow[r, "\alpha_{\equiv_b}"] & \mathsf{PExp}/{\equiv_b} \arrow[r, "{[-]_{\equiv}}"] & \mathsf{PExp}/{\equiv} \arrow[r, dashed, "\mathsf{beh}_d"] & {[0,1]^{A^*}} \end{tikzcd}$$

Proof. By combining Lemma 2.4.2, Lemma 2.4.13, and Lemma 2.4.14, we conclude that $\mathcal{D}[-]_{\equiv_b} \circ \alpha_{\equiv_b} \circ [-]_{\equiv_b}$ is a $\mathcal{G}$-coalgebra homomorphism from $\left(\mathcal{D}\mathsf{PExp}, (\gamma_{\mathsf{PExp}} \circ \partial)^\sharp\right)$ to $(\mathsf{PExp}/{\equiv}, d)$. Composing with a final map beh_d into it, we obtain a $\mathcal{G}$-coalgebra homomorphism rom $\left(\mathcal{D}\mathsf{PExp}, (\gamma_{\mathsf{PExp}} \circ \partial)^\sharp\right)$ into the final coalgebra $\left([0,1]^{A^*}, t\right)$, which, by finality must be equal to $\mathsf{beh}_{(\gamma_{\mathsf{PExp}} \circ \partial)^\sharp}$. $\square$

Since the language-assigning map relies on the homomorphism described above, we can show the following:

Lemma 2.4.19. *The function* $[\![-]\!] : \mathsf{PExp} \to [0,1]^{A^*}$ *assigning each expression to its semantics satisfies:* $[\![-]\!] = \mathsf{beh}_d \circ [-]$.

Proof.

$$\begin{aligned}
[\![-]\!] &= \dagger(\gamma_{\mathsf{PExp}} \circ \partial)^\sharp \circ \eta_{\mathsf{PExp}} && \text{(Def. of } [\![-]\!]\text{)} \\
&= \mathsf{beh}_d \circ [-]_{\equiv} \circ \alpha_{\equiv_b} \circ \mathcal{D}[-]_{\equiv_b} \circ \eta_{\mathsf{PExp}} && \text{(Lemma 2.4.18)} \\
&= \mathsf{beh}_d \circ [-]_{\equiv} \circ \alpha_{\equiv_b} \circ \eta_{\mathsf{PExp}/\equiv_b} \circ [-]_{\equiv_b} && (\eta \text{ is natural}) \\
&= \mathsf{beh}_d \circ [-]_{\equiv} \circ [-]_{\equiv_b} && (\alpha_{\equiv} \text{ is an Eilenberg-Moore algebra}) \\
&= \mathsf{beh}_d \circ [-]
\end{aligned}$$

The last equality in the derivation above follows from the definition of $[-]$. $\square$

We can now immediately conclude that provably equivalent expressions are mapped to the same probabilistic languages, thus establishing soundness.

Theorem 2.4.20 (Soundness)**.** *For all* $e, f \in \mathsf{PExp}$*, if* $e \equiv f$ *then* $[\![e]\!] = [\![f]\!]$.

2.5 Completeness

The completeness proof will follow a pattern of earlier work of Jacobs [Jac06], Silva [Sil10] and Milius [Mil10] and show that the coalgebra structure on the $\mathsf{PExp}/{\equiv}$ is isomorphic to the subcoalgebra of an appropriate final coalgebra, ie. the unique final coalgebra homomorphism from $\mathsf{PExp}/{\equiv}$ is injective. The intuition comes from the coalgebraic modelling of deterministic automata studied in the work of Jacobs [Jac06]. In such a case, the final coalgebra is simply the automaton structure on the set of all formal languages, while the final homomorphism is given by the map taking a state of the automaton to a language it denotes. Restricting the attention to finite-state automata only yields *regular languages*. The set of regular languages can be equipped with an automaton structure, in a way that inclusion map into the final automaton on the set of all formal languages becomes a homomorphism. In such a case, Kozen's completeness proof of Kleene Algebra [Koz94] can be seen as showing isomorphism of the automaton of regular languages and the automaton structure on the regular expressions modulo KA axioms.

Unfortunately, we cannot immediately rely on the identical pattern. Our semantics relies on determinising GPTS, but unfortunately, determinising a finite-state GPTS can yield $\mathcal{G}$-coalgebras with infinite carriers. For example, determinising a single-state GPTS would yield a $\mathcal{G}$-coalgebra over the set of subdistributions over a singleton set, which is infinite. Luckily, all $\mathcal{G}$-coalgebras we work with have additional algebraic structure. This algebraic structure will allow us to rely on the generalisations of the concept of finiteness beyond the category of sets, offered by the theory of locally finite presentable categories [AR94]. Being equipped with those abstract lenses, one can immediately see that the earlier mentioned infinite set of all subdistributions over a singleton set is also a free PCA generated by a single element and thus finitely presentable [SW15].

In particular, we will work with a *rational fixpoint*; a generalisation of the idea of subcoalgebra of regular languages to coalgebras for finitary functors $\mathcal{B} : \mathcal{A} \to \mathcal{A}$ over locally finitely presentable categories. The rational fixpoint provides a semantic domain for the behaviour of coalgebras whose carriers are finitely presentable in the same way as regular languages provide a semantic domain for all finite-state deterministic automata. The completeness proof will essentially rely on establishing that the coalgebra structure on $\mathsf{PExp}/{\equiv}$ satisfies the universal property of the rational fixpoint.

We now move on to showing completeness through the steps described in Section 2.3.3.

2.5.1 Step 1: Algebra structure

Throughout the soundness proof, we have shown that the semantics of any expression $e \in \mathsf{PExp}$ can also be seen as the language of the state corresponding to the equivalence class $[e]$ in the deterministic transition system ($\mathcal{G}$-coalgebra) defined on the set $\mathsf{PExp}/{\equiv}$. The completeness proof will rely on establishing that this coalgebra possesses a universal property of rational fixpoint, that will imply completeness. A first step in arguing so is observing that the coalgebra $(\mathsf{PExp}/{\equiv}, d)$ interacts well with an algebra structure on $\mathsf{PExp}/{\equiv}$ defined in Lemma 2.4.12. Thanks to the fact that we can lift $\mathcal{G} \colon \mathsf{Set} \to \mathsf{Set}$ to $\overline{\mathcal{G}} \colon \mathsf{PCA} \to \mathsf{PCA}$, the set $\mathcal{G}\mathsf{PExp}/{\equiv}$ also carries the algebra structure. In such a setting, the transition function $d \colon \mathsf{PExp}/{\equiv} \to \mathcal{G}\mathsf{PExp}/{\equiv}$ becomes an algebra homomorphism.

Lemma 2.5.1. *$d \colon \mathsf{PExp}/{\equiv} \to \mathcal{G}\mathsf{PExp}/{\equiv}$ is a PCA homomorphism*

$$d \colon (\mathsf{PExp}/{\equiv}, \alpha_{\equiv}) \to \overline{\mathcal{G}}(\mathsf{PExp}/{\equiv}, \alpha_{\equiv})$$

Proof. We need to show that $d\left(\boxplus_{i\in I} p_i \cdot [e_i]\right) = \boxplus_{i\in I} p_i \cdot d([e_i])$. As a consequence of Theorem 2.4.3, we can safely assume that $e_i \equiv q^i \cdot 1 \oplus \bigoplus_{j\in J} r^i_j \cdot a^i_j \,; e^i_j$ for all $i \in I$

and hence $d([e_i]) = \left\langle q^i, \lambda a. \left[\bigoplus_{a=a^i_j} r^i_j \cdot e^i_j\right] \right\rangle$. We show the following:

$$
\begin{aligned}
d\left(\boxplus_{i\in I} p_i \cdot [e_i]\right) &= d\left(\left[\bigoplus_{i\in I} p_i \cdot e_i\right]\right) && \text{(Lemma 2.4.12)}\\
&= d\left(\left[\bigoplus_{i\in I} p_i \cdot \left(q^i \cdot 1 \oplus \bigoplus_{j\in J} r^i_j \cdot a^i_j ; e^i_j\right)\right]\right) && \text{(Def. of } e_i\text{)}\\
&= d\left(\bigoplus_{i\in I} p_i q^i \cdot 1 \oplus \bigoplus_{(i,j)\in I\times J} p_i r^i_j \cdot a^i_j ; e^i_j\right) && \text{(Lemma 2.2.4)}\\
&= \left\langle \sum_{i\in I} p_i q^i, \lambda a. \left[\bigoplus_{a=a^i_j} p_i r^i_j \cdot e^i_j\right] \right\rangle && \text{(Corollary 2.4.17)}\\
&= \left\langle \sum_{i\in I} p_i q^i, \lambda a. \left[\bigoplus_{i\in I} p_i \cdot \left(\bigoplus_{a=a^i_j} r^i_j \cdot e^i_j\right)\right] \right\rangle && \text{(Lemma 2.2.4)}\\
&= \left\langle \sum_{i\in I} p_i q^i, \lambda a. \boxplus_{i\in I} p_i \cdot \left[\bigoplus_{a=a^i_j} r^i_j \cdot e^i_j\right] \right\rangle && \text{(Lemma 2.4.12)}\\
&= \boxplus_{i\in I} p_i \left\langle q^i, \lambda a. \left[\bigoplus_{a=a^i_j} r^i_j \cdot e^i_j\right] \right\rangle && \text{(Def. of } \overline{\mathcal{G}}\text{)}\\
&= \boxplus_{i\in I} p_i \cdot d(e_i) && \text{(Corollary 2.4.17)}
\end{aligned}
$$

This yields the desired result. □

As a consequence of the result showed above, we have that

$$\left(\mathsf{PExp}/\equiv, d\colon (\mathsf{PExp}/\equiv, \alpha_{\equiv}) \to \overline{\mathcal{G}}\,(\mathsf{PExp}/\equiv, \alpha_{\equiv})\right)$$

is a $\overline{\mathcal{G}}$-coalgebra.

2.5.2 Step 2: Proper functors

It can be easily noticed that the generalised determinisation of coalgebras with finite carriers corresponds to algebraically structured coalgebras of a particular, well-behaved kind. Namely, their carriers are algebras which are free finitely generated. We write $\mathsf{Coalg}_{\mathsf{free}}\,\mathcal{B}$ for the subcategory of $\mathsf{Coalg}\,B$ consisting only of $\mathcal{B}$-coalgebras with free finitely generated carriers. The recent work of Milius [Mil18] characterised *proper* functors, for which in order to establish that some $\mathcal{B}$-coalgebra is isomorphic to the rational fixpoint it will suffice to look at coalgebras with free finitely generated carriers.

We start by recalling the following definition:

Definition 2.5.2 ([Mil18, Remark 5.1])**.** A zig-zag in a category $\mathcal{C}$ is a diagram of the form

$$\begin{tikzcd}
Z_0 \arrow[dr, "f_0"] & & Z_2 \arrow[dl, "f_1"'] \arrow[dr, "f_2"] & & \cdots \arrow[dl, "f_3"'] \arrow[dr, "f_{n-2}"] & & Z_n \arrow[dl, "f_{n-1}"'] \\
& Z_1 & & Z_3 & & Z_{n-1} &
\end{tikzcd}$$

For a concrete category $\mathcal{C}$ (equipped with a faithful forgetful functor $\mathcal{U}\colon \mathcal{C} \to \mathsf{Set}$), we say that a zig-zag relates $z_0 \in \mathcal{U}Z_0$ and $z_n \in \mathcal{U}Z_n$ if there exist $z_i \in \mathcal{U}Z_i$, $i = 1, \ldots, n-1$ such that $\mathcal{U}f_i(z_i) = z_{i+1}$ for i even and $\mathcal{U}f_i(z_{i+1}) = z_i$ for i odd.

With the above definition in hand, we can recall the definition of the proper functor:

Definition 2.5.3 ([Mil18, Definition 5.2])**.** Let $\mathbf{T} = (T, \mu, \eta)$ be a finitary monad over Set and let $\mathcal{B}\colon \mathsf{Set}^{\mathbf{T}} \to \mathsf{Set}^{\mathbf{T}}$ be a functor that preserves surjective T-algebra homomorphisms. We call $\mathcal{B}$ proper if for every pair (TB_1, c_1) and (TB_2, c_2) of $\mathcal{B}$-coalgebras with B_1 and B_2 finite sets and each two elements $b_1 \in B_1$ and $b_2 \in B_2$ with $\eta_{B_1}(b_1) \sim_{\mathsf{b}} \eta_{B_2}(b_2)$, there exists a zig-zag in $\mathsf{Coalg}\,\mathcal{B}$, which relates $\eta_{B_1}(b_1)$ and $\eta_{B_2}(b_2)$, and whose nodes Z_i are $\mathcal{B}$ are coalgebras with free and finitely generated carrier.

Proper functors satisfy the following property, which is crucial for the completeness argument presented in this chapter.

Theorem 2.5.4 ([Mil18, Corollary 5.9])**.** *Let $\mathcal{B}\colon \mathsf{Set}^{\mathbf{T}} \to \mathsf{Set}^{\mathbf{T}}$ be a proper functor. Then a $\mathcal{B}$-coalgebra (R, r) is isomorphic to the rational fixpoint if (R, r) is locally finitely presentable and for every $\mathcal{B}$-coalgebra (TX, c) in $\mathsf{Coalg}_{\mathsf{free}}\,\mathcal{B}$ there exists a unique homomorphism from TX to R.*

As much as $\overline{\mathcal{G}}\colon \mathsf{PCA} \to \mathsf{PCA}$ is known to be proper [SW18], not every $\overline{\mathcal{G}}$-coalgebra with a free finitely generated carrier corresponds to a determinisation of some (converted) GPTS. This is simply too general, as some $\overline{\mathcal{G}}$-coalgebras with free finitely generated carriers might be determinisations of RPTS not corresponding to any GPTS. To circumvent that, instead of looking at all coalgebras for the functor $\overline{\mathcal{G}}\colon \mathsf{PCA} \to \mathsf{PCA}$, we can restrict our attention in a way that will exclude determinisations of RPTS not corresponding to any GPTS. To do so, define a functor $\hat{\mathcal{G}}\colon \mathsf{PCA} \to \mathsf{PCA}$. Given a positive convex algebra $\mathbb{X}$ defined on a set X, we define:

$$\begin{aligned}\hat{\mathcal{G}}\mathbb{X} =& \{(o,f) \in [0,1] \times X^A \mid \forall a \in A.\, \exists p_a^i \in [0,1], x_a^i \in X.\\ & f(a) = \boxplus_{i\in I} p_a^i x_a^i \text{ and } \sum_{a\in A}\sum_{i\in I} p_a^i \leq 1-o\}\end{aligned}$$

The PCA structure on $\hat{\mathcal{G}}\mathbb{X}$, as well as the action of $\hat{\mathcal{G}}$ on arrows is defined to be the same as in the case of $\mathcal{G}$. It can be immediately observed that $\hat{\mathcal{G}}$ is a subfunctor of $\overline{\mathcal{G}}$.

Remark 2.5.5. Given that $\overline{\mathcal{G}}$ preserves non-empty monomorphisms (Lemma 2.2.9) and $\hat{\mathcal{G}}$ coincides with $\overline{\mathcal{G}}$ on arrows, it follows that $\hat{\mathcal{G}}$ also preserves non-empty monomorphisms.

Whenever the algebra structure is clear from the context, we write $\hat{\mathcal{G}}X$ for $\hat{\mathcal{G}}\mathbb{X}$. Most importantly for us, thanks to the result of Sokolova and Woracek, we know that $\hat{\mathcal{G}}$ is also proper [SW18]. We can now see the following correspondence:

Lemma 2.5.6. *$\mathcal{DF}$-coalgebras with finite carriers are in one-to-one correspondence with $\hat{\mathcal{G}}$-coalgebras with free finitely generated carriers.*

$$\frac{\beta\colon X \to \mathcal{DF}X \text{ in } \mathsf{Set}}{\xi : (\mathcal{D}X, \mu_X) \to \hat{\mathcal{G}}(\mathcal{D}X, \mu_X) \text{ in } \mathsf{PCA}}$$

In other words, every coalgebra structure map $\xi\colon (\mathcal{D}X, \mu_X) \to \hat{\mathcal{G}}(\mathcal{D}X, \mu_X)$ is given by $\xi = (\gamma_X \circ \beta)^\sharp$ for some unique $\beta\colon X \to \mathcal{DF}X$.

Proof. Since X is finite, we can assume that $X = \{s_i\}_{i\in I}$ for some finite set I. Because of the free-forgetful adjunction between PCA and Set, we have the following correspondence of maps:

$$\frac{\zeta : X \to \mathcal{G}\mathcal{D}X \text{ on Set}}{\xi : (\mathcal{D}X, \mu_X) \to \mathcal{G}(\mathcal{D}X, \mu_X) \text{ on PCA}}$$

First, we show that for all $x \in X$, we have that $\gamma_X \circ \beta(x) \in \hat{\mathcal{G}}\mathcal{D}X$. Pick an arbitrary $x \in X$. For every $a \in A$ define $p_i^a = \beta(x)(a, s_i)$. This implies that $(\pi_2 \circ \gamma_X \circ \beta)(x)(a)(x_i) = p_i^a$ if $x_i \in S$. Therefore, we have

$$(\gamma_X \circ \beta)(x) = \left\langle \beta(x)(\checkmark), \lambda a. \sum_{i \in I} p_i^a \delta_{x_i^a} \right\rangle$$

Using the isomorphism between PCA and $\mathsf{Set}^{\mathcal{D}}$, we can reformulate the above as

$$(\gamma_X \circ \beta)(x) = \left\langle \beta(x)(\checkmark), \lambda a. \bigboxplus_{i \in I} p_i^a \cdot x_i^a \right\rangle$$

where $\boxplus$ denotes the structure map of PCA isomorphic to $(\mathcal{D}X, \mu_X)$, the free Eilenberg-Moore algebra generated by X. From the well-definedness of $\beta(x)$ it follows that:

$$\sum_{a \in A} \sum_{i \in I} p_i^a \leq 1 - \beta(x)(\checkmark)$$

which establishes that $\gamma_X \circ \beta(x) \in \hat{\mathcal{G}}\mathcal{D}X$.

For the converse, observe that every arrow $\xi : (\mathcal{D}X, \mu_X) \to \hat{\mathcal{G}}\mathcal{D}(\mathcal{D}X, \mu_X)$ in PCA arises as an extension of $\zeta : X \to \mathcal{U}\hat{\mathcal{G}}(\mathcal{D}X, \mu_X)$. Furthermore, any such ζ can be expressed as composition $\gamma_X \circ \beta$, where $\beta(x)(\checkmark) = \pi_1 \circ \zeta(x)$ and $\beta(x)(a, x') = \pi_2 \circ \zeta(x)(a)(x')$. The fact that $\zeta(x) \in \hat{\mathcal{G}}\mathcal{D}X$ ensures that $\beta(x)$ is a well-defined subdistribution. Finally, since γ_X is injective (Proposition 2.3.3), β is unique. $\square$

Next, we argue that the coalgebra structure on $\mathsf{PExp}/\equiv$, which is at the centre of attention of the completeness proof, happens also to be a $\hat{\mathcal{G}}$-coalgebra.

Lemma 2.5.7. *$((\mathsf{PExp}/\equiv, \alpha_\equiv), d)$ is a $\hat{\mathcal{G}}$-coalgebra.*

Before we prove the lemma above, we establish two intermediate results. First, we show the lemma allowing to establish that the coalgebra structure on $\mathsf{PExp}/\equiv$ defined in Section 2.4.4 is also a $\hat{\mathcal{G}}$-coalgebra.

Lemma 2.5.8. *Let (X,α) be a* PCA. *Then for every $\zeta \in \mathcal{DF}X$ we have that $\mathcal{G}\alpha \circ \gamma_X(\zeta) \in \hat{\mathcal{G}}(X,\alpha)$.*

Proof. Let $\zeta \in \mathcal{DF}X$. Recall that $\gamma_X(\zeta) = \langle \zeta(\checkmark), \lambda a.\, \lambda x.\, \zeta(a,x) \rangle$. Let $S = \{x \in X \mid \exists a \in A.\, (a,x) \in \mathrm{supp}(\zeta)\}$. Without the loss of generality, we can assume that $S = \{s_i\}_{i\in I}$ for some finite set I. For every $a \in A$ define $p_i^a = \zeta(a,s_i)$. This implies that $(\pi_2 \circ \gamma_X)(\zeta)(a)(x_i) = p_i^a$ if $x_i \in S$ or $(\pi_2 \circ \gamma_X)(\zeta)(a)(x_i) = 0$ otherwise. Therefore, we have the following:

$$(\mathcal{G}\alpha \circ \gamma_X)(\zeta) = \left\langle \zeta(\checkmark), \lambda a. \boxplus_{i\in I} p_i^a \cdot s_i \right\rangle$$

Finally, since $\zeta \in \mathcal{DF}X$ we have that $\sum_{a\in A}\sum_{i\in I} p_i^a \leq 1 - \zeta(\checkmark)$ which proves that indeed the image of $G\alpha \circ \gamma_X$ belongs to $\hat{\mathcal{G}}(X,\alpha)$. $\square$

Next, we show the following preservation result.

Lemma 2.5.9. *$\hat{\mathcal{G}}$-coalgebras are closed under surjective $\overline{\mathcal{G}}$-coalgebra homomorphisms.*

Proof. Let $((X,\alpha_X),\beta_X)$ be a $\hat{\mathcal{G}}$-coalgebra, $((Y,\alpha_Y),\beta_Y)$ be a $\overline{\mathcal{G}}$-coalgebra and let $e\colon X \to Y$ be a surjective $\overline{\mathcal{G}}$-homomorphism $e\colon ((X,\alpha_X),\beta_X) \to ((Y,\alpha_Y),\beta_Y)$. We need to show that for all $y \in Y$, $\beta_Y(y) \in \mathcal{U}\hat{\mathcal{G}}(Y,\alpha_Y)$. Pick an arbitrary $y \in Y$. Since $e\colon X \to Y$ is surjective, we know that $y = e(x)$. Let $\beta_X(x) = \langle o, f\rangle$. We have that:

$$\beta_Y(y) = (\beta_Y \circ e)(x) = (\mathcal{G}e \circ \beta_X)(x) = \langle o, e \circ f\rangle$$

Since $\beta_X(x) \in \hat{\mathcal{G}}X$, we have that for all $a \in A$ there exist $p_a^i \in [0,1]$ and $x_a^i \in X$ such that $f(a) = \boxplus_{i\in I} p_a^i x_a^i$ and $\sum_a \sum_{i\in I} p_a^i \leq 1 - o$. Let $e(x_a^i) = y_a^i$. For all $a \in A$, we have that:

$$(e \circ f)(a) = e\left(\boxplus_{i\in I} p_a^i \cdot x_a^i\right) = \boxplus_{i\in I} p_a^i \cdot e(x_a^i) = \boxplus_{i\in I} p_a^i \cdot y_a^i$$

Therefore, for all $a \in A$ there exist $p_a^i \in [0,1]$ and $y_a^i \in X$ such that $f(a) = \boxplus_{i \in I} p_a^i y_a^i$ and $\sum_a \sum_{i \in I} p_a^i \leq 1 - o$, which completes the proof. □

We are ready to prove the desired result.

Proof of Lemma 2.5.7. Recall that $((\mathsf{PExp}/{\equiv}, \alpha_{\equiv}), d)$ is a quotient coalgebra of

$$(\mathsf{PExp}/{\equiv_b}, \mathcal{G}\alpha_{\equiv_b} \circ \gamma_{\mathsf{PExp}/\equiv_b} \circ [\partial]_{\equiv_b})$$

which by Lemma 2.5.8 is a $\hat{\mathcal{G}}$-coalgebra. Because of Lemma 2.5.9 so is $(\mathsf{PExp}/{\equiv}, d)$. This completes the proof. □

Moreover, when viewed as a $\hat{\mathcal{G}}$-coalgebra, we can show that $((\mathsf{PExp}/{\equiv}, \alpha_{\equiv}), d)$ forms a fixpoint of the functor, i.e. d is an isomorphism. This technical result will play a crucial role in the completeness proof. Specifically, it enables us to establish a correspondence between:

- $\hat{\mathcal{G}}$-coalgebra homomorphisms from $\hat{\mathcal{G}}$-coalgebras with free finitely generated carriers into $((\mathsf{PExp}/{\equiv}, \alpha_{\equiv}), d)$, and
- syntactic solutions to fixpoint systems of equations (that will be formally introduced in Section 2.5.3) describing finite-state GPTS.

Lemma 2.5.10. $d \colon (\mathsf{PExp}, \alpha_{\equiv}) \to \hat{\mathcal{G}}(\mathsf{PExp}, \alpha_{\equiv})$ *is an isomorphism*

Proof. We construct a map $d^{-1} \colon \hat{\mathcal{G}}(\mathsf{PExp}/{\equiv}, \alpha_{\equiv}) \to (\mathsf{PExp}/{\equiv}, \alpha_{\equiv})$ and show that $d \circ d^{-1} = \mathsf{id} = d^{-1} \circ d$. Given that the forgetful functor $\mathcal{U} \colon \mathsf{PCA} \to \mathsf{Set}$ is conservative (reflects isomorphisms), this will immediately imply that d^{-1} is a PCA homomorphism. Given $\langle o, f \rangle \in \hat{\mathcal{G}}X$, such that for any $a \in A$, and $f(a) = \boxplus_{i \in I} p_a^i [e_a^i]$ for some $p_a^i \in [0,1]$, and $[e_a^i] \in \mathsf{PExp}/{\equiv}$, we define the following:

$$d^{-1}\langle o, f \rangle = \left[o \cdot 1 \oplus \left(\bigoplus_{(a,i) \in A \times I} p_a^i \cdot a \,; e_a^i \right) \right]$$

The expression inside the brackets is well defined as $\sum_a \sum_{i \in I} p_a^i \leq 1 - o$. To show that d^{-1} is well-defined, assume that we have $g \colon A \to \mathsf{PExp}/{\equiv}$, such that $f(a) =$

$g(a) = \boxplus_{i\in I} q^i_a[h^i_a]$ for all $a \in A$ and some $q^i_a \in [0,1]$ and $[h^i_a] \in \mathsf{PExp}/\equiv$. To begin, due to the definition of the PCA structure on $\mathsf{PExp}/\equiv$ (Lemma 2.4.12), we have that:

$$\left[\bigoplus_{i\in I} p^i_a \cdot e^i_a\right] = \boxplus_{i\in I} p^i_a[e^i_a] = \boxplus_{i\in I} q^i_a[h^i_a] = \left[\bigoplus_{i\in I} q^i_a \cdot h^i_a\right]$$

Using the above, we can show:

$$\begin{aligned}
d^{-1}\langle o, f\rangle &= \left[o \cdot 1 \oplus \left(\bigoplus_{(a,i)\in A\times I} p^i_a \cdot a\,; e^i_a\right)\right] \\
&= \left[o \cdot 1 \oplus \left(\bigoplus_{(a,i)\in A\times I} q^i_a \cdot a\,; h^i_a\right)\right] && (\equiv \text{ is a congruence}) \\
&= d^{-1}\langle o, g\rangle
\end{aligned}$$

This establishes the well-definedness of d^{-1}. To verify one side of the isomorphism, consider the following:

$$\begin{aligned}
(d \circ d^{-1})(\langle o, f\rangle) &= d\left(\left[o \cdot 1 \oplus \left(\bigoplus_{(a,i)\in A\times I} p^i_a \cdot a\,; e^i_a\right)\right]\right) && (\text{Def. of } d^{-1}) \\
&= \left\langle o, \lambda a.\left[\bigoplus_{i\in I} p^i_a \cdot e^i_a\right]\right\rangle && (\text{Corollary 2.4.17}) \\
&= \left\langle o, \boxplus_{i\in I} p^i_a \cdot [e^i_a]\right\rangle && (\text{Lemma 2.4.12}) \\
&= \langle o, f\rangle
\end{aligned}$$

For the converse direction, let$[e] \in \mathsf{PExp}/\equiv$. By Theorem 2.4.3, we have that:

$$e \equiv o \cdot 1 \oplus \left(\bigoplus_{(a,i)\in A\times I} p^i_a \cdot a\,; e^i_a\right)$$

Next, note that:

$$(d^{-1} \circ d)([e]) = d^{-1}\left\langle o, \lambda a.\left[\bigoplus_{i\in I} p^i_a \cdot e^i_a\right]\right\rangle \qquad (\text{Corollary 2.4.17})$$

$$= d^{-1}\left\langle o, \lambda a. \boxplus_{i\in I} p_a^i \cdot [e_a^i] \right\rangle \qquad \text{(Lemma 2.4.12)}$$

$$= \left[o \cdot 1 \oplus \left(\bigoplus_{(a,i)\in A\times I} p_a^i \cdot a ; e_a^i \right) \right] \qquad \text{(Def. of } d^{-1}\text{)}$$

$$= d([e]) \qquad \square$$

2.5.3 Step 3: Systems of equations

In order to establish that $\mathsf{PExp}/{\equiv}$ is isomorphic to the rational fixpoint (which is the property that will eventually imply completeness), we will show the satisfaction of conditions of Theorem 2.5.4. One of the required things we need to show is that the determinisation of an arbitrary finite-state GPTS admits a unique homomorphism to the coalgebra carried by $\mathsf{PExp}/{\equiv}$. In other words, we need to convert states of an arbitrary finite-state GPTS to language equivalent expressions in a way which is unique up to the axioms of $\equiv$. This can be thought of as an abstract reformulation of one direction of the Kleene theorem to the case of PRE. To make that possible, we give a construction inspired by Brzozowski's equation solving method [Brz64] of converting a DFA to the corresponding regular expression. We start by stating the necessary definitions.

Definition 2.5.11. A left-affine system on a finite non-empty set Q of unknowns is a quadruple

$$\mathcal{S} = (M\colon Q\times Q \to \mathsf{PExp}, p\colon Q\times Q \to [0,1], b\colon Q \to \mathsf{PExp}, r\colon Q \to [0,1])$$

such that for all $q, q' \in Q$, $\sum_{q'\in Q} p_{q,q'} + r_q = 1$ and $E(M_{q,q'}) = 0$.

Example 2.5.12. Consider the set of unknowns $Q = \{q_0, q_1\}$. We define a left-affine system $\mathcal{S} = (M, p, b, r)$ on Q with the following components:

- The function $M\colon Q \times Q \to \mathsf{PExp}$ is defined by:

M	q_0	q_1
q_0	0	a
q_1	0	a

- The function $p\colon Q \times Q \to [0,1]$ is given by:

p	q_0	q_1
q_0	0	1
q_1	0	$\frac{1}{4}$

- Functions $b\colon Q \to \mathsf{PExp}$ and $r\colon Q \to [0,1]$ are defined as:

$$b_x = \begin{cases} 0 & \text{if } x = q_0 \\ 1 & \text{if } x = q_1 \end{cases} \qquad \text{and} \qquad r = \begin{cases} 0 & \text{if } x = q_0 \\ \frac{3}{4} & \text{if } x = q_1 \end{cases}$$

Definition 2.5.13. Let $\equiv_c \subseteq \mathsf{PExp} \times \mathsf{PExp}$ be a congruence relation. A map $h\colon Q \to \mathsf{PExp}$ is $\equiv_c$-solution if for all $q \in Q$ we have that:

$$h(q) \equiv_c \left(\bigoplus_{q' \in Q} p_{q,q'} \cdot M_{q,q'} \,; h(q') \right) \oplus r_q \cdot b_q$$

Example 2.5.14. A $\equiv_c$-solution $h\colon \{q_0, q_1\} \to \mathsf{PExp}$ to the system described in Example 2.5.12, would need to satisfy the following:

$$h(q_0) \equiv_c a\,; h(q_1) \qquad h(q_1) \equiv_c a\,; h(q_1) \oplus_{\frac{1}{4}} 1$$

Definition 2.5.15. A system representing the finite-state $\mathcal{DF}$-coalgebra (X,β) is given by $\mathcal{S}(\beta) = \langle M^\beta, p^\beta, b^\beta, r^\beta \rangle$ where for all $x, x' \in X$ we have:

$$p^\beta_{x,x'} = \sum_{a' \in A} \beta(x)(a', x') \qquad r^\beta_x = 1 - \sum_{(a',x') \in \mathrm{supp}(\beta(x))} \beta(x)(a', x')$$

$$M^{\beta}_{x,x'} = \begin{cases} \bigoplus_{a\in A} \frac{\beta(x)(a,x')}{p^{\beta}_{x,x'}} \cdot a & \text{if } p^{\beta}_{x,x'} \neq 0 \\ 0 & \text{otherwise} \end{cases} \qquad b^{\beta}_{x} = \begin{cases} \frac{\beta(x)(\checkmark)}{r^{\beta}_{x}} \cdot 1 & \text{if } r^{\beta}_{x} \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

Example 2.5.16. The left-affine system described in Example 2.5.12 corresponds to the $\mathcal{DF}$-coalgebra shown on the left-hand side of Example 2.1.2.

The original Brzozowski's equation-solving method is purely semantic, as it crucially relies on Arden's rule [Ard61] by providing solutions up to the language equivalence to the systems of equations. As much as it would be enough for an analogue of the one direction of Kleene's theorem, for the purposes of our completeness argument, we need to argue something stronger. Namely, we show that we can uniquely solve each system purely through the means of syntactic manipulation using the axioms of $\equiv$. This is where the main complexity of the completeness proof is located. We show this property, by re-adapting the key result of Salomaa [Sal66] to the systems of equations of our interest.

The proof of the uniqueness of solutions theorem will proceed by induction on the size of the systems of equations. We first show that systems with only one unknown can be solved via the (Unique) fixpoint axiom. Systems of the size $n+1$ can be reduced to systems of size n, by solving for one of the unknowns and substituting the obtained equation with n unknowns to the remaining n equations. We note that this reduction step is highly reliant on axioms (D2) and (S0). In particular, we will rely on the following property to generalise left distributivity to arbitrary n-ary convex sums.

Lemma 2.5.17. *Let $f \in \mathsf{PExp}$, I be a finite index set and let $\{p_i\}_{i\in I}$ and $\{e_i\}_{i\in I}$ be indexed collections of probabilities and expressions, respectively. Then:*

$$\left(\bigoplus_{i\in I} p_i \cdot e_i\right) ; f \equiv_b \bigoplus_{i\in I} p_i \cdot e_i \,; f$$

Proof. By induction. If $I = \emptyset$, then:

$$\begin{aligned} f\,;\left(\bigoplus_{i\in I} p_i \cdot e_i\right) &\equiv f\,;0 \\ &\equiv 0 && (\mathsf{S0}) \\ &\equiv \bigoplus_{i\in I} p_i \cdot f\,;e_i && (I=\emptyset) \end{aligned}$$

If there exists $j \in I$ such that $p_j = 1$, then:

$$\begin{aligned} f\,;\left(\bigoplus_{i\in I} p_i \cdot e_i\right) &\equiv_b f\,;e_j \\ &\equiv \left(\bigoplus_{i\in I} p_i \cdot f\,;e_i\right) \end{aligned}$$

Finally, for the induction step, we obtain the following:

$$\begin{aligned} f\,;\left(\bigoplus_{i\in I} p_i \cdot e_i\right) &\equiv f\,;\left(e_j \oplus_{p_j} \left(\bigoplus_{i\in I\setminus\{j\}} \frac{p_i}{1-p_j} \cdot e_i\right)\right) \\ &\equiv f\,;e_j \oplus_{p_j} f\,;\left(\bigoplus_{i\in I\setminus\{j\}} \frac{p_i}{1-p_j} \cdot e_i\right) && (\mathbf{D1}) \\ &\equiv f\,;e_j \oplus_{p_j} \left(\bigoplus_{i\in I\setminus\{j\}} \frac{p_i}{1-p_j} \cdot f\,;e_i\right) && \text{(Induction hypothesis)} \\ &\equiv \left(\bigoplus_{i\in I} p_i \cdot f\,;e_i\right) && \square \end{aligned}$$

Before proceeding, we demonstrate how to solve a system of equations with two unknowns.

Example 2.5.18. Consider the transition system from the left-hand side of Example 2.1.2. Recall from Example 2.5.14, that a map $h\colon \{q_0, q_1\} \to \mathsf{PExp}$ is an $\equiv$-solution to the system associated with that transition system, if and only if:

$$h(q_0) \equiv a\,;h(q_1) \qquad h(q_1) \equiv a\,;h(q_1) \oplus_{\frac{1}{4}} 1$$

Since $E(a) = 0$, we can apply the (Unique) axiom and $e\,;1 \equiv e$ to the equation on the right to deduce that $h(q_1) \equiv a^{[\frac{1}{4}]}$. Substituting it into the left equation yields $h(q_0) \equiv a\,;a^{[\frac{1}{4}]}$.

Finally, we proceed to the central result.

Theorem 2.5.19. *Every left-affine system of equations has a unique solution modulo the equivalence relation* $\equiv$.

Proof. We will write $\mathcal{M} = (M, p, b, r)$ for an arbitrary left-affine system of equations on some finite set Q. Since Q is finite and non-empty, we can safely assume that $Q = \{q_1, \dots q_n\}$ for some positive $n \in \mathbb{N}$. We proceed by induction on n.

$\boxed{\text{Base case}}$ If $Q = \{q_1\}$, then we set $h(q_1) = {M_{1,1}}^{[p_{1,1}]}\,;b_1$. To see that it is indeed a $\equiv$-solution, observe the following:

$$
\begin{aligned}
h(q_1) &= {M_{1,1}}^{[p_{1,1}]}\,;b_1 && \text{(Def. of } h\text{)}\\
&\equiv \left(M_{1,1}\,;{M_{1,1}}^{[p_{1,1}]} \oplus_{p_{1,1}} 1\right);b_1 && \text{(Unroll)}\\
&\equiv M_{1,1}\,;{M_{1,1}}^{[p_{1,1}]}\,;b_1 \oplus_{p_{1,1}} b_1 && \text{(D1)}\\
&\equiv M_{1,1}\,;h(q_1) \oplus_{p_{1,1}} b_1 && \text{(Def. of } h\text{)}\\
&\equiv p_{1,1} \cdot (M_{1,1}\,;h(q_1)) \oplus (1 - p_{1,1}) \cdot b_1\\
&\equiv p_{1,1} \cdot (M_{1,1}\,;h(q_1)) \oplus r_1 \cdot b_1 && (r_1 = 1 - p_{1,1})
\end{aligned}
$$

Given an another $\equiv$-solution $g\colon Q \to \mathsf{PExp}$, we have that:

$$
\begin{aligned}
g(q_1) &\equiv p_{1,1} \cdot (M_{1,1}\,;g(q_1)) \oplus r_1 \cdot b_1\\
&\equiv p_{1,1} \cdot (M_{1,1}\,;g(q_1)) \oplus (1 - p_{1,1}) \cdot b_1 && (r_1 = 1 - p_{1,1})\\
&\equiv M_{1,1}\,;g(q_1) \oplus_{p_{1,1}} b_1\\
&\equiv {M_{1,1}}^{[p_{1,1}]}\,;b_1 && \text{((Unique) and } E(M_{1,1}) = 0\text{)}\\
&\equiv h(q_1)
\end{aligned}
$$

$\boxed{\text{Induction step}}$ Assume that all systems of the size n admit a unique $\equiv$-solution. We begin by demonstrating that the problem of finding a $\equiv$-solution for a system with $n+1$ unknowns can be reduced to finding $\equiv$-solution to the system with n unknowns. Let $Q = \{q_1, \ldots, q_{n+1}\}$. For $h\colon Q \to \mathsf{PExp}$ to be a $\equiv$-solution to the system $\mathcal{M}$ of size $n+1$, it must satisfy the following equivalence:

$$h(q_{n+1}) \equiv \left(\bigoplus_{i=1}^{n+1} p_{n+1,i} \cdot M_{n+1,i} \,; h(q_i)\right) \oplus r_{n+1} \cdot b_{n+1}$$

We can expand the $(n+1)$-ary sum in the equation above using Lemma 2.4.5 and obtain the following:

$$\begin{aligned} h(q_{n+1}) \equiv M_{n+1,n+1} \,; h(q_{n+1}) \oplus_{p_{n+1,n+1}} & \\ \left(\left(\bigoplus_{i=1}^{n} \frac{p_{n+1,i}}{1-p_{n+1,n+1}} \cdot M_{n+1,i} \,; h(q_i)\right) \oplus \frac{r_{n+1}}{1-p_{n+1,n+1}} \cdot b_{n+1}\right) & \end{aligned}$$

Since $E(M_{n+1,n+1}) = 0$, we can apply the (Unique) axiom to derive the following:

$$h(q_{n+1}) \equiv M_{n+1,n+1}^{[p_{n+1,n+1}]} \,; \left(\left(\bigoplus_{i=1}^{n} \frac{p_{n+1,i}}{1-p_{n+1,n+1}} \cdot M_{n+1,i} \,; h(q_i)\right) \oplus \frac{r_{n+1}}{1-p_{n+1,n+1}} \cdot b_{n+1}\right)$$

Observe that the above expression depends only on $h(q_1), \ldots h(q_n)$.). We now substitute the equation for $h(q_{n+1})$ into the equations for $h(q_1), \ldots, h(q_n)$. Before proceeding, recall that for any q_j, such that $j \leq n$, we have:

$$h(q_j) \equiv p_{j,n+1} \cdot M_{j,n+1} \,; h(q_{n+1}) \oplus \left(\bigoplus_{i=1}^{n} p_{j,i} \cdot M_{j,i} \,; h(q_i)\right) \oplus r_j \cdot b_j$$

Substituting $h(q_{n+1})$ now yields the following:

$$\begin{aligned} h(q_j) \equiv \left(\bigoplus_{i=1}^{n} p_{j,i} \cdot M_{j,i} \,; h(q_i)\right) \oplus r_j \cdot b_j & \\ \oplus\, p_{j,n+1} \cdot M_{j,n+1} \,; M_{n+1,n+1}^{[p_{n+1,n+1}]} \,; \left(\left(\bigoplus_{i=1}^{n} \frac{p_{n+1,i}}{1-p_{n+1,n+1}} \cdot M_{n+1,i} \,; h(q_i)\right) \oplus \frac{r_{n+1}}{1-p_{n+1,n+1}} \cdot b_{n+1}\right) & \end{aligned}$$

Applying Lemma 2.5.17, we can rewrite the above as:

$$
\begin{aligned}
h(q_j) \equiv & \left(\bigoplus_{i=1}^{n} p_{j,i} \cdot M_{j,i} \, ; h(q_i) \right) \oplus r_j \cdot b_j \\
& \oplus \left(\bigoplus_{i=1}^{n} \frac{p_{j,n+1} p_{n+1,i}}{1 - p_{n+1,n+1}} \cdot M_{j,n+1} \, ; M_{n+1,n+1}^{[p_{n+1,n+1}]} \, ; M_{n+1,i} \, ; h(q_i) \right) \\
& \oplus \frac{p_{j,n+1} r_{n+1}}{1 - p_{n+1,n+1}} \cdot \left(M_{j,n+1} \, ; M_{n+1,n+1}^{[p_{n+1,n+1}]} \, ; b_{n+1} \right)
\end{aligned}
$$

To simplify the expression above, we introduce the following shorthand notation fo $i, j \in \{1, \ldots, n\}$:

$$
s_{j,i} = p_{j,i} + \frac{p_{j,n+1} p_{n+1,i}}{1 - p_{n+1,n+1}}
$$

$$
N_{j,i} = \frac{p_{j,i}}{s_{j,i}} \cdot M_{j,i} \oplus \frac{p_{j,n+1} p_{n+1,i}}{s_{j,i}(1 - p_{n+1,n+1})} \cdot \left(M_{j,n+1} \, ; M_{n+1,n+1}^{[p_{n+1,n+1}]} \, ; M_{n+1,i} \right)
$$

$$
t_j = r_j + \frac{p_{j,n+1} r_{n+1}}{p_{n+1,n+1}}
$$

$$
c_j = \frac{r_j}{t_j} \cdot b_j \oplus \frac{p_{j,n+1} r_{n+1}}{t_j(1 - p_{n+1,n+1})} \cdot M_{j,n+1} \, ; M_{n+1,n+1}^{[p_{n+1,n+1}]} \, ; b_{n+1}
$$

Note that $E(N_{j,i}) = 0$ for all $i, j \in \{1, \ldots, n\}$. Now, applying Lemma 2.2.5 and Lemma 2.4.7, we obtain the following for all $j \in \{1, \ldots, n\}$:

$$
h(q_j) = \left(\bigoplus_{i=1}^{n} s_{j,i} \cdot N_{j,i} \, ; h(q_i) \right) \oplus t_j \cdot c_j
$$

In other words, the restriction of h to $\{q_1, \ldots, q_n\}$ must be a $\equiv$-solution to the left-affine system $\mathcal{T} = (N, s, c, t)$, which, by the induction hypothesis, has a unique solution. This solution can be extended to the entire system $\mathcal{S}$, by defining $h(q_n)$ as follows:

$$h(q_{n+1}) \equiv M_{n+1,n+1}^{[p_{n+1,n+1}]} ; \left(\left(\bigoplus_{i=1}^{n} \frac{p_{n+1,i}}{1-p_{n+1,n+1}} \cdot M_{n+1,i} ; h(q_i) \right) \oplus \frac{r_{n+1}}{1-p_{n+1,n+1}} \cdot b_{n+1} \right)$$

By applying the (Unroll) axiom and reversing the axiomatic manipulations outlined above, it can be shown that this is indeed a $\equiv$-solution to $\mathcal{S}$.

To prove the $\equiv$-uniqueness of h, assume that $g \colon Q \to \mathsf{PExp}$, is another $\equiv$-solution to the system $\mathcal{S}$. Because of (Unique) axiom, we have that:

$$g(q_{n+1}) \equiv M_{n+1,n+1}^{[p_{n+1,n+1}]} ; \left(\left(\bigoplus_{i=1}^{n} \frac{p_{n+1,i}}{1-p_{n+1,n+1}} \cdot M_{n+1,i} ; g(q_i) \right) \oplus \frac{r_{n+1}}{1-p_{n+1,n+1}} \cdot b_{n+1} \right)$$

Substituting this into the equations for $h(q_1), \ldots, h(q_n)$ and following the same steps as before leads to the requirement that, for all $j \in \{1, \ldots, n\}$, we have:

$$g(q_j) = \left(\bigoplus_{i=1}^{n} s_{j,i} \cdot N_{j,i} ; g(q_i) \right) \oplus t_j \cdot c_j$$

By the induction hypothesis, the left-affine system of equations $\mathcal{T}$ admits a unique $\equiv$-solution. Therefore for all $j \in \{1, \ldots, n\}$, we have that $g(q_j) \equiv h(q_j)$. Consequently, we have that:

$$\begin{aligned} g(q_{n+1}) &\equiv M_{n+1,n+1}^{[p_{n+1,n+1}]} ; \left(\left(\bigoplus_{i=1}^{n} \frac{p_{n+1,i}}{1-p_{n+1,n+1}} \cdot M_{n+1,i} ; g(q_i) \right) \oplus \frac{r_{n+1}}{1-p_{n+1,n+1}} \cdot b_{n+1} \right) \\ &\equiv M_{n+1,n+1}^{[p_{n+1,n+1}]} ; \left(\left(\bigoplus_{i=1}^{n} \frac{p_{n+1,i}}{1-p_{n+1,n+1}} \cdot M_{n+1,i} ; h(q_i) \right) \oplus \frac{r_{n+1}}{1-p_{n+1,n+1}} \cdot b_{n+1} \right) \\ &\equiv h(q_{n+1}) \end{aligned}$$

This completes the proof. □

2.5.4 Step 4: Correspondence of solutions and homomorphisms

We are not done yet, as in the last step we only proved properties of systems of equations and their solutions, while our main interest is in appropriate $\hat{\mathcal{G}}$-coalgebras and their homomorphisms. As desired, it turns out that $\equiv$-solutions are in one-to-one correspondence with $\hat{\mathcal{G}}$-coalgebra homomorphisms from determinisations of (converted) finite state $\mathcal{DF}$-coalgebras to the coalgebra structure on $\mathsf{PExp}/{\equiv}$.

Lemma 2.5.20. *For a finite set X, we have the following one-to-one correspondence:*

$$\frac{\hat{\mathcal{G}}\text{-coalgebra homomorphisms } m\colon ((\mathcal{D}X,\mu_X),(\gamma_X\circ\beta)^\sharp)\to((\mathsf{PExp}/{\equiv},\alpha_{\equiv}),d)}{\equiv\text{-solutions } h\colon X\to\mathsf{PExp} \text{ to a system } \mathcal{S}(\beta) \text{ associated with } \mathcal{D}F\text{-coalgebra } (X,\beta)}$$

Before diving into the main argument, we establish the following helper lemma, which provides a concrete characterization of $\equiv$-solutions to systems associated with $\mathcal{DF}$-coalgebras.

Lemma 2.5.21. *Let (X,β) be a finite-state $\mathcal{DF}$-coalgebra. A map $h\colon X\to\mathsf{PExp}$ is a $\equiv$-solution to the system $\mathcal{S}(\beta)$ if and only if for all $x\in X$, we have that:*

$$h(x)\equiv\left(\bigoplus_{(a,x')\in A\times X}\beta(x)(a,x')\cdot a\,;h(x')\right)\oplus\beta(x)(\checkmark)\cdot 1$$

Proof. Fix an arbitrary $x\in X$. Recall that, if $p^{\beta}_{x,x'}=0$, then $M^{\beta}_{x,x'}=0\equiv\bigoplus_{a\in A}0\cdot a$. Because of that, we can safely assume that $M^{\beta}_{x,x'}$ can be always written out in the following form:

$$M^{\beta}_{x,x'}\equiv\bigoplus_{a\in A}s^{a}_{x,x'}\cdot a$$

where $s^{a}_{x,x'}\in[0,1]$. By definition of $\equiv$-solution, we the following:

$$\begin{aligned}
h(x)&\equiv\left(\bigoplus_{x'\in X}p^{\beta}_{x,x'}\cdot M^{\beta}_{x,x'}\,;h(x')\right)\oplus r^{\beta}_{x}\cdot b^{\beta}_{x}\\
&\equiv\left(\bigoplus_{x'\in X}p^{\beta}_{x,x'}\cdot\left(\bigoplus_{a\in A}s^{a}_{x,x'}\cdot a\right);h(x')\right)\oplus r^{\beta}_{x}\cdot b^{\beta}_{x}\\
&\equiv\left(\bigoplus_{x'\in X}p^{\beta}_{x,x'}\cdot\left(\bigoplus_{a\in A}s^{a}_{x,x'}\cdot a\,;h(x')\right)\right)\oplus r^{\beta}_{x}\cdot b^{\beta}_{x} && \text{(Lemma 2.4.7)}\\
&\equiv\left(\bigoplus_{(a,x')\in A\times X}p^{\beta}_{x,x'}s^{a}_{x,x'}\cdot a\,;h(x')\right)\oplus\beta(x)(\checkmark)\cdot 1 && \text{(Lemma 2.2.4)}
\end{aligned}$$

$$\equiv \left(\bigoplus_{(a,x')\in A\times X} \beta(x)(a,x')\cdot a\,; h(x') \right) \oplus \beta(x)(\checkmark)\cdot 1$$

The last step of the proof relies on the observation that for both cases of how $M^{\beta}_{x,x'}$ is defined (Definition 2.5.15), we have that $p^{\beta}_{x,x'}s^{a}_{x,x'} = \beta(x)(a,x')$. Similarly, in the transition between the third and fourth lines, we apply the following identity:

$$r^{\beta}_{x}\cdot b^{\beta}_{x} \equiv \beta(x)(\checkmark)\cdot 1,$$

which holds for both cases in the definition of b^{β}_{x} (see Definition 2.5.15). The necessity of distinguishing these two cases for $M^{\beta}_{x,x'}$ and b^{β}_{x} arises from their definitions: trivial corner cases must be treated separately to avoid division by zero. □

We are now ready to prove the main claim.

Proof of Lemma 2.5.20. First, assume that $h\colon X \to \mathsf{PExp}$ is a solution to the system $\mathcal{S}(\beta)$ associated with $\mathcal{DF}$-coalgebra (X,β). We show that $([-]\circ h)^{\sharp}\colon \mathcal{D}X \to \mathsf{PExp}/{\equiv}$ is a $\hat{\mathcal{G}}$-coalgebra homomorphism. In other words, we will claim the commutativity of the diagram below.

$$\begin{tikzcd}
X \arrow[dr, "\eta_X"] \arrow[drr, "{[-]\circ h}"] \arrow[dd, "\beta"'] & & \\
& \mathcal{D}X \arrow[r, "{([-]\circ h)^{\sharp}}"'] \arrow[ddl, "{(\gamma_X\circ\beta)^{\sharp}}"] & \mathsf{PExp}/{\equiv} \arrow[dd, "d"] \\
\mathcal{DF}X \arrow[d, "\gamma_X"'] & & \\
\hat{\mathcal{G}}\mathcal{D}X \arrow[rr, "{\hat{\mathcal{G}}([-]\circ h)^{\sharp}}"] & & \hat{\mathcal{G}}\mathsf{PExp}/{\equiv}
\end{tikzcd}$$

Because of Lemma 2.5.21, we have that for all $x \in X$:

$$h(x) \equiv \left(\bigoplus_{(a,x')\in A\times X} \beta(x)(a,x')\cdot a\,; h(x') \right) \oplus \beta(x)(\checkmark)\cdot 1$$

Let $\nu \in \mathcal{D}X$. The convex extension of the map $[-]\circ h\colon X \to \mathsf{PExp}/{\equiv}$ is given by the following:

$$([-]\circ h)^{\sharp}(\nu) = \left[\bigoplus_{x\in\mathrm{supp}(\nu)} \nu(x)\cdot h(x)\right]$$

Similarly, given $\beta : X \to \mathcal{DF}X$ we have that:

$$(\gamma_X \circ \beta)^{\sharp}(\nu) = \left\langle \sum_{x\in\mathrm{supp}(\nu)} \nu(x)\beta(x)(\checkmark), \lambda a.\, \lambda x'. \sum_{x\in\mathrm{supp}(\nu)} \nu(x)\beta(x)(a,x')\right\rangle$$

For any distribution $\nu \in \mathcal{D}X$, we have the following:

$$\begin{aligned}
&d\circ([-]\circ h)^{\sharp}(\nu)\\
&= d\left[\bigoplus_{x\in\mathrm{supp}(\nu)} \nu(x)\cdot\left(\left(\bigoplus_{(a,x')\in A\times X} \beta(x)(a,x')\cdot a\,; h(x')\right)\oplus\beta(x)(\checkmark)\cdot 1\right)\right]\\
&= d\left[\left(\bigoplus_{(a,x')\in A\times X}\left(\sum_{x\in\mathrm{supp}(\nu)} \nu(x)\beta(x)(a,x').\right)\cdot a\,; h(x')\right)\right.\\
&\qquad\qquad \left.\oplus\left(\sum_{x\in\mathrm{supp}(\nu)} \nu(x)\beta(x)(\checkmark)\right)\cdot 1\right] \qquad\qquad \text{(Barycenter axiom)}\\
&= \left\langle \sum_{x\in\mathrm{supp}(\nu)} \nu(x)\beta(x)(\checkmark), \lambda a.\left[\bigoplus_{x'\in X}\left(\sum_{x\in\mathrm{supp}(\nu(x))} \nu(x)\beta(x)(a,x')\right)\cdot h(x')\right]\right\rangle
\end{aligned}$$

Now, consider $\hat{\mathcal{G}}([-]\circ h)^{\sharp}\circ(\gamma_X\circ\beta)^{\sharp}(\nu)$. We have the following:

$$\begin{aligned}
&\hat{\mathcal{G}}([-]\circ h)^{\sharp}\circ(\gamma_X\circ\beta)^{\sharp}(\nu)\\
&= \hat{\mathcal{G}}([-]\circ h)^{\sharp}\left\langle \sum_{x\in\mathrm{supp}(\nu)} \nu(x)\beta(x)(\checkmark), \lambda a.\, \lambda x'. \sum_{x\in\mathrm{supp}(\nu)} \nu(x)\beta(x)(a,x')\right\rangle\\
&= \left\langle \sum_{x\in\mathrm{supp}(\nu)} \nu(x)\beta(x)(\checkmark), \lambda a.\,([-]\circ h)^{\sharp}\left(\lambda x'. \sum_{x\in\mathrm{supp}(\nu)} \nu(x)\beta(x)(a,x')\right)\right\rangle\\
&= \left\langle \sum_{x\in\mathrm{supp}(\nu)} \nu(x)\beta(x)(\checkmark), \lambda a.\left[\bigoplus_{x'\in X}\left(\sum_{x\in\mathrm{supp}(\nu)} \nu(x)\beta(x)(a,x')\right)\cdot h(x')\right]\right\rangle
\end{aligned}$$

Hence, we obtain $d\circ([-]\circ h)^{\sharp}(\nu) = \hat{\mathcal{G}}([-]\circ h)^{\sharp}\circ(\gamma_X\circ\beta)^{\sharp}$, thus demonstrating that $([-]\circ h)^{\sharp}$ is indeed a $\hat{\mathcal{G}}$-coalgebra homomorphism.

For the converse, let $((\mathcal{D}X, \mu_X), (\gamma_X \circ \beta)^\sharp)$ be a $\hat{\mathcal{G}}$-coalgebra and let $m \colon \mathcal{D}X \to \mathsf{PExp}/{\equiv}$ be a $\hat{\mathcal{G}}$-coalgebra homomorphism from $((\mathcal{D}X, \mu_X), (\gamma_X \circ \beta)^\sharp)$ to $((\mathsf{PExp}/{\equiv}, \alpha_{\equiv}), d)$. Recall that m arises uniquely as a convex extension of some map $\overline{h} \colon X \to \mathsf{PExp}/{\equiv}$. This map can be factored as $\overline{h} = [-] \circ h$, for some $h \to \mathsf{PExp}$. Observe that any two such factorisations determine the same $\equiv$-solution. In particular, let $s \colon \mathsf{PExp}/{\equiv} \to \mathsf{PExp}$ be a section of $[-] \colon \mathsf{PExp} \to \mathsf{PExp}/{\equiv}$ and define $h = s \circ \overline{h}$.

Since m is a homomorphism, the inner square in the diagram above commutes. Moreover, as the triangle diagrams also commute, it follows that the outer diagram commutes as well. Recall Lemma 2.5.10, which states that $d \colon \mathsf{PExp}/{\equiv} \to \hat{\mathcal{G}}\mathsf{PExp}/{\equiv}$ is an isomorphism. Consequently, for all $x \in X$ we have the following:

$$
\begin{aligned}
([-] \circ h)(x) &= (d^{-1} \circ \hat{\mathcal{G}}([-] \circ h)^\sharp \circ \gamma_x \circ \beta)(x) \\
&= (d^{-1} \circ \hat{\mathcal{G}}([-] \circ h)^\sharp) \left\langle \beta(x)(\checkmark), \lambda a.\, \lambda x'.\, \beta(x)(a, x') \right\rangle \\
&= d^{-1} \left\langle \beta(x)(\checkmark), \lambda a.\, ([-] \circ h)^\sharp(\lambda x'.\, \beta(x)(a, x')) \right\rangle \\
&= d^{-1} \left\langle \beta(x)(\checkmark), \lambda a.\, \left[\bigoplus_{x' \in X} \beta(x)(a, x') \cdot h(x') \right] \right\rangle \\
&= d^{-1} \left\langle \beta(x)(\checkmark), \lambda a.\, \boxplus_{x' \in X} \beta(x)(a, x') \cdot \left[h(x') \right] \right\rangle && \text{(Lemma 2.4.12)} \\
&= \left[\left(\bigoplus_{(a, x') \in A \times X} \beta(x)(a, x') \cdot a \,; h(x') \right) \beta(x)(\checkmark) \cdot 1 \right] && \text{(Def. of } d^{-1}\text{)}
\end{aligned}
$$

This establishes the desired result. □

Aside from the completeness argument, the above result also gives us an analogue of (one direction of) Kleene's theorem for $\mathcal{DF}$-coalgebras as a corollary. The other direction, converting PRE to finite-state $\mathcal{DF}$-coalgebras is given by the Antimirov construction, described in Section 2.3.

Corollary 2.5.22. *Let (X, β) be a finite-state $\mathcal{DF}$-coalgebra. For every state $x \in X$, there exists an expression $e_x \in \mathsf{PExp}$, such that the probabilistic language denoted by x is the same as $[\![e_x]\!]$.*

Proof. Let $h\colon X \to \mathsf{PExp}$ be the solution to the system $\mathcal{S}(\beta)$ associated with a $\mathcal{DF}$-coalgebra (X, β) existing because of Theorem 2.5.19. For each $x \in X$, set $e_x = h(x)$. Recall that because of Lemma 2.5.20, $([-] \circ h)^\sharp\colon \mathcal{D}X \to \mathsf{PExp}/{\equiv}$ is a $\hat{\mathcal{G}}$-coalgebra homomorphism from $(\mathcal{D}X, (\gamma_X \circ \beta)^\sharp)$ to $(\mathsf{PExp}/{\equiv}, d)$. For any $x \in X$, we have the following:

$$\begin{aligned}
[\![e_x]\!] &= [\![h(x)]\!] && \text{(Def. of } e_x\text{)}\\
&= \mathsf{beh}_d([h(x)]) && \text{(Lemma 2.4.19)}\\
&= \mathsf{beh}_d \circ ([-] \circ h)^\sharp(\eta_X(x)) && \text{(Free-forgetful adjunction)}\\
&= \mathsf{beh}_{(\gamma_X \circ \beta)^\sharp}(\eta_X(x)) \qquad = \mathsf{Lang}_{(X,\beta)}(x) && \text{(}([-] \circ h)^\sharp \text{ is a homomorphism)}
\end{aligned}$$

This completes the proof. □

2.5.5 Step 5: Establish the universal property

The only remaining piece allowing us to use Theorem 2.5.4 is to claim that the $\hat{\mathcal{G}}$-coalgebra is locally finitely presentable. We indirectly rely on finiteness of Antimirov derivatives shown in Lemma 2.3.10.

Lemma 2.5.23. *$((\mathsf{PExp}/{\equiv}, \alpha_{\equiv}), d)$ is locally finitely presentable $\hat{\mathcal{G}}$-coalgebra.*

Proof. We establish the simpler conditions of [Mil10, Definition III.7]. Recall that in PCA locally presentable and locally generated objects coincide [SW15]. Because of that, it suffices to show that every finitely generated subalgebra is contained in finitely generated subcoalgebra. Let (Y, α_Y) be a finitely generated subalgebra of $(\mathsf{PExp}/{\equiv}, \alpha_{\equiv})$ generated by $[e_1], \dots, [e_n] \in \mathsf{PExp}/{\equiv}$ where $1 \leq i \leq n$. We will construct a finitely generated subalgebra (Z, α_Y) of $(\mathsf{PExp}/{\equiv}, \alpha_{\equiv})$ such that $[e_i] \in Z$ (hence containing (Y, α_Y) as subalgebra) that is subcoalgebra as well.

Recall that given an expression $e \in \mathsf{PExp}$, we write $\langle e \rangle_\partial \subseteq \mathsf{PExp}$ for the set of all expressions reachable from e. By Lemma 2.3.10 that set is finite. Let (Z, α_Z) be a subalgebra of $(\mathsf{PExp}/{\equiv}, \alpha_{\equiv})$ generated by the following set

$$\left\{ [a\,; e'] \mid a \in A, e' \in \bigcup_{i=1}^{i \leq n} \langle e_i \rangle_\partial \right\} \cup \{1\}$$

Note that this set is finite, as A is finite and there are only finitely many expressions e_i, each with finitely many derivatives. We proceed to showing that Z is closed under the transitions of d. Pick an element $z \in Z$, given by the following:

$$z = p \cdot [1] \boxplus \bigboxplus_{j \in J} p_j \cdot [a_j \,; e'_j]$$

Note that:

$$d([a_k \,; e'_j]) = \langle 0, f_j \rangle \text{ with } f_j(a) = [e'_j] \text{ if } a = a_j \text{ or otherwise } f(a) = [0]$$
$$d([1]) = \langle 1, f \rangle \text{ with } f(a) = [0] \text{ for all } a \in A$$

Therefore, we can conclude that:

$$d(z) = \left\langle p, \lambda a. \left[\bigoplus_{a_j = a} p_j \cdot e'_j \right] \right\rangle$$

As a consequence of Theorem 2.4.3, for all $j \in J$ we have that $[e'_j] = \left[q_j \cdot 1 \oplus \bigoplus_{k \in K} p_{j,k} \cdot a_{j,k} \,; e'_{j,k} \right]$, where $[e'_{i,k}] \in \bigcup_{i=1}^{i \leq n} \langle e_i \rangle_\partial$ and hence $[a_{j,k} \,; e'_{j,k}] \in Z$ for all $k \in K$. To complete the proof, we will argue that $d(z) \in \hat{\mathcal{G}}(Z, \alpha_Z)$. Observe the following:

$$\begin{aligned} d(y) &= \left\langle p, \lambda a. \left[\bigoplus_{a_j = a} p_j \cdot \left(q_j \cdot 1 \oplus \bigoplus_{k \in K} p_{j,k} \cdot a_{j,k} \,; e'_{j,k} \right) \right] \right\rangle \\ &= \left\langle p, \lambda a. \left[\sum_{a_j = a} p_j q_j \cdot 1 \oplus \bigoplus_{\substack{(j,k) \in J \times K \\ a_j = a}} p_j p_{i,k} \cdot a_{j,k} \,; e'_{j,k} \right] \right\rangle && \text{(Lemma 2.2.4)} \\ &= \left\langle p, \lambda a. \left(\sum_{a_j = a} p_j q_j \cdot [1] \boxplus \bigboxplus_{\substack{(j,k) \in J \times K \\ a_j = a}} p_j p_{i,j} \cdot \left[a_{j,k} \,; e'_{j,k} \right] \right) \right\rangle \\ & && \text{(Lemma 2.4.12)} \end{aligned}$$

Now, it remains to observe the following for all $a \in A$:

$$\sum_{\substack{j,k \in J \times K \\ a_j = a}} p_j(p_{i,k} + q_j) \leq \sum_{j \in J} p_j \leq 1 - p$$

Hence $d(y) \in \hat{\mathcal{G}}(Z, \alpha_Z)$, as desired. $\square$

We are now ready to obtain the following result.

Corollary 2.5.24. $((\mathsf{PExp}/{\equiv}, \alpha_{\equiv}), d)$ *is isomorphic to the rational fixpoint of the functor* $\hat{\mathcal{G}}$ *and is a subcoalgebra of the final* $\hat{\mathcal{G}}$*-coalgebra.*

Proof. Follows from Lemma 2.5.6, Theorem 2.5.19, Lemma 2.5.20 and Lemma 2.5.23. Since in PCA finitely presentable and finitely generated objects coincide (Theorem 2.2.7) and $\hat{\mathcal{G}}$ preserves non-empty monomorphisms (Remark 2.5.5), we have that the rational fixpoint of $\hat{\mathcal{G}}$ is fully abstract (Theorem 2.2.8). $\square$

The only thing is to connect the final $\hat{\mathcal{G}}$-coalgebra with the final $\overline{\mathcal{G}}$-coalgebra, which is carried by $[0,1]^{A^*}$.

Lemma 2.5.25. *The final* $\hat{\mathcal{G}}$*-coalgebra is a subcoalgebra of the final* $\overline{\mathcal{G}}$*-coalgebra.*

Proof. Let $\nu\hat{\mathcal{G}}$ be the final $\hat{\mathcal{G}}$-coalgebra and $\nu\overline{\mathcal{G}}$ be the final $\overline{\mathcal{G}}$-coalgera. Since $\nu\hat{\mathcal{G}}$ can be seen as a $\overline{\mathcal{G}}$-coalgebra, there is a unique $\overline{\mathcal{G}}$-coalgebra homomorphism $\mathsf{beh}_{\nu\hat{\mathcal{G}}} \colon \nu\hat{\mathcal{G}} \to \nu\overline{\mathcal{G}}$. Since $\mathcal{D} \colon \mathsf{Set} \to \mathsf{Set}$ preserves epimorphisms [Gum00, Corollary 3.16], we have that epi-mono factorisations in Set carry to epi-mono factorisations in PCA [Wiß22, Proposition 3.7]. Moreover, since $\overline{\mathcal{G}}$ preserves non-empty monomorphisms (Lemma 2.2.9), we can further lift epi-mono factorisations in PCA to epi-mono factorisations in $\mathsf{Coalg}\,\overline{\mathcal{G}}$ [MPW20, Lemma 2.5]. Because of this, we can factorise $\mathsf{beh}_{\nu\overline{\mathcal{G}}}$ in the following way:

$$\begin{tikzcd} \nu\hat{\mathcal{G}} \arrow[r, tail, two heads, "e"] \arrow[rr, bend right, tail, "\mathsf{beh}_{\nu\hat{\mathcal{G}}}"'] & Q \arrow[r, hook, "m"] & \nu\overline{\mathcal{G}} \end{tikzcd}$$

In the above, Q is a $\overline{\mathcal{G}}$-coalgebra, $e \colon \nu\hat{\mathcal{G}} \to Q$ a surjective $\overline{\mathcal{G}}$-coalgebra homomorphism, and $m \colon Q \to \nu\overline{\mathcal{G}}$ an injective $\overline{\mathcal{G}}$-coalgebra homomorphism. We will argue that $e \colon \nu\hat{\mathcal{G}} \to Q$ is an isomorphism.

First, we can use Lemma 2.5.9 to show that Q is a $\hat{\mathcal{G}}$-coalgebra and $e \colon \nu\hat{G} \to Q$ is a $\hat{\mathcal{G}}$-coalgebra homomorphism. Because of this there exists a unique map $\mathsf{beh}_Q \colon Q \to \nu\hat{\mathcal{G}}$ that is a $\hat{\mathcal{G}}$-coalgebra homomorphism. Then, $\mathsf{beh}_Q \circ e \colon \nu\hat{\mathcal{G}} \to \nu\hat{\mathcal{G}}$ is a $\hat{\mathcal{G}}$-homomorphism that by finality of $\nu\hat{\mathcal{G}}$ must be equal to $\mathsf{id}_{\nu\hat{\mathcal{G}}}$.

To see that $e \circ \mathsf{beh}_Q \colon Q \to Q$ is an identity, observe that by finality of $\nu\overline{\mathcal{G}}$, the following two maps must be equal:

$$m \circ e \circ \mathsf{beh}_Q = m \circ \mathsf{id}_Q$$

Since $m \colon Q \to \nu\overline{\mathcal{G}}$ is monic, we can cancel it on the left and obtain $e \circ \mathsf{beh}_Q = \mathsf{id}_Q$, as desired. Since e is an isomorphism, we have that $\mathsf{beh}_{\nu\hat{\mathcal{G}}}$ is injective, which completes the proof. □

A direct consequence of the above is the following:

Corollary 2.5.26. *The map* $\mathsf{beh}_d \colon \mathsf{PExp}/{\equiv} \to [0,1]^{A^*}$ *is injective.*

Proof. Recall that beh_d is a unique $\overline{\mathcal{G}}$-coalgebra homomorphism from $((\mathsf{PExp}/{\equiv}, \alpha_{\equiv}), d)$ to the final $\overline{\mathcal{G}}$-coalgebra carried by the set $[0,1]^{A^*}$. Since $((\mathsf{PExp}/{\equiv}, \alpha_{\equiv}), d)$ is the rational fixpoint of the functor of $\hat{\mathcal{G}}$, beh_d can be factorised as follows:

$$\begin{tikzcd} \mathsf{PExp}/{\equiv} \arrow[r, hook] \arrow[rr, bend right, hook, "\mathsf{beh}_d"'] & \nu\hat{\mathcal{G}} \arrow[r, hook] & {[0,1]^{A^*}} \end{tikzcd}$$

Due to Corollary 2.5.24 and Lemma 2.5.25, the maps involved in the above factorisations are injective, which implies that beh_d is also injective. □

At this point, showing completeness becomes straightforward.

Theorem 2.5.27. *Let* $e, f \in \mathsf{PExp}$. *If* $[\![e]\!] = [\![f]\!]$, *then* $e \equiv f$.

Proof. We have the following:

$$[\![e]\!] = [\![f]\!] \iff \mathsf{beh}_d([e]) = \mathsf{beh}_d([f]) \qquad \text{(Lemma 2.4.19)}$$

$$\Longrightarrow [e] = [f] \qquad (\mathsf{beh}_d \text{ is injective})$$

$$\Longleftrightarrow e \equiv f \qquad \square$$

2.6 Discussion

In this chapter, we introduced probabilistic regular expressions (PRE), a probabilistic counterpart to Kleene's regular expressions. As the main technical contribution, we presented a Salomaa-style inference system for reasoning about probabilistic language equivalence of expressions and proved it sound and complete. Additionally, we gave a two-way correspondence between languages denoted by PRE and finite-state generative probabilistic transition systems. Our approach is coalgebraic and enabled us to reuse several recently proved results on fixpoints of functors and convex algebras. This abstract outlook guided the choice of the right formalisms and enabled us to isolate the key results we needed to prove to achieve completeness while at the same time reusing existing results and avoiding repeating complicated combinatorial proofs. The key technical lemma, on uniqueness of solutions to certain systems of equations, is a generalisation of automata-theoretic constructions from the 60s further exposing the bridge between our probabilistic generalisation and the classical deterministic counterpart.

2.6.1 Related work

Probabilistic process algebras and their axiomatisations have been widely studied [BS01; SS00; MOW03; Ber22] with syntaxes featuring action prefixing and least fixed point operators instead of the regular operations of sequential composition and probabilistic loops. This line of research focussed on probabilistic bisimulation, while probabilistic language equivalence, which we focus on, stems from automata theory, e.g. the work on Rabin automata [Rab63]. Language equivalence of Rabin automata has been studied from an algorithmic point of view [Kie+11; Kie+12].

Stochastic Regular Expressions (SRE) [Ros00; Bee17; GPG18], which were one of the main inspirations for this chapter, can also be used to specify probabilistic languages. The syntax of SRE features probabilistic Kleene star and n-ary probabilistic choice, however, it does not include 0 and 1. The primary context of that line of research was around genetic programming in probabilistic pattern matching, and the topic of axiomatisation was simply not tackled.

PRE can be thought of as a fragment of ProbGKAT [Róż+23], a probabilistic extension of a strictly deterministic fragment of Kleene Algebra with Tests, that was studied only under the finer notion of bisimulation equivalence. The completeness ProbGKAT was obtained through a different approach to ours, as it relied on a powerful axiom scheme to solve systems of equations.

Our soundness result, as well as semantics via generalised determinisation, were inspired by the work of Silva and Sokolova [SS11], who introduced a two-sorted process calculus for reasoning about probabilistic language equivalence of GPTS. Unlike PRE, their language syntactically excludes the possibility of introducing recursion over terms which might immediately terminate. Moreover, contrary to our completeness argument, their result hinges on the subset of axioms being complete with respect to bisimilarity, similarly to the complete axiomatisation of trace congruence of LTS due to Rabinovich [Rab93]. The use of coalgebra to model trace/language semantics is a well-studied topic [JSS15; RJL21] and other approaches besides generalised determinisation [Sil+10; BSS17] included the use of Kleisli categories [HJS07] and coalgebraic modal logic [KR15]. We build on the vast line of work on coalgebraic completeness theorems [Jac06; Sil10; SRS21; Mil10; BMS13], coalgebraic semantics of probabilistic systems [VR99; Sok05] and fixpoints of the functors [MPW20; Mil18; SW18].

2.6.2 Future work

A first natural direction is exploring whether one could obtain an *algebraic* axiomatisation of PRE. Similarly to Salomaa's system, our axiomatisation is unsound under substitution of letters by arbitrary expressions in the case of the termination operator used to give side condition to the unique fixpoint axiom. We are interested if one could give an alternative inference system in the style of Kozen's axiomatisation [Koz94], in which the Kleene star is the least fixpoint wrt the natural order induced by the $+$ operation and thus not requiring the side condition to introduce loops. In the case of PRE, the challenge is that there is no obvious way of defining a natural order on PRE in terms of $\oplus_p$ operation.

Finally, an interesting direction would be to ask if the finer relation $\equiv_b$ is complete with respect to the probabilistic bisimilarity. One could view it as a probabilistic analogue of the problem of completeness of Kleene Algebra modulo bisimilarity posed by Milner [Mil84], which was recently answered positively by Grabmayer [Gra22].

Chapter 3

Conclusions and Future Work

We conclude the thesis, by summarising the key contributions and sketching the potential directions for future work.

3.1 Completeness theorems for behavioural distances

The starting point of the first part of the thesis was a paper by Bacci, Bacci, Larsen and Mardare [Bac+18a], who used a (relaxed version of) quantitative equational theories [MPP16] to axiomatise probabilistic bisimilarity distance between terms of probabilistic process algebra of Stark and Smolka [SS00]. While that result heavily hinged on properties of Kantorovich lifting used to define the behavioural distance, the key observation was that properties necessary for completeness proof are not exclusive to Kantorovich lifting, but rather can be adapted to other instances of behavioural distances stemming from the abstract coalgebraic framework [Bal+18].

In **??**, we have focused on the simplest and most intuitive instantiation of the coalgebraic framework in the case of deterministic automata. As a central contribution, we have obtained a sound and complete axiomatisation of the shortest-distinguishing-word distance between regular expressions. An interesting difference between our result and previous work was the fact that Kleene's star does not break non-expansivity (unlike μ-recursion operator in the case of probabilistic bisimilarity distance [Bac+18a]) that allowed us to rely on the framework of quantitative equational theories, without any ad-hoc modifications to it.

In **??**, we looked at a more involved case of Milner's charts [Mil84], a straightforward generalisation of nondeterministic automata with variable outputs that presents a compelling setting to study behavioural distances, as it shifts focus from linear-time behaviours to branching-time semantics and represents a crucial step towards more complicated models, such as weighted transition systems [LFT11]. Rather than directly following Milner and using an involved process algebraic syntax with binders and μ-recursion operator, we have relied on a compositional, string diagrammatic syntax, building on a previous line of work on string diagrammatic approaches to automata theory [Pie+24; Ant+25].

One of the key contributions of **??** was providing an axiomatic system for reasoning about distances between string diagrams. Besides the recent work of Lobbia et al [Lob+24], who provided basic examples of total variation distance between stochastic matrices and preorders on matrices, our work is the first one to propose a sound and complete quantitative calculus of string diagrams. Despite multiple similarities, our axiomatisation cannot be expressed in the framework of Lobbia et al [Lob+24], which only permits purely equational axioms. Reconciling these two, by providing a more general framework for quantitive axiomatisations of string diagrams permitting implicational rules and axiom schemes is an interesting research direction.

As much as the usage of Hausdorff distance in defining behavioural distance of charts led to a more involved completeness proof, the proof strategy was essentially the same as in **??** and in the work of Bacci, Bacci, Larsen and Mardare [Bac+18a]. This suggests the possibility of developing a more generic framework of axiomatisations of behavioural distances parametric on the branching type of the system and the associated lifting. One of the directions could be following the work of Schmid et al [Sch+22], who generalised Milner's charts and an algebra of regular behaviours to coalgebras for the type functor $T(V + A \times (-))$, where $T \colon \mathsf{Set} \to \mathsf{Set}$ is an underlying functor of a monad $\mathbf{T}$ and a family of process algebras parametric on algebraic operations appearing in the presentation of $\mathbf{T}$. We envision that such a framework would rely on liftings of $T \colon \mathsf{Set} \to \mathsf{Set}$ to $\overline{T} : \mathsf{PMet} \to \mathsf{PMet}$ that are nonexpansive with respect to sup norm and the lifted monad $\overline{\mathbf{T}}$ can be presented as a quantitative theory in which one can arbitrarily closely approximate the distance between terms, from the approximations of distances between the variables. A good starting point could be the class of quantitative theories studied by Mardare et al [MPP16], where all the axioms are so-called *continuous equation schematas*, that precisely enable such approximations. Additionally, it would be interesting to see if such a general framework of axiomatisations of behavioural distances could be reconciled with work on fixpoint extension of quantitative equational theories [MPP21].

3.2 Probabilistic language equivalence

The second part of the thesis explored obtaining a sound and complete axiomatisation of language equivalence of generative probabilistic transition systems [GSS95] through the syntax of probabilistic regular expressions generalising Kleene's regular expressions to a probabilistic setting. Our starting point were recent hard results on properties of convex algebras and fixpoints that Milius [Mil18], Sokolova and Woracek [SW15; SW18]. Those results enabled the use of an abstract framework of proper functors [Mil18] that allowed us to reduce an involved completeness problem to a generalisation of automata-theoretic results that were studied by Salomaa [Sal66], Kleene [Kle51] and Brzozowski [Brz64] more than fifty years ago. Our work provided further evidence that proper functors are a good abstraction for coalgebraic completeness theorems, that cast completeness as proving a certain universal property obtained by uniquely solving finite systems of fixpoint equations.

In the conclusion of Chapter 2, we left several interesting areas for future work, such as obtaining an algebraic axiomatisation in the style of Kozen [Koz94]. Besides this concrete direction, a natural broader research direction would be to extend a generic calculus for weighted automata of Bonsangue et al [BMS13] to coalgebras over arbitrary proper functors. We stipulate that this would crucially rely on extracting the syntax from the presentation of the functor, as was the case in the work of Bonsangue and Kurz [BK06], Silva [Sil10] and more generally in coalgebraic modal logic [Sch08].

Finally, there is a natural research question intersecting both parts of the thesis, namely axiomatising behavioural distance between probabilistic languages denoted by probabilistic regular expressions. The generic framework of coalgebraic behavioural distances can be applied to linear-time behaviours obtained via generalised determinisation [Sil+10] through lifting distributive laws [Bal+18]. In the case of generative probabilistic transition systems, this would yield a variant of a total variation distance between observable words. An immediate obstacle is that one of the key properties used in quantitative completeness theorems in the first part of the thesis was the finiteness of the state-spaces. Unfortunately, as mentioned in Chapter 2, determinising a finite-state generative probabilistic transition system always results in an infinite state-space. The usage of proper functors was crucial in allowing us to reduce completeness to looking at state-spaces that freely generated convex combinations of finitely many elements. We hope that a similar approach could be helpful in a quantitative case.

Bibliography

[AMV06] Jiří Adámek, Stefan Milius, and Jiri Velebil. “Iterative algebras at work”. In: *Math. Struct. Comput. Sci.* 16.6 (2006), pp. 1085–1131 (cit. on p. 35).

[And+14] Carolyn Jane Anderson, Nate Foster, Arjun Guha, Jean-Baptiste Jeannin, Dexter Kozen, Cole Schlesinger, and David Walker. “NetKAT: semantic foundations for networks”. In: *POPL*. 2014, pp. 113–126 (cit. on pp. 9, 20).

[Ant+25] Thibaut Antoine, Robin Piedeleu, Alexandra Silva, and Fabio Zanasi. “A Complete Diagrammatic Calculus for Automata Simulation”. In: *33rd EACSL Annual Conference on Computer Science Logic, CSL 2025, February 10-14, 2025, Amsterdam, Netherlands*. Ed. by Jörg Endrullis and Sylvain Schmitz. Vol. 326. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025, 27:1–27:22 (cit. on pp. 16, 103).

[Ant96] Valentin Antimirov. “Partial derivatives of regular expressions and finite automaton constructions”. In: *Theoretical Computer Science* 155.2 (1996), pp. 291–319. ISSN: 0304-3975 (cit. on pp. 20, 36, 41).

[AR94] J. Adamek and J. Rosicky. *Locally Presentable and Accessible Categories*. London Mathematical Society Lecture Note Series. Cambridge University Press, 1994 (cit. on pp. 27, 74).

[Ard61] Dean N. Arden. “Delayed-logic and finite-state machines”. In: *2nd Annual Symposium on Switching Circuit Theory and Logical Design (SWCT 1961)*. 1961, pp. 133–151 (cit. on pp. 9, 85).

[Bac+18a] Giorgio Bacci, Giovanni Bacci, Kim G. Larsen, and Radu Mardare. "A Complete Quantitative Deduction System for the Bisimilarity Distance on Markov Chains". In: *Log. Methods Comput. Sci.* 14.4 (2018) (cit. on pp. 12, 102, 103).

[Bac+18b] Giorgio Bacci, Giovanni Bacci, Kim G. Larsen, and Radu Mardare. "Complete Axiomatization for the Total Variation Distance of Markov Chains". In: *Proceedings of the Thirty-Fourth Conference on the Mathematical Foundations of Programming Semantics, MFPS 2018, Dalhousie University, Halifax, Canada, June 6-9, 2018*. Ed. by Sam Staton. Vol. 341. Electronic Notes in Theoretical Computer Science. Elsevier, 2018, pp. 27–39 (cit. on p. 12).

[Bac+18c] Giorgio Bacci, Radu Mardare, Prakash Panangaden, and Gordon D. Plotkin. "An Algebraic Theory of Markov Processes". In: *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, Oxford, UK, July 09-12, 2018*. Ed. by Anuj Dawar and Erich Grädel. ACM, 2018, pp. 679–688 (cit. on p. 12).

[Bac76] Roland Carl Backhouse. "Closure algorithms and the star-height problem of regular languages". PhD thesis. Imperial College London, UK, 1976 (cit. on p. 48).

[Bal+18] Paolo Baldan, Filippo Bonchi, Henning Kerstan, and Barbara König. "Coalgebraic Behavioral Metrics". In: *Log. Methods Comput. Sci.* 14.3 (2018) (cit. on pp. 11, 14, 102, 105).

[Bee17] Tobias Beeh. "Transformations between Markov Chains and Stochastic Regular Expressions". MA thesis. University of Stuttgart, 2017 (cit. on pp. 13, 99).

[Ber22] Marco Bernardo. "Probabilistic Trace and Testing Semantics: The Importance of Being Coherent". In: *Found. Trends Program. Lang.* 7.4 (2022), pp. 244–332 (cit. on p. 99).

[BK06] Marcello M. Bonsangue and Alexander Kurz. “Presenting Functors by Operations and Equations”. In: *Foundations of Software Science and Computation Structures, 9th International Conference, FOSSACS 2006, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2006, Vienna, Austria, March 25-31, 2006, Proceedings*. Ed. by Luca Aceto and Anna Ingólfsdóttir. Vol. 3921. Lecture Notes in Computer Science. Springer, 2006, pp. 172–186 (cit. on p. 104).

[BKP18] Filippo Bonchi, Barbara König, and Daniela Petrisan. “Up-To Techniques for Behavioural Metrics via Fibrations”. In: *29th International Conference on Concurrency Theory, CONCUR 2018, September 4-7, 2018, Beijing, China*. Ed. by Sven Schewe and Lijun Zhang. Vol. 118. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018, 17:1–17:17 (cit. on p. 11).

[BMS13] Marcello M. Bonsangue, Stefan Milius, and Alexandra Silva. “Sound and Complete Axiomatizations of Coalgebraic Language Equivalence”. In: *ACM Trans. Comput. Log.* 14.1 (2013), 7:1–7:52 (cit. on pp. 100, 104).

[Bof90] Maurice Boffa. “Une remarque sur les systèmes complets d’identités rationnelles”. In: *RAIRO Theor. Informatics Appl.* 24 (1990), pp. 419–423 (cit. on pp. 9, 19).

[Brz64] Janusz A. Brzozowski. “Derivatives of Regular Expressions”. In: *J. ACM* 11.4 (1964), pp. 481–494 (cit. on pp. 21, 47, 83, 104).

[BS01] Emanuele Bandini and Roberto Segala. “Axiomatizations for Probabilistic Bisimulation”. In: *Automata, Languages and Programming, 28th International Colloquium, ICALP 2001, Crete, Greece, July 8-12, 2001, Proceedings*. Ed. by Fernando Orejas, Paul G. Spirakis, and Jan van Leeuwen. Vol. 2076. Lecture Notes in Computer Science. Springer, 2001, pp. 370–381 (cit. on p. 99).

[BSS17] Filippo Bonchi, Alexandra Silva, and Ana Sokolova. “The Power of Convex Algebras”. In: *28th International Conference on Concurrency Theory (CONCUR 2017)*. Ed. by Roland Meyer and Uwe Nestmann. Vol. 85. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017, 23:1–23:18. ISBN: 978-3-95977-048-4 (cit. on pp. 31, 100).

[BW01] Franck van Breugel and James Worrell. “Towards Quantitative Verification of Probabilistic Transition Systems”. In: *Automata, Languages and Programming, 28th International Colloquium, ICALP 2001, Crete, Greece, July 8-12, 2001, Proceedings*. Ed. by Fernando Orejas, Paul G. Spirakis, and Jan van Leeuwen. Vol. 2076. Lecture Notes in Computer Science. Springer, 2001, pp. 421–432 (cit. on p. 11).

[Che+22] Mingshuai Chen, Joost-Pieter Katoen, Lutz Klinkenberg, and Tobias Winkler. “Does a Program Yield the Right Distribution? - Verifying Probabilistic Programs via Generating Functions”. In: *Computer Aided Verification - 34th International Conference, CAV 2022, Haifa, Israel, August 7-10, 2022, Proceedings, Part I*. Ed. by Sharon Shoham and Yakir Vizel. Vol. 13371. Lecture Notes in Computer Science. Springer, 2022, pp. 79–101 (cit. on p. 20).

[Con12] J.H. Conway. *Regular Algebra and Finite Machines*. Chapman and Hall mathematics series. Dover Publications, Incorporated, 2012. ISBN: 9780486485836 (cit. on p. 9).

[Des+04] Josée Desharnais, Vineet Gupta, Radha Jagadeesan, and Prakash Panangaden. “Metrics for labelled Markov processes”. In: *Theor. Comput. Sci.* 318.3 (2004), pp. 323–354 (cit. on p. 11).

[Dob08] Ernst-Erich Doberkat. “Erratum and Addendum: Eilenberg-Moore algebras for stochastic relations”. In: *Inf. Comput.* 206.12 (2008), pp. 1476–1484 (cit. on pp. 32, 34).

[GPG18] Sinem Getir, Esteban Pavese, and Lars Grunske. “Formal Semantics for Probabilistic Verification of Stochastic Regular Expressions”. In: *Proceedings of the 27th International Workshop on Concurrency, Specification and Programming, Berlin, Germany, September 24-26, 2018*. Ed. by Bernd-Holger Schlingloff and Samira Akili. Vol. 2240. CEUR Workshop Proceedings. CEUR-WS.org, 2018 (cit. on p. 99).

[Gra22] Clemens Armin Grabmayer. “Milner’s Proof System for Regular Expressions Modulo Bisimilarity is Complete: Crystallization: Near-Collapsing Process Graph Interpretations of Regular Expressions”. In: *LICS ’22: 37th Annual ACM/IEEE Symposium on Logic in Computer Science, Haifa, Israel, August 2 - 5, 2022*. Ed. by Christel Baier and Dana Fisman. ACM, 2022, 34:1–34:13 (cit. on p. 101).

[GSS95] Rob J. van Glabbeek, Scott A. Smolka, and Bernhard Steffen. “Reactive, Generative and Stratified Models of Probabilistic Processes”. In: *Inf. Comput.* 121.1 (1995), pp. 59–80 (cit. on pp. 17, 19, 37, 104).

[Gum00] H. Peter Gumm. *Elements Of The General Theory Of Coalgebras*. 2000 (cit. on pp. 13, 35, 66, 97).

[HJS07] Ichiro Hasuo, Bart Jacobs, and Ana Sokolova. “Generic Trace Semantics via Coinduction”. In: *Log. Methods Comput. Sci.* 3.4 (2007) (cit. on p. 100).

[Jac06] Bart Jacobs. “A Bialgebraic Review of Deterministic Automata, Regular Expressions and Languages”. In: *Algebra, Meaning, and Computation, Essays Dedicated to Joseph A. Goguen on the Occasion of His 65th Birthday*. Ed. by Kokichi Futatsugi, Jean-Pierre Jouannaud, and José Meseguer. Vol. 4060. Lecture Notes in Computer Science. Springer, 2006, pp. 375–404 (cit. on pp. 14, 74, 100).

[Jac10] Bart Jacobs. “Convexity, Duality and Effects”. In: *Theoretical Computer Science - 6th IFIP TC 1/WG 2.2 International Conference, TCS 2010, Held as Part of WCC 2010, Brisbane, Australia, September*

20-23, 2010. Proceedings. Ed. by Cristian S. Calude and Vladimiro Sassone. Vol. 323. IFIP Advances in Information and Communication Technology. Springer, 2010, pp. 1–19 (cit. on p. 34).

[JSS15] Bart Jacobs, Alexandra Silva, and Ana Sokolova. "Trace semantics via determinization". In: *J. Comput. Syst. Sci.* 81.5 (2015), pp. 859–879 (cit. on pp. 28, 39, 100).

[Kap+18] Tobias Kappé, Paul Brunet, Alexandra Silva, and Fabio Zanasi. "Concurrent Kleene Algebra: Free Model and Completeness". In: *ESOP*. Vol. 10801. Lecture Notes in Computer Science. Springer, 2018, pp. 856–882 (cit. on p. 9).

[Kie+11] Stefan Kiefer, Andrzej S. Murawski, Joël Ouaknine, Björn Wachter, and James Worrell. "Language Equivalence for Probabilistic Automata". In: *Computer Aided Verification - 23rd International Conference, CAV 2011, Snowbird, UT, USA, July 14-20, 2011. Proceedings*. Ed. by Ganesh Gopalakrishnan and Shaz Qadeer. Vol. 6806. Lecture Notes in Computer Science. Springer, 2011, pp. 526–540 (cit. on pp. 12, 99).

[Kie+12] Stefan Kiefer, Andrzej S. Murawski, Joël Ouaknine, Björn Wachter, and James Worrell. "APEX: An Analyzer for Open Probabilistic Programs". In: *Computer Aided Verification - 24th International Conference, CAV 2012, Berkeley, CA, USA, July 7-13, 2012 Proceedings*. Ed. by P. Madhusudan and Sanjit A. Seshia. Vol. 7358. Lecture Notes in Computer Science. Springer, 2012, pp. 693–698 (cit. on pp. 12, 99).

[KK05] Lucja Kot and Dexter Kozen. "Kleene Algebra and Bytecode Verification". In: *Proceedings of the First Workshop on Bytecode Semantics, Verification, Analysis and Transformation, Bytecode@ETAPS 2005, Edinburgh, UK, April 9, 2005*. Ed. by Fausto Spoto. Vol. 141. Electronic Notes in Theoretical Computer Science 1. Elsevier, 2005, pp. 221–236 (cit. on p. 20).

[Kle51] S.C. Kleene. *Representation of Events in Nerve Nets and Finite Automata*. Memorandum (Rand Corporation). Rand Corporation, 1951 (cit. on pp. 8, 19, 104).

[Koz94] Dexter Kozen. "A Completeness Theorem for Kleene Algebras and the Algebra of Regular Events". In: *Inf. Comput.* 110.2 (1994), pp. 366–390 (cit. on pp. 9, 19, 74, 100, 104).

[KP00] Dexter Kozen and Maria-Christina Patron. "Certification of Compiler Optimizations Using Kleene Algebra with Tests". In: *Computational Logic - CL 2000, First International Conference, London, UK, 24-28 July, 2000, Proceedings*. Ed. by John W. Lloyd, Veronica Dahl, Ulrich Furbach, Manfred Kerber, Kung-Kiu Lau, Catuscia Palamidessi, Luis Moniz Pereira, Yehoshua Sagiv, and Peter J. Stuckey. Vol. 1861. Lecture Notes in Computer Science. Springer, 2000, pp. 568–582 (cit. on p. 20).

[KR15] Bartek Klin and Jurriaan Rot. "Coalgebraic Trace Semantics via Forgetful Logics". In: *Foundations of Software Science and Computation Structures - 18th International Conference, FoSSaCS 2015, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2015, London, UK, April 11-18, 2015. Proceedings*. Ed. by Andrew M. Pitts. Vol. 9034. Lecture Notes in Computer Science. Springer, 2015, pp. 151–166 (cit. on p. 100).

[Kro90] Daniel Krob. "A Complete System of B-Rational Identities". In: *Automata, Languages and Programming, 17th International Colloquium, ICALP90, Warwick University, England, UK, July 16-20, 1990, Proceedings*. Ed. by Mike Paterson. Vol. 443. Lecture Notes in Computer Science. Springer, 1990, pp. 60–73 (cit. on pp. 9, 19).

[KS96] Dexter Kozen and Frederick Smith. "Kleene Algebra with Tests: Completeness and Decidability". In: *CSL*. Vol. 1258. Lecture Notes in Computer Science. Springer, 1996, pp. 244–259 (cit. on p. 9).

[LFT11] Kim G. Larsen, Uli Fahrenberg, and Claus R. Thrane. "Metrics for weighted transition systems: Axiomatization and complexity". In: *Theor. Comput. Sci.* 412.28 (2011), pp. 3358–3369 (cit. on p. 103).

[Lob+24] Gabriele Lobbia, Wojciech Różowski, Ralph Sarkis, and Fabio Zanasi. "Quantitative Monoidal Algebra: Axiomatising Distance with String Diagrams". In: *CoRR* abs/2410.09229 (2024). arXiv: `2410.09229` (cit. on p. 103).

[LS91] Kim Guldstrand Larsen and Arne Skou. "Bisimulation through Probabilistic Testing". In: *Inf. Comput.* 94.1 (1991), pp. 1–28 (cit. on pp. 12, 36, 46, 49).

[Mil10] Stefan Milius. "A Sound and Complete Calculus for Finite Stream Circuits". In: *Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science, LICS 2010, 11-14 July 2010, Edinburgh, United Kingdom*. IEEE Computer Society, 2010, pp. 421–430 (cit. on pp. 14, 35, 74, 95, 100).

[Mil18] Stefan Milius. "Proper Functors and Fixed Points for Finite Behaviour". In: *Log. Methods Comput. Sci.* 14.3 (2018) (cit. on pp. 14, 20, 21, 47, 77, 100, 104).

[Mil84] Robin Milner. "A Complete Inference System for a Class of Regular Behaviours". In: *J. Comput. Syst. Sci.* 28.3 (1984), pp. 439–466 (cit. on pp. 10, 15, 16, 48, 101, 102).

[MOW03] Michael W. Mislove, Joël Ouaknine, and James Worrell. "Axioms for Probability and Nondeterminism". In: *Proceedings of the 10th International Workshop on Expressiveness in Concurrency, EXPRESS 2003, Marseille, France, September 2, 2003*. Ed. by Flavio Corradini and Uwe Nestmann. Vol. 96. Electronic Notes in Theoretical Computer Science. Elsevier, 2003, pp. 7–28 (cit. on p. 99).

[MPP16] Radu Mardare, Prakash Panangaden, and Gordon D. Plotkin. "Quantitative Algebraic Reasoning". In: *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016*. Ed. by Martin Grohe, Eric Koskinen, and Natarajan Shankar. ACM, 2016, pp. 700–709 (cit. on pp. 15, 102, 104).

[MPP21] Radu Mardare, Prakash Panangaden, and Gordon D. Plotkin. "Fixed-Points for Quantitative Equational Logics". In: *36th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2021, Rome, Italy, June 29 - July 2, 2021*. IEEE, 2021, pp. 1–13 (cit. on p. 104).

[MPW20] Stefan Milius, Dirk Pattinson, and Thorsten Wißmann. "A new foundation for finitary corecursion and iterative algebras". In: *Inf. Comput.* 271 (2020), p. 104456 (cit. on pp. 35, 97, 100).

[Par81] David Park. "Concurrency and automata on infinite sequences". In: *Theoretical Computer Science*. Ed. by Peter Deussen. Berlin, Heidelberg: Springer Berlin Heidelberg, 1981, pp. 167–183. ISBN: 978-3-540-38561-5 (cit. on p. 9).

[Pie+24] Robin Piedeleu, Mateo Torres-Ruiz, Alexandra Silva, and Fabio Zanasi. "A Complete Axiomatisation of Equivalence for Discrete Probabilistic Programming". In: *CoRR* abs/2408.14701 (2024). arXiv: `2408.14701` (cit. on p. 103).

[PZ23a] Robin Piedeleu and Fabio Zanasi. "A Finite Axiomatisation of Finite-State Automata Using String Diagrams". In: *Log. Methods Comput. Sci.* 19.1 (2023) (cit. on p. 16).

[PZ23b] Robin Piedeleu and Fabio Zanasi. "An Introduction to String Diagrams for Computer Scientists". In: *CoRR* abs/2305.08768 (2023). arXiv: `2305.08768` (cit. on p. 15).

[Rab63] Michael O. Rabin. “Probabilistic automata”. In: *Information and Control* 6.3 (1963), pp. 230–245. ISSN: 0019-9958 (cit. on pp. 9, 12, 37, 99).

[Rab93] Alexander Moshe Rabinovich. “A Complete Axiomatisation for Trace Congruence of Finite State Behaviors”. In: *Mathematical Foundations of Programming Semantics, 9th International Conference, New Orleans, LA, USA, April 7-10, 1993, Proceedings*. Ed. by Stephen D. Brookes, Michael G. Main, Austin Melton, Michael W. Mislove, and David A. Schmidt. Vol. 802. Lecture Notes in Computer Science. Springer, 1993, pp. 530–543 (cit. on p. 100).

[Red64] V. N. Redko. “On defining relations for the algebra of regular events”. In: *Ukrainskii Matematicheskii Zhurnal* 16 (1 1964), pp. 120–126 (cit. on p. 9).

[RJL21] Jurriaan Rot, Bart Jacobs, and Paul Blain Levy. “Steps and traces”. In: *J. Log. Comput.* 31.6 (2021), pp. 1482–1525 (cit. on p. 100).

[Ros00] Brian J. Ross. “Probabilistic Pattern Matching and the Evolution of Stochastic Regular Expressions”. In: *Appl. Intell.* 13.3 (2000), pp. 285–300 (cit. on pp. 13, 20, 99).

[Róż+23] Wojciech Różowski, Tobias Kappé, Dexter Kozen, Todd Schmid, and Alexandra Silva. “Probabilistic Guarded KAT Modulo Bisimilarity: Completeness and Complexity”. In: *50th International Colloquium on Automata, Languages, and Programming, ICALP 2023, July 10-14, 2023, Paderborn, Germany*. Ed. by Kousha Etessami, Uriel Feige, and Gabriele Puppis. Vol. 261. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023, 136:1–136:20 (cit. on p. 100).

[Róż+25] Wojciech Różowski, Robin Piedeleu, Alexandra Silva, and Fabio Zanasi. “A Diagrammatic Axiomatisation of Behavioural Distance of Nondeterministic Processes”. Under review. 2025 (cit. on p. 16).

[Róż24] Wojciech Różowski. “A Complete Quantitative Axiomatisation of Behavioural Distance of Regular Expressions”. In: *51st International Colloquium on Automata, Languages, and Programming (ICALP 2024)*. Ed. by Karl Bringmann, Martin Grohe, Gabriele Puppis, and Ola Svensson. Vol. 297. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024, 149:1–149:20. ISBN: 978-3-95977-322-5 (cit. on p. 15).

[RS24] Wojciech Różowski and Alexandra Silva. “A Completeness Theorem for Probabilistic Regular Expressions”. In: *Proceedings of the 39th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2024, Tallinn, Estonia, July 8-11, 2024*. Ed. by Pawel Sobocinski, Ugo Dal Lago, and Javier Esparza. ACM, 2024, 66:1–66:14 (cit. on p. 17).

[RS59] M. O. Rabin and D. Scott. “Finite Automata and Their Decision Problems”. In: *IBM Journal of Research and Development* 3.2 (1959), pp. 114–125 (cit. on p. 9).

[Rut00] J.J.M.M. Rutten. “Universal coalgebra: a theory of systems”. In: *Theoretical Computer Science* 249.1 (2000). Modern Algebra, pp. 3–80. ISSN: 0304-3975 (cit. on pp. 13, 20, 50).

[Sal66] Arto Salomaa. “Two Complete Axiom Systems for the Algebra of Regular Events”. In: *J. ACM* 13.1 (Jan. 1966), pp. 158–169. ISSN: 0004-5411 (cit. on pp. 9, 19, 21, 48, 85, 104).

[Sch+21] Todd Schmid, Tobias Kappé, Dexter Kozen, and Alexandra Silva. “Guarded Kleene Algebra with Tests: Coequations, Coinduction, and Completeness”. In: *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, July 12-16, 2021, Glasgow, Scotland (Virtual Conference)*. Ed. by Nikhil Bansal, Emanuela Merelli, and James Worrell. Vol. 198. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021, 142:1–142:14 (cit. on p. 44).

[Sch+22] Todd Schmid, Wojciech Różowski, Alexandra Silva, and Jurriaan Rot. "Processes Parametrised by an Algebraic Theory". In: *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, July 4-8, 2022, Paris, France*. Ed. by Mikolaj Bojanczyk, Emanuela Merelli, and David P. Woodruff. Vol. 229. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022, 132:1–132:20 (cit. on p. 103).

[Sch08] Lutz Schröder. "Expressivity of coalgebraic modal logic: The limits and beyond". In: *Theor. Comput. Sci.* 390.2-3 (2008), pp. 230–247 (cit. on p. 104).

[Sch61] M.P. Schützenberger. "On the definition of a family of automata". In: *Information and Control* 4.2 (1961), pp. 245–270. ISSN: 0019-9958 (cit. on p. 9).

[Sel10] P. Selinger. "A Survey of Graphical Languages for Monoidal Categories". In: *New Structures for Physics*. Springer Berlin Heidelberg, 2010, pp. 289–355. ISBN: 9783642128219 (cit. on p. 15).

[Sil+10] Alexandra Silva, Filippo Bonchi, Marcello M. Bonsangue, and Jan J. M. M. Rutten. "Generalizing the powerset construction, coalgebraically". In: *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2010, December 15-18, 2010, Chennai, India*. Ed. by Kamal Lodaya and Meena Mahajan. Vol. 8. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2010, pp. 272–283 (cit. on pp. 14, 28, 50, 100, 105).

[Sil10] A.M Silva. "Kleene coalgebra". PhD thesis. Radboud Universiteit Nijmegen, 2010 (cit. on pp. 74, 100, 104).

[Sok05] Ana Sokolova. "Coalgebraic analysis of probabilistic systems". PhD thesis. Technische Universiteit Eindhoven, 2005 (cit. on p. 100).

[SRS21] Todd Schmid, Jurriaan Rot, and Alexandra Silva. "On Star Expressions and Coalgebraic Completeness Theorems". In: *Proceedings 37th Conference on Mathematical Foundations of Programming Seman-*

tics, MFPS 2021, Hybrid: Salzburg, Austria and Online, 30th August - 2nd September, 2021. Ed. by Ana Sokolova. Vol. 351. EPTCS. 2021, pp. 242–259 (cit. on p. 100).

[SS00] Eugene W. Stark and Scott A. Smolka. "A complete axiom system for finite-state probabilistic processes". In: *Proof, Language, and Interaction, Essays in Honour of Robin Milner*. Ed. by Gordon D. Plotkin, Colin Stirling, and Mads Tofte. The MIT Press, 2000, pp. 571–596 (cit. on pp. 12, 99, 102).

[SS11] Alexandra Silva and Ana Sokolova. "Sound and Complete Axiomatization of Trace Semantics for Probabilistic Systems". In: *Twenty-seventh Conference on the Mathematical Foundations of Programming Semantics, MFPS 2011, Pittsburgh, PA, USA, May 25-28, 2011*. Ed. by Michael W. Mislove and Joël Ouaknine. Vol. 276. Electronic Notes in Theoretical Computer Science. Elsevier, 2011, pp. 291–311 (cit. on pp. 12, 14, 37–39, 100).

[Sto49] Michael H. Stone. "Postulates for the barycentric calculus". In: *Annali di Matematica Pura ed Applicata* 29 (1949), pp. 25–30 (cit. on p. 24).

[SW15] Ana Sokolova and Harald Woracek. "Congruences of convex algebras". In: *Journal of Pure and Applied Algebra* 219.8 (2015), pp. 3110–3148. ISSN: 0022-4049 (cit. on pp. 20, 32, 34, 74, 95, 104).

[SW18] Ana Sokolova and Harald Woracek. "Proper Semirings and Proper Convex Functors". In: *Foundations of Software Science and Computation Structures - 21st International Conference, FOSSACS 2018, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2018, Thessaloniki, Greece, April 14-20, 2018, Proceedings*. Ed. by Christel Baier and Ugo Dal Lago. Vol. 10803. Lecture Notes in Computer Science. Springer, 2018, pp. 331–347 (cit. on pp. 14, 16, 20, 47, 78, 100, 104).

[Vil09] Cédric Villani. *Optimal Transport*. Springer Berlin Heidelberg, 2009. ISBN: 9783540710509 (cit. on p. 11).

[VR99] Erik P. de Vink and Jan J. M. M. Rutten. "Bisimulation for Probabilistic Transition Systems: A Coalgebraic Approach". In: *Theor. Comput. Sci.* 221.1-2 (1999), pp. 271–293 (cit. on pp. 13, 100).

[Wag+19] Jana Wagemaker, Marcello M. Bonsangue, Tobias Kappé, Jurriaan Rot, and Alexandra Silva. "Completeness and Incompleteness of Synchronous Kleene Algebra". In: *MPC*. 2019, pp. 385–413 (cit. on p. 9).

[Wiß22] Thorsten Wißmann. "Minimality Notions via Factorization Systems and Examples". In: *Log. Methods Comput. Sci.* 18.3 (2022) (cit. on p. 97).