

# 10th Conference on Algebra and Coalgebra in Computer Science

CALCO 2023, June 19–21, 2023, Indiana University Bloomington, IN, USA

Edited by

Paolo Baldan

Valeria de Paiva



*Editors*

**Paolo Baldan** 

University of Padova, Italy  
paolo.baldan@unipd.it

**Valeria de Paiva** 

Topos Institute, Berkeley, CA, USA  
valeria@topos.institute

*ACM Classification 2012*

Theory of computation → Models of computation; Theory of computation → Modal and temporal logics; Theory of computation → Categorical semantics; Theory of computation → Algebraic semantics; Theory of computation → Quantum computation theory; Software and its engineering → Context specific languages

**ISBN 978-3-95977-287-7**

*Published online and open access by*

Schloss Dagstuhl – Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, Saarbrücken/Wadern, Germany. Online available at <https://www.dagstuhl.de/dagpub/978-3-95977-287-7>.

*Publication date*

September, 2023

*Bibliographic information published by the Deutsche Nationalbibliothek*

The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at <https://portal.dnb.de>.

*License*

This work is licensed under a Creative Commons Attribution 4.0 International license (CC-BY 4.0): <https://creativecommons.org/licenses/by/4.0/legalcode>.



In brief, this license authorizes each and everybody to share (to copy, distribute and transmit) the work under the following conditions, without impairing or restricting the authors' moral rights:

- Attribution: The work must be attributed to its authors.

The copyright is retained by the corresponding authors.

Digital Object Identifier: 10.4230/LIPIcs.CALCO.2023.0

**ISBN 978-3-95977-287-7**

**ISSN 1868-8969**

**<https://www.dagstuhl.de/lipics>**

## LIPICs – Leibniz International Proceedings in Informatics

LIPICs is a series of high-quality conference proceedings across all fields in informatics. LIPICs volumes are published according to the principle of Open Access, i.e., they are available online and free of charge.

### *Editorial Board*

- Luca Aceto (*Chair*, Reykjavik University, IS and Gran Sasso Science Institute, IT)
- Christel Baier (TU Dresden, DE)
- Roberto Di Cosmo (Inria and Université de Paris, FR)
- Faith Ellen (University of Toronto, CA)
- Javier Esparza (TU München, DE)
- Daniel Král' (Masaryk University, Brno, CZ)
- Meena Mahajan (Institute of Mathematical Sciences, Chennai, IN)
- Anca Muscholl (University of Bordeaux, FR)
- Chih-Hao Luke Ong (University of Oxford, GB)
- Phillip Rogaway (University of California, Davis, US)
- Eva Rotenberg (Technical University of Denmark, Lyngby, DK)
- Raimund Seidel (Universität des Saarlandes, Saarbrücken, DE and Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Wadern, DE)
- Pierre Senellart (ENS, Université PSL, Paris, FR)

**ISSN 1868-8969**

**<https://www.dagstuhl.de/lipics>**



## ■ Contents

Preface	
<i>Paolo Baldan and Valeria de Paiva</i> .....	0:vii–0:viii
Conference Organization	
.....	0:ix–0:x

### Invited Talks

Integrating Cost and Behavior in Type Theory	
<i>Robert Harper</i> .....	1:1–1:2
Local Completeness for Program Correctness and Incorrectness	
<i>Roberto Bruni</i> .....	2:1–2:2
A Tour on Ecumenical Systems	
<i>Elaine Pimentel and Luiz Carlos Pereira</i> .....	3:1–3:15
The Metatheory of Gradual Typing: State of the Art and Challenges	
<i>Jeremy G. Siek</i> .....	4:1–4:1
Machine-Checked Computational Mathematics	
<i>Assia Mahboubi</i> .....	5:1–5:1

### Regular Papers

Forward and Backward Steps in a Fibration	
<i>Ruben Turkenburg, Harsh Beohar, Clemens Kupke, and Jurriaan Rot</i> .....	6:1–6:18
Structural Operational Semantics for Heterogeneously Typed Coalgebras	
<i>Harald König, Uwe Wolter, and Tim Kräuter</i> .....	7:1–7:17
Interpolation Is (Not Always) Easy to Spoil	
<i>Andrzej Tarlecki</i> .....	8:1–8:19
String Diagram Rewriting Modulo Commutative (Co)Monoid Structure	
<i>Aleksandar Milosavljević, Robin Piedeleu, and Fabio Zanasi</i> .....	9:1–9:17
Strongly Finitary Monads for Varieties of Quantitative Algebras	
<i>Jiří Adámek, Matěj Dostál, and Jiří Velebil</i> .....	10:1–10:14
Generators and Bases for Monadic Closures	
<i>Stefan Zetsche, Alexandra Silva, and Matteo Sammartino</i> .....	11:1–11:19
Bisimilar States in Uncertain Structures	
<i>Jurriaan Rot and Thorsten Wißmann</i> .....	12:1–12:17
A Category for Unifying Gaussian Probability and Nondeterminism	
<i>Dario Stein and Richard Samuelson</i> .....	13:1–13:18
Fractals from Regular Behaviours	
<i>Todd Schmid, Victoria Noquez, and Lawrence S. Moss</i> .....	14:1–14:18

10th Conference on Algebra and Coalgebra in Computer Science (CALCO 2023).

Editors: Paolo Baldan and Valeria de Paiva



Leibniz International Proceedings in Informatics  
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Coinductive Control of Inductive Data Types <i>Paige Randall North and Maximilien Péroux</i> .....	15:1–15:17
Weakly Markov Categories and Weakly Affine Monads <i>Tobias Fritz, Fabio Gadducci, Paolo Perrone, and Davide Trotta</i> .....	16:1–16:17
Many-Valued Coalgebraic Logic: From Boolean Algebras to Primal Varieties <i>Alexander Kurz and Wolfgang Poiger</i> .....	17:1–17:17
Composition and Recursion for Causal Structures <i>Henning Basold and Tanjona Ralaivaosaona</i> .....	18:1–18:17
Aczel-Mendler Bisimulations in a Regular Category <i>Jérémy Dubut</i> .....	19:1–19:18
<b>(Co)algebraic pearls</b>	
Completeness for Categories of Generalized Automata <i>Guido Boccali, Andrea Laretto, Fosco Loregian, and Stefano Luneia</i> .....	20:1–20:14
On Kripke, Vietoris and Hausdorff Polynomial Functors <i>Jiří Adámek, Stefan Milius, and Lawrence S. Moss</i> .....	21:1–21:20
<b>Early Ideas</b>	
CRDTs, Coalgebraically <i>Nathan Liittschwager, Stelios Tsampas, Jonathan Castello, and Lindsey Kuper</i> ...	22:1–22:5
Amortized Analysis via Coinduction <i>Harrison Grodin and Robert Harper</i> .....	23:1–23:6
Higher-Order Mathematical Operational Semantics <i>Sergey Goncharov, Stefan Milius, Lutz Schröder, Stelios Tsampas, and Henning Urbat</i> .....	24:1–24:3

## ■ Preface

This volume contains the proceedings of the 10th Conference on Algebra and Coalgebra in Computer Science (CALCO), held at Indiana University, from June 19th to June 21st, 2023, under the auspices of IFIP WG 1.3 “Foundations of System Specification”. Previous CALCO editions took place in Salzburg (Austria, 2021), London (UK, 2019), Ljubljana (Slovenia, 2017), Nijmegen (the Netherlands, 2015), Warsaw (Poland, 2013), Winchester (UK, 2011), Udine (Italy, 2009), Bergen (Norway, 2007), Swansea (Wales, 2005).

CALCO is a high-level, bi-annual conference formed by joining CMCS (the International Workshop on Coalgebraic Methods in Computer Science) and WADT (the Workshop on Recent Trends in Algebraic Development Techniques). It provides a forum to present and discuss results of theoretical nature on the mathematics of algebras and coalgebras, the way these results can support methods and techniques for software development, as well as experience reports concerning the transfer of the resulting technologies into industrial practice. Typical topics of interest include:

- models and logics
- algebraic and coalgebraic semantics methodologies in software and systems engineering
- specialised models and calculi
- system specification and verification
- tools supporting algebraic and coalgebraic methods
- string diagrams and network theory
- quantum computing.

Following on the tradition started in 2015, also this year’s edition was co-located with the conference Mathematical Foundations of Programming Semantics (MFPS).

The conference featured invited talks by Roberto Bruni, Jeremy Siek and Elaine Pimentel and a Special Session on “*Category theory in Machine Learning*”, organised by Brendan Fong, Brandon Shapiro and Fabio Zanasi, with talks by Jean-Simon Pacaud Lemay on “*Differential Categories and Machine Learning*”, Brandon Shapiro on “*A dynamic monoidal category for deep learning*” and Prakash Panangaden on “*Is there a place for semantics in machine learning?*”. Moreover, Assia Mahboubi and Bob Harper were joint invited speakers for CALCO and MFPS, and there was a joint special session on “*Machine-checked Mathematics*” organised by Assia Mahboubi, with talks by Floris Van Doorn on “*Formalizing sphere eversion using Lean’s mathematical library*”, Yannick Forster on “*Synthetic Computability in Constructive Type Theory*” and Andrei Popescu on “*On the exquisite pleasure of doing coinduction and corecursion in Isabelle*”.

In addition, there were 19 contributed talks, of which 15 were regular papers, 2 (co)algebraic pearls, and 3 early ideas papers. This volume collects the abstracts of the five invited talks, as well as the peer-reviewed papers. We are grateful to the Program Committee members for their hard work in reviewing and selecting the papers.

The Program Committee has also chosen the Best Paper of the conference. The selection process led to the assignment of an ex-aequo award to two papers, namely “*Aczel-Mendler Bisimulations in a Regular Category*” by Jérémy Dubut, and “*Fractals from Regular Behaviours*” by Todd Schmid, Victoria Noquez, Lawrence S. Moss. It was instead the duty of the audience to select the Best Talk. This has been awarded to Dario Stein for his presentation of the paper “*A Category for Unifying Gaussian Probability and Nondeterminism*”, coauthored with Richard Samuelson. Our warmest congratulations to the authors!

10th Conference on Algebra and Coalgebra in Computer Science (CALCO 2023).  
Editors: Paolo Baldan and Valeria de Paiva



Leibniz International Proceedings in Informatics  
LIPIC Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

We would also like to extend our warm thanks to the local organiser, Larry Moss, for his tireless support throughout all phases of the organization, despite the challenges of managing a hybrid conference. We are grateful to Thorsten Wißmann, who served as the publicity chair, as well as Stefan Milius and Alexandra Silva, the former and current chairs of the CALCO steering committee. Additionally, we greatly benefited from the expertise and guidance of Fabio Gadducci and Alexandra Silva, chairs of the previous CALCO edition. Our last acknowledgement goes to Michael Wagner and the LIPIcs team, who provided continuous, accurate and friendly support in the production of these proceedings.

*Paolo Baldan and Valeria de Paiva*



# ■ Conference Organization

## Programme Committee

- Andreas Abel (Gothenburg University)
- Natasha Alechina (Utrecht University)
- Giorgio Bacci (Aalborg University)
- Paolo Baldan (University of Padua)
- Nicolas Behr (CNRS Université Paris Cité)
- Georgiana Caltais (University of Twente)
- Valentina Castiglioni (Reykjavik University)
- Vincenzo Ciancia (Consiglio Nazionale delle Ricerche, Pisa)
- Thomas Colcombet (CNRS, IRIF, Université de Paris)
- Valeria de Paiva (Topos Institute, Berkeley)
- Martin Escardo (University of Birmingham)
- Fabio Gadducci (University of Pisa)
- Holger Giese (Hasso Plattner Institute at the University of Potsdam)
- Patricia Johann (Appalachian State University)
- Michael Johnson (Macquarie University)
- Sara Kalvala (University of Warwick)
- Shin-Ya Katsumata (National Institute of Informatics, Tokyo)
- Sandra Kiefer (University of Oxford)
- Aleks Kissinger (University of Oxford)
- Sławomir Lasota (University of Warsaw)
- Carlos Gustavo Lopez Pombo (University of Buenos Aires)
- Peter Ölveczky (University of Oslo)
- Fernando Orejas (Universitat Politècnica de Catalunya)
- Mehrnoosh Sadrzadeh (University College London)
- Peter Selinger (Dalhousie University)
- Ionut Tutu (Simion Stoilow Institute of Mathematics of the Romanian Academy)
- Henning Urbat (FAU Erlangen-Nürnberg)
- Tarmo Uustalu (Reykjavik University)
- Thorsten Wißmann (FAU Erlangen-Nürnberg)

## Program Committee Chairs

- Paolo Baldan (University of Padua)
- Valeria de Paiva (Topos Institute, Berkeley)

## Additional reviewers

- Christian Adriano
- Benedikt Ahrens
- Quentin Aristote
- Laura Bussi
- Pierre Cagne
- Kostia Chardonnet

10th Conference on Algebra and Coalgebra in Computer Science (CALCO 2023).  
Editors: Paolo Baldan and Valeria de Paiva



Leibniz International Proceedings in Informatics  
LIPICIS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

- Alessandro Di Giorgio
- David Gabelaia
- Bart Jacobs
- Prabhat Jha
- Radu Mardare
- Hernan Melgratti
- Stefan Milius
- Joshua Moerman
- Nima Motamed
- Renato Neves
- Lê Thành Dũng Nguyễn
- Cécilia Pradic
- Exequiel Rivas
- Sven Schneider
- Sven Schneider
- Davide Trotta
- Niccolò Veltri
- Kazuki Watanabe
- Stefan Zetsche

# Integrating Cost and Behavior in Type Theory

Robert Harper  

Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, USA

---

## Abstract

---

The computational view of intuitionistic dependent type theory is as an intrinsic logic of (functional) programs in which types are viewed as specifications of their behavior. Equational reasoning is particularly relevant in the functional case, where correctness can be formulated as equality between two implementations of the same behavior. Besides behavior, it is also important to specify and verify the cost of programs, measured in terms of their resource usage, with respect to both sequential and parallel evaluation. Although program cost can – and has been – verified in type theory using an extrinsic formulation of programs as data objects, what we seek here is, instead, an intrinsic account within type theory itself.

In this talk we discuss Calf, the Cost-Aware Logical Framework, which is an extension of dependent call-by-push-value type theory that provides an intrinsic account of both parallel and sequential resource usage for a variety of problem-specific measures of cost. Thus, for example, it is possible to prove that insertion sort and merge sort are equal as regards behavior, but differ in terms of the number of comparisons required to achieve the same results. But how can equal functions have different cost? To provide an intrinsic account of both intensional and extensional properties of programs, we make use of Sterling’s notion of Synthetic Tait Computability, a generalization of Tait’s method originally developed for the study of higher type theory.

In STC the concept of a “phase” plays a central role: originally as the distinction between the syntactic and semantic aspects of a computability structure, but more recently applied to the formulation of type theories for program modules and for information flow properties of programs. In Calf we distinguish two phases, the intensional and extensional, which differ as regards the significance of cost accounting – extensionally it is neglected, intensionally it is of paramount importance. Thus, in the extensional phase insertion sort and merge sort are equal, but in the intensional phase they are distinct, and indeed one is proved to have optimal behavior as regards comparisons, and the other not. Importantly, both phases are needed in a cost verification – the proof of the complexity of an algorithm usually relies on aspects of its correctness.

We will provide an overview of Calf itself, and of its application in the verification of the cost and behavior of a variety of programs. So far we have been able to verify cost bounds on Euclid’s Algorithm, amortized bounds on batched queues, parallel cost bounds on a joinable form of red-black trees, and the equivalence and cost of the aforementioned sorting methods. In a companion paper at this meeting Grodin and I develop an account of amortization that relates the standard inductive view of instruction sequences with the coinductive view of data structures characterized by the same operations. In ongoing work we are extending the base of verified deterministic algorithms to those taught in the undergraduate parallel algorithms course at Carnegie Mellon, and are extending Calf itself to account for probabilistic methods, which are also used in that course.

(This talk represents joint work with Yue Niu, Harrison Grodin, and Jon Sterling.)

**2012 ACM Subject Classification** Theory of computation → Type theory; Theory of computation → Design and analysis of algorithms; Theory of computation → Logic and verification

**Keywords and phrases** type theory, analysis of algorithms, program verification

**Digital Object Identifier** 10.4230/LIPIcs.CALCO.2023.1

**Category** Invited Talk

**Related Version** *ACM POPL 2022*: <https://doi.org/10.1145/3498670>

**Supplementary Material** *Software*: <https://github.com/jonsterling/agda-calf>  
archived at `swh:1:dir:790153bb2b1e4c1fa942283307af0e0be070d14a8`



© Robert Harper;

licensed under Creative Commons License CC-BY 4.0

10th Conference on Algebra and Coalgebra in Computer Science (CALCO 2023).

Editors: Paolo Baldan and Valeria de Paiva; Article No. 1; pp. 1:1–1:2

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

## 1:2 **Integrating Cost and Behavior in Type Theory**

**Funding** This research was sponsored by the Air Force Office of Scientific Research (AFOSR) under award number A210038S0002 and the National Science Foundation under award number CCF1901381. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

# Local Completeness for Program Correctness and Incorrectness

Roberto Bruni   

Computer Science Department, University of Pisa, Italy

---

## Abstract

Program correctness techniques aim to prove the absence of bugs, but can yield false alarms because they tend to over-approximate program semantics. Vice versa, program incorrectness methods are aimed to detect true bugs, without false alarms, but cannot be used to prove correctness, because they under-approximate program semantics. In this invited talk we will overview our ongoing research on the use of the abstract interpretation framework to combine under- and over-approximation in the same analysis and distill a logic for program correctness and incorrectness.

**2012 ACM Subject Classification** Theory of computation → Logic and verification; Theory of computation → Programming logic; Theory of computation → Hoare logic; Theory of computation → Abstraction

**Keywords and phrases** Program analysis, program verification, Hoare logic, incorrectness logic, abstract interpretation, local completeness

**Digital Object Identifier** 10.4230/LIPIcs.CALCO.2023.2

**Category** Invited Talk

**Funding** Research supported by Italian MIUR PRIN2017 project no. 201784YSZ5 “Analysis of Program Analyses (ASPRA)”, INdAM – GNCS Project, code no. CUP\_E53C22001930001 and by a *Meta research* gift.

## 1 Extended abstract

Floyd-Hoare logic for program correctness [12, 13] was an eye-opening contribution to the use of over-approximation in program verification aimed to prove the absence of errors. From the perspective of programmers, the benefit of the feedback provided by program correctness analyses within the software development ecosystem is appreciated if warnings are reported early and truly [11]. The use of over-approximation is necessary to make the correctness problem tractable and to develop automatic tools, but inevitably it introduces some imprecision. As a consequence verification tools can produce false alarms, i.e., potential errors that are reported by the analysis but that do not correspond to any execution.

Possibly inspired by the consequence rule of Reverse Hoare logic [10], Peter O’Hearn’s recent studies on the use of under-approximation in program analysis have led to the definition of a logic for program incorrectness [17, 18, 19, 16, 14], which, dualising the over-approximation approach of Hoare logic, can be used to exhibit the presence of errors, without false alarms, but not for proving program correctness.

In this talk we will overview our ongoing research [3, 5, 4, 1, 6, 15, 2] on the use of the abstract interpretation framework [8, 9, 7] to combine under- and over-approximation in the same analysis and distill a logic for program correctness and incorrectness. Any triple provable in the logic can be used either to guarantee the correctness of the program or to expose some (true) errors. A key role is played by the notion of locally complete abstraction that provides the necessary proof obligations in logic derivations. Notably different abstract domains can be combined in the same derivation and the logic can be instantiated to different settings, like imperative programming languages and strategy languages for rewrite systems.



© Roberto Bruni;

licensed under Creative Commons License CC-BY 4.0

10th Conference on Algebra and Coalgebra in Computer Science (CALCO 2023).

Editors: Paolo Baldan and Valeria de Paiva; Article No. 2; pp. 2:1–2:2

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

---

**References**

---

- 1 Flavio Ascari, Roberto Bruni, and Roberta Gori. Limits and difficulties in the design of under-approximation abstract domains. In *Proc. of FOSSACS 2022*, volume 13242 of *LNCS*, pages 21–39. Springer, 2022. doi:10.1007/978-3-030-99253-8\_2.
- 2 Flavio Ascari, Roberto Bruni, and Roberta Gori. Logics for extensional, locally complete analysis via domain refinements. In *Proc. of ESOP 2023*, volume 13990 of *LNCS*, pages 1–27. Springer, 2023. doi:10.1007/978-3-031-30044-8\_1.
- 3 Roberto Bruni, Roberto Giacobazzi, Roberta Gori, and Francesco Ranzato. A logic for locally complete abstract interpretations. In *Proc. of LICS 2021*, Distinguished Paper, pages 1–13. IEEE, 2021. doi:10.1109/LICS52264.2021.9470608.
- 4 Roberto Bruni, Roberto Giacobazzi, Roberta Gori, and Francesco Ranzato. Abstract interpretation repair. In *Proc. of PLDI’22*, pages 426–441. ACM, 2022. doi:10.1145/3519939.3523453.
- 5 Roberto Bruni, Roberto Giacobazzi, Roberta Gori, and Francesco Ranzato. A correctness and incorrectness program logic. *J. ACM*, 70(2):15:1–15:45, 2023. doi:10.1145/3582267.
- 6 Marco Campion, Mila Dalla Preda, and Roberto Giacobazzi. Partial (in)completeness in abstract interpretation: limiting the imprecision in program analysis. *Proc. ACM Program. Lang.*, 6(POPL):1–31, 2022. doi:10.1145/3498721.
- 7 Patrick Cousot. *Principles of Abstract Interpretation*. MIT Press, 2021.
- 8 Patrick Cousot and Radhia Cousot. Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Proc. of POPL 1977*, pages 238–252. ACM, 1977. doi:10.1145/512950.512973.
- 9 Patrick Cousot and Radhia Cousot. Systematic design of program analysis frameworks. In *Proc. ACM POPL 1979*, pages 269–282. ACM, 1979. doi:10.1145/567752.567778.
- 10 Edsko de Vries and Vasileios Koutavas. Reverse Hoare logic. In *Proc. of SEFM 2011*, pages 155–171. Springer Berlin Heidelberg, 2011. doi:10.1007/978-3-642-24690-6\_12.
- 11 Dino Distefano, Manuel Fähndrich, Francesco Logozzo, and Peter W. O’Hearn. Scaling static analyses at Facebook. *Commun. ACM*, 62(8):62–70, 2019. doi:10.1145/3338112.
- 12 Robert W. Floyd. Assigning meanings to programs. *Proceedings of Symposium on Applied Mathematics*, 19:19–32, 1967.
- 13 C. A. R. Hoare. An axiomatic basis for computer programming. *Commun. ACM*, 12(10):576–580, October 1969. doi:10.1145/363235.363259.
- 14 Quang Loc Le, Azalea Raad, Jules Villard, Josh Berdine, Derek Dreyer, and Peter W. O’Hearn. Finding real bugs in big programs with incorrectness logic. *Proc. ACM Program. Lang.*, 6(OOPSLA1):1–27, 2022. doi:10.1145/3527325.
- 15 Marco Milanese and Francesco Ranzato. Local completeness logic on Kleene algebra with tests. In *Proc. of SAS 2022*, volume 13790 of *LNCS*, pages 350–371. Springer, 2022. doi:10.1007/978-3-031-22308-2\_16.
- 16 Bernhard Möller, Peter W. O’Hearn, and Tony Hoare. On algebra of program correctness and incorrectness. In *Proc. of RAMiCS 2021*, volume 13027 of *LNCS*, pages 325–343. Springer, 2021. doi:10.1007/978-3-030-88701-8\_20.
- 17 Peter W. O’Hearn. Incorrectness logic. *Proc. ACM Program. Lang.*, 4(POPL):10:1–10:32, 2020. doi:10.1145/3371078.
- 18 Azalea Raad, Josh Berdine, Hoang-Hai Dang, Derek Dreyer, Peter W. O’Hearn, and Jules Villard. Local reasoning about the presence of bugs: Incorrectness separation logic. In *Proc. of CAV 2020, Part II*, volume 12225 of *LNCS*, pages 225–252. Springer, 2020. doi:10.1007/978-3-030-53291-8\_14.
- 19 Azalea Raad, Josh Berdine, Derek Dreyer, and Peter W. O’Hearn. Concurrent incorrectness separation logic. *Proc. ACM Program. Lang.*, 6(POPL):1–29, 2022. doi:10.1145/3498695.

# A Tour on Ecumenical Systems

Elaine Pimentel   

Department of Computer Science, University College London, UK

Luiz Carlos Pereira  

Department of Philosophy, UERJ, Brazil

---

## Abstract

Ecumenism can be understood as a pursuit of unity, where diverse thoughts, ideas, or points of view coexist harmoniously. In logic, ecumenical systems refer, in a broad sense, to proof systems for combining logics. One captivating area of research over the past few decades has been the exploration of seamlessly merging classical and intuitionistic connectives, allowing them to coexist peacefully. In this paper, we will embark on a journey through ecumenical systems, drawing inspiration from Prawitz' seminal work [35]. We will begin by elucidating Prawitz' concept of "ecumenism" and present a pure sequent calculus version of his system. Building upon this foundation, we will expand our discussion to incorporate alethic modalities, leveraging Simpson's meta-logical characterization. This will enable us to propose several proof systems for ecumenical modal logics. We will conclude our tour with some discussion towards a term calculus proposal for the implicational propositional fragment of the ecumenical logic, the quest of automation using a framework based in rewriting logic, and an ecumenical view of proof-theoretic semantics.

**2012 ACM Subject Classification** Theory of computation → Proof theory; Theory of computation → Modal and temporal logics; Theory of computation → Logic and verification; Theory of computation → Type theory

**Keywords and phrases** Intuitionistic logic, classical logic, modal logic, ecumenical systems, proof theory

**Digital Object Identifier** 10.4230/LIPIcs.CALCO.2023.3

**Category** Invited Talk

**Funding** *Elaine Pimentel*: Pimentel has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement Number 101007627.

*Luiz Carlos Pereira*: Pereira is supported by the following projects: CAPES/PRINT, CNPq-313400/2021-0, and CNPq-Gaps and Gluts.

**Acknowledgements** We want to thank Dag Prawitz for the friendship and inspiration, and our co-authors in this endeavour: Valeria de Paiva, Sonia Marin, Emerson Sales, Delia Kesner, Mariana Milicich, Victor Nascimento and Carlos Olarte.

## 1 Introduction

*What is a proof?* In the context of logic and mathematics, a proof is a logical argument that establishes the correctness of a claim based on a set of assumed axioms and definitions, together with previously proven statements. Nevertheless, since the construction methods of these arguments may vary, a proof that appears satisfactory to a classical logician may not necessarily meet the criteria for an intuitionistic logician. For instance, constructive logicians do not accept mathematical proofs that explicitly employ the principle of excluded middle. But does this discrepancy solely pertain to proof methods? What is the real nature of this disagreement?



© Elaine Pimentel and Luiz Carlos Pereira;

licensed under Creative Commons License CC-BY 4.0

10th Conference on Algebra and Coalgebra in Computer Science (CALCO 2023).

Editors: Paolo Baldan and Valeria de Paiva; Article No. 3; pp. 3:1–3:15

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

According to Prawitz [35] the accuracy of an inference relies on the assigned meaning of the *logical constants*, and classical and intuitionistic logicians differ in their interpretations to some of them. The case of disjunction is central in this discussion, since asserting that

$A \vee B$  is valid only if it is possible to give a proof of either  $A$  or  $B$

often claimed to be enough for determining meaning of disjunction in intuitionistic logic, clearly does not correctly determine the meaning of the classical disjunction. In Prawitz' view, classical and intuitionistic logicians would also not agree on the meanings for the implication and existential quantifier, while they would share the same view regarding conjunction, negation, the constant for the absurd and the universal quantifier.

To explore the meanings of all these connectives collectively, Prawitz proposed an all-encompassing language known as *ecumenical logic*, which codifies both classical and intuitionistic reasoning based on a uniform pattern of meaning explanations. In the ecumenical language, the classical and intuitionistic constants coexist harmoniously: the subscript  $c$  is added when denoting the classical meaning, while the subscript  $i$  represents the intuitionistic meaning. This provides a neutral ground for the contestants, as described by Prawitz

“The classical logician is not asserting what the intuitionistic logician denies. For instance, the classical logician asserts  $A \vee_c \neg A$  to which the intuitionist does not object; he objects to the universal validity of  $A \vee_i \neg A$ , which is not asserted by the classical logician.”

We embraced Prawitz's agenda in a series of works, delving into various aspects of ecumenism. In [32], we presented LE, a single-conclusion sequent calculus for Prawitz' original natural deduction ecumenical system. Using proof-theoretic methods, we showed that the ecumenical entailment is intrinsically intuitionistic, but it turns classical in the presence of classical succedents. We then produced a nested sequent version of the original sequent system and showed all of them sound and complete with respect to (first-order extension of) the ecumenical Kripke semantics [31]. Finally, we analysed fragments of the systems presented, coming to well known intuitionistic calculi and a sequent system for classical logic amenable to a treatment by goal directed proof search.

In [22], we lifted this discussion to modal logics, presenting an extension of LE with the alethic modalities of *necessity* and *possibility*. Our proposal for ecumenical modal logics comes in the light of Simpson's meta-logical interpretation of modalities [40] by embedding the expected semantical behavior of the modal operator into ecumenical first-order logic. This resulted in a labelled ecumenical modal system, amenable for modal extensions.

It turns out that the inference rules in the systems presented in [32, 22] are not *pure* [11] or *separable* [25], in the sense that the introduction rules for some connectives strongly depend on the presence of negation. In [23] we presented a *pure label free* calculus for ecumenical modalities, where every basic object of the calculus can be read as a formula in the language of the logic. For that, we used *nested systems* [7, 16, 6, 33] with a stoup [14], together with a new notion of *polarities* for ecumenical formulae.

Recently [24] all these aforementioned studies were revisited, and we started from a pure ecumenical first-order system and naturally expanded it to the modal case. Such pure systems allowed for a clearer notion of the *meaning* for connectives (including modalities), faithfully matching Prawitz' original intention, and the tradition of the *proof-theoretic semantics' school* [38, 39].

Proof-theoretic semantics aims not only to elucidate the meaning of a logical proof, but also to provide means for its use as a basic concept of semantic analysis. Hence while logical ecumenism provides a medium in which meaningful interactions may occur between classical



## INTUITIONISTIC AND NEUTRAL RULES

$$\begin{array}{c}
\frac{A, B, \Gamma \Rightarrow C}{A \wedge B, \Gamma \Rightarrow C} \wedge L \quad \frac{\Gamma \Rightarrow A \quad \Gamma \Rightarrow B}{\Gamma \Rightarrow A \wedge B} \wedge R \quad \frac{A, \Gamma \Rightarrow C \quad B, \Gamma \Rightarrow C}{A \vee_i B, \Gamma \Rightarrow C} \vee_i L \\
\frac{\Gamma \Rightarrow A_j}{\Gamma \Rightarrow A_1 \vee_i A_2} \vee_i R_j \quad \frac{A \rightarrow_i B, \Gamma \Rightarrow A \quad B, \Gamma \Rightarrow C}{\Gamma, A \rightarrow_i B \Rightarrow C} \rightarrow_i L \quad \frac{\Gamma, A \Rightarrow B}{\Gamma \Rightarrow A \rightarrow_i B} \rightarrow_i R \\
\frac{\neg A, \Gamma \Rightarrow A}{\neg A, \Gamma \Rightarrow \perp} \neg L \quad \frac{\Gamma, A \Rightarrow \perp}{\Gamma \Rightarrow \neg A} \neg R \quad \frac{}{\perp, \Gamma \Rightarrow A} \perp L \\
\frac{A[y/x], \forall x.A, \Gamma \Rightarrow C}{\forall x.A, \Gamma \Rightarrow C} \forall L \quad \frac{\Gamma \Rightarrow A[y/x]}{\Gamma \Rightarrow \forall x.A} \forall R \quad \frac{A[y/x], \Gamma \Rightarrow C}{\exists_i x.A, \Gamma \Rightarrow C} \exists_i L \quad \frac{\Gamma \Rightarrow A[y/x]}{\Gamma \Rightarrow \exists_i x.A} \exists_i R
\end{array}$$

## CLASSICAL RULES

$$\begin{array}{c}
\frac{A, \Gamma \Rightarrow \perp \quad B, \Gamma \Rightarrow \perp}{A \vee_c B, \Gamma \Rightarrow \perp} \vee_c L \quad \frac{\Gamma, \neg A, \neg B \Rightarrow \perp}{\Gamma \Rightarrow A \vee_c B} \vee_c R \quad \frac{A \rightarrow_c B, \Gamma \Rightarrow A \quad B, \Gamma \Rightarrow \perp}{A \rightarrow_c B, \Gamma \Rightarrow \perp} \rightarrow_c L \\
\frac{\Gamma, A, \neg B \Rightarrow \perp}{\Gamma \Rightarrow A \rightarrow_c B} \rightarrow_c R \quad \frac{p_i, \Gamma \Rightarrow \perp}{p_c, \Gamma \Rightarrow \perp} L_c \quad \frac{\Gamma, \neg p_i \Rightarrow \perp}{\Gamma \Rightarrow p_c} R_c \\
\frac{A[y/x], \Gamma \Rightarrow \perp}{\exists_c x.A, \Gamma \Rightarrow \perp} \exists_c L \quad \frac{\Gamma, \forall x. \neg A \Rightarrow \perp}{\Gamma \Rightarrow \exists_c x.A} \exists_c R
\end{array}$$

## INITIAL, CUT AND STRUCTURAL RULES

$$\frac{}{p_i, \Gamma \Rightarrow p_i} \text{init} \quad \frac{\Gamma \Rightarrow A \quad A, \Gamma \Rightarrow C}{\Gamma \Rightarrow C} \text{cut} \quad \frac{\Gamma \Rightarrow \perp}{\Gamma \Rightarrow A} \text{W}$$

■ **Figure 1** Ecumenical sequent system LE. In rules  $\forall R, \exists_i L, \exists_c L$ , the eigenvariable  $y$  is fresh;  $p$  is atomic.

and intuitionistic logic, proof-theoretic semantics provides a way of clarifying what is at stake when one accepts or denies *reductio ad absurdum* as a meaningful proof method. In [26] we closed this circle, by showing how to coherently combine both approaches by providing not only a medium in which classical and intuitionistic *logics* may coexist, but also one in which classical and intuitionistic *notions of proof* may coexist.

Finally, building on Girard's original idea of stoup, we presented in [30] an ecumenical pure natural deduction system ( $\text{NE}_p$ ) for the propositional fragment, which seems to be a promising step towards the proposal of a ecumenical term calculus.

In this text, we will synthesise the main aspects of the *op. cit.*, thus providing a tour on ecumenical systems inspired by Prawitz seminal work [35].

## 2 Ecumenical systems

In [35] Dag Prawitz proposed a natural deduction system where classical and intuitionistic logics could coexist in peace. The language  $\mathcal{L}$  used for ecumenical systems is described as follows. We will use a subscript  $c$  for the classical meaning and  $i$  for the intuitionistic one, dropping such subscripts when formulae/connectives can have either meaning.

Classical and intuitionistic  $n$ -ary predicate symbols ( $p_c, p_i, \dots$ ) co-exist in  $\mathcal{L}$  but have different meanings. The neutral logical connectives  $\{\perp, \neg, \wedge, \forall\}$  are common for classical and intuitionistic fragments, while  $\{\rightarrow_i, \vee_i, \exists_i\}$  and  $\{\rightarrow_c, \vee_c, \exists_c\}$  are restricted to intuitionistic and classical interpretations, respectively.

In [32] we presented the system LE (Figure 1), the sequent counterpart of Prawitz' natural deduction system. Sequents are build over  $\mathcal{L}$ -formulae, and have the form  $\Gamma \Rightarrow A$ , where  $\Gamma$  is a multiset. Moving from natural deduction to sequent systems allowed us to carefully analyse the ecumenical notion of *entailment*.

### 3:4 A Tour on Ecumenical Systems

Denoting by  $\vdash_S A$  the fact that the formula  $A$  is a theorem in the proof system  $S$ , we showed that the ecumenical entailment  $\Gamma \Rightarrow A$  is intrinsically intuitionistic, in the following sense.

► **Theorem 1.** *Let  $\Gamma, A$  be a multiset of ecumenical formulae. Then  $\Gamma \Rightarrow A$  is provable in the system LE iff  $\vdash_{LE} \bigwedge \Gamma \rightarrow_i A$*

But when  $A$  is classical, that is, built from classical atomic predicates using only the connectives:  $\rightarrow_c, \vee_c, \exists_c, \neg, \wedge, \forall$  and the unit  $\perp$ , then entailments can be read classically.

► **Theorem 2.** *Let  $A_c$  be a classical formula and  $\Gamma$  be a multiset of ecumenical formulae. Then*

$$\vdash_{LE} \bigwedge \Gamma \rightarrow_c A_c \text{ iff } \vdash_{LE} \bigwedge \Gamma \rightarrow_i A_c.$$

This justifies the ecumenical view of entailments in Prawitz's original proposal.

In [32] the system LE was presented also in a nested sequent version, and all the systems were shown sound and complete w.r.t. (the first-order extension of) the ecumenical Kripke semantics in [31]. Finally, in that work we analysed several fragments of the systems presented.

### 3 The quest for purity

Although being a powerful tool for describing proof-theoretical properties of Prawitz' ecumenical logic, LE is not satisfactory as a logical system since it is not *pure* [11]: the definition of classical connectives depend on other connectives. For example, introducing  $\exists_c$  on the right depends on the presence of negation and the universal quantifier.

One way of *purifying* systems is by introducing the notion of *polarities*. As in linear logic [13], it is possible to polarise formulae [1] into *positive* and *negative* in both classical [14, 18] and intuitionistic [19] logics, where the application of rules is determined by the polarity of the active formula.

The choice of polarization of formulae may vary from system to system, though, as it depends on their intended behaviour. The following rules for the conjunction of positive/negative formulae, represented by  $P, Q$  and  $N, M$  respectively, are characteristic examples of the use of polarities in sequent systems

$$\frac{\Gamma_1 \Rightarrow \Delta_1, P \quad \Gamma_2 \Rightarrow \Delta_2, Q}{\Gamma_1, \Gamma_2 \Rightarrow \Delta_1, \Delta_2, P \wedge Q} \wedge_P \quad \frac{\Gamma \Rightarrow \Delta, N \quad \Gamma \Rightarrow \Delta, M}{\Gamma \Rightarrow \Delta, M \wedge N} \wedge_N$$

In this case, polarities determine the multiplicative/additive behaviour of the rules for conjunction.

Another way of controlling rule applications is by separating the contexts into *bins*. For example, sequents may be restricted for having the form  $\Gamma \Rightarrow \Delta; \Sigma$ , where  $\Gamma, \Delta, \Sigma$  represent sets or multisets of formulae, and the *stoup*  $\Sigma$  is limited to containing *at most* one formula. In such systems, it is common that the active formula in the conclusion of a rule is placed in the stoup.

Usually, in sequent systems polarities and stoup come together. Structural rules then control the movement of formulae in derivations, as in the following *decision* and *store* rules

$$\frac{\Gamma \Rightarrow \Delta; P}{\Gamma \Rightarrow \Delta, P; \cdot} \text{D} \quad \frac{\Gamma \Rightarrow \Delta, N; \cdot}{\Gamma \Rightarrow \Delta; N} \text{store}$$

On a bottom-up reading of these rules, while in D positive formulae can be chosen to be “focused on”, in store negative formulae are stored in the classical context. This often enables for a *two-phase* proof construction, where the focused formula  $P$  is systematically decomposed until reaching a leaf or a negative sub-formula  $N$ . In this last case, focusing is lost and  $N$  is stored, allowing for the beginning of a new focused phase.

Finally, in sequent systems combining polarities and stoup the cut rule can assume different forms, depending on the polarity or the placement of the cut-formula (or both). The following are typical examples of positive and negative cut rules.

$$\frac{\Gamma_1 \Rightarrow \Delta_1; P \quad P, \Gamma_2 \Rightarrow \Delta_2; \Sigma}{\Gamma_1, \Gamma_2 \Rightarrow \Delta_1, \Delta_2; \Sigma} \text{cut}_P \qquad \frac{\Gamma_1 \Rightarrow \Delta_1, N; \cdot \quad N, \Gamma_2 \Rightarrow \Delta_2; \Sigma}{\Gamma_1, \Gamma_2 \Rightarrow \Delta_1, \Delta_2; \Sigma} \text{cut}_N$$

In [24] we made use of polarities and stoups for proposing the pure ecumenical first-order sequent system LCE. Sequents with a stoup in LCE are built over  $\mathcal{L}$ -formulae and have the form  $\Gamma \Rightarrow \Delta; \Sigma$ . Intuitionistic formulae are *positive* and dealt in the stoup, while classical formulae are *negative* and their rules are handled by the classical context  $\Delta$ .

The following states that LCE is correct and complete w.r.t. LE.

► **Theorem 3.** *The sequent  $\Gamma \Rightarrow \Delta; \Sigma$  is provable in LCE iff  $\Gamma, \neg\Delta \Rightarrow \Sigma$  is provable in LE.*

Moreover, it shows that a formula in the classical context actually corresponds to its negated version in the left context. This is justified by the fact that if  $A_c$  is classical, then  $\vdash_{\text{LE}} \Gamma, \neg A_c \Rightarrow \perp$  iff  $\vdash_{\text{LE}} \Gamma \Rightarrow A_c^1$ .

Moving now to the natural deduction setting, in [30] we gave an ecumenical view to Parigot’s natural deduction stoup mechanism [29]. This allowed the definition of the *pure harmonic* natural deduction system  $\text{NE}_p$  (depicted in Figure 2) for the propositional fragment of Prawitz’ ecumenical logic.

While polarities are not considered in  $\text{NE}_p$ , the stoup controls the shape of derivations. The inference rules manipulate *stoups with a context*, which are expressions of the form  $\Delta; \Sigma$ , extensions of natural deduction formulae where  $\Sigma$  is the stoup and  $\Delta$  is its accompanying context (similar to *alternatives* in [36]).

As a derivation example, the following version of Peirce’s Law is provable in  $\text{NE}_p$ .

$$2 \frac{[\cdot; A]^1 \text{ D} \quad \frac{A; \cdot}{A, B; \cdot} \text{W}_c \quad 1 \frac{A, B; \cdot}{A; (A \rightarrow_c B)} \rightarrow_{c\text{-int}} \quad \frac{[\cdot; A]^2 \text{ D}}{A; \cdot} \text{D}}{\frac{A, A; \cdot}{A; \cdot} \text{C} \quad \frac{[\cdot; ((A \rightarrow_c B) \rightarrow_c A)]^3}{A, A; \cdot} \rightarrow_{c\text{-int}}}{\cdot; (((A \rightarrow_c B) \rightarrow_c A) \rightarrow_c A)} \rightarrow_{c\text{-int}} \text{C}$$

More interestingly, any sequent of the form  $((A \rightarrow_j B) \rightarrow_k A) \rightarrow_c A$  with  $j, k \in \{i, c\}$  is provable in  $\text{NE}_p$ . That is, provability is maintained if the outermost implication is classical.

$\text{NE}_p$ ’s normalisation procedure is really interesting, since the presence of stoups enables two kinds of compositions on derivations: in the stoup or in the classical context (see [30] for the details). This reflects, in the natural deduction setting, the two forms of cut for sequent systems with stoup shown above.

<sup>1</sup> Actually the “if” part is valid for any ecumenical formula.

## INTUITIONISTIC AND NEUTRAL RULES

$$\begin{array}{c}
 \frac{\Gamma \quad [;\!; A] \quad \Pi}{\Delta; B} \rightarrow_i\text{-int} \qquad \frac{\Gamma_1 \quad \Gamma_2 \quad \Pi_1 \quad \Pi_2}{\Delta_1; A \rightarrow_i B \quad \Delta_2; A} \rightarrow_i\text{-elim} \\
 \\
 \frac{\Gamma \quad \Pi}{\Delta; A_i} \vee_i\text{-int} \qquad \frac{\Gamma_1 \quad [;\!; A] \quad \Gamma_2 \quad [;\!; B] \quad \Gamma_3 \quad \Pi_1 \quad \Pi_2 \quad \Pi_3}{\Delta_1; A \vee_i B \quad \Delta_2; \Sigma \quad \Delta_3; \Sigma} \vee_i\text{-elim}
 \end{array}$$

## CLASSICAL RULES

$$\begin{array}{c}
 \frac{[;\!; A] \quad \Gamma \quad \Pi}{\Delta, B; \cdot} \rightarrow_c\text{-int} \qquad \frac{\Gamma_1 \quad \Gamma_2 \quad [;\!; B] \quad \Gamma_3 \quad \Pi_1 \quad \Pi_2 \quad \Pi_3}{\Delta_1; A \rightarrow_c B \quad \Delta_2; A \quad \Delta_3; \cdot} \rightarrow_c\text{-elim} \\
 \\
 \frac{\Gamma \quad \Pi}{\Delta, A, B; \cdot} \vee_c\text{-int} \qquad \frac{\Gamma_1 \quad [;\!; A] \quad \Gamma_2 \quad [;\!; B] \quad \Gamma_3 \quad \Pi_1 \quad \Pi_2 \quad \Pi_3}{\Delta_1; A \vee_c B \quad \Delta_2; \cdot \quad \Delta_3; \cdot} \vee_c\text{-elim}
 \end{array}$$

## HYPOTHESIS FORMATION AND STRUCTURAL RULES

$$\begin{array}{c}
 \cdot; A \qquad \frac{\Gamma \quad \Pi}{\Delta; A} \text{D} \qquad \frac{\Gamma \quad \Pi}{\Delta; \cdot} \text{W}_i \qquad \frac{\Gamma \quad \Pi}{\Delta; \Sigma} \text{W}_c \qquad \frac{\Gamma \quad \Pi}{\Delta, A, A; \Sigma} \text{C} \\
 \frac{\Delta; A}{\Delta, A; \cdot} \text{D} \qquad \frac{\Delta; \cdot}{\Delta; A} \text{W}_i \qquad \frac{\Delta; \Sigma}{\Delta, A; \Sigma} \text{W}_c \qquad \frac{\Delta, A, A; \Sigma}{\Delta, A; \Sigma} \text{C}
 \end{array}$$

■ **Figure 2** Ecumenical natural deduction system with stoup  $\text{NE}_p$ .

The design of the proof system is not only a matter of taste: it also allows for adequate proposals for extensions and/or applications. As an example, in [29] Parigot shows that, when trying to establish a link between control operators and classical constructs, a satisfactory notion of reduction for usual natural deduction (with the classical absurdity rule [34]) is hard to achieve. According to him

“The difficulties met in trying to use  $\neg\neg A \rightarrow A$  (or the classical absurdity rule) as a type for control operators is not really due to classical logic, but much more to the deduction system in which it is expressed. It is not easy to find a satisfactory notion of reduction in usual natural deduction because of the restriction to one conclusion which forbids the most natural transformations of proofs (they often generate proofs with more than one conclusion). Of course, as a by-product of our work, we can get possible adequate reductions for usual natural deduction, but none of them can be called “the” canonical one.”

Parigot’s solution for tackling the subject reduction problem was exactly to adopt a system with stoup, where the double negated formulae are stored in the classical context. This served as inspiration to the ongoing work on an ecumenical term calculus, where the  $\lambda\mu$  internalization of *stoups* and the continuation-passing aspect of general rules [37] are naturally mixed together.

On the other hand, the use of polarities and stoup in the sequent setting not only allows for a better proof theoretic view of Prawitz' original proposal, but it also serves as a solid ground for smoothly accommodating modalities [24].

#### 4 Ecumenical modalities

In [22] we lifted the discussion about ecumenism to modal logics, by presenting an extension of EL with the alethic modalities of *necessity* and *possibility*. On doing so, there were many choices to be made and many relevant questions to be asked, *e.g.*: what is the ecumenical interpretation of ecumenical modalities? Should we add classical, intuitionistic, or neutral versions for modal connectives? We proposed an answer for these questions in the light of Simpson's meta-logical interpretation of modalities [40] by embedding the expected semantical behavior of the modal operator into the ecumenical first order logic.

Formally, the language of (*propositional, normal*) *modal formulae* consists of the propositional fragment of the classical language enhanced with the unary *modal operators*  $\Box$  and  $\Diamond$  concerning necessity and possibility, respectively [2]. Given a variable  $x$ , we recall the standard translation  $[\cdot]_x$  from modal formulae into first-order formulae with at most one free variable,  $x$ , as follows: if  $p$  is atomic, then  $[p]_x = p(x)$ ;  $[\perp]_x = \perp$ ; for any binary connective  $\star$ ,  $[A \star B]_x = [A]_x \star [B]_x$ ; for the modal connectives

$$[\Box A]_x = \forall y(R(x, y) \rightarrow [A]_y) \quad [\Diamond A]_x = \exists y(R(x, y) \wedge [A]_y)$$

where  $R(x, y)$  is a binary predicate.  $R(x, y)$  then represents the *accessibility relation*  $R$  in a Kripke frame.

A (object-)modal logic OL is then characterized by the respective interpretation of the modal model in the meta-theory ML (called *meta-logical characterization* [40]) as follows

$$\vdash_{OL} A \quad \text{iff} \quad \vdash_{ML} \forall x.[A]_x$$

Hence, if ML is classical logic (CL), the former definition characterizes the classical modal logic K [2], while if it is intuitionistic logic (IL), then it characterizes the intuitionistic modal logic IK [40]. In [22], we adopted EL as the meta-theory, hence characterizing the ecumenical modal logic EK.

The ecumenical translation  $[\cdot]_x^e$  from propositional ecumenical formulae into LE is defined in the same way as the modal translation  $[\cdot]_x$ . For the case of modal connectives, observe that, due to Theorem 1, the interpretation of ecumenical consequence should be essentially *intuitionistic*. This implies that the box modality is a *neutral connective*. The diamond, on the other hand, has two possible interpretations: classical and intuitionistic, since its leading connective is an existential quantifier. Hence we should have the ecumenical modalities:  $\Box$ ,  $\Diamond_i$ ,  $\Diamond_c$ , determined by the translations

$$\begin{aligned} [\Box A]_x^e &= \forall y(R(x, y) \rightarrow_i [A]_y^e) \\ [\Diamond_i A]_x^e &= \exists_i y(R(x, y) \wedge [A]_y^e) & [\Diamond_c A]_x^e &= \exists_c y(R(x, y) \wedge [A]_y^e) \end{aligned}$$

Setting  $\mathcal{L}_{\mathcal{M}}$  as the ecumenical modal language (that is, built from  $\mathcal{L}$  with ecumenical modalities), the translation above naturally induces the labelled language  $\mathcal{L}_{\mathcal{L}}$  of *labelled modal formulae*, determined by *labelled formulae* of the form  $x : A$  with  $A \in \mathcal{L}_{\mathcal{M}}$  and *relational atoms* of the form  $xRy$ , where  $x, y$  range over a set of variables.

In [22] we proposed a non-pure labelled calculus for ecumenical modal logic. In [24] we achieved purity, as expected, by using polarities and sequents with stoup. *Labelled sequents with stoup* have the form  $\Gamma \Rightarrow \Delta; x : A$ , where  $\Gamma$  is a multiset containing labelled modal

### 3:8 A Tour on Ecumenical Systems

formulae and relational atoms, and  $\Delta$  is a multiset containing labelled modal formulae. The notion of polarities can be lifted from LCE to modalities smoothly, both for labelled and non-labelled calculi. In the former, relational atoms are not polarizable.

In Figure 3 we present the pure, labelled ecumenical modal system **labEK** [24]. Observe that

$$\vdash_{\text{labEK}} x : \diamond_c A \leftrightarrow_i x : \neg \Box \neg A$$

On the other hand,  $\Box$  and  $\diamond_i$  are not inter-definable. However, if  $A_c$  is classical, then

$$\vdash_{\text{labEK}} x : \Box A_c \leftrightarrow_i x : \neg \diamond_c \neg A_c$$

This means that, when restricted to the classical fragment,  $\Box$  and  $\diamond_c$  are duals. This reflects well the ecumenical nature of the defined modalities.

We conclude this section by showing the delicate line separating ecumenical and classical systems. We show how even slight alterations within ecumenical systems can lead to their eventual breakdown and a collapse into the classical framework.

The first example is valid for first-order and modal cases (see [24]).

► **Example 4.** If the cut rule

$$\frac{\Gamma \Rightarrow \Delta, x : A; \Pi^* \quad x : A, \Gamma \Rightarrow \Delta; \Pi}{\Gamma \Rightarrow \Delta; \Pi} \text{ cut}$$

was admissible in **labEK** for an arbitrary formula  $A$ , then  $\cdot \Rightarrow \cdot; x : A \vee_i \neg A$  would have the proof

$$\frac{\frac{\frac{\frac{\frac{\frac{\cdot \Rightarrow x : A \vee_i \neg A; \neg A}{\cdot \Rightarrow x : A \vee_i \neg A; \neg A} \neg R}{\cdot \Rightarrow x : A \vee_i \neg A; x : A \vee_i \neg A} \vee_i D}{\cdot \Rightarrow x : A \vee_i \neg A; \cdot} \vee_i D}{\cdot \Rightarrow x : A \vee_i \neg A; x : A \vee_i \neg A} \vee_i D}{\cdot \Rightarrow x : A \vee_i \neg A; x : A \vee_i \neg A} \vee_i D}{\cdot \Rightarrow \cdot; x : A \vee_i \neg A} \text{ cut}_N$$

Remember that  $x : A \vee_i \neg A$  is positive, hence it can not be the cut-formula in  $\text{cut}_N$ .

The second interesting example regards extensions of the modal logic **EK**, which can be defined by adding extra modal axioms. Many of such axioms can be specified as formulas in first-order logic. For example, in the ecumenical setting, the axiom  $\top : \Box A \rightarrow_i A \wedge A \rightarrow_i \diamond_i A$  is specified by the first-order formula  $\forall x.R(x, x)$ , which corresponds to the rule<sup>2</sup>

$$\frac{xRx, \Gamma \Rightarrow \Delta; \Pi}{\Gamma \Rightarrow \Delta; \Pi} \top$$

The addition of  $\top$  to **EK** yields the system **EKT** [22]. The next example shows that adding the axiom  $\neg \diamond_i \neg A \rightarrow_i \Box A$  to **EKT** has a disastrous propositional consequence.

<sup>2</sup> See [21] for a general framework using polarities and focusing for transforming axioms into rules in the first-order setting and [40, 43, 27] for other seminal works on the subject.

► **Example 5.** The following is a derivation of  $x : A \vee_i \neg A$  in EKT, assuming  $\neg \diamond_i \neg A \rightarrow_i \Box A$  as an axiom<sup>3</sup>

$$\begin{array}{c}
 \frac{xRy, y : A, y : \neg(A \vee_i \neg A) \Rightarrow \cdot; y : A}{xRy, y : A, y : \neg(A \vee_i \neg A) \Rightarrow \cdot; y : \perp} \text{init} \\
 \frac{xRy, y : A, y : \neg(A \vee_i \neg A) \Rightarrow \cdot; y : \perp}{xRy, y : \neg(A \vee_i \neg A) \Rightarrow \cdot; x : \perp} \neg L, \vee_i R_1 \\
 \frac{xRy, y : \neg(A \vee_i \neg A) \Rightarrow \cdot; x : \perp}{x : \diamond_i \neg(A \vee_i \neg A) \Rightarrow \cdot; x : \perp} \neg L, \vee_i R_2, \neg R \\
 \frac{x : \diamond_i \neg(A \vee_i \neg A) \Rightarrow \cdot; x : \perp}{\Rightarrow \cdot; x : \neg \diamond_i \neg(A \vee_i \neg A)} \diamond_i L \\
 \frac{\Rightarrow \cdot; x : \neg \diamond_i \neg(A \vee_i \neg A)}{\Rightarrow \cdot; x : \Box(A \vee_i \neg A)} \neg R \\
 \frac{\Rightarrow \cdot; x : \Box(A \vee_i \neg A)}{\Rightarrow \cdot; x : A \vee_i \neg A} eq \\
 \frac{xRx, x : (A \vee_i \neg A) \Rightarrow \cdot; x : A \vee_i \neg A}{xRx, x : \Box(A \vee_i \neg A) \Rightarrow \cdot; x : A \vee_i \neg A} \text{init} \\
 \frac{xRx, x : \Box(A \vee_i \neg A) \Rightarrow \cdot; x : A \vee_i \neg A}{x : \Box(A \vee_i \neg A) \Rightarrow \cdot; x : A \vee_i \neg A} \Box L \\
 \frac{x : \Box(A \vee_i \neg A) \Rightarrow \cdot; x : A \vee_i \neg A}{\Rightarrow \cdot; x : A \vee_i \neg A} \top \\
 \frac{\Rightarrow \cdot; x : A \vee_i \neg A}{\Rightarrow \cdot; x : A \vee_i \neg A} \text{cut}_P
 \end{array}$$

where *eq* represents the proof steps of the substitution of a boxed formula for its diamond version.<sup>4</sup> That is, if  $\Box$  and  $\diamond_i$  are inter-definable, then  $A \vee_i \neg A$  is a theorem and EKT collapses to classical KT.

## 4.1 A nested system for ecumenical modal logic

In [23] we went one step ahead and proposed a *pure label free* calculus for ecumenical modalities, where every basic object of the calculus can be read as a formula in the language of the logic. The price to pay for getting rid of labels was having to extend sequent systems with *nestings* [7, 16, 6, 33].

This not only allowed for establishing the meaning of modalities via the rules that determine their correct use (*logical inferentialism* [5]), but it also places ecumenical systems as a unifying framework for modalities of which well known modal systems are fragments.

We shall briefly describe the general idea behind a pure label free calculus for ecumenical modalities. First of all, inspired by [41], we adopt the following notation for (one-sided) sequents with stoup:

- formulae in the left context  $\Gamma$  (*left inputs*) will be marked with a full circle  $\bullet$ ;
- formulae in the classical right context  $\Delta$  (*right inputs*) will be marked with a triangle  $\nabla$ ;
- the formula in the stoup  $\Sigma$  (*right output*) will be marked with a white circle  $\circ$ .

Hence, for example, the sequent with stoup  $C \wedge D \Rightarrow \diamond_c A; \neg B$  will be written as  $C \wedge D^\bullet, \diamond_c A^\nabla, \neg B^\circ$ .

Second, we substitute labels for nestings, where a single sequent is replaced with a tree of sequents, whose nodes are multisets of formulae, with the relationship between parent and child in the tree represented by bracketing [·].

For example, the labelled sequent with stoup  $xRy, xRz, z : C \wedge D \Rightarrow x : \diamond_c A; y : \neg B$  corresponds to the nested sequent  $\diamond_c A^\nabla, [\neg B^\circ], [C \wedge D^\bullet]$ , which in turn represents the following tree of sequents with stoup

$$\begin{array}{ccc}
 \Rightarrow \cdot; \neg B & & C \wedge D \Rightarrow \cdot; \cdot \\
 & \searrow & \swarrow \\
 & \Rightarrow \diamond_c A; \cdot &
 \end{array}$$

<sup>3</sup> Observe that  $\vdash_{\text{labEK}} x : \Box A \rightarrow_i \neg \diamond_i \neg A$ .

<sup>4</sup> We have presented a proof with cut for clarity, but remember that labEK has the cut-elimination property [24].

## INTUITIONISTIC AND NEUTRAL RULES

$$\begin{array}{c}
 \frac{\Gamma, x : A, x : B \Rightarrow \Delta; \Pi}{\Gamma, x : A \wedge B \Rightarrow \Delta; \Pi} \wedge L \quad \frac{\Gamma \Rightarrow \Delta; x : A \quad \Gamma \Rightarrow \Delta; x : B}{\Gamma \Rightarrow \Delta; x : A \wedge B} \wedge R \\
 \\
 \frac{\Gamma, x : A \Rightarrow \Delta; \Pi \quad \Gamma, x : B \Rightarrow \Delta; \Pi}{\Gamma, x : A \vee_i B \Rightarrow \Delta; \Pi} \vee_i L \quad \frac{\Gamma \Rightarrow \Delta; x : A_j}{\Gamma \Rightarrow \Delta; x : A_1 \vee_i A_2} \vee_i R_j \\
 \\
 \frac{\Gamma, x : A \rightarrow_i B \Rightarrow \Delta; x : A \quad \Gamma, x : B \Rightarrow \Delta; \Pi}{\Gamma, x : A \rightarrow_i B \Rightarrow \Delta; \Pi} \rightarrow_i L \quad \frac{\Gamma, x : A \Rightarrow \Delta; x : B}{\Gamma \Rightarrow \Delta; x : A \rightarrow_i B} \rightarrow_i R \\
 \\
 \frac{\Gamma, x : \neg A \Rightarrow \Delta; x : A}{\Gamma, x : \neg A \Rightarrow \Delta; \cdot} \neg L \quad \frac{\Gamma, x : A \Rightarrow \Delta; \cdot}{\Gamma \Rightarrow \Delta; x : \neg A} \neg R
 \end{array}$$

## CLASSICAL RULES

$$\begin{array}{c}
 \frac{\Gamma, x : A \rightarrow_c B \Rightarrow \Delta; x : A \quad \Gamma, x : B \Rightarrow \Delta; \cdot}{\Gamma, x : A \rightarrow_c B \Rightarrow \Delta; \cdot} \rightarrow_c L \quad \frac{\Gamma, x : A \Rightarrow x : B, \Delta; \cdot}{\Gamma \Rightarrow x : A \rightarrow_c B, \Delta; \cdot} \rightarrow_c R \\
 \\
 \frac{\Gamma, x : A \Rightarrow \Delta; \cdot \quad \Gamma, x : B \Rightarrow \Delta; \cdot}{\Gamma, x : A \vee_c B \Rightarrow \Delta; \cdot} \vee_c L \quad \frac{\Gamma \Rightarrow x : A, x : B, \Delta; \cdot}{\Gamma \Rightarrow x : A \vee_c B, \Delta; \cdot} \vee_c R \\
 \\
 \frac{\Gamma, x : p_i \Rightarrow \Delta; \cdot}{\Gamma, x : p_c \Rightarrow \Delta; \cdot} L_c \quad \frac{\Gamma \Rightarrow x : p_i, \Delta; \cdot}{\Gamma \Rightarrow x : p_c, \Delta; \cdot} R_c
 \end{array}$$

## MODAL RULES

$$\begin{array}{c}
 \frac{xRy, y : A, x : \Box A, \Gamma \Rightarrow \Delta; \Pi}{xRy, x : \Box A, \Gamma \Rightarrow \Delta; \Pi} \Box L \quad \frac{xRy, \Gamma \Rightarrow \Delta; y : A}{\Gamma \Rightarrow \Delta; x : \Box A} \Box R \quad \frac{xRy, y : A, \Gamma \Rightarrow \Delta; \Pi}{x : \Diamond_i A, \Gamma \Rightarrow \Delta; \Pi} \Diamond_i L \\
 \\
 \frac{xRy, \Gamma \Rightarrow \Delta; y : A}{xRy, \Gamma \Rightarrow \Delta; x : \Diamond_i A} \Diamond_i R \quad \frac{xRy, y : A, \Gamma \Rightarrow \Delta; \cdot}{x : \Diamond_c A, \Gamma \Rightarrow \Delta; \cdot} \Diamond_c L \quad \frac{xRy, \Gamma \Rightarrow y : A, x : \Diamond_c A, \Delta; \cdot}{xRy, \Gamma \Rightarrow x : \Diamond_c A, \Delta; \cdot} \Diamond_c R
 \end{array}$$

## INITIAL, DECISION AND STRUCTURAL RULES

$$\begin{array}{c}
 \frac{}{\Gamma, x : A \Rightarrow \Delta; x : A} \text{init}_i \quad \frac{}{\Gamma, x : A \Rightarrow x : A, \Delta; \Pi} \text{init}_c \\
 \\
 \frac{\Gamma \Rightarrow x : P, \Delta; x : P}{\Gamma \Rightarrow x : P, \Delta; \cdot} D \quad \frac{\Gamma \Rightarrow x : N, \Delta; \cdot}{\Gamma \Rightarrow \Delta; x : N} \text{store} \quad \frac{\Gamma \Rightarrow \Delta; \cdot}{\Gamma \Rightarrow \Delta; x : A} W
 \end{array}$$

## CUT RULES

$$\frac{\Gamma \Rightarrow \Delta; x : P \quad x : P, \Gamma \Rightarrow \Delta; \Pi}{\Gamma \Rightarrow \Delta; \Pi} \text{cut}_P \quad \frac{\Gamma \Rightarrow \Delta, x : N; \Pi^* \quad x : N, \Gamma \Rightarrow \Delta; \Pi}{\Gamma \Rightarrow \Delta; \Pi} \text{cut}_N$$

■ **Figure 3** Ecumenical pure modal labelled system **labEK**. In rules  $\Box R, \Diamond_i L, \Diamond_c L$ , the eigenvariable  $y$  does not occur free in any formula of the conclusion;  $N$  is negative and  $P$  is positive;  $p$  is atomic;  $\Pi^*$  is either empty or some  $z : P \in \Delta$ .



The modal rules in nested systems then govern the transfer of (modal) formulae between the different sequents, and they are *local*, in the sense that it is sufficient to transfer only one formula at a time.

In [24] we presented the nested ecumenical modal system **nEK**. We will highlight next some of its rules. Starting with modalities, the nested rules for the intuitionistic diamond are

$$\frac{\Gamma\{A^\bullet\}}{\Gamma\{\diamond_i A^\bullet\}} \diamond_i^\bullet \quad \frac{\Lambda_1\{A^\circ, \Lambda_2\}}{\Lambda_1\{\diamond_i A^\circ, [\Lambda_2]\}} \diamond_i^\circ$$

where  $\Lambda$  represents a nested context containing only input formulae<sup>5</sup>. In the worlds-as-nestings interpretation [12], doing proof search in a system containing these rules actually corresponds to moving bottom-up on a Kripke structure: in rule  $\diamond_i^\bullet$ , assuming  $\diamond_i A$  in a certain nesting (corresponding to a world, say,  $x$ ) is equivalent to creating a new nesting (corresponding to a fresh world, say,  $y$  related to  $x$ ) and assuming  $A$  there (compare with rule  $\diamond_i L$  in Figure 3).

Polarities determine the mobility of formulae between contexts, via the decision and store rules.

$$\frac{\Gamma^*\{P^\nabla, P^\circ\}}{\Gamma^{\perp^\circ}\{P^\nabla\}} D \quad \frac{\Lambda\{N^\nabla, \perp^\circ\}}{\Lambda\{N^\circ\}} \text{store}$$

In a bottom-up reading, a positive formula is chosen to be “focused on” in the decision rule  $D$ , while a negative formula in the stoup can be stored in the classical context by using the rule  $\text{store}$ , just as described in Section 3.

Finally, the positive and negative nested versions of the cut rule are given by

$$\frac{\Gamma^*\{P^\circ\} \quad \Gamma\{P^\bullet\}}{\Gamma\{\emptyset\}} \text{cut}^\circ \quad \frac{\Gamma^P\{N^\nabla\} \quad \Gamma\{N^\bullet\}}{\Gamma\{\emptyset\}} \text{cut}^\nabla$$

where  $\Gamma^P$  denotes that the context contains either  $\perp^\circ$  or  $P^\circ$  for some  $P^\nabla \in \Gamma$ . In [24] showed that both cut rules are admissible in **nEK**. Moreover, **nEK** was shown to be sound and complete w.r.t. an ecumenical birelational model. Since the same result holds for the labelled system **labEK**, the two systems are equivalent. Finally, the *op.cit.* also brings a discussion about fragments, axioms and extensions of ecumenical modal logics.

## 5 What is next?

There are still many paths to be traversed on this journey. We finish this text by discussing some future ideas and presenting related work.

**Computational interpretation.** As mentioned at the end of Section 3, we have been exploring the computational counterpart of the implicational fragment of the ecumenical logic, extending the paradigm “proofs-as-programs” to ecumenical proofs. There are two main challenges on this enterprise: (i) finding an adequate deduction system in which the classical and intuitionistic logical behaviours can be faithfully captured in a term calculus; (ii) dealing with general ecumenical natural deduction rules. In [30] we tackled part (i) by proposing the ecumenical pure natural deduction system  $\mathbf{NE}_p$ , where the  $\lambda\mu$  internalization of *stoups* can

<sup>5</sup> Observe that rules are applied anywhere in the nesting structure, which is represented by *contexts with a hole* of the form  $\Gamma\{\}$ .

be easily adapted to the ecumenical case. Regarding, (ii), we are currently investigating the possibility of formulating an ecumenical version of the call-by-name lambda-calculus with generalized applications presented in [37] which integrates a notion of distant reduction that allows to unblock  $\beta$ -redexes without resorting to the permutative conversions of generalized applications.

**Automated theorem proving.** In [28] we developed an algorithmic-based approach for proving inductive properties of propositional sequent systems such as admissibility, invertibility, cut-elimination, and identity expansion. The proposed algorithms are based on rewrite and narrowing techniques. They have been fully mechanized in the **L-Framework**, thus offering both proof-search algorithms and semi-decision procedures for proving theorems and meta-theorems of several logical systems. We have started implementing the sequent-based systems mentioned in this text in the **L-Framework**, proving proof-theoretic properties to some of them. The next step is to specify nested sequent systems, which turns out to be a real challenge.

**Proof-theoretic semantics.** Together with logical ecumenism, *proof-theoretic semantics* [38, 39] is another approach to logic currently providing interesting contributions to the debate concerning philosophical grounds for the validity of classical and intuitionistic logics. While logical ecumenism proposes an unified framework in which two “rival” logics may peacefully coexist, proof-theoretic semantics aims not only to elucidate the meaning of a logical proof, but also to provide means for its use as a basic concept of semantic analysis. In [26] we showed how to coherently combine both approaches by providing not only a medium in which classical and intuitionistic *logics* may coexist, but also one in which classical and intuitionistic *notions of proof* may coexist. We did not, however, provided a proof-theoretic semantics for Prawitz’ original system, or any of the systems presented here – this is future work.

### Related work

Given that ecumenical systems refer, in a broad sense, to proof systems for combining logics, the related work on this subject is extensive and encompasses numerous other works. We will mention few which serve as reference to the present work.

Peter Krauss [17] and Gilles Dowek [10] explored the same ecumenical ideas as the ones shown in this text. Their main motivation was mathematical: to explore the possibility of hybrid readings of axioms and proofs in mathematical theories, *i.e.*, the occurrences of classical and intuitionistic operators in mathematical axioms and proofs, in order to propose a new and original method of constructivisation of classical mathematics. Krauss applied these ideas in basic algebraic number theory and Dowek considered the example of an ecumenical proof of a simple theorem in basic set theory.

Dowek’s original work has been further explored in [3] and [4]. In that works, a (type) theory in  $\lambda\Pi$ -calculus modulo theory is investigated, where proofs of several logical systems can be expressed.

Regarding proof systems, there is the seminal work of Girard in [15] and the more recent work of Liang and Miller [18]. Their work is based on *polarities* and *focusing*, using translations into linear logic.

A complete different approach comes from the school of combining logics [9, 20, 8], where Hilbert like systems are built from a combination of axiomatic systems.

Finally, we would like to cite Tesi and Negri's work on an ecumenical approach to infinitary logic [42], where a labelled sequent calculus combining classical and intuitionistic connectives is proposed.

---

## References

- 1 Jean-Marc Andreoli. Logic programming with focusing proofs in linear logic. *J. Log. Comput.*, 2(3):297–347, 1992. doi:10.1093/logcom/2.3.297.
- 2 Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2001. doi:10.1017/CB09781107050884.
- 3 Frédéric Blanqui, Gilles Dowek, Émilie Grienenberger, Gabriel Hondet, and François Thiré. Some axioms for mathematics. In Naoki Kobayashi, editor, *6th International Conference on Formal Structures for Computation and Deduction, FSCD 2021, July 17-24, 2021, Buenos Aires, Argentina (Virtual Conference)*, volume 195 of *LIPICs*, pages 20:1–20:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICs.FSCD.2021.20.
- 4 Frédéric Blanqui, Gilles Dowek, Émilie Grienenberger, Gabriel Hondet, and François Thiré. A modular construction of type theories. *Log. Methods Comput. Sci.*, 19(1), 2023. doi:10.46298/lmcs-19(1:12)2023.
- 5 Robert Brandom. *Articulating Reasons: An Introduction to Inferentialism*. Harvard University Press, 2000.
- 6 Kai Brünnler. Deep sequent systems for modal logic. *Arch. Math. Log.*, 48:551–577, 2009.
- 7 Robert A. Bull. Cut elimination for propositional dynamic logic without \*. *Zeitschr. f. math. Logik und Grundlagen d. Math.*, 38:85–100, 1992.
- 8 Carlos Caleiro and Jaime Ramos. Combining classical and intuitionistic implications. In Boris Konev and Frank Wolter, editors, *Frontiers of Combining Systems, 6th International Symposium, FroCoS 2007, Liverpool, UK, September 10-12, 2007, Proceedings*, volume 4720 of *Lecture Notes in Computer Science*, pages 118–132. Springer, 2007. doi:10.1007/978-3-540-74621-8\_8.
- 9 Luis Fariñas del Cerro and Andreas Herzig. Combining classical and intuitionistic logic, or: Intuitionistic implication as a conditional. In Franz Baader and Klaus U. Schulz, editors, *Frontiers of Combining Systems, First International Workshop FroCoS 1996, Munich, Germany, March 26-29, 1996, Proceedings*, volume 3 of *Applied Logic Series*, pages 93–102. Kluwer Academic Publishers, 1996.
- 10 Gilles Dowek. On the definition of the classical connectives and quantifiers. *Why is this a Proof?*, *Festschrift for Luiz Carlos Pereira*, 27:228–238, 2016.
- 11 Michael Dummett. *The Logical Basis of Metaphysics*. Cambridge, Mass.: Harvard University Press, 1991.
- 12 Melvin Fitting. Nested sequents for intuitionistic logics. *Notre Dame Journal of Formal Logic*, 55(1):41–61, 2014. doi:10.1215/00294527-2377869.
- 13 Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.
- 14 Jean-Yves Girard. A new constructive logic: Classical logic. *Math. Struct. Comput. Sci.*, 1(3):255–296, 1991. doi:10.1017/S0960129500001328.
- 15 Jean-Yves Girard. On the unity of logic. *Ann. Pure Appl. Logic*, 59(3):201–217, 1993. doi:10.1016/0168-0072(93)90093-S.
- 16 Ryo Kashima. Cut-free sequent calculi for some tense logics. *Studia Logica*, 53(1):119–136, 1994. doi:10.1007/BF01053026.
- 17 Peter Krauss. A constructive refinement of classical logic. Draft, 1992.
- 18 Chuck Liang and Dale Miller. A focused approach to combining logics. *Ann. Pure Appl. Logic*, 162(9):679–697, 2011. doi:10.1016/j.apal.2011.01.012.

- 19 Chuck C. Liang and Dale Miller. Focusing and polarization in intuitionistic logic. In Jacques Duparc and Thomas A. Henzinger, editors, *Computer Science Logic, 21st International Workshop, CSL 2007, 16th Annual Conference of the EACSL, Lausanne, Switzerland, September 11-15, 2007, Proceedings*, volume 4646 of *Lecture Notes in Computer Science*, pages 451–465. Springer, 2007. doi:10.1007/978-3-540-74915-8\_34.
- 20 Paqui Lucio. Structured sequent calculi for combining intuitionistic and classical first-order logic. In Hélène Kirchner and Christophe Ringeissen, editors, *Frontiers of Combining Systems, Third International Workshop, FroCoS 2000, Nancy, France, March 22-24, 2000, Proceedings*, volume 1794 of *Lecture Notes in Computer Science*, pages 88–104. Springer, 2000. doi:10.1007/10720084\_7.
- 21 Sonia Marin, Dale Miller, Elaine Pimentel, and Marco Volpe. From axioms to synthetic inference rules via focusing. *Ann. Pure Appl. Log.*, 173(5):103091, 2022. doi:10.1016/j.apal.2022.103091.
- 22 Sonia Marin, Luiz Carlos Pereira, Elaine Pimentel, and Emerson Sales. Ecumenical modal logic. In *DaLi 2020*, volume 12569 of *LNCS*, pages 187–204. Springer, 2020. doi:10.1007/978-3-030-65840-3\_12.
- 23 Sonia Marin, Luiz Carlos Pereira, Elaine Pimentel, and Emerson Sales. A pure view of ecumenical modalities. In Alexandra Silva, Renata Wassermann, and Ruy J. G. B. de Queiroz, editors, *Logic, Language, Information, and Computation - 27th International Workshop, WoLLIC 2021, Virtual Event, October 5-8, 2021, Proceedings*, volume 13038 of *Lecture Notes in Computer Science*, pages 388–407. Springer, 2021. doi:10.1007/978-3-030-88853-4\_24.
- 24 Sonia Marin, Luiz Carlos Pereira, Elaine Pimentel, and Emerson Sales. Separability and harmony in ecumenical systems. *CoRR*, abs/2204.02076, 2022. doi:10.48550/arXiv.2204.02076.
- 25 Julien Murzi. Classical harmony and separability. *Erkenntnis*, 85:391–415, 2020.
- 26 Victor Nascimento, Luiz Carlos Pereira, and Elaine Pimentel. An ecumenical view of proof-theoretic semantics, 2023. arXiv:2306.03656.
- 27 Sara Negri. Proof analysis in modal logic. *J. Philosophical Logic*, 34(5-6):507–544, 2005. doi:10.1007/s10992-005-2267-3.
- 28 Carlos Olarte, Elaine Pimentel, and Camilo Rocha. A rewriting logic approach to specification, proof-search, and meta-proofs in sequent systems. *CoRR*, abs/2101.03113, 2021. arXiv:2101.03113.
- 29 Michel Parigot.  $\lambda\mu$ -calculus: An algorithmic interpretation of classical natural deduction. In *LPAR: Logic Programming and Automated Reasoning, International Conference*, volume 624 of *LNCS*, pages 190–201. Springer, 1992.
- 30 Luiz Carlos Pereira and Elaine Pimentel. On an ecumenical natural deduction with stoup - part I: the propositional case. *CoRR*, abs/2204.02199, 2022. doi:10.48550/arXiv.2204.02199.
- 31 Luiz Carlos Pereira and Ricardo Oscar Rodriguez. Normalization, soundness and completeness for the propositional fragment of Prawitz’ ecumenical system. *Revista Portuguesa de Filosofia*, 73(3-3):1153–1168, 2017.
- 32 Elaine Pimentel, Luiz Carlos Pereira, and Valeria de Paiva. An ecumenical notion of entailment. *Synthese*, 198(22-S):5391–5413, 2021. doi:10.1007/s11229-019-02226-5.
- 33 Francesca Poggiolesi. The method of tree-hypersequents for modal propositional logic. In *Towards Mathematical Philosophy*, volume 28 of *Trends In Logic*, pages 31–51. Springer, 2009.
- 34 Dag Prawitz. *Natural Deduction, volume 3 of Stockholm Studies in Philosophy*. Almqvist and Wiksell, 1965.
- 35 Dag Prawitz. Classical versus intuitionistic logic. In Bruno Lopes Edward Hermann Haeusler, Wagner de Campos Sanz, editor, *Why is this a Proof?, Festschrift for Luiz Carlos Pereira*, volume 27, pages 15–32. College Publications, 2015.
- 36 Greg Restall. Speech acts & the quest for a natural account of classical proof. Available at <https://consequently.org/writing/>, 2021.

- 37 José Espírito Santo, Delia Kesner, and Loïc Peyrot. A faithful and quantitative notion of distant reduction for generalized applications. In Patricia Bouyer and Lutz Schröder, editors, *Foundations of Software Science and Computation Structures - 25th International Conference, FOSSACS 2022, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2022, Munich, Germany, April 2-7, 2022, Proceedings*, volume 13242 of *Lecture Notes in Computer Science*, pages 285–304. Springer, 2022. doi: 10.1007/978-3-030-99253-8\_15.
- 38 Peter Schroeder-Heister. Uniform proof-theoretic semantics for logical constants (abstract). *Journal of Symbolic Logic*, 56:1142, 1991.
- 39 Peter Schroeder-Heister. Validity concepts in proof-theoretic semantics. *Synthese*, 148(3):525–571, 2006.
- 40 Alex K. Simpson. *The Proof Theory and Semantics of Intuitionistic Modal Logic*. PhD thesis, College of Science and Engineering, School of Informatics, University of Edinburgh, 1994.
- 41 Lutz Straßburger. Cut elimination in nested sequents for intuitionistic modal logics. In *Proceedings of FOSSACS 2013*, pages 209–224, 2013. doi:10.1007/978-3-642-37075-5\_14.
- 42 Matteo Tesi and Sara Negri. Neighbourhood semantics and labelled calculus for intuitionistic infinitary logic. *J. Log. Comput.*, 31(7):1608–1639, 2021. doi:10.1093/logcom/exab040.
- 43 Luca Viganò. *Labelled Non-Classical Logics*. Kluwer Academic Publishers, 2000.



# The Metatheory of Gradual Typing: State of the Art and Challenges

Jeremy G. Siek   

Computing Laboratory, Indiana University, Bloomington, IN, USA

---

## Abstract

Gradually typed languages offer both static and dynamic checking of program invariants, from simple properties such as type safety, to more advanced ones such as information flow control (security), relational parametricity (theorems for free), and program correctness. To ensure that gradually typed languages behave as expected, researchers prove theorems about their language designs. For example, the Gradual Guarantee Theorem states that a programmer can migrate their program to become more or less statically checked and the resulting program will behave the same (modulo errors). As another example, the Noninterference Theorem (for information flow control) states that high security inputs do not affect the low security outputs of a program. These theorems are often proved using simulation arguments or via syntactic logical relations and modal logics. Sometimes the proofs are mechanized in a proof assistant, but often they are simply written in LaTeX. However, as researchers consider gradual languages of growing complexity, the time to conduct such proofs, and/or the likelihood of errors in the proofs, also grows. As a result there is a need for improved proof techniques and libraries of mechanized results that would help to streamline the development of the metatheory of gradually typed languages.

**2012 ACM Subject Classification** Software and its engineering → Semantics; Theory of computation → Operational semantics

**Keywords and phrases** gradual typing, type safety, gradual guarantee, noninterference, simulation, logical relation, mechanized metatheory

**Digital Object Identifier** 10.4230/LIPIcs.CALCO.2023.4

**Category** Invited Talk

**Funding** Parts of this work were funded by NSF award 1763922.



© Jeremy G. Siek;

licensed under Creative Commons License CC-BY 4.0

10th Conference on Algebra and Coalgebra in Computer Science (CALCO 2023).

Editors: Paolo Baldan and Valeria de Paiva; Article No. 4; pp. 4:1–4:1

Leibniz International Proceedings in Informatics






LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany





# Machine-Checked Computational Mathematics

Assia Mahboubi   

Nantes Université, École Centrale Nantes, CNRS, INRIA, LS2N, UMR 6004, F-44000 Nantes, France

Vrije Universiteit Amsterdam, The Netherlands

---

## Abstract

This talk shall discuss the potential impact of formal methods, and in particular, of interactive theorem proving, on computational mathematics.

Geared with increasingly fast computer algebra libraries and scientific computing software, computers have become amazing instruments for mathematical guesswork. In fact, computer calculations are even sometimes used to substantiate actual reasoning steps in proofs, later published in major venues of the mathematical literature. Yet surprisingly, little of the now standard techniques available today for verifying critical software (e.g., cryptographic components, airborne commands, etc.) have been applied to the programs used to produce mathematics. In this talk, we propose to discuss this state of affairs.

**2012 ACM Subject Classification** Computing methodologies → Theorem proving algorithms; Theory of computation → Type theory

**Keywords and phrases** Type theory, computer algebra, interactive theorem proving

**Digital Object Identifier** 10.4230/LIPIcs.CALCO.2023.5

**Category** Invited Talk

**Funding** This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 101001995).



© Assia Mahboubi;

licensed under Creative Commons License CC-BY 4.0

10th Conference on Algebra and Coalgebra in Computer Science (CALCO 2023).

Editors: Paolo Baldan and Valeria de Paiva; Article No. 5; pp. 5:1–5:1

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



# Forward and Backward Steps in a Fibration

Ruben Turkenburg  



Radboud University, Nijmegen, The Netherlands

Harsh Beohar  

University of Sheffield, UK

Clemens Kupke  

Strathclyde University, Glasgow, UK

Jurriaan Rot  

Radboud University, Nijmegen, The Netherlands

---

## Abstract

Distributive laws of various kinds occur widely in the theory of coalgebra, for instance to model automata constructions and trace semantics, and to interpret coalgebraic modal logic. We study steps, which are a general type of distributive law, that allow one to map coalgebras along an adjunction. In this paper, we address the question of what such mappings do to well known notions of equivalence, e.g., bisimilarity, behavioural equivalence, and logical equivalence.

We do this using the characterisation of such notions of equivalence as (co)inductive predicates in a fibration. Our main contribution is the identification of conditions on the interaction between the steps and liftings, which guarantees preservation of fixed points by the mapping of coalgebras along the adjunction. We apply these conditions in the context of lax liftings proposed by Bonchi, Silva, Sokolova (2021), and generalise their result on preservation of bisimilarity in the construction of a belief state transformer. Further, we relate our results to properties of coalgebraic modal logics including expressivity and completeness.

**2012 ACM Subject Classification** Theory of computation → Categorical semantics

**Keywords and phrases** Coalgebra, Fibration, Bisimilarity

**Digital Object Identifier** 10.4230/LIPIcs.CALCO.2023.6

**Funding** *Ruben Turkenburg, Jurriaan Rot:* Partially supported by the NWO grant OCENW.M20.053.

*Harsh Beohar:* Partially supported by the EPSRC grant EP/X019373/1.

*Clemens Kupke:* Partially supported by Leverhulme Trust Research Project Grant RPG-2020-232.

## 1 Introduction

The theory of coalgebras provides a general perspective on state-based systems, parametric in an endofunctor which models the type of system [19]. Accordingly, many interesting constructions on state-based systems arise as functors between categories of coalgebras.

These functors between categories of coalgebras often arise as liftings of left or right adjoints between the underlying base categories. Such liftings are central to, for instance, coalgebraic approaches to trace semantics and determinisation [13, 20, 33, 6, 22, 32] as well as testing semantics and algebraic semantics of modal logics [27, 31, 5, 24, 8, 23].

A central question is how these constructions on coalgebras affect behavioural equivalence. For instance, determinisation of automata turns a coalgebra on, e.g., the category *Set* of sets and functions, into a coalgebra on the category of Eilenberg-Moore algebras for a monad, so that the canonical notion of behavioural equivalence changes from bisimilarity to language semantics. Subsequently, the algebraic structure may be forgotten again, turning the determinised coalgebra back into a *Set* coalgebra, and this simple operation does not



© Ruben Turkenburg, Harsh Beohar, Clemens Kupke, and Jurriaan Rot;  
licensed under Creative Commons License CC-BY 4.0

10th Conference on Algebra and Coalgebra in Computer Science (CALCO 2023).

Editors: Paolo Baldan and Valeria de Paiva; Article No. 6; pp. 6:1–6:18

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

affect behavioural equivalence. But when determinisation is not given by a distributive law, such as in the construction of belief-state transformers in [4], proving that this “forgetting” preserves and reflects behavioural equivalence can be non-trivial (see *op. cit.*, [35]).

A different type of example of a construction given by a lifting of an adjoint is the algebraic semantics of modal logic, where the semantics yields a transformation that takes a coalgebra (e.g., a Kripke model) and turns it into an algebra (here viewed as a coalgebra on an opposite category, for uniformity of the presentation). As we show later, one form of preservation of behavioural equivalence amounts to adequacy and expressivity of the logic.

We propose an abstract framework to analyse whether a coalgebra lifting of an adjoint preserves behavioural equivalence. The basic infrastructure is as follows.

- We use functor liftings in fibrations, which is a standard approach to define coalgebraic bisimilarity [17] and other (co)inductive predicates [14]. This approach to define coinductive predicates beyond bisimilarity has recently been used, for instance, in general expressivity proofs of modal logics [29, 26], closely connected to the current paper.
- We use the notion of a *step* to lift left and right adjoints to categories of coalgebras. Steps are a variant of distributive laws (also known as morphisms of endofunctors) over a left or right adjoint, named as such in [32] but widely used before. They are relevant in all of the above-mentioned examples on language semantics, determinisation and modal logic.

This paper connects steps and fibrations, to speak generally about preservation of coinductive (and inductive) predicates by coalgebra constructions. The key technical idea is to use a variant of fibred adjunctions [21]. We start with an adjunction and a step, and assume a fibration and functor lifting on both sides of the adjunction to formulate the coinductive predicates that we wish to relate. We then lift this adjunction to the total categories of the fibrations involved [16]. This setting allows us to formulate sufficient conditions for preservation of coinductive predicates by coalgebra constructions induced by steps.

There are two main variants of this abstract story: one that starts from a step that lifts the *left* adjoint to coalgebras, and one for lifting the *right* adjoint. The first allows us, for instance, to speak about adequacy and expressivity of modal logics, without referring to initial algebras. This connects to recent work that uses Galois connections [1], and in fact we recover those Galois connections from our adjunctions between fibrations. We also study an example that has not occurred in previous abstract frameworks for expressivity: proving expressivity of a logic by relating it to *apartness* instead of bisimilarity [12, 11].

The second variant – constructions arising as liftings of right adjoints – includes preservation of bisimilarity on belief-state transformers [4]. More generally, it follows from our results that any right adjoint in a *lax lifting* situation preserves and reflects bisimilarity (assuming split monos instead of injectivity), generalising the result for belief-state transformers. By using opposite categories we also get a very different example in this context, which connects preservation of coinductive predicates to completeness of coalgebraic modal logic.

## 2 Preservation of coinductive predicates in lattices

Before moving to the general theory of fibrations and steps, we start with introductory examples on preservation of (co)inductive predicates in the context of lattices, forming a special (and well-known) case of steps on Galois connections. In subsequent sections, we will use the structure of steps, being certain natural transformations allowing us to transform coalgebras along an adjunction. A similar structure is already known in order theory, where we may consider inequalities between compositions of monotone maps, as in:

$$b \curvearrowright \Delta \begin{array}{c} \xrightarrow{f} \\ \perp \\ \xleftarrow{g} \end{array} \Gamma \curvearrowright l \quad (1)$$

with  $\Delta$  and  $\Gamma$  lattices, and  $f, g, b, l$  monotone maps. We call the inequality  $bg \leq gl$  a *forward* step, and  $gl \leq bg$  a *backward* step. Central to our study of steps is the following standard result found in, e.g., [9], relating them to preservation of greatest fixed points (which we denote here using the operator  $\nu$ ).

► **Lemma 1.** *Given the setting of (1):*

1. *If  $gl \leq bg$ , then  $g(\nu l) \leq \nu b$ ;*
2. *If  $bg \leq gl$ , then  $g(\nu l) \geq \nu b$ .*

Now, if  $bg = gl$  in the setting of (1), the inequalities combine to give  $g(\nu l) = \nu b$ . This has been shown more generally in the context of coalgebras in [17], where the equality  $bg = gl$  is instead a natural isomorphism  $BG \xrightarrow{\sim} GL$ , with  $F, G, B, L$  functors. It is shown in *op. cit.* that this allows the lifting of the adjunction  $F \dashv G$  to coalgebras (generalising post-fixed points) so that the right adjoint preserves the final coalgebra (generalising the greatest fixed point) as right adjoints preserve limits.

## 2.1 Example: Closed and Convex Relations

We will first consider two instances where the lattices consist of relations on sets on one side, and relations with either topological or convex structure on the other, i.e.:

$$b \curvearrowright \text{Rel}_X \begin{array}{c} \xrightarrow{c} \\ \perp \\ \xleftarrow{u} \end{array} \text{CRel}_{\mathbb{X}} \curvearrowleft v \quad b \curvearrowright \text{Rel}_A \begin{array}{c} \xrightarrow{h} \\ \perp \\ \xleftarrow{u} \end{array} \text{ConRel}_{\mathbb{A}} \curvearrowleft d \quad (2)$$

where  $\text{CRel}_{\mathbb{X}}$  consists of closed relations on a compact Hausdorff space  $\mathbb{X}$  and  $\text{ConRel}_{\mathbb{A}}$  consists of convex relations on a convex algebra  $\mathbb{A}$ . The monotone maps  $v, d$  will be such that the post-fixed points are bisimulations and the greatest fixed points are bisimilarity on systems with topological ( $\mathbb{X}$ ) or convex ( $\mathbb{A}$ ) structure, and the maps  $b$  characterise bisimulations/bisimilarity on systems where this structure has been forgotten ( $X$  and  $A$  respectively). Lemma 1, thus, tells us how bisimilarity on each side can be related via the right adjoint.

These settings arise in examples of ultrafilter extensions for coalgebras and the transformation of probabilistic automata into belief-state transformers. In the first instance, the closed relations can in fact be restricted to Stone topological spaces (those compact Hausdorff spaces which are zero-dimensional), where we consider coalgebras for the Vietoris functor in **Stone**. It is shown in [2] that these coalgebras correspond to descriptive frames, which arise in the first stage of the construction of the ultrafilter extension of a Kripke frame given in [28]. The second stage given there is to transport these back to a coalgebra in **Set**. The construction of a belief-state transformer from a probabilistic automaton (PA) has a similar structure, where the second stage is to transport a system with extra algebraic structure back to a **Set** coalgebra. In each case, it is important that behavioural equivalence is preserved and reflected in the second stage, shown in [2, 4] respectively for the above examples. These results are recovered already in [35]. However, the approach taken there does not immediately apply to the examples of adequacy and expressivity of modal logics, so we prefer the conditions given in the current paper for their generality.

## 2.2 Example: Expressivity

Another example relates to work on expressivity of coalgebraic logics [29, 1, 24, 23], where we wish to relate bisimilarity and logical equivalence (or indistinguishability). The lattices involved are equivalence relations on the carrier  $X$  of a coalgebra and predicates on  $2^X$ .

$$b \curvearrowright \text{ERel}_X \begin{array}{c} \xrightarrow{f} \\ \perp \\ \xleftarrow{g} \end{array} \text{Pred}_{2^X}^{\text{op}} \curvearrowleft l \quad (3)$$

## 6:4 Forward and Backward Steps in a Fibration

This gives us the setting shown in (3), where we define  $g(\Gamma \subseteq 2^X) = \{(x, y) \mid \forall S \in \Gamma. x \in S \iff y \in S\}$ . In words, we relate those elements which are in exactly the same sets of  $\Gamma$ . Next, the action of  $f$  on an equivalence relation  $R$  is to give those subsets which are closed under  $R$ , i.e., they are a union of equivalence classes of  $R$ . More formally, we have  $f(R) = \{\Gamma \subseteq X \mid \forall (x, y) \in R. x \in \Gamma \iff y \in \Gamma\}$ .

The monotone map  $b$  is taken to be such that the greatest fixed point is bisimilarity. The map  $l$  is dual to the map whose *least* fixed point we can think of as those predicates obtainable as the interpretation of a formula of a modal logic. In essence, these *are* the formulas which we generate from some propositional constants and applications of the operators of our logic.

Applying  $g$  to these “reachable” predicates gives an equivalence relating states which satisfy exactly the same formulas. This is exactly logical equivalence, and the above picture then allows us to relate this to bisimilarity. Namely, if  $g(\nu l) \geq \nu b$ , then bisimilarity implies logical equivalence, which is precisely adequacy of the logic. If, conversely,  $g(\nu l) \leq \nu b$ , then logical equivalence implies bisimilarity, called expressivity of the logic.

Now, Lemma 1 gives us a way to obtain these inclusions via inequalities capturing the interaction between the logic and behaviour in a rather general way. Later, we will show in more detail how these conditions relate to existing approaches to the semantics of coalgebraic modal logic and the properties of adequacy and expressivity.

### 3 Fibrations and Bisimulations

We give the basic definitions related to fibrations (for details see [21]).

Given a functor  $p: \mathcal{E} \rightarrow \mathcal{C}$ , a morphism  $b: R \rightarrow S$  in  $\mathcal{E}$  is ( $p$ -)Cartesian over  $f: X \rightarrow Y$  in  $\mathcal{C}$ , if  $pb = f$  and for every  $c: T \rightarrow S$  s.t.  $pc = f \circ g$  for some  $g: pT \rightarrow X$ , there is a unique  $d: T \rightarrow R$  with  $c = b \circ d$ . A functor  $p: \mathcal{E} \rightarrow \mathcal{C}$  is now a (Grothendieck) fibration if for all objects  $S \in \mathcal{E}$  and arrows  $f: X \rightarrow pS$ , there is a Cartesian arrow  $b: R \rightarrow S$  in  $\mathcal{E}$  with  $pb = f$  (and thus also  $pR = X$ ). We say that  $R$  is *above*  $pR$  and  $b: R \rightarrow S$  is *above*  $pb: pR \rightarrow pS$ . We will also call  $\mathcal{C}$  the base category and  $\mathcal{E}$  the total category of the fibration.

For an object  $X \in \mathcal{C}$ , the *fibre above*  $X$  is the category  $\mathcal{E}_X$  whose objects are those objects in  $\mathcal{E}$  above  $X$ , and arrows are above the identity on  $X$ . A choice of Cartesian lifting for every  $f: X \rightarrow Y$  in  $\mathcal{C}$  is called a *cleavage*, and any cleavage defines, for each such  $f$ , a *reindexing* functor  $f^*: \mathcal{E}_Y \rightarrow \mathcal{E}_X$  defined on objects exactly by the choice of Cartesian arrow  $\bar{f}(S): f^*(S) \rightarrow S$ . We assume below that reindexing functors have left adjoints  $\coprod_f \dashv f^*$  (called *direct-image*). This is equivalent to the condition that both  $p: \mathcal{E} \rightarrow \mathcal{C}$  and  $p^{\text{op}}: \mathcal{E}^{\text{op}} \rightarrow \mathcal{C}^{\text{op}}$  are fibrations, in which case,  $p$  is also called a *bifibration*.

Given fibrations  $p: \mathcal{E} \rightarrow \mathcal{C}$  and  $q: \mathcal{F} \rightarrow \mathcal{D}$ , a *morphism of fibrations* from  $p$  to  $q$  is a pair of functors  $(\bar{F}: \mathcal{E} \rightarrow \mathcal{F}, F: \mathcal{C} \rightarrow \mathcal{D})$  such that  $q \circ \bar{F} = F \circ p$ . In that case, for every object we have a restriction  $\bar{F}_X: \mathcal{E}_X \rightarrow \mathcal{E}_{FX}$ , denoted by  $\bar{F}$  if the type is clear from the context. We will also call  $\bar{F}$  a *lifting* of  $F$ . If  $\bar{F}$  preserves Cartesian morphisms, it is called *fibred*. This is equivalent to having the equation  $\bar{F}_X \circ f^* = (Ff)^* \circ \bar{F}_Y$  for all morphisms  $f: X \rightarrow Y$ .

We will work with fibrations with the additional assumption that the fibres form complete lattices and reindexing preserves meets, i.e., the fibrations have *fibred* meets:

► **Assumption 2.** *We assume that for any fibration  $p: \mathcal{E} \rightarrow \mathcal{C}$ , the fibres  $\mathcal{E}_X$  form complete lattices and reindexing preserves meets. We will also call such a fibration a  $\text{CLat}_\wedge$ -fibration.*

This ensures that the fibrations have many desirable properties, while being general enough for our purposes of defining coinductive predicates. In particular, such fibrations are always bifibrations. For a more detailed treatment of  $\text{CLat}_\wedge$ -fibrations see, e.g., [25, 34].

### 3.1 Subobject and Relation fibrations

Take a category  $\mathcal{C}$  which is complete and well-powered (subobjects of a given object form a set). Then, the category  $\text{Pred}(\mathcal{C})$  is defined as follows: objects are subobjects  $f: S \twoheadrightarrow X$ , i.e., equivalence classes of monos; and morphisms are maps  $u: X \rightarrow Y$  in  $\mathcal{C}$  such that there is a (necessarily unique) arrow making the diagram on the left in (4) commute. Then the functor  $p: \text{Pred}(\mathcal{C}) \rightarrow \mathcal{C}$  sending a subobject  $f: S \twoheadrightarrow X$  to  $X$ , is a fibration, with reindexing given by pullbacks, referred to as the *predicate* fibration. Since the base category is complete and well-powered, the fibres  $\text{Pred}(\mathcal{C})_X$  are complete lattices [7, Cor. 4.2.5]. Since reindexing is defined by pullback, it preserves meets, so that  $p$  is a  $\text{CLat}_\wedge$ -fibration.

$$\begin{array}{ccccc}
 S & \dashrightarrow & T & & \\
 \downarrow & & \downarrow & & \\
 X & \xrightarrow{u} & Y & & \\
 & & & & \\
 \text{Rel}(\mathcal{C}) & \longrightarrow & \text{Rel}'(\mathcal{C}) & \longrightarrow & \text{Pred}(\mathcal{C}) \\
 \downarrow & \lrcorner & \downarrow & \lrcorner & \downarrow \\
 \mathcal{C} & \xrightarrow{\Delta} & \mathcal{C} \times \mathcal{C} & \xrightarrow{\times} & \mathcal{C}
 \end{array} \tag{4}$$

As  $\mathcal{C}$  is complete, it furthermore has products, so we can form the relation fibration  $\text{Rel}(\mathcal{C})$  via the pullbacks as on the right in (4). The fibration  $\text{Rel}'(\mathcal{C})$  consists of relations on *all* pairs of objects  $(X, Y) \in \mathcal{C} \times \mathcal{C}$ , whereas we will use the fibration  $\text{Rel}(\mathcal{C})$  consisting of relations for which  $X = Y$ . By this definition, we obtain that an object of  $\text{Rel}(\mathcal{C})$  is a subobject  $R \twoheadrightarrow X \times X$  of the product of  $X$  with itself. The functor part of the fibration sends a relation  $R \twoheadrightarrow X \times X$  to  $X$  and a morphism to the underlying arrow  $u: X \rightarrow Y$ , analogously to (4). By the same argument as for  $\text{Pred}(\mathcal{C})$ ,  $\text{Rel}(\mathcal{C})$  is a  $\text{CLat}_\wedge$ -fibration.

► **Example 3.** In  $\text{Set}$ , subobjects are just subsets  $U \subseteq X$ , and reindexing is inverse image, i.e.,  $f^*(V \subseteq Y) = \{x \in X \mid f(x) \in V\}$ . Similarly, relations are subsets  $R \subseteq X \times X$ , with  $f^*(S \subseteq Y \times Y) = \{(x_1, x_2) \in X \times X \mid (f(x_1), f(x_2)) \in S\}$ . Notice  $\text{Set}$  is complete, and the collection of subsets of a set is its powerset which is again a set. Each powerset is thus a complete lattice with join and meet given by union and intersection, respectively.

For a monad  $T: \text{Set} \rightarrow \text{Set}$ , let  $\text{EM}(T)$  be the category of Eilenberg-Moore algebras. Then  $\text{Pred}(\text{EM}(T))$  consists of subalgebras, and  $\text{Rel}(\text{EM}(T))$  consists of congruences, i.e., relations that are closed under the algebra structure (not necessarily equivalence relations).

We can restrict  $\text{Rel}(\mathcal{C})$  to the category  $\text{ERel}(\mathcal{C})$  of equivalence relations, defined internally (e.g., [21]), and define reindexing and meets for equivalence relations in the same way as for relations since these are defined as pullbacks. This turns  $\text{ERel}(\mathcal{C}) \rightarrow \mathcal{C}$  into a  $\text{CLat}_\wedge$ -fibration.

### 3.2 Predicate and Relation liftings

Here we recall a method for lifting functors to predicates and relations based on factorisation systems. For a factorisation system  $(\mathcal{E}, \mathcal{M})$ , we refer to elements of  $\mathcal{E}$  as *abstract epis* and write them as  $\cdot \twoheadrightarrow \cdot$ , and maps in  $\mathcal{M}$  *abstract monos* written as  $\cdot \dashv \rightarrow \cdot$ . As an example, for  $\text{Set}$  we can take  $\mathcal{E}$  to be the class of all surjective functions, and  $\mathcal{M}$  to be the class of all injective functions. The factorisation of a function  $f: X \rightarrow Y$  is the image factorisation, where  $e: X \twoheadrightarrow \text{Im}(f)$  acts as  $f$  and  $m: \text{Im}(f) \dashv \rightarrow Y$  embeds the image of  $f$  into the original codomain  $Y$ . Another important example is that of *regular categories*, where maps factorise as a *regular epi* (i.e., an epi which is the coequaliser of some parallel pair of morphisms) followed by a mono. In fact, the existence of such factorisations is part of the defining property of a regular category.

## 6:6 Forward and Backward Steps in a Fibration

Assuming a category  $\mathcal{D}$  with a factorisation system  $(\mathcal{E}, \mathcal{M})$  such that all maps in  $\mathcal{M}$  are monos, we can define the (canonical) predicate and relation liftings  $\text{Pred}(F)$  and  $\text{Rel}(F)$  of a functor  $F: \mathcal{C} \rightarrow \mathcal{D}$  via the following factorisations, where  $p: P \rightarrow X$  is a predicate and  $r: R \rightarrow X \times X$  a relation:

$$\begin{array}{ccc}
 FP & \xrightarrow{Fp} & FX \\
 \downarrow e & & \uparrow m \\
 & \text{Pred}(F)(P) & 
 \end{array}
 \quad
 \begin{array}{ccc}
 FR & \xrightarrow{Fr} & F(X \times X) \xrightarrow{\langle F\pi_1, F\pi_2 \rangle} FX \times FX \\
 \downarrow e & & \uparrow m \\
 & \text{Rel}(F)(R) & 
 \end{array}
 \quad (5)$$

By the assumption that all maps in  $\mathcal{M}$  are monos, the above constructions define functors  $\text{Pred}(F): \text{Pred}(\mathcal{C}) \rightarrow \text{Pred}(\mathcal{D})$  and  $\text{Rel}(F): \text{Rel}(\mathcal{C}) \rightarrow \text{Rel}(\mathcal{D})$  respectively, with the actions on arrows defined by orthogonality.

### 3.3 Invariants and Coinductive Predicates

We will now recall the notion of coinductive invariants and predicates, defined as post and greatest fixed points of certain monotone maps respectively (see also [19, 14]). Assumption 2 ensures that the monotone maps always have such fixpoints.

Assuming a fibration  $p: \mathcal{E} \rightarrow \mathcal{C}$ , a coalgebra  $f: X \rightarrow BX$  with  $X$  in  $\mathcal{C}$ , and a lifting  $\overline{B}: \mathcal{E} \rightarrow \mathcal{E}$  of  $B: \mathcal{C} \rightarrow \mathcal{C}$ , we can define a monotone map  $f^* \circ \overline{B}_X: \mathcal{E}_X \rightarrow \mathcal{E}_X$  using reindexing. Instantiating this to the category  $\text{Set}$ , and the fibration with  $\text{Rel}(\text{Set})$  as total category, we can consider the canonical relation lifting (5) of  $B$ , given explicitly by  $\text{Rel}(B)(R) = \{(y_1, y_2) \in BX \times BX \mid \exists z \in BR. B\pi_1(z) = y_1 \wedge B\pi_2(z) = y_2\}$ . As mentioned earlier, reindexing for the relation fibration is given by pullbacks, which amounts to taking the inverse image, so that:

$$f^* \circ \text{Rel}(B)(R) = \{(x_1, x_2) \in X \times X \mid \exists z \in BR. B\pi_1(z) = f(x_1) \wedge B\pi_2(z) = f(x_2)\} \quad (6)$$

Taking a post-fixed point  $R \leq f^* \circ \text{Rel}(B)(R)$  of such a monotone map (also called an *invariant*), we recover the usual notion of *coalgebraic bisimulation*. The greatest fixed point  $\nu(f^* \circ \text{Rel}(B)(-))$  is then bisimilarity.

More generally, for a lifting  $\overline{B}$  of  $B$ , we call such a greatest fixed point the *coinductive predicate* defined by  $\overline{B}$  on  $f$ . This covers many more examples than bisimilarity. For a simple instance, take  $B$  to be the powerset functor  $\mathcal{P}: \text{Set} \rightarrow \text{Set}$ , and  $\overline{\mathcal{P}}: \text{Pred} \rightarrow \text{Pred}$  with  $\overline{\mathcal{P}}(P \subseteq X) = \{S \subseteq X \mid P \cap S \neq \emptyset\}$ . A  $\mathcal{P}$ -coalgebra is a transition system, and the coinductive predicate on it defined by  $\overline{\mathcal{P}}$  is the set of all states which have an infinite path. For other examples of coinductive predicates defined in this way see, e.g., [14, 3, 34, 18, 25].

## 4 Lifting adjunctions in a fibration

Let  $F \dashv G: \mathcal{D} \rightarrow \mathcal{C}$  be an adjunction, and assume fibrations  $p: \mathcal{E} \rightarrow \mathcal{C}$  and  $q: \mathcal{F} \rightarrow \mathcal{D}$ . Further, assume an adjunction  $\overline{F} \dashv \overline{G}: \mathcal{F} \rightarrow \mathcal{E}$ , as in (7). If we have  $q \circ \overline{F} = F \circ p$ ,  $p \circ \overline{G} = G \circ q$ , and the unit and counit of the adjunction  $\overline{F} \dashv \overline{G}$  are above the unit and counit of the adjunction  $F \dashv G$  respectively, then we say that  $\overline{F} \dashv \overline{G}$  is a *lifting of the adjunction*  $F \dashv G$  (alternatively, this is an adjunction in  $\text{Cat}^{\rightarrow}$ , the 2-category of functors and commuting squares [16]).

$$\begin{array}{ccc}
 \mathcal{E} & \xrightarrow{\overline{F}} & \mathcal{F} \\
 \downarrow p & \begin{array}{c} \dashv \\ \perp \\ \dashv \end{array} & \downarrow q \\
 \mathcal{C} & \xrightarrow{F} & \mathcal{D} \\
 & \begin{array}{c} \dashv \\ \perp \\ \dashv \end{array} & \\
 & G & 
 \end{array}
 \quad (7)$$



This definition differs from the usual notion of a *fibred adjunction*, as we do not assume fibredness of either adjoint. However, it has been shown in [36, Lemma 4.5] for fibrations over a single base category, and later generalised in [16, Lemma 3.3.3] to fibrations over arbitrary base categories, that the right adjoint in a lifting of an adjunction is in fact always fibred. Dually, we have that the left adjoint is co-fibred (i.e., it preserves op-Cartesian maps).

A family of instances is given by predicate and relation liftings.

► **Lemma 4.** *Let  $\mathcal{C}, \mathcal{D}$  be complete and well-powered categories with factorisation systems  $(\mathcal{E}_1, \mathcal{M}_1), (\mathcal{E}_2, \mathcal{M}_2)$  with  $\mathcal{M}_1$  and  $\mathcal{M}_2$  consisting of monos. Given an adjunction  $F \dashv G: \mathcal{D} \rightarrow \mathcal{C}$  such that  $F(\mathcal{E}_1) \subseteq \mathcal{E}_2$ , the predicate and relation liftings form liftings of the adjunction:*

$$\begin{array}{ccc}
 \begin{array}{ccc}
 \text{Pred}(\mathcal{C}) & \begin{array}{c} \xrightarrow{\text{Pred}(F)} \\ \perp \\ \xleftarrow{\text{Pred}(G)} \end{array} & \text{Pred}(\mathcal{D}) \\
 \downarrow & & \downarrow \\
 \mathcal{C} & \begin{array}{c} \xrightarrow{F} \\ \perp \\ \xleftarrow{G} \end{array} & \mathcal{D}
 \end{array} & & \begin{array}{ccc}
 \text{Rel}(\mathcal{C}) & \begin{array}{c} \xrightarrow{\text{Rel}(F)} \\ \perp \\ \xleftarrow{\text{Rel}(G)} \end{array} & \text{Rel}(\mathcal{D}) \\
 \downarrow & & \downarrow \\
 \mathcal{C} & \begin{array}{c} \xrightarrow{F} \\ \perp \\ \xleftarrow{G} \end{array} & \mathcal{D}
 \end{array} \\
 & & (8)
 \end{array}$$

From these liftings, especially the case for predicates, we can lift also to quotients and equivalence relations given some extra conditions. For a category  $\mathcal{C}$ , we denote by  $\text{Quot}(\mathcal{C})$  the category of co-subobjects of objects of  $\mathcal{C}$ , that is, equivalence classes of epimorphisms. This is exactly the category  $\text{Pred}(\mathcal{C}^{\text{op}})^{\text{op}}$ , so that also  $\text{Pred}(\mathcal{C}^{\text{op}}) \simeq \text{Quot}(\mathcal{C})^{\text{op}}$  and  $\text{Quot}(\mathcal{C}^{\text{op}}) \simeq \text{Pred}(\mathcal{C})^{\text{op}}$ . Further, we can define quotient lifting dually to predicate lifting. The following is then the dual of the above result.

► **Corollary 5.** *Let  $\mathcal{C}, \mathcal{D}$  be co-complete and co-well-powered categories with factorisation systems  $(\mathcal{E}_1, \mathcal{M}_1), (\mathcal{E}_2, \mathcal{M}_2)$  with  $\mathcal{E}_1, \mathcal{E}_2$  consisting of epis. Then, given an adjunction  $F \dashv G: \mathcal{D} \rightarrow \mathcal{C}$ , s.t.  $G(\mathcal{M}_1) \subseteq \mathcal{M}_2$ , the quotient liftings form a lifting of the adjunction:*

$$\begin{array}{ccc}
 \begin{array}{ccc}
 \text{Quot}(\mathcal{C}) & \begin{array}{c} \xrightarrow{\text{Quot}(F)} \\ \perp \\ \xleftarrow{\text{Quot}(G)} \end{array} & \text{Quot}(\mathcal{D}) \\
 \downarrow & & \downarrow \\
 \mathcal{C} & \begin{array}{c} \xrightarrow{F} \\ \perp \\ \xleftarrow{G} \end{array} & \mathcal{D}
 \end{array} & & (9)
 \end{array}$$

As discussed above, predicates and quotients in the opposite category are (as objects) exactly quotients and predicates in the original category respectively. In the following result, we take a dual adjunction, so that the lifting gives an adjunction between predicates and quotients. We further give conditions under which the quotients correspond to equivalence relations (ERel). We then have adjunctions between predicates and equivalence relations, which we require for our applications to modal logic in Section 6.

► **Corollary 6.** *Let  $\mathcal{C}$  and  $\mathcal{D}$  be complete, well-powered and co-complete, co-well-powered categories respectively, with factorisation systems  $(\mathcal{E}_1, \mathcal{M}_1), (\mathcal{E}_2, \mathcal{M}_2)$  with  $\mathcal{M}_1$  consisting of monos, and  $\mathcal{E}_2$  consisting of epis. Suppose also an adjunction  $F \dashv G: \mathcal{D}^{\text{op}} \rightarrow \mathcal{C}$ , such that  $F(\mathcal{E}_1) \subseteq \mathcal{M}_2$ . If  $\mathcal{D}$  is an (Barr) exact category in which all epis are regular, we obtain a lifting of the adjunction as on the left below. If instead  $\mathcal{C}$  is exact and all epis are regular we obtain the lifting of the adjunction as on the right:*

## 6:8 Forward and Backward Steps in a Fibration

$$\begin{array}{ccc}
 \begin{array}{ccc}
 \text{Pred}(\mathcal{C}) & \begin{array}{c} \xrightarrow{\text{Pred}(F)} \\ \perp \\ \xleftarrow{\text{Pred}(G)} \end{array} & \text{ERel}(\mathcal{D})^{\text{op}} \\
 \downarrow & & \downarrow \\
 \mathcal{C} & \begin{array}{c} \xrightarrow{F} \\ \perp \\ \xleftarrow{G} \end{array} & \mathcal{D}^{\text{op}}
 \end{array} & & 
 \begin{array}{ccc}
 \text{ERel}(\mathcal{C}) & \begin{array}{c} \xrightarrow{\text{Quot}(F)} \\ \perp \\ \xleftarrow{\text{Quot}(G)} \end{array} & \text{Pred}(\mathcal{D})^{\text{op}} \\
 \downarrow & & \downarrow \\
 \mathcal{C} & \begin{array}{c} \xrightarrow{F} \\ \perp \\ \xleftarrow{G} \end{array} & \mathcal{D}^{\text{op}}
 \end{array}
 \end{array} \tag{10}$$

Our goal is now to relate liftings of adjunctions to adjunctions defined between fibres. In [21] it is shown how this can be done for fibrations over a single base category. Also studied in [15, 16] is how adjunctions between fibrations with distinct base categories arise from adjunctions between fibrations with a common base category.

► **Lemma 7.** *Suppose we have a lifted adjunction as in (7). Then we also have the following adjunctions between fibres, for all objects  $X$  of  $\mathcal{C}$  and  $Y$  of  $\mathcal{D}$ , where  $\eta$  and  $\varepsilon$  are the unit and counit of the adjunction  $F \dashv G$  respectively.*

$$\begin{array}{ccc}
 \mathcal{E}_X & \begin{array}{c} \xrightarrow{\overline{F}} \\ \perp \\ \xleftarrow{\eta_X^* \circ \overline{G}} \end{array} & \mathcal{F}_{FX} & & 
 \mathcal{E}_{GY} & \begin{array}{c} \xrightarrow{\coprod_{\varepsilon_Y} \circ \overline{F}} \\ \perp \\ \xleftarrow{\overline{G}} \end{array} & \mathcal{F}_Y
 \end{array} \tag{11}$$

Returning to the example of adjunctions for predicate and relation liftings (Lemma 4), Lemma 7 allows us to obtain adjunctions between fibres, which are of interest when we study invariants and coinductive predicates in the coming section. In particular, we recover the adjunctions from Section 2.

► **Example 8** (Eilenberg-Moore). For the case of an adjunction monadic over  $\text{Set}$ , each category ( $\text{Set}$  and  $\text{EM}(T)$  for a monad  $T$ ) has a  $(\text{RegEpi}, \text{Mono})$ -factorisation system as they are regular. Also, the abstract epis are preserved by left adjoints and these categories are complete and well-powered. We thus obtain a lifting of any monadic adjunction to predicates, relations and quotients by Lemma 4 and Corollary 5. Furthermore, the adjunction between fibres as on the right in Equation (11) then exactly instantiates to the adjunctions discussed in Section 2.1 for the cases of compact Hausdorff spaces and convex algebras. The left adjoint in each case takes the closure or convex hull of a relation on a set.

In fact, each of these “local” adjunctions implies the existence of the other. Note that we do not assume a (global) lifted adjunction, so we must assume (co-)fibredness explicitly.

► **Lemma 9.** *Suppose we have an adjunction  $F \dashv G: \mathcal{D} \rightarrow \mathcal{C}$  and bifibrations  $p: \mathcal{E} \rightarrow \mathcal{C}$ ,  $q: \mathcal{F} \rightarrow \mathcal{D}$ . Also, suppose  $\overline{G}$  is a fibred lifting of  $G$  and  $\overline{F}$  is a co-fibred lifting of  $F$ . Then we have a adjunctions  $\overline{F}_X \dashv \eta_X^* \circ \overline{G}_{FX}$  for all  $X$  iff we have adjunctions  $\coprod_{\varepsilon_Y} \circ \overline{F}_{GY} \dashv \overline{G}_Y$  for all  $Y$ , that is, the adjunctions in (11) can be derived from each other.*

Due to results of [16, 15] on factorisation of fibred adjunctions, we can also go from the existence of adjunctions between fibres (above all objects of our base category) to an adjunction between the total categories. As mentioned before, we drop the requirement of fibredness as much as possible.

► **Lemma 10.** *Suppose we have an adjunction  $F \dashv G: \mathcal{D} \rightarrow \mathcal{C}$  and fibrations  $p: \mathcal{E} \rightarrow \mathcal{C}$  and  $q: \mathcal{F} \rightarrow \mathcal{D}$ . Then the following are equivalent*

1. A lifting of the adjunction to  $\overline{F} \dashv \overline{G}: \mathcal{F} \rightarrow \mathcal{E}$
2. A fibred lifting  $\overline{G}$  of  $G$  and for each object  $Y$  of  $\mathcal{D}$ , a left adjoint to  $\overline{G}: \mathcal{F}_Y \rightarrow \mathcal{E}_{GY}$
3. A fibred lifting  $\overline{G}$  of  $G$  and for each object  $Y$  of  $\mathcal{D}$ ,  $\overline{G}: \mathcal{F}_Y \rightarrow \mathcal{E}_{GY}$  preserves meets.

This allows us, under certain conditions, to lift an adjunction to *equivalence* relations.

► **Lemma 11.** *In the context of Lemma 4 and assuming that we use a factorisation system on  $\mathcal{C}$  with all abstract epis being split epi,  $\text{Rel}(G)$  maps equivalence relations to equivalence relations. Further, its restriction to equivalence relations has a left adjoint  $\overline{F}$ , forming a lifting of the adjunction between base categories.*

► **Remark 12.** The condition on abstract epis of Lemma 11 is stronger than our earlier assumption that the left adjoint preserves abstract epis, as having a factorisation system  $(\mathcal{E}, \mathcal{M})$  with  $\mathcal{E} \subseteq \text{SplitEpi}$  and  $\mathcal{M} \subseteq \text{Mono}$  implies that in fact  $\mathcal{E} = \text{SplitEpi}$  (cf. [19, Exercise 4.4.2]) and split epis are absolute in the sense that *all* functors preserve them.

## 5 Comparing coinductive predicates along steps

In this section, we consider endofunctors in the setting of an adjunction, and will study coalgebras for these endofunctors – and sometimes algebras, viewed as coalgebras in an opposite category. These endofunctors are connected via the notion of a *step* [32], which is a natural transformation that allows one to transport coalgebras along the adjunction. More formally, steps give rise to liftings of the right and left adjoint (depending on which kind of step) to categories of coalgebras. The key question that we address in this section is whether these liftings to categories of coalgebras preserve a coinductive predicate of interest.

► **Definition 13.** *Consider an adjunction with endofunctors as follows:*

$$B \curvearrowright \mathcal{C} \begin{array}{c} \xrightarrow{F} \\ \perp \\ \xleftarrow{G} \end{array} \mathcal{D} \curvearrowleft L \quad (12)$$

*Then a forward step is a natural transformation  $\delta: BG \rightarrow GL$  and a backward step is a natural transformation  $\iota: GL \rightarrow BG$ .*

Due to the adjunction  $F \dashv G$ , a natural transformation  $\delta: BG \rightarrow GL$  has a *mate*  $\hat{\delta}: FB \rightarrow LF$  given by  $\hat{\delta} = \varepsilon_{LF} \circ F\delta \circ \eta_{FB}$ . This then gives rise to liftings of  $F$  and  $G$  to coalgebras, called *step-induced coalgebra liftings* and denoted  $\hat{F}: \text{CoAlg}(B) \rightarrow \text{CoAlg}(L)$  and  $\hat{G}: \text{CoAlg}(L) \rightarrow \text{CoAlg}(B)$  respectively. These are defined on objects by

$$f: X \rightarrow BX \mapsto \hat{\delta}_X \circ Ff: FX \rightarrow LFX \quad (13)$$

$$g: Y \rightarrow LY \mapsto \iota_Y \circ Gg: GY \rightarrow BGY \quad (14)$$

On arrows, they act simply as  $F$  and  $G$ . This is well defined due to functoriality of  $F$  and  $G$  and naturality of the involved steps.

► **Remark 14.** The names “forward” and “backward” steps are from [35], where they are assumed to be one-sided inverses. In the current paper, we make no such assumption and study forward and backward steps independently from each other. In [32] only what we refer to as a forward step appears. There is a clear asymmetry between the two; forward steps have a mate correspondence, and at least two other equivalent presentations via transposing along the adjunction. For backward steps there seem to be no such equivalent characterisations, as the left adjoint is on the “wrong” side.

► **Example 15.** An example of such transformations occurs in a determinisation procedure for probabilistic automata given in [4]. There, the functors  $B$  and  $L$  are taken to be  $B = \mathcal{P}^A$  and  $L = \mathcal{P}_c^A$  with  $A$  a set of labels and  $\mathcal{P}_c$  the convex powerset on  $\text{EM}(\mathcal{D})$ , equivalent to

## 6:10 Forward and Backward Steps in a Fibration

the category of convex algebras. Note that we allow the empty set in the definition of  $\mathcal{P}_c(X) = \{S \subseteq X \mid S \text{ convex}\}$ . It is shown in *op. cit.* that there is an injective natural transformation  $\iota: \mathcal{U} \circ \mathcal{P}_c^A \rightarrow \mathcal{P}^A \circ \mathcal{U}$ , induced by an analogous transformation for  $B = \mathcal{P}$  and  $L = \mathcal{P}_c$ . Such a transformation without labels simply includes convex subsets into the powerset. This has a componentwise inverse which forms, for each subset, its convex hull.

Aside from this example, which we will revisit later, steps occur, e.g., as the one-step semantics of coalgebraic modal logics (more usually, the mate of a forward step) [31, 24, 32], and have been used to construct ultrafilter extensions of coalgebras [28]. In the case of ultrafilter extensions for powerset coalgebras, the steps are those defining a weak lifting in the sense of Garner [10]; the forward step forms the topological closure of all subsets, and the backward step includes closed subsets into the powerset.

### 5.1 Comparing coinductive predicates

We have now seen how steps defined for an adjunction with endofunctors on each of the categories allow us to map coalgebras along this adjunction. Further, when we have fibrations on each of the categories of the adjunction, and liftings of the involved functors, we can define predicates on these coalgebras. Next, we will combine these transformations and give conditions on the steps and liftings, which allow us to link predicates on a coalgebra with predicates on the coalgebra obtained by applying a step-induced lifting.

► **Assumption 16.** *Throughout this section, we assume a lifting  $\overline{F} \dashv \overline{G}: \mathcal{F} \rightarrow \mathcal{E}$  of an adjunction  $F \dashv G: \mathcal{D} \rightarrow \mathcal{C}$ , together with endofunctors  $B, L$  on  $\mathcal{C}$  and  $\mathcal{D}$  and liftings  $\overline{B}$  and  $\overline{L}$  to  $\mathcal{E}$  and  $\mathcal{F}$  respectively.*

These assumptions give us coinductive predicates on  $B$ -coalgebras, using  $\overline{B}$ , and on  $L$ -coalgebras, using  $\overline{L}$ . This setting allows us to put conditions on forward and backward steps. These conditions, in turn, allow us to obtain steps at the level of the induced adjunctions between fibres, which puts us back into the setting of Section 2. In particular, it allows us to apply Lemma 1 to preserve the relevant coinductive predicates. We now explain this in more detail for backward and forward steps separately.

#### 5.1.1 Preservation via backward steps

Consider a backward step  $\iota: GL \rightarrow BG$ . Given an  $L$ -coalgebra  $(Y, g)$  together with this backward step and Assumption 16, we have the following setting.

$$(\iota_Y \circ Gg)^* \circ \overline{B} \hookrightarrow \mathcal{E}_{GY} \begin{array}{c} \xrightarrow{\overline{F}} \\ \perp \\ \xleftarrow{\overline{G}} \end{array} \mathcal{F}_Y \hookrightarrow g^* \circ \overline{L} \quad (15)$$

The greatest fixed point  $\nu(g^* \circ \overline{L})$  is a coinductive predicate on the  $L$ -coalgebra  $(Y, g)$ . The greatest fixed point  $\nu((\iota_Y \circ Gg)^* \circ \overline{B})$  is instead a coinductive predicate on the  $B$ -coalgebra obtained by applying the lifting  $\hat{G}: \text{CoAlg}(L) \rightarrow \text{CoAlg}(B)$  induced by the step  $\iota$  to  $(Y, g)$ . Like in Section 2, we ask whether the right adjoint  $\overline{G}_Y$  preserves the greatest fixed point, that is, maps  $\nu(g^* \circ \overline{L})$  to  $\nu((\iota_Y \circ Gg)^* \circ \overline{B})$ .

The following result gives a sufficient condition for constructing a step in the above adjunction between fibres; this condition is in terms of the backward step  $\iota$  and the liftings.

► **Lemma 17.** For a (backward) step  $\iota: GL \rightarrow BG$  and an  $L$ -coalgebra  $g: Y \rightarrow LY$ :

1. If  $\overline{G} \circ \overline{L} \leq \iota^* \circ \overline{B} \circ \overline{G}$ , then  $\overline{G}_Y \circ g^* \circ \overline{L}_Y \leq (\iota_Y \circ Gg)^* \circ \overline{B}_{GY} \circ \overline{G}_Y$ ;
2. If  $\iota^* \circ \overline{B} \circ \overline{G} \leq \overline{G} \circ \overline{L}$ , then  $\overline{G}_Y \circ g^* \circ \overline{L}_Y \geq (\iota_Y \circ Gg)^* \circ \overline{B}_{GY} \circ \overline{G}_Y$ .

We note that the condition of Item 1 is equivalent to having a lifting  $\bar{\iota}: \overline{G}\overline{L} \rightarrow \overline{B}\overline{G}$  of  $\iota$ , using the existence of a Cartesian lifting of  $\iota$ . The inequality of Item 2 often requires further assumptions and more work. We will give some instances where it is satisfied in Section 6. Together, the assumptions are equivalent to  $\bar{\iota}$  itself being a Cartesian map. Using Lemmas 1, 7, and 17 we obtain the following preservation result for coinductive predicates.

► **Corollary 18.** Suppose we have a (backward) step  $\iota: GL \rightarrow BG$ . Then for any  $g: Y \rightarrow LY$ :

1. If  $\overline{G} \circ \overline{L} \leq \iota^* \circ \overline{B} \circ \overline{G}$ , then  $\overline{G}_Y(\nu(g^* \circ \overline{L}_Y)) \leq \nu((\iota_Y \circ Gg)^* \circ \overline{B}_{GY})$ ;
2. If  $\iota^* \circ \overline{B} \circ \overline{G} \leq \overline{G} \circ \overline{L}$ , then  $\overline{G}_Y(\nu(g^* \circ \overline{L}_Y)) \geq \nu((\iota_Y \circ Gg)^* \circ \overline{B}_{GY})$ .

Corollary 18 thus gives sufficient conditions for  $\overline{G}_Y$  to map the greatest fixed point of the coinductive predicate on an  $L$ -coalgebra  $(Y, g)$  to the greatest fixed point of the coinductive predicate on the  $B$ -coalgebra  $\hat{G}(Y, g)$  constructed via  $\iota$ , in the setting of (15).

► **Remark 19.** It is in fact not necessary that  $\iota$  is natural; that is, Lemma 17 and Corollary 18 go through even if  $\iota$  is just a collection of arrows.

### 5.1.2 Preservation via forward steps

We proceed to focus on forward steps. Recall from Lemma 7 that the lifted adjunction between fibrations induces two types of adjunctions between fibres; for backward steps we used one of them, for forward steps we use the other. We thus work in the following setting:

$$f^* \circ \overline{B} \circ \mathcal{C}_X \begin{array}{c} \xrightarrow{\overline{F}} \\ \perp \\ \xleftarrow{\eta_X^* \circ \overline{G}} \end{array} \mathcal{F}_{FX} \begin{array}{c} \xrightarrow{(\hat{\delta}_X \circ Ff)^* \circ \overline{L}} \\ \xleftarrow{\eta_X^* \circ \overline{G}} \end{array} \quad (16)$$

where  $(X, f)$  is a  $B$ -coalgebra. We have the following result on constructing steps in this adjunction between fibres.

► **Lemma 20.** Suppose we have a (forward) step  $\delta: BG \rightarrow GL$ , then for  $f: X \rightarrow BX$ :

1. If  $\delta^* \circ \overline{G} \circ \overline{L} \leq \overline{B} \circ \overline{G}$  and  $\overline{B}$  is fibred, then  $\eta_X^* \circ \overline{G}_{FX} \circ (\hat{\delta}_X \circ Ff)^* \circ \overline{L}_{FX} \leq f^* \circ \overline{B}_X \circ \eta_X^* \circ \overline{G}_{FX}$ ;
2. If  $\overline{B} \circ \overline{G} \leq \delta^* \circ \overline{G} \circ \overline{L}$ , then  $\eta_X^* \circ \overline{G}_{FX} \circ (\hat{\delta}_X \circ Ff)^* \circ \overline{L}_{FX} \geq f^* \circ \overline{B}_X \circ \eta_X^* \circ \overline{G}_{FX}$ .

Similarly to backward steps, we get the following result from Lemmas 1, 7, and 20, giving a sufficient condition for preservation of the coinductive predicate by the right adjoint in (16).

► **Corollary 21.** Suppose we have a (forward) step  $\delta: BG \rightarrow GL$  and a lifting of the adjunction as in Equation (7). Then for  $f: X \rightarrow BX$ :

1. If  $\delta^* \circ \overline{G} \circ \overline{L} \leq \overline{B} \circ \overline{G}$  and  $\overline{B}$  is fibred, then  $\eta_X^* \circ \overline{G}_{FX}(\nu((\hat{\delta}_X \circ Ff)^* \circ \overline{L}_{FX})) \leq \nu(f^* \circ \overline{B}_X)$ ;
2. If  $\overline{B} \circ \overline{G} \leq \delta^* \circ \overline{G} \circ \overline{L}$ , then  $\eta_X^* \circ \overline{G}_{FX}(\nu((\hat{\delta}_X \circ Ff)^* \circ \overline{L}_{FX})) \geq \nu(f^* \circ \overline{B}_X)$ .

► **Remark 22.** Contrary to the case of backward steps (see Remark 19), for forward steps we use naturality, in the proof of Lemma 20. That proof is more involved than that of Lemma 17, emphasising again the asymmetry between forward and backward steps.

► **Remark 23.** We have assumed that the adjunction between base categories lifts to the total categories of the fibrations, even though the results in Corollary 21 and Corollary 18 are about the adjunctions between fibres. Therefore, one might be tempted to only assume these adjunctions between fibres instead of an adjunction between total categories. However,

## 6:12 Forward and Backward Steps in a Fibration

in Lemma 17 and Lemma 20 (on which the aforementioned results rely) we also use that the right adjoint  $\overline{G}$  is fibred, and if we additionally assume this then it is equivalent to having an adjunction between the total categories (Lemma 10).

► **Remark 24.** Our focus is on the separate analysis of  $\iota$  and  $\delta$ . If we assume instead that:  $\iota$  and  $\delta$  both exist; their liftings form an isomorphism  $\bar{\iota}: \overline{G}\overline{L} \simeq \overline{B}\overline{G}$ ; and we have a fibred lifting  $\overline{G}$  of  $G$  such that its restriction to fibres preserves meets, then we have  $\overline{G}_Y(\nu(g^* \circ \overline{L}_Y)) = \nu((\iota_Y \circ Gg)^* \circ \overline{B}_{GY})$  where  $\iota = p\bar{\iota}$ , for any  $L$ -coalgebra  $(Y, g)$ . Restricting ourselves to fibrations over a single base category,  $B = L$ , and a lifting of the identity between the total categories, we recover [34, Prop. 6.2].

## 6 Examples

### 6.1 Lax liftings

Our first application of the results of the previous section continues on from Example 15, where we are now able to apply Corollary 18 to show the preservation and reflection of bisimilarity by the second stage of the construction given in the example.

That construction goes from probabilistic automata, which combine probabilistic transitions with non-deterministic choice, to belief-state transformers, where the probabilities occur in the state space rather than on the transitions. It has its roots in the generalised determinisation procedure of [33], but requires an alternative approach due to the non-existence of a lifting of the powerset monad to convex algebras. The determinisation starts from a monadic adjunction over  $\mathbf{Set}$ , and then proceeds in two steps: first a lifting of the left adjoint gives a “determinised” system with algebraic structure, then a lifting of the right adjoint forgets this structure to give a system in  $\mathbf{Set}$ . Here, we consider the second stage and take a lifting of the adjunction and endofunctors  $B$  and  $L$  to  $\mathbf{Rel}$  fibrations as in (17), so that we may apply our earlier results to show preservation and reflection of bisimilarity.

$$\begin{array}{ccc}
 \text{Rel}(B) \curvearrowright \text{Rel}(\mathbf{Set}) & \begin{array}{c} \xrightarrow{\text{Rel}(\mathcal{F})} \\ \perp \\ \xleftarrow{\text{Rel}(\mathcal{U})} \end{array} & \text{Rel}(\mathbf{EM}(T)) \curvearrowright \text{Rel}(L) \\
 \downarrow & & \downarrow \\
 B \curvearrowright \mathbf{Set} & \begin{array}{c} \xrightarrow{\mathcal{F}} \\ \perp \\ \xleftarrow{\mathcal{U}} \end{array} & \mathbf{EM}(T) \curvearrowright L
 \end{array} \tag{17}$$

The lifting of the right adjoint to coalgebras uses a  $\iota$  which comes from a so-called *lax lifting* [4]. Given a functor  $B: \mathbf{Set} \rightarrow \mathbf{Set}$ , a lax lifting of  $B$  is a functor  $L: \mathbf{EM}(T) \rightarrow \mathbf{EM}(T)$  such that there is an injective natural transformation  $\iota: \mathcal{U} \circ L \rightarrow B \circ \mathcal{U}$ . We show the following result for transformations that are componentwise split mono, and then show how this applies to the example of probabilistic automata.

► **Lemma 25.** *The lifting  $\hat{\mathcal{U}}: \mathbf{CoAlg}(L) \rightarrow \mathbf{CoAlg}(B)$  induced by a componentwise split mono transformation  $\iota: \mathcal{U} \circ L \rightarrow B \circ \mathcal{U}$  preserves and reflects bisimilarity.*

Taking  $B = \mathcal{P}^A$  and  $L = \mathcal{P}_c^A$  in the setting of Equation (17) (recall also Example 15), we have a componentwise split mono  $\iota$  because we have an injective transformation, and  $\mathcal{U} \circ \mathcal{P}_c^A(X)$  is only empty when  $A$  is empty, in which case also  $\mathcal{P}^A \circ \mathcal{U}(X)$  is empty. In [35], a similar result is shown for behavioural equivalence instead of bisimilarity in case the functor  $B$  preserves weak pullbacks.

## 6.2 Expressivity

In this subsection, we will establish the well-known expressivity of modal logic with respect to  $\mathcal{P}$ -bisimilarity using our abstract framework. Note that for simplicity we consider (unlabelled) transition systems modelled as  $\mathcal{P}$ -coalgebra with  $\mathcal{P}: \mathbf{Set} \rightarrow \mathbf{Set}$  the (full) powerset functor.

We relate bisimilarity to the logic defined by the following grammar:

$$\phi ::= \diamond \left( \bigwedge_{i \in I} \phi_i \wedge \bigwedge_{i \in J} \neg \phi_i \right)$$

There are no size restrictions on  $I$  and  $J$ , so that the collection of formulas forms a proper class. As a consequence, the usual syntax based on initial algebras (living in  $\mathbf{Set}$ ) is not well founded. While this expressivity result is not new – [30] shows expressivity of a similar infinitary modal logic for  $\mathcal{P}$ -bisimilarity – we include this example as it demonstrates that this fundamental expressivity result of modal logic fits into our general framework. So, these considerations leads to the contravariant adjoint situation between  $\mathbf{Set}$  and  $\mathbf{Set}^{\text{op}}$  as depicted in (18).

Now taking inspiration from [24] where this is done for the finitary case, we consider the coalgebraic modal logic  $(L, \delta)$ , where the syntax is given by the endofunctor  $L = \mathcal{P}(2 \times -)$  on  $\mathbf{Set}$  and the “one-step” semantics  $\delta: \mathcal{P}2^- \rightarrow 2^L$  is defined as follows:

$$\delta_X(S)(U) = \bigvee_{\varphi \in S} \left( \bigwedge_{(1,x) \in U} \varphi(x) \wedge \bigwedge_{(0,x) \in U} \neg \varphi(x) \right).$$

Note that the step-induced coalgebra lifting of  $\delta$  turns a transition system with set  $X$  of states into an  $L$ -algebra on  $2^X$  (cf. (13)). This gives an abstract notion of definability: precisely those sets  $\varphi \in 2^X$  which are “reachable”, that is, contained in the least subalgebra of all predicates on  $LX$ . So we consider the fibrations  $\mathbf{Pred}$  and  $\mathbf{ERel}$  of predicates and equivalence relations (see (18)) on  $\mathbf{Set}$  (note  $\mathbf{ERel}$  is chosen since  $\mathcal{P}$ -bisimilarity is an equivalence).

$$\begin{array}{ccc} \text{Pred}(F) & & \\ \text{Rel}(\mathcal{P}) \curvearrowright \mathbf{ERel} & \begin{array}{c} \xrightarrow{\perp} \\ \text{Pred}(G) \\ \xleftarrow{\perp} \end{array} & \mathbf{Pred}^{\text{op}} \curvearrowleft \text{Pred}(L) \\ & & \\ \mathcal{P} \curvearrowright \mathbf{Set} & \begin{array}{c} \xrightarrow{F=2^-} \\ \perp \\ \xleftarrow{G=2^-} \end{array} & \mathbf{Set}^{\text{op}} \curvearrowleft L=\mathcal{P}(2 \times -) \end{array} \quad (18)$$

Next we define the corresponding liftings of the functor in order to invoke Corollary 21 in proving expressivity of our logic. For a predicate  $P \mapsto Y$  we fix  $\mathbf{Pred}(2^-)(P) = \{(\varphi, \psi) \mid \forall p \in P. \varphi(p) \leftrightarrow \psi(p)\}$  and  $\mathbf{Pred}(L)(P) = (\mathcal{P}(2 \times P) \mapsto \mathcal{P}(2 \times Y))$ .

► **Remark 26.** It should be noted that the Galois connection (cf. Section 2) between the lattices  $\mathbf{ERel}_X, \mathbf{Pred}_{2^X}^{\text{op}}$  can be reconstructed from the adjunction between the total categories  $\mathbf{ERel}$  and  $\mathbf{Pred}^{\text{op}}$ . In particular, Lemma 7 gives:  $\overline{F} \dashv \eta_X^* \circ \overline{G}: \mathbf{Pred}_{F^X}^{\text{op}} \rightarrow \mathbf{ERel}_X$ . Moreover,  $\eta_X^* \circ \overline{G} = g$  (recall  $g$  from Section 2).

Now adequacy and expressivity of our logic  $L$  follows by proving their corresponding sufficient condition (cf. Corollary 21) as in the following proposition.

► **Proposition 27.** *For any  $P \mapsto X$ , we have  $\delta^*(\mathbf{Pred}(G) \mathbf{Pred}(L)(P)) = \text{Rel}(\mathcal{P}) \mathbf{Pred}(G)(P)$ .*

## 6.3 Apartness

In this subsection, we again consider (unlabelled) transition systems and rather show how our framework allows us to prove the dual of the Hennessy-Milner theorem: *two states are  $\mathcal{P}$ -apart [12, 11] (i.e., not bisimilar) iff there is a distinguishing formula between them.*

## 6:14 Forward and Backward Steps in a Fibration

$$\begin{array}{ccccc}
 \text{ERel}^{\text{fop}} & \xrightarrow{\neg} & \text{ERel} & \xrightarrow{\text{Pred}(F)} & \text{Pred}^{\text{op}} & \xrightarrow{\text{Pred}(L)} \\
 \downarrow & \xleftarrow{\neg} & \downarrow & \begin{array}{c} \perp \\ \text{Pred}(G) \end{array} & \downarrow & \\
 \mathcal{P} \curvearrowright \text{Set} & \xlongequal{\quad} & \text{Set} & \xrightarrow{F=2^-} & \text{Set}^{\text{op}} & \xrightarrow{L=\mathcal{P}(2 \times -)} \\
 & & & \begin{array}{c} \perp \\ G=2^- \end{array} & & 
 \end{array} \tag{19}$$

Recall that an apartness relation  $R$  on a set  $X$  is an irreflexive, symmetric, and co-transitive relation (i.e.,  $\forall x, y \in X. x R y \rightarrow \forall z \in Z. (x R z \vee y R z)$ ). Following [12], the fibration of apartness relations on  $\text{Set}$  can be seen as the fibred opposite of  $\text{ERel}$ . In particular, the functor  $\neg$  maps a tuple  $(X, R)$  (when  $R$  is an equivalence/apartness on  $X$ ) to the tuple  $(X, \neg R)$ . Note that, alternatively, one can also directly recover the above adjoint situation from (10). Moreover, on fibres, the functor  $\neg \circ \text{Pred}(G)$  takes a predicate  $P \mapsto X$  and produces an apartness relation  $P_{\neg G}$  on  $2^X$  given as follows:

$$\varphi P_{\neg G} \varphi' \iff \exists x \in P. (\varphi \#_x \varphi' \vee \varphi' \#_x \varphi), \quad \text{where } \varphi \#_x \varphi' \iff \varphi(x) \wedge \neg \varphi'(x).$$

For the lifting  $\text{ERel}^{\text{fop}}(\mathcal{P})$ , we consider the following definition<sup>1</sup>:

$$U \text{ERel}^{\text{fop}}(\mathcal{P})(R) V \iff \exists x \in U. \forall y \in V. x R y \vee \exists y \in V. \forall x \in U. x R y$$

Now we are in the position to use Corollary 21 and establish the dual of Hennessy-Milner theorem, which was also recently shown in [11] though for image-finite transition systems.

► **Proposition 28.** *For any  $P \mapsto X$ ,  $\text{ERel}^{\text{fop}}(\mathcal{P})(P_{\neg G}) = \delta^*(\neg \text{Pred}(G) \text{Pred}(L)(P))$ .*

### 6.4 Completeness

We now turn to the example of completeness of (finitary) basic modal logic by using a backward step  $\iota$ . Consider the functor  $B = \mathcal{P}$  with the *dual* adjunction of Equation (20) for  $F = \text{hom}_{\text{Set}}(-, 2)$  and  $G = \text{hom}_{\text{BA}}(-, 2)$ .

$$B \curvearrowright \text{Set} \begin{array}{c} \xrightarrow{F} \\ \perp \\ \xleftarrow{G} \end{array} \text{BA}^{\text{op}} \curvearrowright L \tag{20}$$

We obtain basic modal logic as coalgebraic modal logic for  $B$  using the predicate lifting  $\blacksquare : F \rightarrow F \circ B$  where for  $X \in \text{Set}$  and  $U \in FX$  we put  $\blacksquare_X(U) = \{V \in BX \mid V \subseteq U\}$ . Consider a sound and complete deduction system  $D$  for propositional logic. We define modal derivability  $\vdash_{ML}$  by extending  $D$  with the derivation rules

$$\frac{a \leftrightarrow b \wedge c}{\Box a \leftrightarrow \Box b \wedge \Box c} \qquad \frac{a \leftrightarrow \top}{\Box a \leftrightarrow \top}$$

We call a set of formulas  $\Phi$  inconsistent if there are formulas  $\varphi_1, \dots, \varphi_n \in \Phi$  such that  $\vdash_{ML} \varphi_1 \wedge \dots \wedge \varphi_n \rightarrow \perp$ , otherwise  $\Phi$  is consistent. Our goal is to prove completeness of the logic, i.e., we would like to show that any consistent set of formulas  $\Phi$  is satisfied in some  $B$ -coalgebra. The proof usually proceeds via a canonical model construction, that equips the set of maximally consistent sets of formulas (“theories”) with a  $B$ -coalgebra structure.

<sup>1</sup> Note that our definition differs from the lifting of an apartness relation given in [12], where the two logical formulae ( $\exists x \in U. \forall y \in V. x R y$  and  $\exists y \in V. \forall x \in U. x R y$ ) are composed incorrectly by conjunction.



We will adjust this by viewing canonical models as fixpoints of a construction that defines models on possibly inconsistent theories and iteratively removes inconsistent theories until only consistent ones are left. An issue is that inconsistent theories will not have a model and thus we cannot define a meaningful  $B$ -coalgebra structure on them. Instead, we leave the coalgebra structure “undefined”. To model such a partial  $B$ -coalgebra structure we switch to  $B_{\perp}$ -coalgebras with  $B_{\perp} = 1 + B$ . The intuition behind our construction is that the coalgebra structure maps a theory to  $\text{inl}(\ast)$  iff it is inconsistent. Ultimately we are left with a  $B$ -coalgebra based on the set of all consistent theories. The full setup is as in Equation (21).

$$\begin{array}{ccc}
 \overline{B_{\perp}} \hookrightarrow \text{Pred} & \xrightarrow{\overline{F}} & \text{Cong}^{\text{op}} \hookrightarrow \overline{L} \\
 \downarrow & \begin{array}{c} \perp \\ \overline{G} \end{array} & \downarrow \\
 B_{\perp} = 1 + \mathcal{P} \hookrightarrow \text{Set} & \xrightarrow{F} & \text{BA}^{\text{op}} \hookrightarrow L \\
 & \begin{array}{c} \perp \\ G \end{array} & 
 \end{array} \tag{21}$$

Here  $\text{Cong}$  denotes the category of congruences over Boolean algebras, i.e., objects are pairs  $(A, \equiv)$  with  $A$  being a Boolean algebra and  $\equiv \subseteq A \times A$  being a congruence on  $A$ . It is well known that  $\text{Cong}$  is isomorphic to the category  $\text{Quot}$  of quotients of Boolean algebras. Therefore the above situation can be seen to meet the requirements of Cor. 6 and we obtain suitable liftings  $\overline{F}$  and  $\overline{G}$  of  $F$  and  $G$ , respectively.

Given a congruence  $\equiv \subseteq A \times A$ , the predicate  $\overline{G}(\equiv)$  on  $GA$  can be given by  $u \in \overline{G}(\equiv)$  iff  $\forall a \in u. a \not\equiv \perp$  (equivalently to the “canonical” lifting of Corollary 6, so that we have a left adjoint). Intuitively,  $\overline{G}(\equiv)$  contains all ultrafilters that are consistent with respect to  $\equiv$ . The lifting  $\overline{B_{\perp}}$  of  $B_{\perp} = 1 + B$  is defined using the canonical predicate lifting for  $B$ , i.e., for all  $t \in B_{\perp}X$  and a predicate  $U \subseteq X$  we have  $t \in \overline{B_{\perp}}(U)$  iff  $t = \text{inr}(V)$  for some  $V \subseteq U$ . Finally, the lifting  $\overline{L}$  of  $L$  is defined by letting  $(LA, \equiv_{LA})$  be the smallest Boolean congruence containing  $\Box a \wedge \Box b \equiv_{LA} \Box c$  for  $a \wedge b \equiv c$  and  $\Box a \equiv_{LA} \top$  for  $a \equiv \top$ . We turn now to the definition of a suitable backward step  $\iota : GL \rightarrow B_{\perp}G$  that will allow us to prove completeness. To this aim we let  $u \in GLA$  and consider the following intersection:

$$\text{sem}(u) = \bigcap_{\Box a \in u} \blacksquare \hat{a} \cap \bigcap_{\Box a \notin u} BGA \setminus \blacksquare \hat{a}$$

where  $\hat{a} = \{v \in GA \mid a \in v\}$ . We define a  $\iota$  by selecting for each  $u \in GLA$  an element  $t \in \text{sem}(u)$  if such an element exists and by putting  $\iota_A(u) := \text{inr}(t)$ . Otherwise we put  $\iota_A(u) = \text{inl}(\ast)$ . Note that with this definition  $\iota$  will not necessarily be natural, but this is not required in our setting. In addition, using topological machinery, we could ensure naturality of  $\iota$  by requiring  $\iota(u)$  to be closed in the Vietoris topology (cf. e.g. [28]).

We now show that  $\iota_A(u) = \text{inl}(\ast)$  iff  $u \notin \overline{G}(\equiv_{LA}) = \overline{G}(\equiv_A)$ , by case distinction:

**Case**  $u \notin \overline{G}(\equiv_{LA})$  because there exists  $a, b, c \in A$  with  $a \wedge b \equiv c$  but  $\Box a, \Box b \in u$  and  $\Box c \notin u$ . Then  $\text{sem}(u) \subseteq \blacksquare \hat{a} \cap \blacksquare \hat{b} \cap BGA \setminus \blacksquare \hat{c}$  and the latter is empty because any element would need to contain an ultrafilter  $v \in GA$  with  $a \in v, b \in v$  and  $c \notin v$  which contradicts the assumption that  $a \wedge b \equiv c$  and  $v \in \overline{G}(\equiv_A)$ .

**Case** All other cases with  $u \notin \overline{G}(\equiv_{LA})$  can be proven in the same way as the first case.

**Case**  $u \in \overline{G}(\equiv_{LA})$ . In this case one can use compactness to argue that  $\text{sem}(u)$  is non-empty: by compactness and the definition of  $\blacksquare$ , if  $\text{sem}(u) = \emptyset$ , there would need to be some  $\Box a \in u$  and  $\{\Box a_1, \dots, \Box a_k\} \subseteq LA \setminus u$  such that  $\blacksquare \hat{a} \cap \bigcap_{j=1}^k BGA \setminus \blacksquare \hat{a}_j = \emptyset$  which can be seen to entail that  $a \leq a_j$  for some  $j \in \{1, \dots, k\}$ . By monotonicity of  $\Box$  (a well-known consequence of the axiomatisation above) we obtain  $\Box a \leq_{LA} \Box a_j$ . Therefore, as  $\Box a \in u$

and  $u \in \overline{G}(\equiv_{LA})$  by assumption, we get  $\Box a_j \in u$  which is a contradiction. This shows that  $\text{sem}(u) \neq \emptyset$  as required. Note that in the standard completeness proof of coalgebraic modal logic this case is the key step, proving so-called one-step completeness of the logic.

The claim above can be used to show that  $\overline{G} \circ \overline{L} = \iota^* \circ \overline{B}_\perp \circ \overline{G}$ . Furthermore, for a given  $g : LA \rightarrow A$  we observe that  $\nu((\iota_A \circ Gg)^* \circ (\overline{B}_\perp)_A)$  consists precisely of those ultrafilters in  $GA$  on which  $(\iota_A \circ Gg)$  restricts to a  $B$ -coalgebra structure, i.e., that cannot reach a state  $u \in GA$  for which  $(\iota_A \circ Gg)(u) = \text{inl}(\ast)$ . On the other hand, spelling out the definitions one can show that for an  $L$ -algebra  $g : LA \rightarrow A$ ,  $\mu(\coprod_g \circ \overline{L}_A)$  yields the least congruence  $\equiv$  over  $A$  that satisfies the modal axioms. Corollary 18 then implies that any ultrafilter  $u \in GA$  satisfying  $\nu((\iota_A \circ Gg)^* \circ (\overline{B}_\perp)_A)$  is  $\equiv$ -consistent (“soundness”) and that any  $\equiv$ -consistent ultrafilter of  $A$  is satisfiable (“completeness”). Here satisfiable simply means that there is  $B$ -coalgebra structure defined on  $u$ . To establish a precise connection with the standard notions of soundness and completeness, one would need to define the usual semantics of  $\Box$  via a forward step  $\delta$ . Standard completeness then follows when starting from the Boolean algebra consisting of all modal formulas quotiented by equivalence in propositional logic.

## 7 Related and future work

In [35] we studied a preservation result assuming both a forward step  $\delta$  and a backward step  $\iota$ , which form one-sided inverses, that is,  $\delta \circ \iota = \text{id}$ . In the current paper, we treat preservation of coinductive predicates by forward and backward steps as separate cases, which we realise by formulating the conditions in a purely fibrational way instead of assuming inverses. This allows us, for instance, to provide a general preservation result for lax liftings (Section 6.1), which can not be obtained from the results in *op. cit.*: the latter requires a *natural* inverse  $\delta$ , which is not part of the assumptions of a lax lifting (only componentwise inverses are assumed), and can in fact be non-trivial to provide; for instance, in [35] the argument for existence went via weak distributive laws. Moreover, in the current paper we are more general by moving from the relation fibration to general  $\text{CLat}_\wedge$ -fibrations; this allows us, for instance, to use fibrations of predicate and equivalence relation fibrations, as we do in the analysis of expressivity and completeness.

In [29] a general approach to expressivity of logics with respect to coinductive predicates is proposed. In that paper, there is a fibration only on one of the two categories, and the coinductive predicate of interest is related to logical equivalence. Logical equivalence is defined there via the semantics of the logic, which is in turn obtained via the universal property of an initial algebra. In contrast, in the current paper, we do not use initial algebras and instead obtain logical equivalence by characterising “modally definable” on the coalgebra of interest, which yields a notion of logical equivalence by applying the right adjoint in a Galois connection between equivalence relations and predicates. This Galois connection was also used in [23], and in the recent [1] as the starting point for proving expressivity. Here, instead, we obtain this Galois connection from an adjunction between fibrations.

**Future Work.** In [1] it was shown how the functional characterising bisimilarity can be synthesised from a “logic” function. Using the notations of this paper, this meant defining  $\overline{B}$  in terms of  $\overline{L}$ . This question and its symmetric one (constructing  $\overline{L}$  from  $\overline{B}$ ) are of interest at the global level of contravariant adjunctions. An answer to these questions would pave the way not only for sufficient conditions for expressivity, but also provide the means to establish them in a more structured manner. Last, it would be interesting to try and apply our results on comparing coinductive predicates and lifting adjunctions in the quantitative setting of (pseudo-)metrics.

---

**References**

---

- 1 Harsh Beohar, Sebastian Gurke, Barbara König, and Karla Messing. Hennessy-Milner theorems via Galois connections. In *CSL*, volume 252 of *LIPICs*, pages 12:1–12:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.
- 2 Nick Bezhanishvili, Gaëlle Fontaine, and Yde Venema. Vietoris bisimulations. *J. Log. Comput.*, 20(5):1017–1040, 2010.
- 3 Filippo Bonchi, Barbara König, and Daniela Petrisan. Up-to techniques for behavioural metrics via fibrations. In *CONCUR*, volume 118 of *LIPICs*, pages 17:1–17:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.
- 4 Filippo Bonchi, Alexandra Silva, and Ana Sokolova. Distribution bisimilarity via the power of convex algebras. *Log. Methods Comput. Sci.*, 17(3), 2021.
- 5 Marcello M. Bonsangue and Alexander Kurz. Duality for logics of transition systems. In *FoSSaCS*, volume 3441 of *Lecture Notes in Computer Science*, pages 455–469. Springer, 2005.
- 6 Marcello M. Bonsangue, Stefan Milius, and Alexandra Silva. Sound and complete axiomatizations of coalgebraic language equivalence. *ACM Trans. Comput. Log.*, 14(1):7:1–7:52, 2013.
- 7 Francis Borceux. *Handbook of categorical algebra: volume 1, Basic category theory*, volume 1. Cambridge University Press, 1994.
- 8 Liang-Ting Chen and Achim Jung. On a categorical framework for coalgebraic modal logic. In *MFPS*, volume 308 of *Electronic Notes in Theoretical Computer Science*, pages 109–128. Elsevier, 2014.
- 9 Brian A. Davey and Hilary A. Priestley. *Introduction to Lattices and Order, Second Edition*. Cambridge University Press, 2002.
- 10 Richard Garner. The Vietoris monad and weak distributive laws. *Appl. Categorical Struct.*, 28(2):339–354, 2020.
- 11 Herman Geuvers. Apartness and distinguishing formulas in hennessy-milner logic. In *A Journey from Process Algebra via Timed Automata to Model Learning*, volume 13560 of *Lecture Notes in Computer Science*, pages 266–282. Springer, 2022.
- 12 Herman Geuvers and Bart Jacobs. Relating apartness and bisimulation. *Log. Methods Comput. Sci.*, 17(3), 2021.
- 13 Ichiro Hasuo, Bart Jacobs, and Ana Sokolova. Generic trace semantics via coinduction. *Log. Methods Comput. Sci.*, 3(4), 2007.
- 14 Ichiro Hasuo, Toshiki Kataoka, and Kenta Cho. Coinductive predicates and final sequences in a fibration. *Math. Struct. Comput. Sci.*, 28(4):562–611, 2018.
- 15 Claudio Hermida. On fibred adjunctions and completeness for fibred categories. In *COMPASS/ADT*, volume 785 of *Lecture Notes in Computer Science*, pages 235–251. Springer, 1992.
- 16 Claudio Hermida. *Fibrations, logical predicates and indeterminates*. PhD thesis, University of Edinburgh, UK, 1993.
- 17 Claudio Hermida and Bart Jacobs. Structural induction and coinduction in a fibrational setting. *Inf. Comput.*, 145(2):107–152, 1998.
- 18 Jesse Hughes and Bart Jacobs. Simulations in coalgebra. *Theor. Comput. Sci.*, 327(1-2):71–108, 2004.
- 19 Bart Jacobs. *Introduction to Coalgebra: Towards Mathematics of States and Observation*, volume 59 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2016.
- 20 Bart Jacobs, Alexandra Silva, and Ana Sokolova. Trace semantics via determinization. *J. Comput. Syst. Sci.*, 81(5):859–879, 2015.
- 21 Bart P. F. Jacobs. *Categorical Logic and Type Theory*, volume 141 of *Studies in logic and the foundations of mathematics*. North-Holland, 2001.

- 22 Henning Kerstan, Barbara König, and Bram Westerbaan. Lifting adjunctions to coalgebras to (re)discover automata constructions. In *CMCS*, volume 8446 of *Lecture Notes in Computer Science*, pages 168–188. Springer, 2014.
- 23 Bartek Klin. The least fibred lifting and the expressivity of coalgebraic modal logic. In *CALCO*, volume 3629 of *Lecture Notes in Computer Science*, pages 247–262. Springer, 2005.
- 24 Bartek Klin. Coalgebraic modal logic beyond sets. In *MFPS*, volume 173 of *Electronic Notes in Theoretical Computer Science*, pages 177–201. Elsevier, 2007.
- 25 Yuichi Komorida, Shin-ya Katsumata, Nick Hu, Bartek Klin, and Ichiro Hasuo. Codensity games for bisimilarity. In *LICS*, pages 1–13. IEEE, 2019.
- 26 Yuichi Komorida, Shin-ya Katsumata, Clemens Kupke, Jurriaan Rot, and Ichiro Hasuo. Expressivity of quantitative modal logics : Categorical foundations via codensity and approximation. In *LICS*, pages 1–14. IEEE, 2021.
- 27 Clemens Kupke, Alexander Kurz, and Dirk Pattinson. Algebraic semantics for coalgebraic logics. In *CMCS*, volume 106 of *Electronic Notes in Theoretical Computer Science*, pages 219–241. Elsevier, 2004.
- 28 Clemens Kupke, Alexander Kurz, and Dirk Pattinson. Ultrafilter extensions for coalgebras. In *CALCO*, volume 3629 of *Lecture Notes in Computer Science*, pages 263–277. Springer, 2005.
- 29 Clemens Kupke and Jurriaan Rot. Expressive logics for coinductive predicates. *Log. Methods Comput. Sci.*, 17(4), 2021.
- 30 Lawrence S. Moss. Coalgebraic logic. *Ann. Pure Appl. Log.*, 96(1-3):277–317, 1999.
- 31 Dusko Pavlovic, Michael W. Mislove, and James Worrell. Testing semantics: Connecting processes and process logics. In *AMAST*, volume 4019 of *Lecture Notes in Computer Science*, pages 308–322. Springer, 2006.
- 32 Jurriaan Rot, Bart Jacobs, and Paul Blain Levy. Steps and traces. *J. Log. Comput.*, 31(6):1482–1525, 2021.
- 33 Alexandra Silva, Filippo Bonchi, Marcello M. Bonsangue, and Jan J. M. M. Rutten. Generalizing determinization from automata to coalgebras. *Log. Methods Comput. Sci.*, 9(1), 2013.
- 34 David Sprunger, Shin-ya Katsumata, Jérémy Dubut, and Ichiro Hasuo. Fibrational bisimulations and quantitative reasoning. In *CMCS*, volume 11202 of *Lecture Notes in Computer Science*, pages 190–213. Springer, 2018.
- 35 Ruben Turkenburg, Clemens Kupke, Jurriaan Rot, and Ezra Schoen. Preservation and reflection of bisimilarity via invertible steps. In *FoSSaCS*, volume 13992 of *Lecture Notes in Computer Science*, pages 328–348. Springer, 2023.
- 36 Glynn Winskel. A compositional proof system on a category of labelled transition systems. *Inf. Comput.*, 87(1/2):2–57, 1990.

# Structural Operational Semantics for Heterogeneously Typed Coalgebras



Harald König  

Fachhochschule für die Wirtschaft Hannover, Germany

Western Norway University of Applied Sciences, Bergen, Norway

Uwe Wolter  

University of Bergen, Norway

Tim Kräuter  

Western Norway University of Applied Sciences, Bergen, Norway

---

## Abstract

Concurrently interacting components of a modular software architecture are heterogeneously structured behavioural models. We consider them as coalgebras based on different endofunctors. We formalize the composition of these coalgebras as specially tailored segments of distributive laws of the bialgebraic approach of Turi and Plotkin. The resulting categorical rules for structural operational semantics involve many-sorted algebraic specifications, which leads to a description of the components together with the composed system as a single holistic behavioural system. We evaluate our approach by showing that observational equivalence is a congruence with respect to the algebraic composition operation.

**2012 ACM Subject Classification** Theory of computation → Semantics and reasoning

**Keywords and phrases** Coalgebra, Bialgebra, Structural operational semantics, Compositionality

**Digital Object Identifier** 10.4230/LIPIcs.CALCO.2023.7

**Funding** *Harald König*: The author thanks the University of Bergen for support of this project.

**Acknowledgements** The authors thank the anonymous referees for their helpful suggestions that have helped to improve this article.

## 1 Introduction

In a modular and component-based software architecture of a compound system the individual components interact concurrently. Categorically, these individual state-based components are modelled as *coalgebras*. However, in a landscape of multiple interacting systems these behavioural models are *heterogeneously* typed: There are deterministic or non-deterministic labelled transition systems as well as probabilistic systems, systems with or without termination, with or without output and so on, see [24], Chapter 3. Hence the coalgebras of the individual components are based on different endofunctors.

Reasoning about the correct behaviour of a compound system often requires establishing correctness of each local component and furthermore using theoretical means, which guarantee that global behaviour is determined by local behaviours. In [9], this modular method is called *compositionality* and a precise formulation of it requires the use of a framework, which captures the operational semantics of concurrent processes. Such a framework is given by transition rules of structural operational semantics (SOS). Conditional rules of the form

$$\frac{x \xrightarrow{a} x' \quad y \xrightarrow{b} y'}{\text{op}(x, y) \xrightarrow{c} \text{op}(x', y')} \quad (1)$$



© Harald König, Uwe Wolter, and Tim Kräuter;

licensed under Creative Commons License CC-BY 4.0

10th Conference on Algebra and Coalgebra in Computer Science (CALCO 2023).

Editors: Paolo Baldan and Valeria de Paiva; Article No. 7; pp. 7:1–7:17

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

generate systems, whose states are closed terms over an *algebraic* signature [1]. Well-known rule formats are GSOS<sup>1</sup> [3] and tyft/tyxt [8]. For some of these rule formats one can prove compositionality to hold, whereas counterexamples can be provided for other formats [8].

In this paper, we propose a formal structure, which describes the composition of *heterogeneously typed* coalgebras with the help of structural operational semantics. For this, it is important to provide a suitable rule format, which guarantees compositionality (and hopefully other similar requirements) in heterogeneous environments. Since we deal with supposedly arbitrarily varying behavioural specifications, we need more general rule formats, which cannot be expected to be homogeneous like GSOS or tyft/tyxt. An adequate generalization of transition rules in the context of coalgebraic specifications are *natural transformations* between functors, whose domain and codomain reflect the transition from  $n (\geq 2)$  local systems to one compound system, i.e., functors of type  $\mathcal{SET}^n \rightarrow \mathcal{SET}$ . We will show that these so-called *interaction laws* (see Def. 12) can be embedded into distributive laws

$$\lambda : \vec{\Sigma} \vec{\mathcal{B}} \Rightarrow \vec{\mathcal{B}} \vec{\Sigma}$$

for suitable endofunctors  $\vec{\Sigma}$  and  $\vec{\mathcal{B}}$ . Distributive laws are part of a *bialgebraic* approach, which has been described in [14], but was originally proposed by Turi and Plotkin [27]. Here  $\vec{\mathcal{B}}$  (and also  $\vec{\Sigma}$ ) is a  $\mathcal{SET}^{n+1}$ -endofunctor, which simultaneously covers the behaviours of the  $n$  heterogeneously typed coalgebras *and* a specification of the compound system, which has to comprise the commonalities of the local system behaviours. The algebraic syntax functor  $\vec{\Sigma}$  contains the operation(s), which realize(s) the transition from the local components (input of the operation) to the global view of the composed system (output). We evaluate our approach by proving compositionality to hold for interaction laws.

Whereas in process algebras like CCS or CSP<sup>2</sup> this transfer of observational indistinguishability during syntactical build-up of process terms has to be guaranteed [8], we rather want compositionality, when individual software components are composed into a global compound network. Whereas [14] circumscribes compositionality as observational equivalence (w.r.t. final semantics) being a congruence (i.e. the coinductive extension is an algebra homomorphism), we propose a slightly adapted definition tailored to the specific situation of heterogeneously typed interacting systems.

Our work was inspired by practical scenarios, where the coupling of behavioural models with other executable models like test runners or event injectors is of crucial importance [19]. Furthermore, recently, systematic approaches to co-simulation for large-scale system assessment have gained popularity [20]. Here, a typical scenario is the interaction with probabilistic systems [2, 26], which requires a concrete language for their interaction [5, 19]. While these DSLs<sup>3</sup> are already well-established, they lack theoretical underpinning in the form of transition rules to reason about correctness properties.

Hence, we answer the main question

*How can we apply (parts of) the bialgebraic theory to understand the interaction of heterogeneously typed behavioural components?*

by providing the following contributions and novelties:

- A proof for the preservation of observational equivalence, when  $n$  local components are based on *different* behavioural specifications  $\mathcal{B}_1, \dots, \mathcal{B}_n$ , by embedding interaction laws into distributive laws.

<sup>1</sup> *General Structured Operational Semantics*

<sup>2</sup> Calculus of Communicating Systems [21], Communicating Sequential Processes [10]

<sup>3</sup> Domain Specific Languages

- An ensemble of  $n$  separated individual components together with the specification of its composition is formalised in one holistic *many-sorted* approach, i.e., as coalgebras for an endofunctor  $\vec{\mathcal{B}} : \mathcal{SET}^{n+1} \rightarrow \mathcal{SET}^{n+1}$ .

The paper is organized as follows: Sect. 2 clarifies notation, Sect. 3 presents the general setting based on practical scenarios as well as a motivating example, Sect. 4 recapitulates the survey [14] in some detail to make the content complete and comprehensible, and Sect. 5 presents the above mentioned novelties in detail: The linkage of the definitions of Interaction Law (in Def.12) and Congruence (adapted to the heterogeneous case in Def. 17) yield an adequate definition of compositionality and we can obtain our main statements: Theorem 21 proves compositionality to hold for interaction laws and Corollary 22 adapts the statement of the theorem to practical needs.

## 2 Basic Notation

We use the following notations:  $\mathcal{SET}$  is the category of sets and total mappings. For two sets  $A$  and  $X$  we write  $X^A$  for the set of all total maps from  $A$  to  $X$ . A special set is  $1$ , which denotes any singleton set, e.g.  $(1 + X)^A$  is the set of all partial maps from  $A$  to  $X$ .

For functors we will use calligraphic letters like  $\mathcal{F}$ ,  $\mathcal{G}$ , and, especially, letter  $\mathcal{B}$  for behavioural and greek letter  $\Sigma$  for algebraic specifications. Categories are denoted  $\mathbb{C}$  or  $\mathbb{D}$ , an application of a functor  $\mathcal{F} : \mathbb{C} \rightarrow \mathbb{D}$  will be written  $\mathcal{F}(X)$  for  $X \in |\mathbb{C}|$  (or short  $X \in \mathbb{C}$ ), the collection of objects of  $\mathbb{C}$ , whereas an application of  $\mathcal{F}$  to a morphism  $\alpha : X \rightarrow Y$  does not use parentheses:  $\mathcal{F}\alpha : \mathcal{F}(X) \rightarrow \mathcal{F}(Y)$ . Composition of functors  $\mathcal{F}$  and  $\mathcal{G}$  (if it is possible) is always written  $\mathcal{G}\mathcal{F}$  ( $\mathcal{G}$  applied after  $\mathcal{F}$ ). Special functors are  $\mathcal{ID}_{\mathbb{C}}$ , the identity functor on  $\mathbb{C}$ , and  $\wp_{fin}$ , the powerset functor assigning to a set the set of its finite subsets.

It is often convenient to give a definition (on objects) of a functor without explicitly naming its formal parameters, e.g. a functor  $\mathcal{B}$  mapping a set  $X$  to the set  $(1+X)^A$  (see above) is often denoted  $\mathcal{B} = (1 + \_)^A$ . Furthermore, when we give the complete definition of functors  $\mathcal{F}$ , we often combine object and morphism mapping by writing  $X \xrightarrow{f} Y \mapsto \mathcal{F}(X) \xrightarrow{\mathcal{F}f} \mathcal{F}(Y)$ .

As usual, a natural transformation  $\nu$  between functors  $\mathcal{F}$  and  $\mathcal{G}$  with common domain and codomain, written  $\nu : \mathcal{F} \Rightarrow \mathcal{G}$ , is a family  $(\nu_X : \mathcal{F}(X) \rightarrow \mathcal{G}(X))_{X \in |\mathbb{C}|}$  compatible with morphism mapping. For appropriate functors  $\mathcal{H}$  and  $\mathcal{H}'$  we denote with  $\mathcal{H}\nu$  the family  $(\mathcal{H}\nu_X : \mathcal{H}\mathcal{F}(X) \rightarrow \mathcal{H}\mathcal{G}(X))_{X \in |\mathbb{C}|}$  and with  $\nu_{\mathcal{H}'}$  the family  $(\nu_{\mathcal{H}'(X)} : \mathcal{F}\mathcal{H}'(X) \rightarrow \mathcal{G}\mathcal{H}'(X))_{X \in |\mathbb{C}|}$ .

For an endofunctor  $\mathcal{B} : \mathbb{C} \rightarrow \mathbb{C}$  a  $\mathcal{B}$ -coalgebra is a  $\mathbb{C}$ -morphism  $X \xrightarrow{\alpha} \mathcal{B}(X)$ , called the structure map and written  $(X, \alpha)$  or - if  $X$  is clear from the context - just  $\alpha$ . A coalgebra morphism from  $(X, \alpha)$  to  $(Y, \beta)$  is a  $\mathbb{C}$ -morphism  $f : X \rightarrow Y$  such that  $\beta \circ f = \mathcal{B}f \circ \alpha$ . Instead of  $f$  we sometimes write  $(f, \mathcal{B}f)$  to stress the fact that commutativity involves  $\mathcal{B}f$ , as well. The resulting category of all coalgebras for  $\mathcal{B} : \mathbb{C} \rightarrow \mathbb{C}$  will be denoted  $\mathcal{B}\text{-Coalg}$ . If it admits a final object  $(Z, \zeta)$  and if  $(X, \alpha) \in \mathcal{B}\text{-Coalg}$ , we denote with  $u_\alpha : X \rightarrow Z$  the *coinductive extension* of  $\alpha$ , i.e. the unique  $\mathcal{B}\text{-Coalg}$ -morphism into the final object.

Likewise for an endofunctor  $\Sigma : \mathbb{C} \rightarrow \mathbb{C}$  a  $\Sigma$ -algebra is a  $\mathbb{C}$ -morphism  $\Sigma(X) \xrightarrow{a} X$  written  $(a, X)$ . An algebra morphism from  $(a, X)$  to  $(b, Y)$  is a  $\mathbb{C}$ -morphism  $f : X \rightarrow Y$  such that  $b \circ \Sigma f = f \circ a$ . Instead of  $f$  we sometimes write  $(\Sigma f, f)$ . The resulting category of all algebras will be denoted  $\Sigma\text{-Alg}$ .

For morphisms in combination with cartesian products, we use the following notations: If  $f : A \rightarrow B$  and  $g : A \rightarrow B'$ , then  $\langle f, g \rangle : A \rightarrow B \times B'$  denotes the uniquely determined resulting morphism. Likewise for  $g : A' \rightarrow B'$ ,  $f \times g : A \times A' \rightarrow B \times B'$ .

### 3 General Setting and Example

Multiple interacting components of software architectures collectively realize the requirements of business domains. Describing the interactions between these systems and checking their global behavioural consistency is a general, well-known challenge in software engineering [4]. To address this challenge, model-driven software engineering utilizes abstract representations of the constituting systems and their interactions. Such a setting thus consists of an ensemble of *heterogeneously structured* components, which must guarantee the desired global behaviour.

In the sequel, we will speak of *local* or *individual* components, which are assembled into a *global* or *compound* system. As in [24], "system" is also used as a superordinate term for all kinds of artifacts, whether they are composite or not.

Using a general and formal coordination language for the interaction of behavioural components in the form of transition rules requires agreement on key concepts of behavioural systems. It turns out that the concepts "State" and (observational) "State Change" are common to almost all behavioural specifications, cf. the introductory remarks of [12]. Coalgebras  $(X, \alpha)$  for some endofunctor  $\mathcal{B} : \mathbb{C} \rightarrow \mathbb{C}$  on some category  $\mathbb{C}$  comprise exactly these concepts: The structure map  $\alpha$  assigns to each  $x$  in the state space  $X$  the observable causality exhibited in state  $x$ . The different natures of causalities (behaviour) are specified by different endofunctors  $\mathcal{B}$ .

Towards a formal underpinning for the described setting, we need to understand how aligning individual components by specifying their interactions on the one hand, and automatic generation (computation) of global execution behaviour of the compound system, on the other hand, are carried out. For this, we assume  $n$  behavioural specifications  $\mathcal{B}_i : \mathcal{SET} \rightarrow \mathcal{SET}$  to be given for some  $n \geq 2$  and fix individual behavioural systems  $(S_i, \alpha_i) \in \mathcal{B}_i\text{-Coalg}$ .

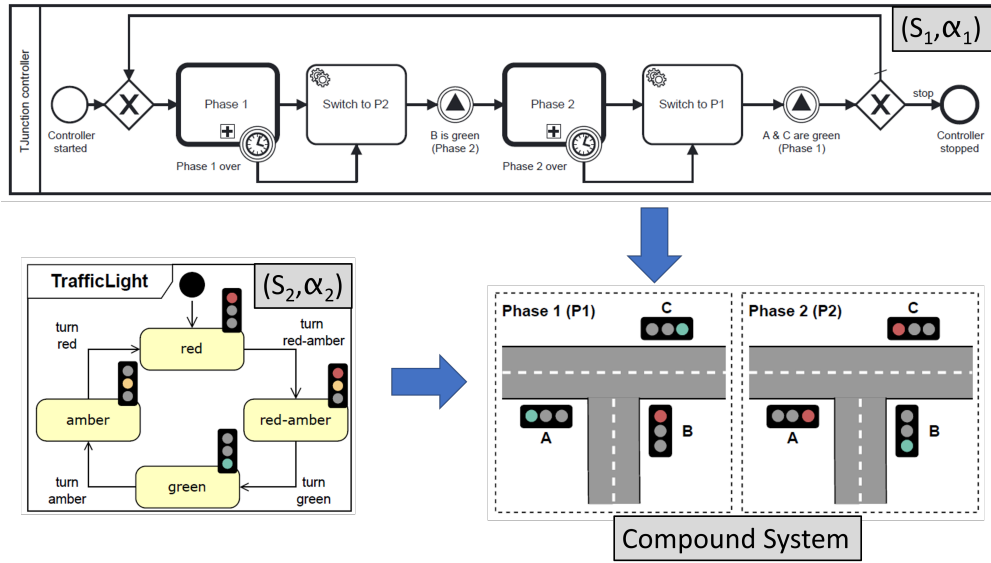
As an example, we refer to the use case depicted in Fig. 1, where an instance of a **T-Junction-Controller** regulates the interaction of three **TrafficLights** A, B, and C. The T-junction controller (component  $(S_1, \alpha_1)$ ) and the behaviour of one traffic light (e.g., component  $(S_2, \alpha_2)$ ) are shown in the top and the bottom left part of Fig. 1. The resulting compound system is hinted at in the bottom right part. The operation, which takes as input the local components and "generates" the semantics of the compound system is visualized by arrows between the systems (blue in a colour display).

Whereas each traffic light is specified as a labelled transition system, the **TJunction** may be modelled as a BPMN<sup>4</sup>-model. The BPMN model specifies different phases to handle ( $P1$  and  $P2$ ). They are shown in the BPMN model and also in the two different snapshots of the compound system. The interaction with approaching vehicles may be modelled with a third formalism, e.g., a probabilistic transition system, which simulates exponentially distributed arrivals of buses or cars at one of the traffic lights. Aligning individual components by means of coordination languages, cf. [5], requires specifying coordination points (communication channels), e.g., if a request  $e$  of some approaching bus triggers the switch to phase 2 in the **TJunction** controller (an observation  $o$ ), this transition must synchronize with input  $i = \text{turnRed}$  of traffic light A and C. Moreover, B must simultaneously turn green. These synchronisations can be formalized with *synchronization algebras*, cf. [22], in this simple case, a partial map  $\varphi : O \times I \rightarrow \text{Act}$ , where  $O$  is the set of outputs of the controller like throw events or service calls in automatic tasks,  $I$  is the set of possible inputs to the respective traffic light, and  $\text{Act}$  is the set of observable actions of the compound system.

---

<sup>4</sup> Business Process Model and Notation





■ **Figure 1** TJunction traffic control system and its individual components.

The global execution behaviour can be described by *premises* (transitions of the local components) and *conclusions* (the resulting actions taken by the compound system). If, for instance,  $x \xrightarrow{e/o} x'$  in the BPMN-model and  $y \xrightarrow{i} y'$  are possible, then  $z \xrightarrow{\varphi(o,i)} z'$  is a global interaction of the components. Note that we obtain a respective conditional rule, but with different formats in its premises and conclusions: There is the Mealy-like notation in the first premise specifying output  $o$  in the BPMN process, when event  $e$  occurs, whereas the second premise specifies that the labelled transition system behaves like  $y'$ , if, in state  $y$ , input  $i$  occurred. Furthermore, the compound system may be non-deterministic, such that the conclusion reads, “The system *may* behave like  $z'$ , if in state  $z$ , action  $\varphi(o, i)$  was performed”.

Formally the interaction operation, which takes as input  $n$  states of the local components and outputs a state of the compound system, is based on an  $n$ -ary operation symbol  $\text{op} = \text{interact} : s_1 s_2 \dots s_n \rightarrow s_{n+1}$  of a suitable algebraic signature  $\Sigma$ , where sorts  $s_1, s_2, \dots$  reflect the structurally separated but interacting local components, and the compound system is based on a new behavioural specification  $\mathcal{B}$  and requires a new sort  $s_{n+1} \notin \{s_1, \dots, s_n\}$ .

We summarise the transfer from practical concepts to the bialgebraic formalism:

1. The individual components are based on behavioural endofunctors  $\mathcal{B}_1, \dots, \mathcal{B}_n$  and the compound system's behaviour is specified by another endofunctor  $\mathcal{B}$ . The individual components are coalgebras  $(S_i, \alpha_i) \in \mathcal{B}_i\text{-Coalg}$ , from which the states of the compound system  $(S, \alpha) \in \mathcal{B}\text{-Coalg}$  arise as output of an application of an  $n$ -ary algebraic operation  $\text{op}$  which has as input the states of the individual components.
2. The semantics of the compound system is formalized by SOS rules of the form

$$\frac{x_1 \xrightarrow{E_1} x'_1 \dots x_n \xrightarrow{E_n} x'_n}{\text{op}(x_1, \dots, x_n) \xrightarrow{F} \text{op}(x'_1, \dots, x'_n)}$$

where  $E_i$  and  $F$  are differently structured terms over the coordination points, and  $x_i, x'_i$  are states of the individual component  $(S_i, \alpha_i)$  for all  $i \in \{1, \dots, n\}$ .

#### 4 Background: Distributive Laws and Bialgebras

In this section, we recapitulate parts of [14] which are necessary to make the content of Sect. 5 complete and comprehensible. As in [18], we extend behavioural specifications  $\mathcal{B}$  only to copointed coalgebras (see Def. 2 below), i.e. we consider the assignment  $X \mapsto X \times \mathcal{B}(X)$  instead of  $\mathcal{B}(X)$  in order to be able to use current states in the formulation of the conclusion of SOS rules. However, we do not generalise it further, i.e. we neither make use of free extensions for the algebraic specification functor as in the original work [27] nor cofree extensions of the behaviour functor, cf. [14].

Let  $\mathbb{C}$  be an arbitrary category with finite products. The classical theory works with one fixed behavioural specification  $\mathcal{B} : \mathbb{C} \rightarrow \mathbb{C}$  and an algebraic specification  $\Sigma : \mathbb{C} \rightarrow \mathbb{C}$  which usually specifies the syntactical assembly of process terms (such as prefixing, alternative, and parallel, as well as interaction). In contrast to that, in Sect. 5, we will use  $\Sigma$  for the assembly of the compound system from the local components. Furthermore, let's define the functor

$$\mathcal{H} : \begin{cases} \mathbb{C} & \rightarrow & \mathbb{C} \\ X \xrightarrow{f} Y & \mapsto & X \times \mathcal{B}(X) \xrightarrow{f \times \mathcal{B}f} Y \times \mathcal{B}(Y). \end{cases} \quad (2)$$

Pairs  $(\mathcal{H} : \mathbb{C} \rightarrow \mathbb{C}, \varepsilon : \mathcal{H} \Rightarrow ID_{\mathbb{C}})$  are usually called *copointed* functors in the literature, e.g. [14], i.e.  $\mathcal{H}$  comes equipped with a comonadic "counit"  $\varepsilon$ . In our particular definition  $\mathcal{H}$  is accompanied with counit  $\pi_1 : \mathcal{H} \Rightarrow ID_{\mathbb{C}}$ , where  $\pi_1 = ((\pi_1)_X : X \times \mathcal{B}(X) \rightarrow X)$  is the componentwise first projection. We will use only these special copointed functors.

► **Definition 1** (Distributive Law over  $\mathcal{H}$ ). *A Distributive Law of  $\Sigma$  over  $\mathcal{H}$  is a natural transformation*

$$\lambda : \Sigma\mathcal{H} \Rightarrow \mathcal{H}\Sigma$$

*which is compatible with the counit, i.e. such that  $(\pi_1)_{\Sigma} \circ \lambda = \Sigma\pi_1 : \Sigma\mathcal{H} \Rightarrow \Sigma$ .* ┘

The extension from the original behavioural specification functor  $\mathcal{B}$  to  $\mathcal{H}$  also requires to consider special coalgebras for  $\mathcal{H}$  [18]:

► **Definition 2** (Copointed  $\mathcal{H}$ -Coalgebra). *Let  $\mathcal{H}$  be given as above. The category of copointed  $\mathcal{H}$ -coalgebras, written  $\mathcal{H}\text{-Coalg}_{co}$ , is the full subcategory of  $\mathcal{H}\text{-Coalg}$  with those objects  $(X, \alpha)$  satisfying  $(\pi_1)_X \circ \alpha = id_X$ .* ┘

► **Proposition 3** (Copointed  $\mathcal{H}$ -Coalgebras are  $\mathcal{B}$ -Coalgebras). *The assignment  $(X, \alpha) \mapsto (X, (\pi_2)_X \circ \alpha)$  extends to an isomorphism between categories  $\mathcal{H}\text{-Coalg}_{co}$  and  $\mathcal{B}\text{-Coalg}$ .* ┘

There is a canonical assignment from distributive laws over  $\mathcal{H}$  to natural transformations

$$\rho : \Sigma\mathcal{H} \Rightarrow \mathcal{B}\Sigma \quad (3)$$

given by  $\lambda \mapsto (\pi_2)_{\Sigma} \circ \lambda$ . Using counit compatibility from Def. 1, the assignment

$$\rho \mapsto \langle \Sigma\pi_1, \rho \rangle \quad (4)$$

turns out to be inverse to the former, see Theorem 10 in [18]. Thus

► **Proposition 4** (Equivalent Representation of Distributive Laws). *The assignments (3) and (4) yield a bijection between distributive laws over  $\mathcal{H}$  and natural transformations  $\rho : \Sigma\mathcal{H} \Rightarrow \mathcal{B}\Sigma$ .* ┘

Note that natural transformations as in (3) are special cases of GSOS laws, where the syntax functor  $\Sigma$  is replaced by its free extension  $\Sigma^*$  in the codomain of  $\rho$ , thus enabling arbitrary terms in the target of the SOS rule conclusion.

Because  $\lambda$  is a natural transformation,

$$\Sigma_\lambda : \begin{cases} \mathcal{H}\text{-Coalg}_{co} & \rightarrow & \mathcal{H}\text{-Coalg}_{co} \\ (X, h) & \mapsto & (\Sigma(X), \lambda_X \circ \Sigma h) \end{cases}$$

and its dual construction

$$\mathcal{H}^\lambda : \begin{cases} \Sigma\text{-Alg} & \rightarrow & \Sigma\text{-Alg} \\ (g, X) & \mapsto & (\mathcal{H}g \circ \lambda_X, \mathcal{H}(X)) \end{cases}$$

extend to endofunctors, where the first indeed maps to  $\mathcal{H}\text{-Coalg}_{co}$  by the compatibility of counits in Def. 1.  $\Sigma_\lambda$  applied to a coalgebra yields behaviour of algebraically composed states and will play a major role in Sect. 5.  $\mathcal{H}^\lambda$  will be used only in the present section.

For a distributive law  $\lambda$ , there is a new category, which yields a combination of operational and denotational models w.r.t. functors  $\mathcal{B}$  (and thus  $\mathcal{H}$ ) and  $\Sigma$ :

► **Definition 5** (Category of  $\lambda$ -Bialgebras). *Let  $\lambda : \Sigma\mathcal{H} \Rightarrow \mathcal{H}\Sigma$  be a distributive law according to Def.1. The category  $\lambda$ -Bialg has objects arrow-pairs  $\Sigma(X) \xrightarrow{g} X \xrightarrow{h} \mathcal{H}(X)$  with copointed  $(X, h)$  and for which*

$$\mathcal{H}g \circ \lambda_X \circ \Sigma h = h \circ g \tag{5}$$

Morphisms are those  $f : X \rightarrow Y$ , which are simultaneously  $\Sigma\text{-Alg}$ - and  $\mathcal{H}\text{-Coalg}_{co}$ -morphisms.

Using (5), one obtains

► **Proposition 6** ([14], Prop. 12). *There are the isomorphisms*

$$\Sigma_\lambda\text{-Alg} \cong \lambda\text{-Bialg} \cong \mathcal{H}^\lambda\text{-Coalg}$$

where e.g. the second one is based on the assignment

$$(\Sigma(X) \xrightarrow{g} X \xrightarrow{h} \mathcal{H}(X)) \mapsto ((g, X) \xrightarrow{(\Sigma h, h)} \mathcal{H}^\lambda(g, X))$$

on objects of the respective categories. ┘

A consequence of this fact is the following proposition, for which we include a proof, because we need parts of it in Sect. 5:

► **Proposition 7** (Initial and Final Bialgebras, [14], Sect. 4.3). *If  $\Sigma$  admits an initial algebra  $(a, A)$  and if  $\mathcal{H}\text{-Coalg}_{co}$  has the final (copointed) coalgebra  $(Z, \zeta)$ , then the former uniquely extends to an initial object  $\Sigma(A) \xrightarrow{a} A \xrightarrow{h_\lambda} \mathcal{H}(A)$  of  $\lambda$ -Bialg and the latter uniquely extends to a final object  $\Sigma(Z) \xrightarrow{g^\lambda} Z \xrightarrow{\zeta} \mathcal{H}(Z)$  of  $\lambda$ -Bialg.*

**Proof.** By Prop. 6 we can look for an initial object in  $\mathcal{H}^\lambda\text{-Coalg}$ . But for any endofunctor  $\overline{\mathcal{H}} : \mathbb{D} \rightarrow \mathbb{D}$  the carrier of the initial object in  $\overline{\mathcal{H}}\text{-Coalg}$  is just the initial object in  $\mathbb{D}$ , if it exists. Hence for  $\overline{\mathcal{H}} = \mathcal{H}^\lambda$  and  $\mathbb{D} = \Sigma\text{-Alg}$ , we obtain the initial  $\mathcal{H}^\lambda\text{-Coalg}$ -object  $a \xrightarrow{(\Sigma h_\lambda, h_\lambda)} \mathcal{H}^\lambda a$ , where  $h_\lambda : A \rightarrow \mathcal{H}(A)$  is the unique  $\Sigma\text{-Alg}$ -morphism out of the initial object. By the definition of  $\mathcal{H}^\lambda$  this yields the commutative diagram

$$\begin{array}{ccc} \Sigma(A) & \xrightarrow{a} & A \\ \Sigma h_\lambda \downarrow & & \downarrow h_\lambda \\ \Sigma \mathcal{H}(A) & \xrightarrow{\mathcal{H}a \circ \lambda_A} & \mathcal{H}(A) \end{array} \tag{6}$$

## 7:8 SOS for Heterogeneously Typed Coalgebras

turning  $\Sigma(A) \xrightarrow{a} A \xrightarrow{h_\lambda} \mathcal{H}(A)$  into a  $\lambda$ -*Bialg*-object (because  $h_\lambda$  is copointed by the following Prop. 8) but also the *initial*  $\lambda$ -*Bialg*-object due to the assignment given in Prop. 6. The unique extension of the final object is dually obtained yielding the final  $\lambda$ -bialgebra  $\Sigma(Z) \xrightarrow{g^\lambda} Z \xrightarrow{\zeta} \mathcal{H}(Z)$  for the unique  $\mathcal{H}$ -*Coalg*<sub>co</sub>-morphism  $g^\lambda$  to the final object. ◀

Using counit compatibility of  $\pi_1$  in Def. 1 and initiality of  $(a, A)$ , one also obtains

► **Proposition 8** (Copointedness of  $h_\lambda$ ).  *$h_\lambda$  is a copointed  $\mathcal{H}$ -coalgebra.* ◻

The initial and final bialgebras from the proof of Prop. 7 yield a unique arrow  $f : A \rightarrow Z$  in the commutative diagram

$$\begin{array}{ccccc} \Sigma(A) & \xrightarrow{a} & A & \xrightarrow{h_\lambda} & \mathcal{H}(A) \\ \Sigma f \downarrow & & \downarrow f & & \downarrow \mathcal{H}f \\ \Sigma(Z) & \xrightarrow{g^\lambda} & Z & \xrightarrow{\zeta} & \mathcal{H}(Z) \end{array} \quad (7)$$

$f$  is simultaneously the coinductive extension of  $h_\lambda$ , i.e. its behavioural semantics, and the inductive extension of  $g^\lambda$ , i.e. the evaluation of  $\Sigma$ -terms in  $(g^\lambda, Z)$ . The former statement is the important one. It gives the key statement of this section:

► **Observation 9.** The coinductive extension of copointed  $h_\lambda$  is an algebra homomorphism from the initial  $\Sigma$ -algebra. ◻

## 5 From Local Components to Compound Systems Coalgebraically

### 5.1 The Theoretical Setting

Let  $(S_1, \alpha_1) \in \mathcal{B}_1\text{-Coalg}, \dots, (S_n, \alpha_n) \in \mathcal{B}_n\text{-Coalg}$  be  $n$  individual, local components and  $\mathcal{B}$  a behavioural specification for the compound system, see item 1 in the summary on page 5. Related local components' interactions are based on an  $n$ -ary operation symbol  $\text{op} := \text{interact}$  and coordination points of  $\mathcal{B}_1, \dots, \mathcal{B}_n$  together with a synchronisation algebra<sup>5</sup>  $\varphi$  establish transition rules, cf. item 2 of the summary.

► **Example 10** (See Sect. 3). A BPMN model can be encoded with the functor

$$\mathcal{B}_1 = (1 + O \times \_)^E,$$

where  $E$  are state-changing events like a timer event in the Tjunction Controller, cf. Fig. 1. The set  $O$  defines outputs, e.g.,  $\text{SwitchToP2} \in O$ , which must be synchronized with a call to a traffic light to turn red. Let  $(S_1, \alpha_1) \in \mathcal{B}_1\text{-Coalg}$  be such a component. Traffic lights are deterministic labelled transition systems based on

$$\mathcal{B}_2 = (1 + \_)^I,$$

where, for instance,  $I = \{\mathbf{A}.\text{turnRed}, \mathbf{A}.\text{turnGreen}, \dots\}$  is the input set  $I$  of traffic light  $\mathbf{A}$ . Let  $(S_2, \alpha_2) \in \mathcal{B}_2\text{-Coalg}$  be such a component.

<sup>5</sup> For arbitrary  $n$ , these synchronisation descriptions will no longer be binary.

We define  $Act := O + I + \{\tau\}$  and expect the compound system to change state depending on the used coordination points<sup>6</sup>. For this, we extend the involved sets by an idle action  $*$ , i.e.  $X_* := X + \{*\}$  for  $X \in \{O, I, Act\}$ , cf. [22]. As usual, a transition with  $*$  from state  $x$  to  $x'$  is possible, if and only if  $x = x'$ . We assume a synchronisation algebra  $\varphi : O_* \times I_* \rightarrow Act_*$  for the synchronisation of a BPMN model with one traffic light. In the above example, we might have  $\varphi(\text{SwitchToP2}, \text{A.turnRed}) = \tau$  (modelling a silent synchronisation), whereas some other value combinations like  $\varphi(\text{SwitchToP2}, \text{A.turnGreen})$  are undefined. Whenever an output  $o$  is *uncoupled*, i.e., whenever the first component can evolve independently from the second for a transition with  $o$ , we let  $\varphi$  be undefined for pairs  $(o, i)$  for all  $i \in I$ , and define  $\varphi(o, *) := o$ . This is true, for example, if  $o$  is an outgoing signal like "B is green", which does not change the state of any traffic light, cf. Fig. 1. Similarly  $\varphi$  is undefined for  $(o, i)$  for all  $o \in O$  and  $\varphi(*, i) = i$  for uncoupled  $i$ . Finally,  $\varphi(o, i) = * \iff o = i = *$ , cf. [22].

The underlying algebraic signature will have three sorts:  $s_1$  and  $s_2$  for the states of the two local components and  $s_3$  for the compound system. Because resulting transitions can be silent for different coordinations and hence result in non-determinism of the compound system, we define

$$\mathcal{B} := \wp_{fin}(Act \times \_) \quad \lrcorner$$

The original work of [27] shows a one-to-one correspondence between sets of GSOS laws and natural transformations. We will show how we can follow this approach along the above-stated example. When we mention SOS rules, we exemplarily use notations in the context of our examples. Furthermore, whenever we write down  $\varphi(o, i)$ , we automatically assume this value to be defined.

The family of SOS-rules

$$\left( \frac{x \xrightarrow{e/o} x' \quad y \xrightarrow{i} y'}{op(x, y) \xrightarrow{\varphi(o, i)} op(x', y')} \right)_{o \in O_*, i \in I_*} \quad (8)$$

describes the operational semantics of the compound system as a *heterogenous* interaction law. E.g., the controller's command to make traffic light A change to red is the law for  $b := \text{SwitchToP2}$  and  $i = \text{A.turnRed}$ .

In the example, the state space  $S$  of the compound system must take into consideration the original state spaces by pairing  $S_1$  and  $S_2$ , i.e., in this example

$$S = S_1 \times S_2. \quad (9)$$

Of course, in the general case,  $S$  can depend arbitrarily on the state spaces  $S_1, \dots, S_n$  of the local components.

## 5.2 Interaction Laws and Induced Coalgebra

In this section, we formalize the construction of the compound system from the local components, if its operational semantics is given as an SOS rule like in (8). This rule does not depend on concrete state spaces, hence it can be seen as an interaction law between

<sup>6</sup> As usual,  $\tau$  models silent (unobservable) transitions.

## 7:10 SOS for Heterogeneously Typed Coalgebras

systems of arbitrary state spaces  $X$  and  $Y$ . We claim that it can be encoded as a map, which decomposes into two factors, the first reflecting the premises given by the transitions of  $\alpha_1$  and  $\alpha_2$ , and a second factor  $\rho_{X,Y}$ , which reflects the conclusions:

$$X \times Y \xrightarrow{\langle id, \alpha_1 \rangle \times \langle id, \alpha_2 \rangle} X \times \mathcal{B}_1(X) \times Y \times \mathcal{B}_2(Y) \xrightarrow{\rho_{X,Y}} \mathcal{B}(X \times Y) \quad (10)$$

We first define  $\rho_{X,Y}$  in the context of our example:

► **Example 11** (Example 10 ctd). Recall that we call  $o \in O$  *coupled*, if there is  $i \in I$  such that  $\varphi(o, i)$  is defined and vice versa for  $i \in I$ . Otherwise, it is called *uncoupled*. Then, for  $\mathcal{B}_1 = (1 + O \times \_)^E$ ,  $\mathcal{B}_2 = (1 + \_)^I$ , and  $\mathcal{B} = \wp_{fin}(Act \times \_)$ , we can define

$$\begin{aligned} \rho_{X,Y}(x, f_1, y, f_2) &= \{(\varphi(o, i), (x', y')) \mid o \neq * \neq i, (o, x') \in f_1(E), y' = f_2(i)\} \\ &\cup \{(o, (x', y)) \mid (o, x') \in f_1(E), o \text{ uncoupled}\} \\ &\cup \{(i, (x, y')) \mid y' = f_2(i), i \text{ uncoupled}\} \end{aligned}$$

where  $f_1 : E \rightarrow 1 + O \times X$  and  $f_2 : I \rightarrow 1 + Y$ . It is easy to see that  $(\rho_{X,Y})_{(X,Y) \in |\mathcal{SET}^2|}$  is natural in its parameters  $X$  and  $Y$ .  $\lrcorner$

As in the classical theory, natural transformations as in Example 10 can now be used to *define* SOS-rules. For this let's define the copointed versions  $\mathcal{H}_i := \mathcal{SET} \rightarrow \mathcal{SET}$  of the functors  $\mathcal{B}_i$  as in (2) for  $i \in \{1, \dots, n\}$ . The special assignment  $(S_1, S_2) \mapsto S_1 \times S_2$  from (9) extends to a functor  $\Sigma : \mathcal{SET}^2 \rightarrow \mathcal{SET}$  which yields natural transformation  $\rho : \Sigma(\mathcal{H}_1 \times \mathcal{H}_2) \Rightarrow \mathcal{B}\Sigma : \mathcal{SET}^2 \rightarrow \mathcal{SET}$  in Example 11. However, we don't want to exclude additional dependencies, when constructing states of the compound system. E.g. additional supervising components or intermediate components like message queues may let the overall state space differ from the pure cartesian product of the local state spaces. Hence, we are interested in an arbitrary functor  $\Sigma : \mathcal{SET}^n \rightarrow \mathcal{SET}$  for some  $n \geq 2$  and correspondingly adapted natural transformations. Thus the appropriate definition in our context is

► **Definition 12** (Interaction Law). *Let  $\Sigma : \mathcal{SET}^n \rightarrow \mathcal{SET}$  be an arbitrary functor,  $\mathcal{B}_1, \dots, \mathcal{B}_n$ , and  $\mathcal{B}$  be  $\mathcal{SET}$ -endofunctors, and functors  $\mathcal{H}_i : \mathcal{SET} \rightarrow \mathcal{SET}$  be defined as in (2), i.e.  $\mathcal{H}_i(X) = X \times \mathcal{B}_i(X)$  for all  $X \in \mathcal{SET}$  and all  $i \in \{1, \dots, n\}$ . An interaction law is a natural transformation*

$$\rho : \Sigma(\mathcal{H}_1 \times \dots \times \mathcal{H}_n) \Rightarrow \mathcal{B}\Sigma : \mathcal{SET}^n \rightarrow \mathcal{SET}. \quad \lrcorner$$

Similarly to the definition of  $\Sigma_\lambda$  in Sect. 4, this yields an assignment

$$\Sigma_\rho : \left\{ \begin{array}{l} \mathcal{B}_1\text{-Coalg} \times \dots \times \mathcal{B}_n\text{-Coalg} \longrightarrow \mathcal{B}\text{-Coalg} \\ ((S_1, \alpha_1), \dots, (S_n, \alpha_n)) \mapsto (\Sigma(S_1, \dots, S_n), \rho_{S_1, \dots, S_n} \circ \Sigma(\langle id_{S_1}, \alpha_1 \rangle, \dots, \langle id_{S_n}, \alpha_n \rangle)) \end{array} \right. \quad (11)$$

which becomes a functor, because  $\rho$  is a natural transformation. Any  $n$ -tuple  $(f_1, \dots, f_n)$  with  $f_i$  a  $\mathcal{B}_i$ -Coalg-morphism is mapped by  $\Sigma_\rho$  to the  $\mathcal{B}$ -Coalg-morphism  $\Sigma(f_1, \dots, f_n)$ .

► **Definition 13** ( $\rho$ -Induced Coalgebra). *Given  $(S_1, \alpha_1) \in \mathcal{B}_1\text{-Coalg}, \dots, (S_n, \alpha_n) \in \mathcal{B}_n\text{-Coalg}$  and an interaction law  $\rho$ , the  $\mathcal{B}$ -coalgebra  $\Sigma_\rho((S_1, \alpha_1), \dots, (S_n, \alpha_n))$  is called the  $\rho$ -induced coalgebra of  $(S_1, \alpha_1), \dots, (S_n, \alpha_n)$ . If the carrier sets are clear from the context, we just write  $\Sigma_\rho(\alpha_1, \dots, \alpha_n)$  for the  $\rho$ -induced coalgebra.*  $\lrcorner$

Hence, the  $\rho$ -induced coalgebra is the compound system arising from the local components, when an SOS rule like (8), which is reflected in interaction law  $\rho$ , is applied.

► **Example 14** (Example 10 ctd). In the example, we obtain the desired compound system, a  $\mathcal{B}$ -coalgebra with state space  $S_1 \times S_2$  behaving as specified by the local components and the SOS-laws from (8).  $\lrcorner$

### 5.3 Compositionality

Verification of correctness of composed systems should be guaranteed, if its components are already correct. Moreover, semantics preserving refactorings of local components should also preserve the semantics of the compound system. These behavioural correctness issues are often based on observational equivalence, hence we want observational equivalence to be preserved after the construction of the compound system from the local components. In this section, we formally define these aspects in the context of our setting.

► **Definition 15** (Observational Equivalence). *Let  $\mathcal{F} : \mathcal{SET} \rightarrow \mathcal{SET}$ , such that  $\mathcal{F}$ -Coalg admits a final object  $(Z, \zeta)$ . Let  $(X, \alpha) \in \mathcal{F}$ -Coalg and  $u_\alpha : X \rightarrow Z$  be its coinductive extension. Two states  $x, x' \in X$  are said to be observationally equivalent, written  $x \sim_\alpha x'$ , if  $(x, x')$  is contained in the kernel relation  $\ker(u_\alpha)$ , i.e. if  $u_\alpha(x) = u_\alpha(x')$ .  $\lrcorner$*

For future use, we state the following proposition, which easily follows, because for any  $\mathcal{F}$ -Coalg-morphism  $f : (X, \alpha) \rightarrow (Y, \beta)$ , the coinductive extension satisfies  $u_\alpha = u_\beta \circ f$ :

► **Proposition 16** (Observational Equivalence). *With the same ingredients as in Def. 15, two states  $x_1, x_2 \in X$  are observationally equivalent, if there is an  $\mathcal{F}$ -Coalg-morphism  $f : (X, \alpha) \rightarrow (Y, \beta)$  such that  $f(x_1) = f(x_2)$ .<sup>7</sup>  $\lrcorner$*

Let  $(u_{\alpha_i})_{i \in \{1, \dots, n\}}$  be the coinductive extensions of our local components,

$$\sim_i = \ker(u_{\alpha_i}), \quad (12)$$

and some operation  $\text{op} : S_1 \times \dots \times S_n \rightarrow A$  be given for some state set  $A$  of some  $\mathcal{B}$ -coalgebra. Furthermore, let  $\sim$  be the kernel relation of its coinductive extension, then preservation of observational equivalence under  $\text{op}$  means

$$\forall i \in \{1, \dots, n\} : x_i \sim_i x'_i \Rightarrow \text{op}(x_1, \dots, x_n) \sim \text{op}(x'_1, \dots, x'_n), \quad (13)$$

i.e. observational equivalence is a compatible w.r.t. operation  $\text{op}$ . It is well-known that a general definition of congruence on an algebra  $a : \mathcal{F}(A) \rightarrow A$  for an endofunctor  $\mathcal{F} : \mathbb{C} \rightarrow \mathbb{C}$  is as follows: A monomorphism  $R \xrightarrow{\langle \pi_1, \pi_2 \rangle} A \times A$  is a congruence on  $a$ , if there is an algebra  $r : \mathcal{F}(R) \rightarrow R$ , for which the diagram

$$\begin{array}{ccccc} \mathcal{F}(A) & \xleftarrow{\mathcal{F}(\pi_1)} & \mathcal{F}(R) & \xrightarrow{\mathcal{F}(\pi_2)} & \mathcal{F}(A) \\ a \downarrow & & r \downarrow & & \downarrow a \\ A & \xleftarrow{\pi_1} & R & \xrightarrow{\pi_2} & A \end{array} \quad (14)$$

commutes, cf. Theorem 3.3.5. in [12].

However, in the case of separated heterogeneously typed state sets of the local systems, a general definition of congruence must be based on the above-defined functor  $\Sigma : \mathcal{SET}^n \rightarrow \mathcal{SET}$ .

<sup>7</sup> Note that this proposition can even better be taken as the definition for observational equivalence, because it does not depend on the existence of a final object. We found it, however, more demonstrative to use Def. 15 for it.

► **Definition 17** (*a-compatibility*). Let  $A_1, \dots, A_n, A$  be sets and

$$a : \Sigma(A_1, \dots, A_n) \rightarrow A$$

be a map. Furthermore let  $(R_i \subseteq A_i \times A_i)_{i \in \{1, \dots, n\}}$  and  $R \subseteq A \times A$  be a collection of  $n + 1$  binary relations with projections  $\pi_1^i, \pi_2^i : R_i \rightarrow A_i$  for all  $i$  and  $\pi_1, \pi_2 : R \rightarrow A$ . The relation tuple  $(R_1, \dots, R_n, R)$  is said to be *a-compatible*, if there is a map  $r$ , such that the following diagram commutes:

$$\begin{array}{ccccc} \Sigma(A_1, \dots, A_n) & \xleftarrow{\Sigma(\pi_1^1, \dots, \pi_1^n)} & \Sigma(R_1, \dots, R_n) & \xrightarrow{\Sigma(\pi_2^1, \dots, \pi_2^n)} & \Sigma(A_1, \dots, A_n) \\ \downarrow a & & \downarrow r & & \downarrow a \\ A & \xleftarrow{\pi_1} & R & \xrightarrow{\pi_2} & A \end{array}$$

┘

► **Example 18** (*op-compatibility*). Let  $R_i = \sim_i$  and  $R = \sim$ , cf. (12), then it is easy to see that in the case  $\Sigma(X_1, \dots, X_n) = X_1 \times \dots \times X_n$  *op-compatibility* yields (13). ┘

► **Observation 19.** *a-compatibility* of  $(R_1, \dots, R_n, R)$  can thus be read as an implication: If pairs  $(a_i, a'_i)$  are related via  $R_i$ , then *a-images* of corresponding elements of the set  $\Sigma(A_1, \dots, A_n)$  are related as well. ┘

Of course, the meaning of the term "corresponding" depends on the action of  $\Sigma$ .

It is not self-evident that observational equivalence is compatible with the syntactic structure of process terms in transition rules, see the counterexamples in [8] or violations of compositionality in the context of the  $\pi$ -calculus [21], Chapt. 12.4. However, in our setting, we can prove that interaction laws preserve observational equivalence. Note that this is almost evident in the above example, where  $n = 2$  and  $\Sigma(X, Y) = X \times Y$ , because the image of the pair of coinductive extensions  $u_{\alpha_1} : S_1 \rightarrow \dots$  and  $u_{\alpha_2} : S_2 \rightarrow \dots$  of functor  $\Sigma_\rho$  is the  $\mathcal{B}$ -coalgebra-morphism  $u = u_{\alpha_1} \times u_{\alpha_2}$ , for which  $((x_1, x_2), (x'_1, x'_2)) \in \ker(u_{\alpha_1} \times u_{\alpha_2})$ , if  $(x_1, x'_1) \in \ker(u_{\alpha_1})$  and  $(x_2, x'_2) \in \ker(u_{\alpha_2})$ , which yields the desired result by Prop. 16.

The proof idea for the general case is to keep the local state spaces and the state space for the compound system separated as systems in their own right by assigning different sorts of the underlying algebraic specification to them and then apply (7) (i.e. observation 9). For this, we must formalise the whole setting of system components and their interaction in one holistic *many-sorted* approach as follows. Recall that we assume  $(S_1, \alpha_1) \in \mathcal{B}_1\text{-Coalg}, \dots, (S_n, \alpha_n) \in \mathcal{B}_n\text{-Coalg}$  to be  $n$  individual, local components, then we define the endofunctor

$$\vec{\Sigma} : \begin{cases} \mathcal{SET}^{n+1} & \rightarrow & \mathcal{SET}^{n+1} \\ (X_1, \dots, X_n, X_{n+1}) & \mapsto & (S_1, \dots, S_n, \Sigma(X_1, \dots, X_n)) \end{cases}$$

with  $\vec{\Sigma}(h_1, \dots, h_n, h_{n+1}) := (\underbrace{id, \dots, id}_{n \text{ times}}, \Sigma(h_1, \dots, h_n))$  on function tuples. Intuitively, we define

an algebraic signature with sorts  $s_1, \dots, s_n, s_{n+1}$  and "constants" of sort  $s_i$  the elements of  $S_i$  (for  $1 \leq i \leq n$ ), as well as operation symbols with codomain  $s_{n+1}$ . Thus the term algebra has carrier sets  $S_1, \dots, S_n$ , whereas the carrier of sort  $s_{n+1}$  comprises all terms arising from a single application of an operation symbol. We obtain

► **Proposition 20** (Initial Object of  $\vec{\Sigma}$ ).  $\vec{\Sigma}\text{-Alg}$  possesses an initial object with carrier  $\mathbf{0} := (S_1, \dots, S_n, \Sigma(S_1, \dots, S_n)) = \vec{\Sigma}(\mathbf{0})$  and structure map  $id_{\mathbf{0}} : \vec{\Sigma}(\mathbf{0}) \rightarrow \mathbf{0}$ .



**Proof.** Given a  $\vec{\Sigma}$ -algebra  $(f_1, \dots, f_n, f_{n+1}) : \vec{\Sigma}(X_1, \dots, X_n, X_{n+1}) \rightarrow (X_1, \dots, X_n, X_{n+1})$ , it is easy to see that

$$\mathbf{0} \xrightarrow{(f_1, \dots, f_n, f_{n+1} \circ \Sigma(f_1, \dots, f_n))} (X_1, \dots, X_n, X_{n+1})$$

establishes the unique algebra homomorphism from  $id_{\mathbf{0}}$  to the given algebra.  $\blacktriangleleft$

► **Theorem 21** (Interaction Laws preserve Observational Equivalence). *Let  $\mathcal{B}_1, \dots, \mathcal{B}_n$ , and  $\mathcal{B}$  be  $n+1$   $\mathcal{SET}$ -endofunctors, such that all corresponding categories of coalgebras admit a final coalgebra. Let  $(S_1, \alpha_1) \in \mathcal{B}_1\text{-Coalg}, \dots, (S_n, \alpha_n) \in \mathcal{B}_n\text{-Coalg}$  and  $u_{\alpha_1}, \dots, u_{\alpha_n}$  be their coinductive extensions. For functor  $\Sigma : \mathcal{SET}^n \rightarrow \mathcal{SET}$  let an interaction law  $\rho$  be given as in Def. 12, and  $\Sigma_\rho(\alpha_1, \dots, \alpha_n)$  be the  $\rho$ -induced  $\mathcal{B}$ -coalgebra, cf. (11) and Def. 13, together with its coinductive extension  $u$ . Then the family  $(\ker(u_{\alpha_1}), \dots, \ker(u_{\alpha_n}), \ker(u))$  of kernel relations is  $id_{\Sigma(S_1, \dots, S_n)}$ -compatible.*

Thus by observation 19: If pairs  $(x_i, x'_i)$  are observationally equivalent w.r.t.  $\alpha_i$ , then corresponding elements in the set  $\Sigma(S_1, \dots, S_n)$  are observationally equivalent w.r.t. the  $\rho$ -induced coalgebra  $\Sigma_\rho(\alpha_1, \dots, \alpha_n)$ . Thus observational equivalence carries over from the local components to the compound system.

**Proof.** To make reading easier, we give the proof of Theorem 21 for the special case  $n = 2$ . It easily carries over to the general case. Let  $\mathcal{H}_1, \mathcal{H}_2$ , and  $\mathcal{H}$  be the copointed versions of  $\mathcal{B}_1, \mathcal{B}_2$ , and  $\mathcal{B}$  as in (2) and with this

$$\vec{\mathcal{H}} := \mathcal{H}_1 \times \mathcal{H}_2 \times \mathcal{H} : \mathcal{SET}^3 \rightarrow \mathcal{SET}^3$$

Let  $\vec{\mathcal{B}} := \mathcal{B}_1 \times \mathcal{B}_2 \times \mathcal{B} : \mathcal{SET}^3 \rightarrow \mathcal{SET}^3$ , then we have obtained the setting of Sect. 4 with  $\mathbb{C} = \mathcal{SET}^3$ ,  $\mathcal{H} := \vec{\mathcal{H}}$ , and  $\mathcal{B} := \vec{\mathcal{B}}$ . Furthermore, we define

$$\vec{\rho} := (\alpha_1, \alpha_2, \rho) : \vec{\Sigma}\vec{\mathcal{H}} \Rightarrow \vec{\mathcal{B}}\vec{\Sigma} : \mathcal{SET}^3 \rightarrow \mathcal{SET}^3. \quad (15)$$

which is a natural transformation, because  $(\alpha_1, X_1 = \alpha_1)_{X_1 \in |\mathcal{SET}|}$  and  $(\alpha_2, X_2 = \alpha_2)_{X_2 \in |\mathcal{SET}|}$  are independent of their parameters  $X_1, X_2$ , resp. By Prop. 4 it corresponds to a distributive law  $\vec{\lambda} : \vec{\Sigma}\vec{\mathcal{H}} \Rightarrow \vec{\mathcal{H}}\vec{\Sigma}$  of  $\vec{\Sigma}$  over  $\vec{\mathcal{H}}$ , where by (4)

$$\vec{\lambda} = \langle \vec{\Sigma}\pi_1, \vec{\rho} \rangle. \quad (16)$$

Following the notation of Prop. 20, (6) becomes

$$\begin{array}{ccc} \vec{\Sigma}(\mathbf{0}) & \xlongequal{\quad} & \mathbf{0} \\ \vec{\Sigma}h_{\vec{\lambda}} \downarrow & & \downarrow h_{\vec{\lambda}} \\ \vec{\Sigma}\vec{\mathcal{H}}(\mathbf{0}) & \xrightarrow{\vec{\lambda}_{\mathbf{0}}} & \vec{\mathcal{H}}(\mathbf{0}) \end{array} \quad (17)$$

The first component  $h_{\vec{\lambda}}^1$  of  $h_{\vec{\lambda}}$  is the composition of the first components of the left and bottom arrow:  $h_{\vec{\lambda}}^1 = \vec{\lambda}_{\mathbf{0}}^1 \circ (\vec{\Sigma}h_{\vec{\lambda}})^1 = \langle id_{S_1}, \alpha_1 \rangle \circ id_{S_1}$ , because  $\vec{\Sigma}$  is constant in the first component and similarly for the second component, hence

$$(h_{\vec{\lambda}}^1, h_{\vec{\lambda}}^2) = (\langle id_{S_1}, \alpha_1 \rangle, \langle id_{S_1}, \alpha_2 \rangle). \quad (18)$$

Thus, the third component of  $\vec{\Sigma}h_{\vec{\lambda}}$  equals  $\Sigma(\langle id_{S_1}, \alpha_1 \rangle, \langle id_{S_2}, \alpha_2 \rangle)$ . By (16) the third component of  $\vec{\lambda}_{\mathbf{0}}$  is the pair of the third component of  $(\vec{\Sigma}\pi_1)_{\mathbf{0}}$  and  $\rho_{(S_1, S_2)}$ , hence

$$h_{\vec{\lambda}}^3 = \langle id_{\Sigma(S_1, S_2)}, \rho_{S_1, S_2} \circ \Sigma(\langle id_{S_1}, \alpha_1 \rangle, \langle id_{S_2}, \alpha_2 \rangle) \rangle = \langle id_{\Sigma(S_1, S_2)}, \Sigma_\rho(\alpha_1, \alpha_2) \rangle \quad (19)$$

## 7:14 SOS for Heterogeneously Typed Coalgebras

by Def. 13. Let  $(\mathbf{1}, \zeta)$  be the final  $\vec{B}$ -coalgebra (which exists, because it is taken componentwise), then by Prop. 3  $(\mathbf{1}, \langle id, \zeta \rangle)$  is final in  $\vec{\mathcal{H}}\text{-Coalg}_{co}$  and (7) is reflected in the left two squares in

$$\begin{array}{ccccc}
 \vec{\Sigma}(\mathbf{0}) & \xlongequal{\quad} & \mathbf{0} & \xrightarrow{h_{\vec{x}}} & \vec{\mathcal{H}}(\mathbf{0}) & \xrightarrow{(\pi_2)_0} & \vec{B}(\mathbf{0}) \\
 \vec{\Sigma}\vec{u} \downarrow & & \downarrow \vec{u} & & \downarrow \vec{\mathcal{H}}\vec{u} & & \downarrow \vec{B}\vec{u} \\
 \vec{\Sigma}(\mathbf{1}) & \xrightarrow{g_{\vec{x}}} & \mathbf{1} & \xrightarrow{\langle id, \zeta \rangle} & \vec{\mathcal{H}}(\mathbf{1}) & \xrightarrow{(\pi_2)_1} & \vec{B}(\mathbf{1}) \\
 & & & \searrow \zeta & & & 
 \end{array} \tag{20}$$

with  $\vec{u} = (u_{\alpha_1}, u_{\alpha_2}, u)$ , see observation 9. By (15), (18), and (19) the triangle in the top right commutes.

Thus the coinductive extension  $\vec{u}$  of the  $\vec{B}$ -coalgebra  $(\alpha_1, \alpha_2, \Sigma_\rho(\alpha_1, \alpha_2))$  is a  $\vec{\Sigma}$ -algebra homomorphism and it is well-known that this makes  $\vec{u}$ 's kernel relation a congruence in the sense of (14) for  $\mathcal{F} := \vec{\Sigma}$ , see [12], Sect. 3.2., where  $A = (S_1, S_2, \Sigma(S_1, S_2)) = \mathcal{F}(A)$ ,  $R = (\ker(u_{\alpha_1}), \ker(u_{\alpha_2}), \ker(u))$ , hence  $\mathcal{F}(R) = (S_1, S_2, \Sigma(\ker(u_{\alpha_1}), \ker(u_{\alpha_2}))$ . Considering the third components only, shows that the family of kernel relations  $(R_1 := \ker(u_{\alpha_1}), R_2 := \ker(u_{\alpha_2}), R := \ker(u))$  is  $id_{\Sigma(S_1, S_2)}$ -compatible as desired.  $\blacktriangleleft$

From Theorem 21 we also obtain

**► Corollary 22** (Sufficient Criterion for Compositionality). *Let  $\mathcal{B}_1, \dots, \mathcal{B}_n, \mathcal{B}$  and  $\Sigma : \mathcal{SET}^n \rightarrow \mathcal{SET}$  be given as above. Let for all  $i \in \{1, \dots, n\}$  the  $\mathcal{SET}$ -endofunctors  $\mathcal{H}_i$  and  $\mathcal{H}$  be given as in (2), then compositionality holds for the heterogeneous scenario, if the computation of the compound system can be described by a natural transformation*

$$\rho : \Sigma(\mathcal{H}_1 \times \dots \times \mathcal{H}_n) \Rightarrow \mathcal{B}\Sigma : \mathcal{SET}^n \rightarrow \mathcal{SET}. \quad \lrcorner$$

## 6 Related Work

**Practical approaches.** The general idea of transforming different behavioural formalisms to a single semantic domain in order to reason about crosscutting concerns is nothing new [6]. We mention only a few approaches: [17] developed consistency checking for sequence diagrams and statecharts based on CSP, while Petri nets were used for the same scenario in [29]. Nevertheless, all approaches utilize fixed types of transition systems and no common framework, which can capture all possible types of transition structures. In recent years, *co-simulation* of coupled heterogeneous systems has become popular and there is already a plethora of work on that topic [7]. In particular [5] tackles the problem of coordinating different models using a dedicated coordination language. However, the majority of these approaches lack theoretical underpinnings, and, to the best of our knowledge, co-simulated comprehensive behaviour has not been formulated coalgebraically.

**SOS Framework, Distributive Laws and Compositionality.** All important variants of SOS rules are described in [1] and we took most of its coalgebraic abstraction from the original work [27], further elaborated in [14], especially for copointed functors in [18], and probably formulated in the most general way in [12]. All important variations of distributive laws and connected aspects of compositionality are surveyed in Chapter 8 of [14]. Moreover, compositionality in the bialgebraic approach is a facet of the microcosm principle: The

behavior of a composed system involves an outer operator on  $\mathcal{B}\text{-Coalg}$ , the composition of behaviors is an inner operator on the final object of  $\mathcal{B}\text{-Coalg}$ , see [9], where the compositionality property is derived from a formalization of the microcosm principle for Lawvere theories.

*Heterogeneity* appears whenever different behavioral paradigms shall be combined. One of the first examples are hybrid systems, which combine discrete and continuous dynamics [11]. However, reasoning about operational semantics of arbitrary heterogeneously typed transition structures is usually treated by common abstractions of the different systems: E.g. the coordination of a Mealy machine and a probabilistic system can be investigated by reducing both systems to labelled transition systems and formulating interactions with LTS-based SOS rules. A different approach, which is closer to ours, is described in [13], where the combination of two distributive laws based on different behavioral specifications is investigated: So-called *heterogeneous* transition systems simultaneously carry two different coalgebraic structures  $\mathcal{B}$  and  $\mathcal{B}'$  and behavioural descriptions are based on natural transformations of the form  $\Sigma(\mathcal{B} \times \mathcal{B}') \Rightarrow (\mathcal{B} \times \mathcal{B}')\Sigma$ . However, the authors do not pick up the holistic view of our approach and do not investigate compositionality.

*Categorically*, heterogeneity leads to the general theory of *(co-)institutions*. [23] proves three different types of logics for coalgebras to be institutions. Another approach are parametrized endofunctors as comprehensive behavioural specifications, where the overall structure can be studied in terms of cofibrations [16]. [28] investigates co-institutions purely dual to classical institutions [25].

## 7 Future Work

We investigated the synchronisation of  $n$  local components to obtain a compound system. The idea was to introduce  $n + 1$  sorts, which reflects the fact that the resulting compound system is obtained *in one step* from the locals. That excludes step-by-step synchronisation, i.e. the assembly of some components to an intermediate composed system, which in a later step is combined with other components, before the resulting global operational semantics is reached. The challenge in future work is to cope with an unsteady number of sorts for the arising intermediate systems. Similarly, our approach cannot directly be applied to asynchronous communications via intermediate components like message queues, object spaces, etc. It is a goal to derive formal underpinnings also in these cases.

Moreover, it is worth thinking about other types of extensions or refinements of local components and how they cause an impact on the composed system. If, for instance, a local system is conservatively extended [1], then we can ask the question whether the compound system is also conservatively extended. Furthermore, it is an open question, whether extensive refinements of the local systems and their interaction specifications can still be handled with interaction laws.

Finally, if additional system properties are imposed on the local behavioural models by modal logic formulae, the question arises, whether the use of co-forgetful functors in the translation of these formulae to the compound system [15] matches the framework proposed in the present paper. Altogether, the goal is to extend the first iteration of our work and, in future steps, develop more insight into the topic.

## References

- 1 Luca Aceto, Wan J. Fokkink, and Chris Verhoef. Structural operational semantics. In Jan A. Bergstra, Alban Ponse, and Scott A. Smolka, editors, *Handbook of Process Algebra*, pages 197–292. North-Holland / Elsevier, 2001. doi:10.1016/b978-044482830-9/50021-7.
- 2 Falk Bartels. On generalised coinduction and probabilistic specification formats: distributive laws in coalgebraic modelling. *Academisch Proefschrift, Vrije Universiteit Amsterdam*, 2004.
- 3 Bard Bloom, Sorin Istrail, and Albert R. Meyer. Bisimulation can’t be traced. In Jeanne Ferrante and Peter Mager, editors, *Conference Record of the Fifteenth Annual ACM Symposium on Principles of Programming Languages, San Diego, California, USA, January 10-13, 1988*, pages 229–239. ACM Press, 1988. doi:10.1145/73560.73580.
- 4 Marco Brambilla, Jordi Cabot, and Manuel Wimmer. Model-driven software engineering in practice. *Synthesis lectures on software engineering*, 3(1):1–207, 2017.
- 5 Julien Deantoni. Modeling the behavioral semantics of heterogeneous languages and their coordination. In *2016 Architecture-Centric Virtual Integration (ACVI)*, pages 12–18. IEEE, 2016.
- 6 Gregor Engels, Jochen Malte Küster, Reiko Heckel, and Luuk Groenewegen. A methodology for specifying and analyzing consistency of object-oriented behavioral models. In A Min Tjoa and Volker Gruhn, editors, *Proceedings of the 8th European Software Engineering Conference held jointly with 9th ACM SIGSOFT International Symposium on Foundations of Software Engineering 2001, Vienna, Austria, September 10-14, 2001*, pages 186–195. ACM, 2001. doi:10.1145/503209.503235.
- 7 Cláudio Gomes, Casper Thule, David Broman, Peter Gorm Larsen, and Hans Vangheluwe. Co-simulation: a survey. *ACM Computing Surveys (CSUR)*, 51(3):1–33, 2018.
- 8 Jan Friso Groote and Frits Vaandrager. Structured operational semantics and bisimulation as a congruence. *Information and computation*, 100(2):202–260, 1992.
- 9 Ichiro Hasuo, Bart Jacobs, and Ana Sokolova. The microcosm principle and concurrency in coalgebra. In *International Conference on Foundations of Software Science and Computational Structures*, pages 246–260. Springer, 2008.
- 10 Charles Antony Richard Hoare. Communicating sequential processes. *Communications of the ACM*, 21(8):666–677, 1978.
- 11 Bart Jacobs. Object-oriented hybrid systems of coalgebras plus monoid actions. *Theoretical Computer Science*, 239(1):41–95, 2000.
- 12 Bart Jacobs. *Introduction to Coalgebra: Towards Mathematics of States and Observation*, volume 59 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2016. doi:10.1017/CB09781316823187.
- 13 Marco Kick, John Power, and Alex Simpson. Coalgebraic semantics for timed processes. *Information and Computation*, 204(4):588–609, 2006.
- 14 Bartek Klin. Bialgebras for structural operational semantics: An introduction. *Theor. Comput. Sci.*, 412(38):5043–5069, 2011. doi:10.1016/j.tcs.2011.03.023.
- 15 Harald König and Uwe Wolter. Consistency of heterogeneously typed behavioural models: A coalgebraic approach. In Yamine Aït Ameur and Florin Craciun, editors, *Theoretical Aspects of Software Engineering - 16th International Symposium, TASE 2022, Cluj-Napoca, Romania, July 8-10, 2022, Proceedings*, volume 13299 of *Lecture Notes in Computer Science*, pages 308–325. Springer, 2022. doi:10.1007/978-3-031-10363-6\_21.
- 16 Alexander Kurz and Dirk Pattinson. *Coalgebras and modal logic for parameterised endofunctors*. Centrum voor Wiskunde en Informatica, 2000.
- 17 Jochen Malte Küster. Towards Inconsistency Handling of Object-Oriented Behavioral Models. *Electron. Notes Theor. Comput. Sci.*, 109:57–69, 2004. doi:10.1016/j.entcs.2004.02.056.
- 18 Marina Lenisa, John Power, and Hiroshi Watanabe. Category theory for operational semantics. *Theor. Comput. Sci.*, 327(1-2):135–154, 2004. doi:10.1016/j.tcs.2004.07.024.

- 19 Dorian Leroy, Erwan Bousse, Manuel Wimmer, Tanja Mayerhofer, Benoit Combemale, and Wieland Schwinger. Behavioral interfaces for executable DSLs. *Software and Systems Modeling*, 19(4):1015–1043, 2020.
- 20 Giovanni Liboni and Julien Deantoni. Cosim20: An integrated development environment for accurate and efficient distributed co-simulations. In *Proceedings of the 2020 International Conference on Big Data in Management*, pages 76–83, 2020.
- 21 Robin Milner. *Communicating and Mobile Systems: The  $\pi$ -Calculus*. Cambridge University Press, 1999.
- 22 Mogens Nielsen and Glynn Winskel. Models for concurrency. In *Handbook of Logic in Computer Science, Volume 4*. Oxford Science Publications, 1995.
- 23 Dirk Pattinson. Translating logics for coalgebras. In *International Workshop on Algebraic Development Techniques*, pages 393–408. Springer, 2002.
- 24 Jan J. M. M. Rutten. Universal coalgebra: a theory of systems. *Theor. Comput. Sci.*, 249(1):3–80, 2000. doi:10.1016/S0304-3975(00)00056-6.
- 25 Donald Sannella and Andrzej Tarlecki. *Foundations of Algebraic Specification and Formal Software Development*. Monographs in Theoretical Computer Science. An EATCS Series. Springer, 2012. doi:10.1007/978-3-642-17336-3.
- 26 Ana Sokolova. Probabilistic systems coalgebraically: A survey. *Theoretical Computer Science*, 412(38):5095–5110, 2011.
- 27 Daniele Turi and Gordon Plotkin. Towards a mathematical operational semantics. In *Proceedings of Twelfth Annual IEEE Symposium on Logic in Computer Science*, pages 280–291. IEEE, 1997.
- 28 U. Wolter. (Co)Institutions for Coalgebras. Reports in Informatics 415, Dep. of Informatics, University of Bergen, 2016.
- 29 Shuzhen Yao and Sol M Shatz. Consistency checking of UML dynamic models based on Petri net techniques. In *2006 15th International Conference on Computing*, pages 289–297. IEEE, 2006.



# Interpolation Is (Not Always) Easy to Spoil

Andrzej Tarlecki ✉🏠

Institute of Informatics, University of Warsaw, Poland

---

## Abstract

We study a version of the Craig interpolation theorem as formulated in the framework of the theory of institutions. This formulation proved crucial in the development of a number of key results concerning foundations of software specification and formal development. We investigate preservation of interpolation under extensions of institutions by new models and sentences. We point out that some interpolation properties remain stable under such extensions, even if quite arbitrary new models or sentences are permitted. We give complete characterisations of such situations for institution extensions by new models, by new sentences, as well as by new models and sentences, respectively.

**2012 ACM Subject Classification** Theory of computation → Logic; Theory of computation → Logic and verification

**Keywords and phrases** interpolation, institutions, institutional abstract model theory, specification theory

**Digital Object Identifier** 10.4230/LIPIcs.CALCO.2023.8

**Acknowledgements** Thanks to the anonymous reviewers for a number of useful comments.

## 1 Introduction

The Craig interpolation theorem [12] states that when an implication  $\varphi \Rightarrow \psi$  between premise  $\varphi$  and conclusion  $\psi$  holds then there is an interpolant  $\theta$  built using the symbols the premise and the conclusion have in common that witnesses this implication, that is, such that both  $\varphi \Rightarrow \theta$  and  $\theta \Rightarrow \psi$  hold. This is one of the fundamental properties of the classical first-order logic, with numerous consequences and links with other key properties developed in the framework of classical model theory [11].

In the area of foundations of system specification and formal development, interpolation proved indispensable for a number of most fundamental features of various approaches. This was perhaps first pointed out in [27], where it was used to ensure composability of subsequent implementation steps (later refined in various forms of the so-called modularisation theorem [42, 41]). In the work on module algebra [3] the interpolation was necessary to obtain crucial distributive laws for their export operator ([31] joined the two threads). The proofs of completeness of proof calculi for consequences of structured specifications rely on interpolation [10, 5] (in fact, no “good” sound and complete such proof calculus may exist without an appropriate interpolation property for the underlying logic [36]). These and further results concerning completeness of various reasoning systems necessary in the process of reliable software development involve interpolation explicitly, but the same idea that showing properties of a union of a number of extensions of a basic theory must rely on some form of interpolation (perhaps disguised as the Robinson consistency [32]) is omnipresent in both practical and foundational aspects of computing.

Applications of logic in computer science face the problem of dealing with numerous logical systems. This follows from the real needs of software development, based on the multitude of application areas as well as of programming paradigms, features and languages. This led to various attempts to abstract away from a specific logical system in use. Such an independence of the foundations for software specification has been successfully achieved by



© Andrzej Tarlecki;

licensed under Creative Commons License CC-BY 4.0

10th Conference on Algebra and Coalgebra in Computer Science (CALCO 2023).

Editors: Paolo Baldan and Valeria de Paiva; Article No. 8; pp. 8:1–8:19

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

relying on the concept of an institution, introduced by Goguen and Burstall as a formalisation of the concept of a logical system [25]. See for instance [35] for an exhaustive account of such ideas, with further examples in the development of specification formalisms such as CASL [1].

It has been realised quite early that institutions also offer a framework for developing a very abstract version of model theory, going beyond what has been studied within abstract model theory following [2]. This was noted in [37] and expanded in many crucial directions by Diaconescu and his group; his monograph [13] offers an overview of this work, with later developments scattered through numerous articles (see e.g. [16] and references there).

In the institutional model theory the interpolation property is formulated so that it can be studied (and used) for logical systems departing considerably from the first-order logic. This was put forward in [37], but we use here a still more refined formulation of interpolation given in [34, 14]. This formulation uses logical entailment (rather than implication), sets of sentences (rather than individual sentences) and, most crucially, works over arbitrary commutative squares of signature morphisms (rather than over union/intersection squares only), and so caters for instance for the logical systems where one lacks compactness, conjunction and other classical connectives, and even the concept of the set of symbols used in a formula and union/intersection of signatures may not be directly available. The key point of many of the applications mentioned above is the need to abstract away from signature inclusions and deal with interpolation properties with other signature morphisms considered. Subsequent work included development of generic model-theoretic proof techniques to establish interpolation for institutions satisfying a number of structural properties. This led to new interpolation results concerning various logical systems, as well as to studying interpolation in even more general context of non-standard entailment relations [14, 6, 24, 30, 15, 22, 23, 17].

The need for the use of many logical systems leads to the need for establishing their properties, including the interpolation property we study here. Rather than doing this for each system anew, it is desirable to ensure the required properties in the course of systematic construction of new logics, perhaps along the lines aimed at for instance in [29, 28] or [8, 7, 9]. Typically, the new logics are linked with the original ones by institution (co)morphisms [25, 26]. An important line of research was to clarify sufficient conditions on the institution (co)morphisms involved to allow interpolation properties to be “transferred” between the institutions they link [18, 22].

We address a perhaps more basic question that arises in this framework: namely, when interpolation properties can be spoiled by extending a logic by new abstract models or sentences. Looking at the standard formulation, it seems that the answer is always positive. To spoil an interpolant for the premise and the conclusion of a true implication, just add a new model that satisfies the premise but not the interpolant, or the interpolant but not the conclusion, thus spoiling the required implication between the premise and the interpolant, or between the interpolant and the conclusion. This should work, except for the trivial cases when the signature of the premise includes or is included in the signature of the conclusion. At a closer look though, when one considers arbitrary signature morphisms, adding new models for the signature of the premise or for the signature of the conclusion may result in new models for their union signature, and ruin the implication considered.

We explore the consequences of this observation, and give exact characterisations of the situations where interpolation is stable under extensions of institutions. Equivalently, looking at the other side of this coin, we obtain the exact characterisation of the situations where new models or sentences may spoil the interpolation property. More precisely: we consider separately institution extensions where only new models, only new sentences, and both new models and sentences, respectively, are permitted. In each of these three cases



complete characterisations are given, formulating necessary and sufficient conditions for a commutative square of signature morphisms under which no such institution extension may spoil interpolation properties over this square.

## 2 Institutions

### 2.1 Notational preliminaries

For any function  $f: X \rightarrow Y$ , given a set  $X' \subseteq X$ ,  $f(X') = \{f(x) \mid x \in X'\} \subseteq Y$  is the image of  $X'$  w.r.t.  $f$ , and for  $Y' \subseteq Y$ ,  $f^{-1}(Y') = \{x \in X \mid f(x) \in Y'\}$  is the coimage of  $Y'$  w.r.t.  $f$ .

Throughout the paper we freely use the basic notions from category theory (*category*, *functor*, *natural transformation*, *pushout*, etc). Composition in any category is denoted by “;” (semicolon) and written in the diagrammatic order. For instance,  $f: A \rightarrow B$  is a *retraction* if for some  $g: B \rightarrow A$  we have  $g;f = id_B$ , and  $f: A \rightarrow B$  is a *coretraction* if for some  $g: B \rightarrow A$  we have  $f;g = id_A$ . The collection of objects of any category  $\mathbf{K}$  is written as  $|\mathbf{K}|$ . The category of sets is denoted by **Set**, and the (quasi-)category of classes by **Class**.

### 2.2 Institutions

In the foundations of software specification and development [35] it is standard by now to abstract away from the details of the logical system in use, relying on the formalisation of a logical system as an *institution* [25]. An institution **INS** consists of:

- a category  $\mathbf{Sign}_{\mathbf{INS}}$  of *signatures*;
- a functor  $\mathbf{Sen}_{\mathbf{INS}}: \mathbf{Sign}_{\mathbf{INS}} \rightarrow \mathbf{Set}$ , giving a set  $\mathbf{Sen}_{\mathbf{INS}}(\Sigma)$  of  $\Sigma$ -*sentences* for each signature  $\Sigma \in |\mathbf{Sign}_{\mathbf{INS}}|$ ;
- a functor  $\mathbf{Mod}_{\mathbf{INS}}: \mathbf{Sign}_{\mathbf{INS}}^{op} \rightarrow \mathbf{Class}$ , giving a class (or a discrete category)<sup>1</sup>  $\mathbf{Mod}_{\mathbf{INS}}(\Sigma)$  of  $\Sigma$ -*models* for each signature  $\Sigma \in |\mathbf{Sign}_{\mathbf{INS}}|$ ; and
- a family  $\langle \models_{\mathbf{INS}, \Sigma} \subseteq \mathbf{Mod}_{\mathbf{INS}}(\Sigma) \times \mathbf{Sen}_{\mathbf{INS}}(\Sigma) \rangle_{\Sigma \in |\mathbf{Sign}_{\mathbf{INS}}|}$  of *satisfaction relations* such that for any signature morphism  $\sigma: \Sigma \rightarrow \Sigma'$  the induced translations  $\mathbf{Mod}_{\mathbf{INS}}(\sigma)$  of models and  $\mathbf{Sen}_{\mathbf{INS}}(\sigma)$  of sentences preserve the satisfaction relation, that is, for any  $\varphi \in \mathbf{Sen}_{\mathbf{INS}}(\Sigma)$  and  $M' \in \mathbf{Mod}_{\mathbf{INS}}(\Sigma')$  the following *satisfaction condition* holds:

$$M' \models_{\mathbf{INS}, \Sigma'} \mathbf{Sen}_{\mathbf{INS}}(\sigma)(\varphi) \quad \text{iff} \quad \mathbf{Mod}_{\mathbf{INS}}(\sigma)(M') \models_{\mathbf{INS}, \Sigma} \varphi.$$

The subscripts **INS** and  $\Sigma$  are typically omitted. For any signature morphism  $\sigma: \Sigma \rightarrow \Sigma'$ , the *translation*  $\mathbf{Sen}(\sigma): \mathbf{Sen}(\Sigma) \rightarrow \mathbf{Sen}(\Sigma')$  is denoted by  $\sigma: \mathbf{Sen}(\Sigma) \rightarrow \mathbf{Sen}(\Sigma')$ , and the *reduct*  $\mathbf{Mod}(\sigma): \mathbf{Mod}(\Sigma') \rightarrow \mathbf{Mod}(\Sigma)$  by  $\_ |_{\sigma}: \mathbf{Mod}(\Sigma') \rightarrow \mathbf{Mod}(\Sigma)$ . For instance, the satisfaction condition may be re-stated as:  $M' \models \sigma(\varphi)$  iff  $M' |_{\sigma} \models \varphi$ , and given the notation for image and coimage, for  $\Phi \subseteq \mathbf{Sen}(\Sigma)$ ,  $\sigma(\Phi) = \{\sigma(\varphi) \mid \varphi \in \Phi\} \subseteq \mathbf{Sen}(\Sigma')$ , and for  $\mathcal{M} \subseteq \mathbf{Mod}(\Sigma)$ ,  $\mathcal{M} |_{\sigma}^{-1} = \{M' \in \mathbf{Mod}(\Sigma') \mid M' |_{\sigma} \in \mathcal{M}\}$ . For any signature  $\Sigma$ , the satisfaction relation extends to sets of  $\Sigma$ -sentences and classes of  $\Sigma$ -models. For  $\Phi \subseteq \mathbf{Sen}(\Sigma)$ , the *class of models of  $\Phi$*  is  $Mod(\Phi) = \{M \in \mathbf{Mod}(\Sigma) \mid M \models \Phi\}$ , and for  $\mathcal{M} \subseteq \mathbf{Mod}(\Sigma)$ , the *theory of  $\mathcal{M}$*  is  $Th(\mathcal{M}) = \{\varphi \in \mathbf{Sen}(\Sigma) \mid \mathcal{M} \models \varphi\}$ . The latter notation is also used for the *theory generated by a set of sentences*: for  $\Phi \subseteq \mathbf{Sen}(\Sigma)$ ,  $Th(\Phi) = Th(Mod(\Phi))$ .

Each satisfaction relation determines a (semantic) entailment between sets of sentences:  $\Phi \subseteq \mathbf{Sen}(\Sigma)$  *entails*  $\Psi \subseteq \mathbf{Sen}(\Sigma)$  (or  $\Psi$  is a *consequence of  $\Phi$* ), written  $\Phi \models \Psi$ , when  $\Psi \subseteq Th(\Phi)$ . The satisfaction condition implies that the semantic entailment is preserved under

<sup>1</sup> We disregard here model morphisms, irrelevant for the purposes of this paper.

## 8:4 Interpolation Is (Not Always) Easy to Spoil

translation along signature morphisms: for any  $\sigma: \Sigma \rightarrow \Sigma'$ , if  $\Phi \models \Psi$  then  $\sigma(\Phi) \models \sigma(\Psi)$ . If the opposite implication holds as well, i.e.  $\Phi \models \Psi$  iff  $\sigma(\Phi) \models \sigma(\Psi)$  for all  $\Phi, \Psi \subseteq \mathbf{Sen}(\Sigma)$ , we say that  $\sigma: \Sigma \rightarrow \Sigma'$  is *conservative*. It is well-known that if the reduct  $\_ |_{\sigma}: \mathbf{Mod}(\Sigma') \rightarrow \mathbf{Mod}(\Sigma)$  is surjective then  $\sigma: \Sigma \rightarrow \Sigma'$  is conservative.<sup>2</sup>

We typically decorate the names for institution components and for other derived notions by primes, indices, etc, to identify the institution they refer to, and rely on this convention whenever the institution is clear from the context. So, for instance,  $\mathbf{Mod}_1$  is the model functor in an institution  $\mathbf{INS}_1$ ,  $\models'$  is the satisfaction relation (and entailment) in  $\mathbf{INS}'$ , etc.

Examples of institutions abound, see e.g. [35, 13]. We just sketch three standard examples.

► **Example 1.** The institution **FO** of (many-sorted) first-order logic has signatures that consist of sets of sort names, of operation names with indicated arities and result sorts, and of predicate names with indicated arities. Terms and atomic formulae are defined as usual, and first-order formulae are built using the usual Boolean connectives (including nullary **false**) and quantification. First-order sentences are closed formulae (i.e. formulae with no free occurrences of variables). First-order models consist of many-sorted carrier sets (one set for each sort name), functions to interpret operation names and relations to interpret predicate names, in accordance with their arities and result sorts. Satisfaction of first-order sentences in first-order models is defined as usual. Signature morphisms map sort names to sort names, operation names to operation names and predicate names to predicate names preserving their arities and result sorts. For any such morphism, translation of sentences is defined by renaming sort, operation and predicate names as indicated by the morphism, and model reducts are defined by interpreting the symbols of the source signature as the symbols they are mapped to in the target signature are interpreted in the argument model. This indeed defines an institution [25]. We assume that carrier sets in first-order models are nonempty. The variant of **FO** where empty carrier sets are allowed in models is denoted by  $\mathbf{FO}_{\emptyset}$ . Another variant is the institution  $\mathbf{FO}_{\mathbf{EQ}}$  of first-order logic with equality, with a binary equality predicate for each sort, interpreted as the identity relation in all models.

The institution **EQ** of (many-sorted) equational logic may be defined as the restriction of  $\mathbf{FO}_{\mathbf{EQ}}$  to the signatures with no predicates other than equalities (models are usually called algebras then), and sentences limited to universally quantified equalities.  $\mathbf{EQ}_{\emptyset}$  is the variant of **EQ** with empty carriers permitted. See [35, 13] for an explicit definition.

The institution **PL** of propositional logic has finite sets of *propositional variables* as signatures, with signature morphisms being arbitrary functions between those sets. Propositional sentences are built from propositional variables using the usual Boolean connectives (with obvious translations under functions renaming propositional variables). Models over a signature are given as subsets of this signature (consisting of the propositional variables that are satisfied in the model) with reducts w.r.t. signature morphisms given as their coimage. With the usual satisfaction of propositional sentences in such models, the satisfaction condition is easy to check. In fact, the institution **PL** of propositional logic may be viewed as a restriction of the institution of first-order logic to finite signatures with no sort names (and hence no operation names and nullary predicates only).

In the institutions **FO**, **EQ**, and **PL** all injective signature morphisms induce surjective reducts, and so are conservative. This need not be the case for non-injective morphisms. In  $\mathbf{FO}_{\emptyset}$  in  $\mathbf{EQ}_{\emptyset}$ , the variants of **FO** and of **EQ** where empty carriers are permitted, not all injective signature morphisms are conservative.

<sup>2</sup> Some authors use “conservative” for signature morphisms that induce surjective reducts [26]. Our more permissive definition seems closer to the standard definition of a conservative theory interpretation [11].

In the above examples all the signatures, sentences and models are quite familiar, and link with many intuitions and implicit assumptions. However, when exploiting the generality of the concept and working with an arbitrary institution, such connotations should be dropped. All the entities involved (signatures, their morphisms, sentences, models, satisfaction relations) are considered entirely abstract, with completely unknown structure and properties. It is perhaps surprising how far one can go with developments of the foundations for software specification [35] and an abstract version of model theory [13] in such an abstract setting.

### 2.3 Extending institutions by models and sentences

We introduce two basic ways of extending institutions, by adding new “abstract” models, and new “abstract” sentences, respectively. The definitions are shaped after the definition of *constraints* in [25, 35]. The basic observation is that when a new sentence is added over a signature, with some predefined notion of satisfaction in the institution models, it must also be “fitted” to other signatures to mimic its translation along signature morphisms with this signature as a source. Hence, together with each new sentence, we also add its “formal translations” along signature morphisms. The satisfaction of such formal translations is determined by the satisfaction condition. Similarly, when we add a new model over a signature – apart from the model itself, we must also add its “formal reducts”.

Consider an arbitrary institution  $\mathbf{INS} = \langle \mathbf{Sign}, \mathbf{Sen}, \mathbf{Mod}, \langle \models_{\Sigma} \rangle_{\Sigma \in |\mathbf{Sign}|} \rangle$ .

Suppose that for each signature we are given a set of (new) “sentences” with predefined satisfaction relation in the  $\mathbf{INS}$ -models, which may be organised as a signature-indexed family of sets with relations:  $\mathcal{NS} = \langle \mathcal{NS}_{\Sigma}, \models_{\Sigma}^{\mathcal{NS}} \subseteq \mathbf{Mod}(\Sigma) \times \mathcal{NS}_{\Sigma} \rangle_{\Sigma \in |\mathbf{Sign}|}$ .<sup>3</sup>

The *extension of  $\mathbf{INS}$  by sentences  $\mathcal{NS}$*  is  $\mathbf{INS}^+ = \langle \mathbf{Sign}, \mathbf{Sen}^+, \mathbf{Mod}, \langle \models_{\Sigma}^+ \rangle_{\Sigma \in |\mathbf{Sign}|} \rangle$ , where for  $\Sigma \in |\mathbf{Sign}|$ ,  $\mathbf{Sen}^+(\Sigma) = \mathbf{Sen}(\Sigma) \cup \{ \lceil \tau(\varphi') \rceil \mid \varphi' \in \mathcal{NS}_{\Sigma'}, \tau: \Sigma' \rightarrow \Sigma \}$ .<sup>4</sup> Then for  $M \in \mathbf{Mod}(\Sigma)$ ,  $M \models_{\Sigma}^+ \varphi$  iff  $M \models_{\Sigma} \varphi$  for  $\varphi \in \mathbf{Sen}(\Sigma)$ , and for  $\varphi' \in \mathcal{NS}_{\Sigma'}, \tau: \Sigma' \rightarrow \Sigma$ ,  $M \models_{\Sigma}^+ \lceil \tau(\varphi') \rceil$  iff  $M|_{\tau} \models_{\Sigma'}^{\mathcal{NS}} \varphi'$ . Finally, for  $\sigma: \Sigma \rightarrow \Sigma''$ ,  $\mathbf{Sen}^+(\sigma)(\varphi) = \mathbf{Sen}(\sigma)(\varphi)$  for  $\varphi \in \mathbf{Sen}(\Sigma)$ , and for  $\varphi' \in \mathcal{NS}_{\Sigma'}, \tau: \Sigma' \rightarrow \Sigma$ ,  $\mathbf{Sen}^+(\sigma)(\lceil \tau(\varphi') \rceil) = \lceil (\tau;\sigma)(\varphi') \rceil$ .

This defines an institution where for  $\Sigma \in |\mathbf{Sign}|$ , the new sentences  $\varphi \in \mathcal{NS}_{\Sigma}$  are present as  $\lceil id_{\Sigma}(\varphi) \rceil$ . Such an extension does not affect entailments between sets of  $\mathbf{INS}$ -sentences.

Suppose then that for each signature we are given a class of (new) “models” with predefined satisfaction relation for the  $\mathbf{INS}$ -sentences, organised as a signature-indexed family of classes with relations:  $\mathcal{NM} = \langle \mathcal{NM}_{\Sigma}, \models_{\Sigma}^{\mathcal{NM}} \subseteq \mathcal{NM}_{\Sigma} \times \mathbf{Sen}(\Sigma) \rangle_{\Sigma \in |\mathbf{Sign}|}$ .

The *extension of  $\mathbf{INS}$  by models  $\mathcal{NM}$*  is  $\mathbf{INS}^+ = \langle \mathbf{Sign}, \mathbf{Sen}, \mathbf{Mod}^+, \langle \models_{\Sigma}^+ \rangle_{\Sigma \in |\mathbf{Sign}|} \rangle$ , where for  $\Sigma \in |\mathbf{Sign}|$ ,  $\mathbf{Mod}^+(\Sigma) = \mathbf{Mod}(\Sigma) \cup \{ \lceil M' \rceil_{\tau} \mid M' \in \mathcal{NM}_{\Sigma'}, \tau: \Sigma \rightarrow \Sigma' \}$ .<sup>5</sup> Then for  $\varphi \in \mathbf{Sen}(\Sigma)$ ,  $M \models_{\Sigma}^+ \varphi$  iff  $M \models_{\Sigma} \varphi$  for  $M \in \mathbf{Mod}(\Sigma)$ , and for  $M' \in \mathcal{NM}_{\Sigma'}, \tau: \Sigma \rightarrow \Sigma'$ ,  $\lceil M' \rceil_{\tau} \models_{\Sigma}^+ \varphi$  iff  $M' \models_{\Sigma'}^{\mathcal{NS}} \tau(\varphi)$ . Finally, for  $\sigma: \Sigma'' \rightarrow \Sigma$ ,  $\mathbf{Mod}^+(\sigma)(M) = M|_{\sigma}$  for  $M \in \mathbf{Mod}(\Sigma)$ , and for  $M' \in \mathcal{NM}_{\Sigma'}, \tau: \Sigma \rightarrow \Sigma'$ ,  $\mathbf{Mod}^+(\sigma)(\lceil M' \rceil_{\tau}) = \lceil M' \rceil_{\sigma;\tau}$ .

This defines an institution where for  $\Sigma \in |\mathbf{Sign}|$ , the new models  $M \in \mathcal{NM}_{\Sigma}$  are present as  $\lceil M \rceil_{id_{\Sigma}}$ . Such an extension may spoil some entailments between sets of  $\mathbf{INS}$ -sentences: for  $\Phi, \Psi \subseteq \mathbf{Sen}(\Sigma)$  if  $\Phi \models^+ \Psi$  then  $\Phi \models \Psi$  but the opposite implication may fail.

<sup>3</sup> To avoid any foundational problems below, we may assume that  $\mathbf{Sign}$  is small, or that it is locally small and  $\mathcal{NS}_{\Sigma} \neq \emptyset$  for a set of signatures  $\Sigma$  only.

<sup>4</sup>  $\lceil \tau(\varphi') \rceil$  is just our syntax for the sentence  $\varphi' \in \mathcal{NS}_{\Sigma'}$  formally “fitted” by  $\tau: \Sigma' \rightarrow \Sigma$  to the signature  $\Sigma$ ; we assume that no sentences of the form  $\lceil \tau(\varphi') \rceil$  are present in  $\mathbf{INS}$ .

<sup>5</sup>  $\lceil M' \rceil_{\tau}$  is just our syntax for the model  $M' \in \mathcal{NM}_{\Sigma'}$  formally “fitted” by  $\tau: \Sigma \rightarrow \Sigma'$  to the signature  $\Sigma$ ; we assume that no models of the form  $\lceil M' \rceil_{\tau}$  are present in  $\mathbf{INS}$ .

## 8:6 Interpolation Is (Not Always) Easy to Spoil

When using these constructions, we often present new sentences  $\mathcal{NS}$  and new models  $\mathcal{NM}$  somewhat informally, avoiding much of the notational burden. We disregard the formal distinction between  $\varphi \in \mathcal{NS}_\Sigma$  and  $\lceil id_\Sigma(\varphi) \rceil$ , and between  $M \in \mathcal{NM}_\Sigma$  and  $\lceil M|_{id_\Sigma} \rceil$ . For  $\Sigma \in |\mathbf{Sign}|$ , we may define the satisfaction relations  $\models_\Sigma^{\mathcal{NS}}$  indirectly by defining  $Mod^+(\varphi) \subseteq \mathbf{Mod}(\Sigma)$  for  $\varphi \in \mathcal{NS}_\Sigma$  (then for  $M \in \mathbf{Mod}(\Sigma)$ ,  $M \models_\Sigma^{\mathcal{NS}} \varphi$  iff  $M \in Mod^+(\varphi)$ ), and  $\models_\Sigma^{\mathcal{NM}}$  by defining  $Th^+(M) \subseteq \mathbf{Sen}(\Sigma)$  for  $M \in \mathcal{NM}_\Sigma$  (then for  $\varphi \in \mathbf{Sen}(\Sigma)$ ,  $M \models_\Sigma^{\mathcal{NM}} \varphi$  iff  $\varphi \in Th^+(M)$ ).

► **Example 2.** We may define an extension of the institution **PL** of propositional logic by sentences, adding for each signature  $\Sigma$  a new sentence  $\mathbf{even}_\Sigma$  defined to hold in models that contain an even number of propositional variables. In the resulting extension  $\mathbf{PL}^+$  of **PL**, for any  $\sigma: \Sigma \rightarrow \Sigma'$ ,  $\mathbf{Sen}^+(\sigma)(\mathbf{even}_\sigma)$  is  $\lceil \sigma(\mathbf{even}_\Sigma) \rceil$ , which is distinct from  $\mathbf{even}_{\Sigma'}$ . Indeed, putting  $\mathbf{Sen}^+(\sigma)(\mathbf{even}_\Sigma) = \mathbf{even}_{\Sigma'}$  would violate the satisfaction condition for some  $\sigma$ .

► **Example 3.** We may also define an extension of **PL** by models, adding for each signature  $\Sigma$  and  $\Sigma$ -model  $M$ , a new model  $\widetilde{M}$ , where the satisfaction of propositional sentences in  $\widetilde{M}$  is defined by interpreting propositional connectives as usual, but the truth of propositional variables is determined separately for each occurrence, from left to right, and after each occurrence the values of all propositional variables are “swapped” (from true to false and vice versa). Thus, for instance the sentence  $p \wedge q$  holds in  $\widetilde{M}$  if  $p \in M$  and  $q \notin M$ , and  $p \vee p$  holds in any model  $\widetilde{M}$ . In the resulting extension  $\mathbf{PL}^+$ , for any signature  $\Sigma$  and  $M \in \mathbf{Mod}(\Sigma)$ , for any  $\sigma: \Sigma' \rightarrow \Sigma$ ,  $\widetilde{M}|_\sigma$  (that is,  $\mathbf{Mod}^+(\sigma)(\widetilde{M})$ ) and  $\widetilde{M}|_\sigma$  are distinct  $\Sigma'$ -models, even though they are logically equivalent (satisfy exactly the same propositional sentences).

## 2.4 Institution morphisms

There are a number of standard notions to capture relationships between different institutions, with institution morphisms [25] and comorphisms [26] perhaps the most common.

For any institutions **INS** and **INS'**, an *institution morphism*  $\mu: \mathbf{INS} \rightarrow \mathbf{INS}'$  consists of:

- a functor  $\mu^{Sign}: \mathbf{Sign} \rightarrow \mathbf{Sign}'$ ,
- a natural transformation  $\mu^{Sen}: \mu^{Sign}; \mathbf{Sen}' \rightarrow \mathbf{Sen}$ , i.e., a family of functions  $\mu_\Sigma^{Sen}: \mathbf{Sen}'(\mu^{Sign}(\Sigma)) \rightarrow \mathbf{Sen}(\Sigma)$  natural in  $\Sigma \in |\mathbf{Sign}|$ , and
- a natural transformation  $\mu^{Mod}: \mathbf{Mod} \rightarrow (\mu^{Sign})^{op}; \mathbf{Mod}'$ , i.e., a family of functions  $\mu_\Sigma^{Mod}: \mathbf{Mod}(\Sigma) \rightarrow \mathbf{Mod}'(\mu^{Sign}(\Sigma))$  natural in  $\Sigma \in |\mathbf{Sign}|$

such that for all  $\Sigma \in |\mathbf{Sign}|$ ,  $\varphi' \in \mathbf{Sen}'(\mu^{Sign}(\Sigma))$  and  $M \in \mathbf{Mod}(\Sigma)$ ,  $M \models_\Sigma \mu_\Sigma^{Sen}(\varphi')$  iff  $\mu_\Sigma^{Mod}(M) \models'_{\mu^{Sign}(\Sigma)} \varphi'$  (this is referred to as the satisfaction condition for  $\mu$ ). Institution morphisms compose in the obvious, component-wise manner [25].

Semantic entailment is preserved by translation under institution morphisms: for any signature  $\Sigma \in |\mathbf{Sign}|$  and sets of sentences  $\Phi', \Psi' \subseteq \mathbf{Sen}'(\mu^{Sign}(\Sigma))$ , if  $\Phi' \models' \Psi'$  then  $\mu_\Sigma^{Sen}(\Phi') \models \mu_\Sigma^{Sen}(\Psi')$ . If the translation of models  $\mu_\Sigma^{Mod}: \mathbf{Mod}(\Sigma) \rightarrow \mathbf{Mod}'(\mu^{Sign}(\Sigma))$  is surjective then also the opposite implication holds, and  $\Phi' \models' \Psi'$  iff  $\mu_\Sigma^{Sen}(\Phi') \models \mu_\Sigma^{Sen}(\Psi')$ .

For instance, there is an obvious institution morphism from the institution **FO** of first-order logic to the institution **PL** of propositional logic (removing from signatures everything but nullary predicates). For further examples of institution morphisms we refer to [35, 13].

In this paper we deal only with institution morphisms that leave the signature category intact, that is, where the signature functor is the identity. This also allows us to disregard institution comorphisms, since essentially they are the same as institution morphisms then.

An institution morphism  $\mu: \mathbf{INS} \rightarrow \mathbf{INS}'$  is *logically trivial* if it is the identity on signatures and surjective on sentences and models, that is,  $\mathbf{Sign}' = \mathbf{Sign}$  and  $\mu^{Sign} = id_{\mathbf{Sign}}$ , and for all signatures  $\Sigma \in |\mathbf{Sign}|$ , the functions  $\mu_{\Sigma}^{Sen}: \mathbf{Sen}'(\Sigma) \rightarrow \mathbf{Sen}(\Sigma)$  and  $\mu_{\Sigma}^{Mod}: \mathbf{Mod}(\Sigma) \rightarrow \mathbf{Mod}'(\Sigma)$  are surjective. The following fact justifies this terminology:<sup>6</sup>

► **Fact 4.** *Logically trivial institution morphisms identify only sentences and models that are logically equivalent.*

Special institution morphisms relate institutions with their extensions by new sentences and by new models, respectively, introduced in Sect. 2.3. If  $\mathbf{INS}_{NS}^+$  is the extension of  $\mathbf{INS}$  by new sentences  $\mathcal{NS}$  then there is an institution morphism  $\mu_{NS}: \mathbf{INS}_{NS}^+ \rightarrow \mathbf{INS}$ , where  $\mu_{NS}^{Sign}$  and  $\mu_{NS}^{Mod}$  are identities (the former is the identity functor on  $\mathbf{Sign}$ , the latter is the identity natural transformation on  $\mathbf{Mod}: \mathbf{Sign}^{op} \rightarrow \mathbf{Class}$ ), and for  $\Sigma \in |\mathbf{Sign}|$ ,  $\mu_{NS}^{Sen}: \mathbf{Sen}(\Sigma) \rightarrow \mathbf{Sen}_{NS}^+(\Sigma)$  are inclusions. Similarly, if  $\mathbf{INS}_{NM}^+$  is the extension of  $\mathbf{INS}$  by new models  $\mathcal{NM}$  then there is an institution morphism  $\mu_{NM}: \mathbf{INS} \rightarrow \mathbf{INS}_{NM}^+$ , where  $\mu_{NM}^{Sign}$  and  $\mu_{NM}^{Sen}$  are identities, and for  $\Sigma \in |\mathbf{Sign}|$ ,  $\mu_{NM}^{Mod}: \mathbf{Mod}(\Sigma) \rightarrow \mathbf{Mod}_{NM}^+(\Sigma)$  are inclusions.

► **Fact 5.** *Let  $\mathbf{INS}'$  and  $\mathbf{INS}''$  be institutions with a common signature category  $\mathbf{Sign}$ . Consider an institution morphism  $\mu: \mathbf{INS}' \rightarrow \mathbf{INS}''$  with  $\mu^{Sign} = id_{\mathbf{Sign}}$ . Then for some institution  $\mathbf{INS}$ , extension  $\mathbf{INS}_{NS}^+$  of  $\mathbf{INS}$  by new sentences, extension  $\mathbf{INS}_{NM}^+$  of  $\mathbf{INS}$  by new models, and logically trivial institution morphisms  $\mu': \mathbf{INS}' \rightarrow \mathbf{INS}_{NS}^+$  and  $\mu'': \mathbf{INS}_{NM}^+ \rightarrow \mathbf{INS}''$*

*we have  $\mu = \underbrace{\mu'; \mu_{NS}; \mu_{NM}; \mu''}_{\mu}: \mathbf{INS}' \xrightarrow{\mu'} \mathbf{INS}_{NS}^+ \xrightarrow{\mu_{NS}} \mathbf{INS} \xrightarrow{\mu_{NM}} \mathbf{INS}_{NM}^+ \xrightarrow{\mu''} \mathbf{INS}''$*

Proof (hint): Use  $\mathbf{INS} = \langle \mathbf{Sign}, \mathbf{Sen}'', \mathbf{Mod}', \langle \models_{\Sigma} \rangle_{\Sigma \in |\mathbf{Sign}|} \rangle$ , where for  $\Sigma \in \mathbf{Sign}$ ,  $M' \in \mathbf{Mod}'(\Sigma)$  and  $\varphi'' \in \mathbf{Sen}''(\Sigma)$ ,  $M' \models_{\Sigma} \varphi''$  iff  $M' \models_{\Sigma} \mu_{\Sigma}^{Sen}(\varphi')$ .

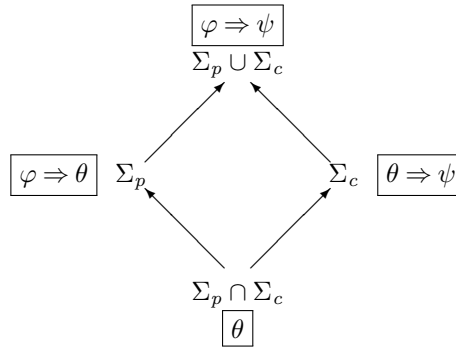
## 3 Interpolation

### 3.1 Classical interpolation

The well-known Craig interpolation theorem [12] states that if an implication between two first-order formulae  $\varphi \Rightarrow \psi$  holds then there is a formula  $\theta$  that uses only the symbols common to  $\varphi$  and  $\psi$  such that both  $\varphi \Rightarrow \theta$  and  $\theta \Rightarrow \psi$  hold;  $\theta$  is then called an *interpolant* for  $\varphi$  and  $\psi$ . This is one of the key properties of first-order logic, with numerous applications, including simpler proofs of similarly famous and important results like the Robinson consistency [32] and Beth definability [4] theorems. The interpolation property has been investigated (and proved or disproved) for many standard extensions (and fragments) of first-order logic [40] as well as for other logical systems, for instance for various modal and intuitionistic logics [21].

The above statement of the interpolation property implicitly involves the following union/intersection square of signatures:

<sup>6</sup> Due to the page limit imposed, proofs are either omitted here or reduced to hints only.



where  $\Sigma_p$  and  $\Sigma_c$  are (first-order) signatures for  $\varphi$  and  $\psi$ , respectively, and the arrows indicate signature inclusions.

As hinted at in Sect. 1, interpolation proved indispensable for many foundational aspects of computer science, in particular in the area of software specification and development. However, the classical formulation of Craig’s interpolation for many applications requires some generalisations, which perhaps do not bring much new insight in the framework of first-order logic, but may be important when other logical systems are considered.

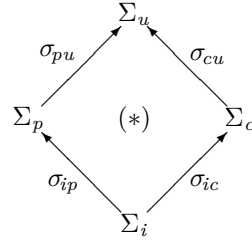
To begin with, the use of implication should be replaced by entailment. Then, we should deal with entailments between sets of sentences, rather than between individual sentences (strictly speaking, this is needed for the premise  $\varphi$  and especially for the interpolant  $\theta$  – for notational symmetry, we do this for the conclusion  $\psi$  as well). Both these generalisations are irrelevant for first-order logic, where implication captures semantic entailment, and a set of sentences in the premise of each single-conclusion entailment may always be replaced by a single sentence (since the logic is compact and has conjunction). However, for instance, working in equational logic we have no implication available, and interpolants cannot be always expressed as a single equation – even though the interpolation property holds if sets of equations are permitted as interpolants [33].

Perhaps most importantly, for instance in applications where parameterised specifications and their “pushout-style” instantiations [42, 20] are involved, we have to go beyond union/intersection squares of signatures and inclusions to relate the signatures. More general classes of signature squares are needed, with non-injective signature morphisms necessary to capture for instance morphisms used to “fit” actual to formal parameters. Typically in applications at least pushouts of signature morphisms are involved, sometimes additionally restricted to indicated classes of morphisms permitted at the “bottom-left” and “bottom-right” of the squares, respectively [42, 41, 5, 13, 30]. However, for the purposes of this paper we will consider interpolation properties for an arbitrary commutative square of signature morphisms.

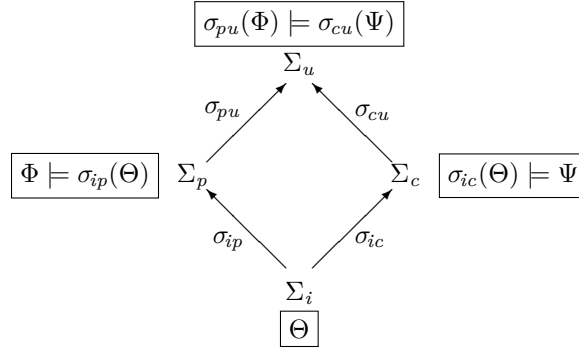
### 3.2 Interpolation in an institution

Throughout the rest of this paper, let  $\mathbf{INS} = \langle \mathbf{Sign}, \mathbf{Sen}, \mathbf{Mod}, \langle \models_{\Sigma} \rangle_{\Sigma \in |\mathbf{Sign}|} \rangle$  be an arbitrary institution, and we study interpolation properties over the following commutative square (\*) of signature morphisms:<sup>7</sup>

<sup>7</sup> To help memorising the notation:  $p$  for *premise*,  $c$  for *conclusion*,  $u$  for *union* and  $i$  for *intersection* (or *interpolant*).



Let  $\Phi \subseteq \mathbf{Sen}(\Sigma_p)$  and  $\Psi \subseteq \mathbf{Sen}(\Sigma_c)$  be such that  $\sigma_{pu}(\Phi) \models_{\Sigma_u} \sigma_{cu}(\Psi)$ . An *interpolant* for  $\Phi$  and  $\Psi$  (over diagram  $(*)$ ) is a set  $\Theta \subseteq \mathbf{Sen}(\Sigma_i)$  of  $\Sigma_i$ -sentences such that  $\Phi \models_{\Sigma_p} \sigma_{ip}(\Theta)$  and  $\sigma_{ic}(\Theta) \models_{\Sigma_c} \Psi$ .



To simplify some further statements, if  $\sigma_{pu}(\Phi) \not\models_{\Sigma_u} \sigma_{cu}(\Psi)$  then we say that any set  $\Theta \subseteq \mathbf{Sen}(\Sigma_i)$  is an interpolant for  $\Phi$  and  $\Psi$  (over diagram  $(*)$ ).

We say that a commutative square  $(*)$  of signature morphisms *admits interpolation* if there is an interpolant for every  $\Phi \subseteq \mathbf{Sen}(\Sigma_p)$  and  $\Psi \subseteq \mathbf{Sen}(\Sigma_c)$  such that  $\sigma_{pu}(\Phi) \models_{\Sigma_u} \sigma_{cu}(\Psi)$ .

► **Example 6.** In the institution **FO** of first-order logic, and in any of its variants mentioned in Example 1, if the square  $(*)$  is a pushout and at least one of  $\sigma_{ip}: \Sigma_i \rightarrow \Sigma_p$ ,  $\sigma_{ic}: \Sigma_i \rightarrow \Sigma_c$  is injective on sorts then  $(*)$  admits interpolation; otherwise interpolation may fail for  $(*)$  (see [6]). In the institution **EQ** of equational logic if the square  $(*)$  is a pushout and  $\sigma_{ic}: \Sigma_i \rightarrow \Sigma_c$  is injective then  $(*)$  admits interpolation; otherwise interpolation may fail for  $(*)$ , and in **EQ<sub>∅</sub>**, where empty carriers are permitted, interpolation may fail even for union/intersection squares of signatures (see [39]). In the institution **PL** of propositional logic, all pushouts admit interpolation.

It is well known that the interpolation property of a logical system is fragile. When the logic is extended, when new models or sentences are added, the interpolation property may easily be spoiled. Clearly, this may happen when entirely new signatures are added, with new models and sentences over them. Therefore, in this paper we consider the category of signatures to be fixed, and consider only such extensions of institutions that preserve it.

Throughout the rest of the paper we study how the interpolation property may be spoiled by adding new models or sentences. This will be done from a “local” perspective, for particular commutative squares of signature morphisms, as well as for particular interpolants.

We say that an interpolant  $\Theta \subseteq \mathbf{Sen}(\Sigma_i)$  for  $\Phi \subseteq \mathbf{Sen}(\Sigma_p)$  and  $\Psi \subseteq \mathbf{Sen}(\Sigma_c)$  (over diagram  $(*)$ ) is *stable* under extensions of the institution by models if for every extension  $\mathbf{INS}^+$  of  $\mathbf{INS}$  by new models,  $\Theta$  is an interpolant for  $\Phi$  and  $\Psi$  in  $\mathbf{INS}^+$ ; otherwise we say that the interpolant  $\Theta$  is *fragile*. Note that adding new sentences cannot spoil a particular interpolant, but may spoil interpolation property for a given diagram.

### 3.3 Interpolants may be stable

► **Lemma 7.** *Consider the diagram (\*) of signature morphisms.*

1. *If  $\sigma_{ip}: \Sigma_i \rightarrow \Sigma_p$  is such that  $\mathbf{Sen}(\sigma_{ip}): \mathbf{Sen}(\Sigma_i) \rightarrow \mathbf{Sen}(\Sigma_p)$  is surjective and  $\sigma_{cu}: \Sigma_c \rightarrow \Sigma_u$  is conservative then (\*) admits interpolation.*
2. *If  $\sigma_{ic}: \Sigma_i \rightarrow \Sigma_c$  is such that  $\mathbf{Sen}(\sigma_{ic}): \mathbf{Sen}(\Sigma_i) \rightarrow \mathbf{Sen}(\Sigma_c)$  is surjective and  $\sigma_{pu}: \Sigma_p \rightarrow \Sigma_u$  is conservative then (\*) admits interpolation.*

Proof (hint): *An interpolant for  $\Phi \subseteq \mathbf{Sen}(\Sigma_p)$  and  $\Psi \subseteq \mathbf{Sen}(\Sigma_c)$  is  $\sigma_{ip}^{-1}(\Phi)$  under 1., or  $\sigma_{ic}^{-1}(\Psi)$  under 2.*

A trivial special case here is when  $\sigma_{ip}$  and  $\sigma_{cu}$ , or  $\sigma_{ic}$  and  $\sigma_{pu}$ , are isomorphisms, which can be further refined as follows:

► **Corollary 8.** *The diagram (\*) of signature morphisms admits interpolation if*

1.  *$\sigma_{ip}: \Sigma_i \rightarrow \Sigma_p$  is a retraction and  $\sigma_{cu}: \Sigma_c \rightarrow \Sigma_u$  is a coretraction, or*
2.  *$\sigma_{ic}: \Sigma_i \rightarrow \Sigma_c$  is a retraction and  $\sigma_{pu}: \Sigma_p \rightarrow \Sigma_u$  is a coretraction.*

Proof (hint): *The requirements here imply the respective conditions in Lemma 7.*

This shows that if the signature morphisms in (\*) satisfy the premises of Cor. 8 then the diagram enjoys a stable interpolation property, which cannot be spoiled by any institution extension that leaves the category of signatures unchanged! No matter how we add new models or sentences, the interpolation property is ensured by the properties of the signature morphisms involved, and the implied properties of the translations of sentences and reducts of models they induce in the institution and in any of its extensions.

The conditions stated in Cor. 8 are in fact quite strong and in many practical situations do not depart too far from the trivial case when  $\Sigma_p$  is (up to isomorphism) included in  $\Sigma_c$  or vice versa. Namely, when the diagram (\*) is a pushout then condition 1. implies that  $\sigma_{cu}: \Sigma_c \rightarrow \Sigma_u$  is an isomorphism, and condition 2. implies that  $\sigma_{pu}: \Sigma_p \rightarrow \Sigma_u$  is an isomorphism. Dually, when (\*) is a pullback then condition 1. implies that  $\sigma_{ip}: \Sigma_i \rightarrow \Sigma_p$  is an isomorphism, and condition 2. implies that  $\sigma_{ic}: \Sigma_i \rightarrow \Sigma_c$  is an isomorphism.

► **Fact 9.** *Let  $\mu: \mathbf{INS} \rightarrow \mathbf{INS}'$  be a logically trivial institution morphism. Diagram (\*) in the category of signatures admits interpolation in  $\mathbf{INS}$  iff it admits interpolation in  $\mathbf{INS}'$ .*

Facts 5 and 9 imply that for our study of the fragility of interpolation institution extensions by new models and by new sentences are of primary importance.

## 4 Spoiling an interpolant by new models

Recall that we study interpolation over a commutative square of signature morphisms (\*) in an institution  $\mathbf{INS} = \langle \mathbf{Sign}, \mathbf{Sen}, \mathbf{Mod}, \langle \models_{\Sigma} \rangle_{\Sigma \in |\mathbf{Sign}|} \rangle$ . Throughout this section, let  $\Phi \subseteq \mathbf{Sen}(\Sigma_p)$  and  $\Psi \in \mathbf{Sen}(\Sigma_c)$  be such that  $\sigma_{pu}(\Phi) \models \sigma_{cu}(\Psi)$ , and let  $\Theta \subseteq \mathbf{Sen}(\Sigma_i)$  be an interpolant for  $\Phi$  and  $\Psi$  in  $\mathbf{INS}$ .

► **Lemma 10.** *Suppose that there exists a set of  $\Sigma_p$ -sentences  $\Phi^{\bullet} \supseteq \Phi$  such that  $\sigma_{ip}(\Theta) \not\subseteq \Phi^{\bullet}$  and for all signature morphisms  $\tau: \Sigma_u \rightarrow \Sigma_p$ , if  $\tau(\sigma_{pu}(\Phi)) \subseteq \Phi^{\bullet}$  then  $\tau(\sigma_{cu}(\Psi)) \subseteq \Phi^{\bullet}$ . Then the interpolant  $\Theta$  for  $\Phi$  and  $\Psi$  is not stable under extensions of  $\mathbf{INS}$  by models.*

Proof (hint): *Extend  $\mathbf{INS}$  by a new  $\Sigma_p$ -model  $M$  with  $\text{Th}^+(M) = \Phi^{\bullet}$ . Then still  $\sigma_{pu}(\Phi) \models^+ \sigma_{cu}(\Psi)$ , but  $\Phi \not\models^+ \sigma_{ip}(\Theta)$ .*

The key property of the set  $\Phi^{\bullet}$  in the above lemma is that it cannot be used to separate  $\sigma_{pu}(\Phi)$  from  $\sigma_{cu}(\Psi)$  via any morphism  $\tau: \Sigma_u \rightarrow \Sigma_p$ . More formally, for any signatures  $\Sigma, \Sigma' \in |\mathbf{Sign}|$ , we say that  $\Upsilon \subseteq \mathbf{Sen}(\Sigma)$  never separates  $\Phi' \subseteq \mathbf{Sen}(\Sigma')$  from  $\Psi' \subseteq \mathbf{Sen}(\Sigma')$



when for all morphisms  $\tau: \Sigma' \rightarrow \Sigma$ , if  $\tau(\Phi') \subseteq \Upsilon$  then  $\tau(\Psi') \subseteq \Upsilon$ . For any set  $\Phi \subseteq \mathbf{Sen}(\Sigma)$ , we denote by  $[\Phi \xrightarrow[\Sigma]{\Sigma'} \Psi'](\Phi)$  the least set of  $\Sigma$ -sentences that contains  $\Phi$  and never separates  $\Phi'$  from  $\Psi'$  (it exists since the family of such sets is closed under intersection and is nonempty).

► **Corollary 11.** *If  $\sigma_{ip}(\Theta) \not\subseteq [\sigma_{pu}(\Phi) \xrightarrow[\Sigma_p]{\Sigma_u} \sigma_{cu}(\Psi)](\Phi)$  then the interpolant  $\Theta$  for  $\Phi$  and  $\Psi$  is not stable under extensions of **INS** by models.*

► **Lemma 12.** *Suppose that there exists a set of  $\Sigma_c$ -sentences  $\Psi^\circ \subseteq \mathbf{Sen}(\Sigma_c)$  such that  $\Psi \cap \Psi^\circ \neq \emptyset$ ,  $\sigma_{ic}(\Theta) \cap \Psi^\circ = \emptyset$  and for all signature morphisms  $\tau: \Sigma_u \rightarrow \Sigma_c$ , if  $\tau(\sigma_{cu}(\Psi)) \cap \Psi^\circ \neq \emptyset$  then  $\tau(\sigma_{pu}(\Phi)) \cap \Psi^\circ \neq \emptyset$ . Then the interpolant  $\Theta$  for  $\Phi$  and  $\Psi$  is not stable under extensions of **INS** by models.*

Proof (hint): *Extend **INS** by a new  $\Sigma_c$ -model  $N$  with  $Th^+(N) = \mathbf{Sen}(\Sigma_c) \setminus \Psi^\circ$ . Then still  $\sigma_{pu}(\Phi) \models^+ \sigma_{cu}(\Psi)$ , but  $\sigma_{ip}(\Theta) \not\models^+ \Psi$ .*

To refine Lemma 12 in the style of Cor. 11, notice that the requirement on  $\Psi^\circ \subseteq \mathbf{Sen}(\Sigma_c)$  that for  $\tau: \Sigma_u \rightarrow \Sigma_c$ , if  $\tau(\sigma_{cu}(\Psi)) \cap \Psi^\circ \neq \emptyset$  then  $\tau(\sigma_{pu}(\Phi)) \cap \Psi^\circ \neq \emptyset$ , means that the set  $\mathbf{Sen}(\Sigma_c) \setminus \Psi^\circ$  never separates  $\sigma_{pu}(\Phi)$  from  $\sigma_{cu}(\Psi)$ .

► **Corollary 13.** *If  $\Psi \not\subseteq [\sigma_{pu}(\Phi) \xrightarrow[\Sigma_c]{\Sigma_u} \sigma_{cu}(\Psi)](\sigma_{ic}(\Theta))$  then the interpolant  $\Theta$  for  $\Phi$  and  $\Psi$  is not stable under extension of **INS** by models.*

Corollaries 11 and 13 present sufficient conditions that ensure that a particular interpolant may be spoiled by an extension of the institution by new models. In fact, these conditions jointly are also necessary:

► **Theorem 14.** *The interpolant  $\Theta$  for  $\Phi$  and  $\Psi$  is stable under extensions of **INS** by models if and only if the following conditions hold:*

1.  $\sigma_{ip}(\Theta) \subseteq [\sigma_{pu}(\Phi) \xrightarrow[\Sigma_p]{\Sigma_u} \sigma_{cu}(\Psi)](\Phi)$ , and
2.  $\Psi \subseteq [\sigma_{pu}(\Phi) \xrightarrow[\Sigma_c]{\Sigma_u} \sigma_{cu}(\Psi)](\sigma_{ic}(\Theta))$ .

Proof (hint): *In any extension **INS**<sup>+</sup> of **INS** by models such that  $\sigma_{pu}(\Phi) \models^+ \sigma_{cu}(\Psi)$ , if  $\Phi \not\models^+ \sigma_{ip}(\Theta)$  then 1. fails, and if  $\sigma_{ic}(\Theta) \not\models^+ \Psi$  then 2. fails, which proves the “if” part. The “only if” part follows by Corollaries 11 and 13.*

The above theorem gives precise conditions that ensure stability of a particular interpolant under extensions of the institution by new models. Equivalently, this is a precise characterisation of specific interpolation properties that can be spoiled by adding new abstract models. It should be stressed that the conditions in use are purely “syntactic” – they do not refer to the semantic properties of the sets of sentences involved, and depend on a specific syntactic form of the sentences, and the conclusions may change when the sentences considered are replaced by semantically equivalent sentences that are of a different syntactic form.

► **Example 15.** Consider a trivial example in the institution **PL** of propositional logic. In the diagram (\*), let  $\Sigma_p = \{p, r\}$ ,  $\Sigma_c = \{p, q\}$ ,  $\Sigma_u = \Sigma_p \cup \Sigma_c = \{r, p, q\}$ ,  $\Sigma_i = \Sigma_p \cap \Sigma_c = \{p\}$ , and the four signature morphisms are inclusions.

Let  $\varphi$  be  $r \wedge p$  and  $\psi$  be  $p \vee q$ .<sup>8</sup> Clearly,  $\varphi \models \psi$ , and  $\varphi$  and  $\psi$  have a number of distinct interpolants in **PL**. One such interpolant for  $\varphi$  and  $\psi$  is  $p$ . Consider the **PL**-model  $M = \{r\} \in \mathbf{Mod}_{\mathbf{PL}}(\Sigma_p)$ . Let **PL**<sup>+</sup> be an extension of **PL** by a new  $\Sigma_p$ -model  $\tilde{M}$  (with

<sup>8</sup> When convenient, we write  $\varphi$  for  $\{\varphi\}$ , relying on the context to impose such identification of a sentence with the one-element set that contains it.

## 8:12 Interpolation Is (Not Always) Easy to Spoil

interpretation of propositional sentences “swapping” the valuation of propositional variables, as in Example 3). Then  $\widetilde{M} \models^+ r \wedge p$  while  $\widetilde{M} \not\models^+ p$ , and so  $p$  is not an interpolant for  $\varphi$  and  $\psi$  in  $\mathbf{PL}^+$ . In fact,  $\Phi^\bullet = \{\varphi \in \mathbf{Sen}_{\mathbf{PL}}(\Sigma_p) \mid \widetilde{M} \models^+ \varphi\}$  satisfies the premises of Lemma 10.

Moreover, one can easily calculate that  $[r \wedge p \xrightarrow[\Sigma_p]{\Sigma_u} p \vee q](r \wedge p) = \{r \wedge p, p \vee r, p \vee p\} \subseteq \mathbf{Sen}_{\mathbf{PL}}(\Sigma_p)$  (there are exactly two morphisms from  $\Sigma_u$  to  $\Sigma_p$  that map  $r \wedge p$  to  $r \wedge p$ , they are identities on  $\{p, r\}$  and map  $q$  to any of the symbols in  $\Sigma_p$ ). Thus, by Cor. 11, any interpolant for  $\varphi$  and  $\psi$  other than  $p \vee p$  may be spoiled by extending  $\mathbf{PL}$  by new models.

Indeed,  $p \vee p$  is an interpolant for  $\varphi$  and  $\psi$ . Since no morphism from  $\Sigma_u$  to  $\Sigma_c$  maps  $r \wedge p$  to  $p \vee p$ , we have  $[r \wedge p \xrightarrow[\Sigma_c]{\Sigma_u} p \vee q](p \vee p) = \{p \vee p\} \subseteq \mathbf{Sen}_{\mathbf{PL}}(\Sigma_c)$ , and so by Cor. 13 in some extension of  $\mathbf{PL}$  by new models  $p \vee p$  is not an interpolant for  $\varphi$  and  $\psi$ . For instance, consider  $\mathbf{PL}$ -model  $N = \{q\} \in \mathbf{Mod}_{\mathbf{PL}}(\Sigma_c)$ . Let  $\mathbf{PL}^+$  be the extension of  $\mathbf{PL}$  by a new  $\Sigma_c$ -model  $\widetilde{N}$  (with interpretation of propositional sentences “swapping” the valuation of propositional variables, as in Example 3). Then  $\widetilde{N} \models^+ p \vee p$  while  $\widetilde{N} \not\models^+ p \vee q$ , and so  $p \vee p$  is not an interpolant for  $\varphi$  and  $\psi$  in  $\mathbf{PL}^+$ . Summing up: none of the interpolants for  $\varphi$  and  $\psi$  in  $\mathbf{PL}$  is stable under extensions of  $\mathbf{PL}$  by new models.

Let now  $\varphi'$  be  $(p \vee r) \wedge (p \vee \neg r)$  and  $\psi'$  be  $(p \vee q) \wedge (p \vee \neg p)$ . Perhaps the most obvious interpolant for  $\varphi'$  and  $\psi'$  is  $p$ . This interpolant, however, is fragile. Namely,

$$[\varphi' \xrightarrow[\Sigma_p]{\Sigma_u} \psi']((p \vee r) \wedge (p \vee \neg r)) = \{(p \vee r) \wedge (p \vee \neg r), (p \vee p) \wedge (p \vee \neg p)\} \subseteq \mathbf{Sen}_{\mathbf{PL}}(\Sigma_p).$$

Thus, by Cor. 11,  $p$  is not an interpolant for  $\varphi'$  and  $\psi'$  in an extension of  $\mathbf{PL}$  by new models.

Another interpolant for  $\varphi'$  and  $\psi'$  in  $\mathbf{PL}$  is  $(p \vee p) \wedge (p \vee \neg p)$ . Since  $(p \vee p) \wedge (p \vee \neg p) \in [\varphi' \xrightarrow[\Sigma_p]{\Sigma_u} \psi']((p \vee r) \wedge (p \vee \neg r))$ , Cor. 11 cannot be used here to conclude that this interpolant gets spoiled in an extension of  $\mathbf{PL}$  by new models. Moreover,

$$[\varphi' \xrightarrow[\Sigma_c]{\Sigma_u} \psi']((p \vee p) \wedge (p \vee \neg p)) = \{(p \vee p) \wedge (p \vee \neg p), (p \vee q) \wedge (p \vee \neg q)\} \subseteq \mathbf{Sen}_{\mathbf{PL}}(\Sigma_c).$$

Consequently, Cor. 13 does not apply here either, and by Thm. 14 the interpolant  $(p \vee p) \wedge (p \vee \neg p)$  for  $\varphi'$  and  $\psi'$  in  $\mathbf{PL}$  is stable under extensions of  $\mathbf{PL}$  by new models.

## 5 Spoiling interpolation by new models

As in the previous section, consider an institution  $\mathbf{INS} = \langle \mathbf{Sign}, \mathbf{Sen}, \mathbf{Mod}, \langle \models_\Sigma \rangle_{\Sigma \in |\mathbf{Sign}|}, \text{commutative square of signature morphisms } (*), \text{ and sets of sentences } \Phi \subseteq \mathbf{Sen}(\Sigma_c) \text{ and } \Psi \in \mathbf{Sen}(\Sigma_c) \text{ such that } \sigma_{pu}(\Phi) \models \sigma_{cu}(\Psi) \rangle$ . Theorem 14 gives the exact characterisation of interpolants that are stable under extensions of  $\mathbf{INS}$  by new models. Of course, this also characterises interpolants that are fragile. In this section we characterise situations where all interpolants for the premise  $\Phi$  and conclusion  $\Psi$  may be spoiled at once.

► **Corollary 16.** *Let  $\Phi^* = [\sigma_{pu}(\Phi) \xrightarrow[\Sigma_p]{\Sigma_u} \sigma_{cu}(\Psi)](\Phi)$  and  $\Psi^* = [\sigma_{pu}(\Phi) \xrightarrow[\Sigma_c]{\Sigma_u} \sigma_{cu}(\Psi)](\sigma_{ic}(\sigma_{ip}^{-1}(\Phi^*)))$ .*

*If  $\Psi \not\subseteq \Psi^*$  then there is an extension  $\mathbf{INS}^+$  of  $\mathbf{INS}$  by models such that there is no interpolant for  $\Phi$  and  $\Psi$  in  $\mathbf{INS}^+$ .*

*Proof (hint): Extend  $\mathbf{INS}$  by a new  $\Sigma_p$ -model  $M$  with  $Th(M)^+ = \Phi^*$  and a new  $\Sigma_c$ -model  $N$  with  $Th^+(N) = \Psi^*$ .*

The converse of Cor. 16 does not hold, since the conclusion follows as well when we limit our attention to consequences of  $\Phi$ , rather than all sentences in  $\Phi^* = [\sigma_{pu}(\Phi) \xrightarrow[\Sigma_p]{\Sigma_u} \sigma_{cu}(\Psi)](\Phi)$ .

To avoid repetition, for the rest of this section let  $\Theta^* = \sigma_{ip}^{-1}([\sigma_{pu}(\Phi) \xrightarrow[\Sigma_p]{\Sigma_u} \sigma_{cu}(\Psi)](\Phi) \cap Th(\Phi))$  (that is, more explicitly:  $\Theta^* = \{\theta \in \mathbf{Sen}(\Sigma_i) \mid \sigma_{ip}(\theta) \in [\sigma_{pu}(\Phi) \xrightarrow[\Sigma_p]{\Sigma_u} \sigma_{cu}(\Psi)](\Phi), \Phi \models \sigma_{ip}(\theta)\}$ ).

► **Lemma 17.** *If  $\Psi \not\subseteq [\sigma_{pu}(\Phi) \xrightarrow[\Sigma_c]{\Sigma_u} \sigma_{cu}(\Psi)](\sigma_{ic}(\Theta^*))$  then no interpolant for  $\Phi$  and  $\Psi$  is stable under extensions of **INS** by models.*

Proof (hint): *For any interpolant  $\Theta$  for  $\Phi$  and  $\Psi$ , if  $\Theta \not\subseteq \Theta^*$  then  $\Theta$  is not stable by Cor. 11, and if  $\Theta \subseteq \Theta^*$  by Cor. 13.*

The thesis of Lemma 17 seems weaker than that of Cor. 11 – but only superficially so:

► **Lemma 18.** *If no interpolant for  $\Phi$  and  $\Psi$  is stable under extensions of **INS** by models then in some extension of **INS** by models  $\Phi$  and  $\Psi$  have no interpolant at all.*

► **Corollary 19.** *If  $\Psi \not\subseteq [\sigma_{pu}(\Phi) \xrightarrow[\Sigma_c]{\Sigma_u} \sigma_{cu}(\Psi)](\sigma_{ic}(\Theta^*))$  then in some extension of **INS** by models  $\Phi$  and  $\Psi$  have no interpolant at all.*

► **Theorem 20.** *There is an interpolant for  $\Phi$  and  $\Psi$  in every extension of **INS** by models if and only if  $\Psi \subseteq [\sigma_{pu}(\Phi) \xrightarrow[\Sigma_c]{\Sigma_u} \sigma_{cu}(\Psi)](\sigma_{ic}(\Theta^*))$  and  $\sigma_{ic}(\Theta^*) \models \Psi$ .*

Proof (hint): *For the “if” part: under the assumptions,  $\Theta^*$  is a stable interpolant for  $\Phi$  and  $\Psi$ . For the “only if” part: any stable interpolant  $\Theta$  for  $\Phi$  and  $\Psi$  satisfies  $\Theta \subseteq \Theta^*$ .*

► **Example 21.** Recall Example 15. As argued there, every interpolant for  $r \wedge p$  and  $p \vee q$  in **PL** is fragile. Theorem 20 leads to the same conclusion, of course. Namely, as in Example 15,  $[r \wedge p \xrightarrow[\Sigma_p]{\Sigma_u} q \vee p](r \wedge p) = \{r \wedge p, p \vee r, p \vee p\}$ . Then, using the notation  $\Theta^*$  defined above for the case at hand,  $\Theta^* = \{p \vee p\}$ . Recalling from Example 15 again:  $[r \wedge p \xrightarrow[\Sigma_c]{\Sigma_u} p \vee q](\Theta^*) = \{p \vee p\}$ , and so  $p \vee q \notin [r \wedge p \xrightarrow[\Sigma_c]{\Sigma_u} p \vee q](\Theta^*)$ . Thus, by Thm. 20 and Lemma 18, there is an extension of **PL** by models in which  $r \wedge p$  and  $p \vee q$  have no interpolant.

As in Example 15, let now  $\varphi'$  be  $(p \vee r) \wedge (p \vee \neg r)$  and  $\psi'$  be  $(p \vee q) \wedge (p \vee \neg q)$ , and we get  $[\varphi' \xrightarrow[\Sigma_p]{\Sigma_u} \psi'](\varphi') = \{(p \vee r) \wedge (p \vee \neg r), (p \vee p) \wedge (p \vee \neg p)\}$ . Therefore, using the notation  $\Theta^*$  for the current case,  $\Theta^* = \{(p \vee p) \wedge (p \vee \neg p)\}$ , and then  $(p \vee q) \wedge (p \vee \neg q) \in [\varphi' \xrightarrow[\Sigma_p]{\Sigma_u} \psi'](\Theta^*)$ . Thus, by Thm. 20,  $(p \vee r) \wedge (p \vee \neg r)$  and  $(p \vee q) \wedge (p \vee \neg q)$  have an interpolant in every extension of **PL** by models, and indeed, in Example 15 we argued independently that  $(p \vee p) \wedge (p \vee \neg p)$  is such an interpolant.

## 6 Spoiling interpolation by new sentences

As before, in an institution  $\mathbf{INS} = \langle \mathbf{Sign}, \mathbf{Sen}, \mathbf{Mod}, \langle \models_{\Sigma} \rangle_{\Sigma \in |\mathbf{Sign}|} \rangle$  we study interpolation over a commutative square of signature morphisms (\*).

Changes to a logical system that may arise when new sentences are introduced are in no sense dual to those resulting from extending the logical system by new models. In particular, new sentences do not modify the entailments between the sentences of the original system, so we cannot expect that we may spoil interpolants for old sentences. However, new sentences (over the premise and conclusion signatures) may lead to new entailments  $\sigma_{pu}(\Phi) \models^+ \sigma_{cu}(\Psi)$  with no interpolant for  $\Phi$  and  $\Psi$ . On the other hand, adding appropriate new sentences (over the interpolant signature) may restore (or establish) the interpolation property.

## 8:14 Interpolation Is (Not Always) Easy to Spoil

The first rough idea (see for instance the semantic characterisation of interpolation in [13]) is that to spoil interpolation for the diagram (\*), we look for a class  $\mathcal{K} \subseteq \mathbf{Mod}(\Sigma_i)$  that is not definable in **INS**, and then build an extension **INS**<sup>+</sup> of **INS** by new sentences  $\varphi \in \mathbf{Sen}^+(\Sigma_p)$  and  $\psi \in \mathbf{Sen}^+(\Sigma_c)$  such that  $Mod^+(\varphi) = \mathcal{K}|_{\sigma_p}^{-1}$  and  $Mod^+(\psi) = \mathcal{K}|_{\sigma_c}^{-1}$ . Then  $\sigma_{pu}(\varphi) \models^+ \sigma_{cu}(\psi)$ , and it may seem that there should be no interpolant for  $\varphi$  and  $\psi$ , since such an interpolant would have to define  $\mathcal{K}$ . However, the latter need not be true in general.

One technical nuance is that a set  $\Theta \subseteq \mathbf{Sen}^+(\Sigma_i)$  of sentences may be an interpolant for  $\varphi$  and  $\psi$  if  $Mod^+(\Theta) \supset \mathcal{K}$  but no model in  $Mod^+(\Theta) \setminus \mathcal{K}$  has a  $\sigma_{ic}$ -expansion to a model in  $\mathbf{Mod}(\Sigma_c)$ . Another technicality is that the requirement that  $Mod^+(\varphi) = \mathcal{K}|_{\sigma_p}^{-1}$  may be weakened to  $Mod^+(\varphi)|_{\sigma_p} = \mathcal{K}$ . At the conclusion side, it is enough to assume that all  $\sigma_{ic}$ -expansions of the models in  $\mathcal{K}$  are in  $Mod(\psi)$ ,  $\mathcal{K}|_{\sigma_c}^{-1} \subseteq Mod(\psi)$ , or equivalently, no model in  $\mathcal{K}$  is a  $\sigma_{ic}$ -reduct of a  $\Sigma_c$ -model outside  $Mod(\psi)$ ,  $\mathcal{K} \subseteq \mathbf{Mod}(\Sigma_i) \setminus ((\mathbf{Mod}(\Sigma_c) \setminus Mod(\psi))|_{\sigma_c})$ . We may also permit a gap between  $Mod^+(\varphi)|_{\sigma_p}$  and  $\mathbf{Mod}(\Sigma_i) \setminus ((\mathbf{Mod}(\Sigma_c) \setminus Mod(\psi))|_{\sigma_c})$  as long as no definable class separates them.

Most importantly though, adding new sentences over signatures  $\Sigma_p$  and  $\Sigma_c$  may result in adding new  $\Sigma_i$ -sentences (as translations of the added sentences), and some  $\Sigma_i$ -model classes that are not definable in **INS** may become definable in **INS**<sup>+</sup>. The following notion will be used to take care of this: for any signature  $\Sigma \in |\mathbf{Sign}|$  and collection  $\mathcal{F} = \{\langle \Sigma_j, \mathcal{M}_j \rangle \mid \Sigma_j \in |\mathbf{Sign}|, \mathcal{M}_j \subseteq \mathbf{Mod}(\Sigma_j), j \in \mathcal{J}\}$ ,<sup>9</sup> we say that a class  $\mathcal{M} \subseteq \mathbf{Mod}(\Sigma)$  of  $\Sigma$ -models is *definable in **INS** from  $\mathcal{F}$*  if for some family of signature morphisms  $\tau_l: \Sigma_{j_l} \rightarrow \Sigma$  where  $j_l \in \mathcal{J}$ ,  $l \in \mathcal{L}$ , and a set  $\Phi \subseteq \mathbf{Sen}(\Sigma)$  of  $\Sigma$ -sentences we have  $\mathcal{M} = \bigcap_{l \in \mathcal{L}} \mathcal{M}_{j_l}|_{\tau_l}^{-1} \cap Mod(\Phi)$ .

► **Theorem 22.** *There is an extension **INS**<sup>+</sup> of **INS** by new sentences in which the diagram (\*) does not admit interpolation if and only if there are classes of models  $\mathcal{M} \subseteq \mathbf{Mod}(\Sigma_p)$  and  $\mathcal{N} \subseteq \mathbf{Mod}(\Sigma_c)$  such that*

1.  $\mathcal{M}|_{\sigma_{pu}}^{-1} \subseteq \mathcal{N}|_{\sigma_{cu}}^{-1}$  and
2. *no class of models  $\mathcal{K} \subseteq \mathbf{Mod}(\Sigma_i)$  such that  $\mathcal{M}|_{\sigma_p} \subseteq \mathcal{K}$  and  $\mathcal{K}|_{\sigma_c}^{-1} \subseteq \mathcal{N}$  is definable in **INS** from  $\{\langle \Sigma_p, \mathcal{M} \rangle, \langle \Sigma_c, \mathcal{N} \rangle\}$ .*

Proof (hint): For the “if” part, extend **INS** by new sentences that define  $\mathcal{M}$  and  $\mathcal{N}$ , respectively. For the “only if” part, if in an extension **INS**<sup>+</sup> of **INS** by new sentences there is no interpolant for  $\Phi^+ \subseteq \mathbf{Sen}^+(\Sigma_p)$  and  $\Psi^+ \subseteq \mathbf{Sen}^+(\Sigma_c)$  then  $\mathcal{M} = Mod(\Phi^+)$  and  $\mathcal{N} = Mod(\Psi^+)$  satisfy 1. and 2.

► **Example 23.** Consider an example in the institution **FO<sub>EQ</sub>** of first-order logic with equality. Let all the signatures in the diagram (\*) extend  $\Sigma_i$ , which has sort *Nat*, constant 0: *Nat* and operation  $s: \text{Nat} \rightarrow \text{Nat}$ . In addition,  $\Sigma_p$  has  $bop: \text{Nat} \times \text{Nat} \rightarrow \text{Nat}$  and  $\Sigma_c$  has  $\_ + \_ : \text{Nat} \times \text{Nat} \rightarrow \text{Nat}$ . Finally,  $\Sigma_u = \Sigma_p \cup \Sigma_c$ , and all morphisms in (\*) are inclusions.

Let  $\mathcal{M} \subseteq \mathbf{Mod}(\Sigma_p)$  be the class of all models with the carrier set freely generated by 0 and  $s$  (where each element is the value of exactly one of the terms of the form  $s^n(0)$ ). Let then  $\mathcal{N} \subseteq \mathbf{Mod}(\Sigma_c)$  be the class of models that satisfy the following implication:

$$\psi \equiv (\forall x, y: \text{Nat}. x + 0 = x \wedge x + s(y) = s(x + y)) \Rightarrow \forall x, y: \text{Nat}. x + y = y + x$$

Let **FO<sub>EQ</sub>**<sup>+</sup> be the extension of **FO<sub>EQ</sub>** by a new  $\Sigma_p$ -sentence  $\varphi$  (and its formal translations) such that  $Mod^+(\varphi) = \mathcal{M}$ . No  $\Sigma_c$ -sentence is added, since  $\mathcal{N}$  is already definable in **FO<sub>EQ</sub>**. Clearly,  $\mathcal{M}|_{\sigma_{pu}}^{-1} \subseteq \mathcal{N}|_{\sigma_{cu}}^{-1}$ , and so  $\sigma_{pu}(\varphi) \models^+ \sigma_{cu}(\psi)$ .

<sup>9</sup>  $\mathcal{J}$  is a set of indices; we introduce such sets of indices whenever convenient.

However, no class of models  $\mathcal{K} \subseteq \mathbf{Mod}(\Sigma_i)$  that is definable by first-order sentences excludes non-standard models of natural numbers (with “infinitary” elements). Moreover, there is no signature morphism from  $\Sigma_p$  to  $\Sigma_i$ . Therefore, if  $\mathcal{M}|_{\sigma_{ip}} \subseteq \mathcal{K} \subseteq \mathbf{Mod}(\Sigma_i)$  and  $\mathcal{K}$  is definable in  $\mathbf{FO}_{\mathbf{EQ}}$  from  $\{\langle \Sigma_p, \mathcal{M} \rangle\}$  then  $\mathcal{K}|_{\sigma_{ic}}^{-1} \not\models^+ \psi$  (addition need not commute on “infinitary” arguments). Consequently,  $\varphi$  and  $\psi$  have no interpolant in  $\mathbf{FO}_{\mathbf{EQ}}^+$ .

If we remove the binary operation *bop* from  $\Sigma_p$  (and replace it by a unary operation *uop*:  $\mathit{Nat} \rightarrow \mathit{Nat}$ ) the situation becomes quite different. We have then a (unique) signature morphism  $\tau: \Sigma_p \rightarrow \Sigma_i$ , and the sentence  $[\tau(\varphi)] \in \mathbf{Sen}^+(\Sigma_i)$  defines up to isomorphism the standard model of natural numbers, and therefore is an interpolant for  $\varphi$  and  $\psi$ .

For institutions like **PL**, where all classes of models are definable, it might seem that all commutative squares of signature morphisms admit interpolation, and no extension by sentences may spoil this property. However, this need not be the case in general, since for classes of models  $\mathcal{M} \subseteq \mathbf{Sen}(\Sigma_p)$  and  $\mathcal{N} \subseteq \mathbf{Sen}(\Sigma_c)$  such that  $\mathcal{M}|_{\sigma_{pu}}^{-1} \subseteq \mathcal{N}|_{\sigma_{cu}}^{-1}$ , the inclusion  $\mathcal{M}|_{\sigma_{ip}} \subseteq \mathbf{Mod}(\Sigma_i) \setminus ((\mathbf{Mod}(\Sigma_c) \setminus \mathcal{N})|_{\sigma_{ic}})$  may fail, and then no class  $\mathcal{K} \subseteq \mathbf{Mod}(\Sigma_i)$  satisfies  $\mathcal{M}|_{\sigma_{ip}} \subseteq \mathcal{K}$  and  $\mathcal{K}|_{\sigma_{ic}}^{-1} \subseteq \mathcal{N}$ .

The diagram (\*) admits *weak amalgamation* if for all models  $M \in \mathbf{Mod}(\Sigma_p)$  and  $N \in \mathbf{Mod}(\Sigma_c)$  such that  $M|_{\sigma_{ip}} = N|_{\sigma_{ic}}$  there is a model  $K \in \mathbf{Mod}(\Sigma_u)$  such that  $K|_{\sigma_{pu}} = M$  and  $K|_{\sigma_{cu}} = N$ . The diagram (\*) admits *amalgamation* if such a model  $K \in \mathbf{Mod}(\Sigma_u)$  is always unique. This is a standard property used extensively in “institutional” foundations of software specifications. Amalgamation (and hence weak amalgamation) holds for pushouts in all the sample institutions and their variants we defined in Example 1; it fails for some non-pushout diagrams though.

► **Corollary 24.** *If the diagram (\*) does not admit weak amalgamation then it does not admit interpolation in some extension of the institution by new sentences, nor in its further extensions by new sentences.*

Proof (hint): If  $M \in \mathbf{Mod}(\Sigma_p)$  and  $N \in \mathbf{Mod}(\Sigma_c)$  give a counterexample to the weak amalgamation for (\*) then  $\{M\}$  and  $\mathbf{Mod}(\Sigma_c) \setminus \{N\}$  satisfy 1. and 2. in Thm. 22.

► **Theorem 25.** *Assume that in **INS** each class of  $\Sigma_i$ -models is definable by a set of  $\Sigma_i$ -sentences. Then the diagram (\*) admits interpolation in every extension of **INS** by new sentences if and only if it admits weak amalgamation.*

Proof (hint): Assuming weak amalgamation for (\*), for  $\mathcal{M} \subseteq \mathbf{Mod}(\Sigma_p)$  and  $\mathcal{N} \subseteq \mathbf{Mod}(\Sigma_c)$ , if  $\mathcal{M}|_{\sigma_{pu}}^{-1} \subseteq \mathcal{N}|_{\sigma_{cu}}^{-1}$  then  $(\mathcal{M}|_{\sigma_{ip}})|_{\sigma_{ic}}^{-1} \subseteq \mathcal{N}$ .

## 7 Spoiling interpolation by new models and sentences

As so far, we study interpolation over a commuting diagram of signature morphisms (\*) in an institution  $\mathbf{INS} = \langle \mathbf{Sign}, \mathbf{Sen}, \mathbf{Mod}, \langle \models_{\Sigma} \rangle_{\Sigma \in |\mathbf{Sign}|} \rangle$ . In this section we consider the stability of interpolation under institution extensions by new models and sentences.

An extension of an institution **INS** by new models and sentences is an extension  $\mathbf{INS}^{++}$  by new sentences of an extension  $\mathbf{INS}^+$  by new models of the institution **INS**.

The order of the extensions in the above definition is irrelevant. To see this, suppose that  $\mathbf{INS}^+$  extends **INS** by models  $\mathcal{NM} = \langle \mathcal{NM}_{\Sigma}, \models_{\Sigma}^{\mathcal{NM}} \subseteq \mathcal{NM}_{\Sigma} \times \mathbf{Sen}(\Sigma) \rangle_{\Sigma \in |\mathbf{Sign}|}$ , and  $\mathbf{INS}^{++}$  extends  $\mathbf{INS}^+$  by sentences  $\mathcal{NS} = \langle \mathcal{NS}_{\Sigma}, \models_{\Sigma}^{\mathcal{NS}} \subseteq \mathbf{Mod}^+(\Sigma) \times \mathcal{NS}_{\Sigma} \rangle_{\Sigma \in |\mathbf{Sign}|}$  (see Sect. 2.3 for the definitions and notation). Then define  $\mathbf{INS}'$  as the extension of **INS** by sentences  $\mathcal{NS}' = \langle \mathcal{NS}_{\Sigma}, \models_{\Sigma}^{\mathcal{NS}'} \subseteq \mathbf{Mod}(\Sigma) \times \mathcal{NS}_{\Sigma} \rangle_{\Sigma \in |\mathbf{Sign}|}$ , where  $M \models_{\Sigma}^{\mathcal{NS}'} \varphi$  iff  $M \models_{\Sigma}^{\mathcal{NS}} \varphi$  for  $\Sigma \in |\mathbf{Sign}|$ ,  $M \in \mathbf{Mod}(\Sigma)$  and  $\varphi \in \mathcal{NS}_{\Sigma}$ . Then  $\mathbf{INS}^{++}$  coincides with the extension of

## 8:16 Interpolation Is (Not Always) Easy to Spoil

$\mathbf{INS}'$  by models  $\mathcal{NM}' = \langle \mathcal{NM}_\Sigma, \models_{\Sigma}^{\mathcal{NM}'} \subseteq \mathcal{NM}_\Sigma \times \mathbf{Sen}'(\Sigma) \rangle_{\Sigma \in |\mathbf{Sign}|}$ , where for  $\Sigma \in |\mathbf{Sign}|$  and  $M \in \mathcal{NM}_\Sigma$ ,  $M \models_{\Sigma}^{\mathcal{NM}'} \varphi$  iff  $M \models_{\Sigma}^{\mathcal{NM}} \varphi$  for  $\varphi \in \mathbf{Sen}(\Sigma)$ , and for  $\tau: \Sigma' \rightarrow \Sigma$ ,  $\varphi' \in \mathcal{NS}_{\Sigma'}$ ,  $M \models_{\Sigma}^{\mathcal{NM}'} [\tau(\varphi')]$  iff  $[M]_{\tau} \models_{\Sigma'}^{\mathcal{NS}} \varphi'$ .

Obviously, we have “sinks” of institution morphisms that link institution  $\mathbf{INS}$  and its extension  $\mathbf{INS}^{++}$  by models and sentences, but in general there is no institution morphism between  $\mathbf{INS}$  and  $\mathbf{INS}^{++}$ . Their relationship can be captured by another kind of mapping between institutions, where sentences and models translate covariantly [38, 26].

Corollary 8 gives sufficient conditions that ensure that interpolation over a diagram (\*) is stable under extensions of the institution by new models and sentences. The key result here is that these conditions are necessary:

► **Theorem 26.** *The diagram (\*) admits interpolation in all extensions of  $\mathbf{INS}$  by new sentences and models if and only if at least one of the following conditions holds:*

1.  $\sigma_{ip}: \Sigma_i \rightarrow \Sigma_p$  is a retraction and  $\sigma_{cu}: \Sigma_c \rightarrow \Sigma_u$  is a coretraction, or
2.  $\sigma_{ic}: \Sigma_i \rightarrow \Sigma_c$  is a retraction and  $\sigma_{pu}: \Sigma_p \rightarrow \Sigma_u$  is a coretraction.

Proof (hint): For the “only if” part, let  $\mathbf{INS}^{++}$  extend  $\mathbf{INS}$  by a new  $\Sigma_p$ -model  $M$  and a new  $\Sigma_c$ -model  $N$  that do not satisfy any  $\mathbf{INS}$ -sentences and then by a new  $\Sigma_p$ -sentence  $\varphi$  and a new  $\Sigma_c$ -sentence  $\psi$  such that

$$\begin{aligned} \blacksquare \quad \text{Mod}^{++}(\varphi) &= \{M\} \cup \{[N]_{\tau_{pi};\sigma_{ic}} \mid \tau_{pi}: \Sigma_p \rightarrow \Sigma_i, \tau_{pi};\sigma_{ip} = id_{\Sigma_p}\} \\ \blacksquare \quad \text{Mod}^{++}(\psi) &= \{[M]_{\sigma_{cu};\tau_{up}} \mid \tau_{up}: \Sigma_u \rightarrow \Sigma_c, \sigma_{pu};\tau_{up} = id_{\Sigma_p}\} \cup \\ &\quad \{[N]_{\tau_{cc}} \mid \tau_{cc}: \Sigma_c \rightarrow \Sigma_c, \tau_{cc} \neq id_{\Sigma_c}\} \end{aligned}$$

If condition 1. fails then  $\sigma_{pu}(\varphi) \models^{++} \sigma_{cu}(\psi)$ . If condition 2. fails then for any  $\Theta \subseteq \mathbf{Sen}^{++}(\Sigma_i)$ , if  $\varphi \models^{++} \sigma_{ip}(\Theta)$  then  $\sigma_{ic}(\Theta) \not\models^{++} \psi$ .

## 8 Final remarks

In this paper we deal with a general interpolation property, recalling its formulation for an arbitrary logical system formalised as an institution. We study behaviour of interpolation properties over an arbitrary commutative square of signature morphisms under extensions of the institution by new models and sentences. We give an exact characterisation of the situations when a particular interpolant for a premise and a conclusion remains stable under institution extensions by new models (Thm. 14), or looking at this from the other side, when a particular interpolant for a premise and a conclusion is spoiled in some extension of the institution by new models. Another result (Thm. 20) gives sufficient and necessary conditions under which no interpolant for a given premise and conclusion may survive all extensions of the institution by new models, or turning to the positive view, when no extension by new models may spoil the interpolation property for a given premise and conclusion. Then we turn to institution extensions by new sentences, and give an exact characterisation of commutative squares of signature morphisms where adding new sentences may lead to the lack of interpolation (Thm. 22). Incidentally, we clarify here the role of the weak amalgamation property as a necessary condition without which interpolation fails if adding new sentences is permitted (Cor. 24). Finally, we give exact characterisation of commutative squares of signature morphisms where interpolation is ensured for all extensions of the institution by new models and sentences (Thm. 26).

We have carried out our study for the Craig interpolation property. However, in applications a stronger formulation of interpolation is needed: so-called Craig-Robinson (or parameterised) interpolation [19, 13, 35], where the conclusion is required to follow only when an additional “parameter” set of sentences over the signature of the conclusion is added

to the premise and, respectively, to the interpolant. In first-order logic Craig-Robinson interpolation can easily be derived from the Craig interpolation property, but in general, in logical systems that lack compactness and standard logical connectives, this need not be the case. We do not treat explicitly Craig-Robinson interpolation here, to avoid extra complication of notation, but the concepts and techniques we use carry over to this case as well, and so the results may easily be adjusted to cover this more general property.

In many applications, the class of signature morphisms and of their commutative squares for which the interpolation property is required may be considerably restricted. Typically, signature pushouts are of the utmost importance, with further restrictions on the classes of morphisms used. In fact, this is often necessary, since many institutions involved (including the many-sorted first-order logic **FO** and equational logic **EQ**) simply do not admit interpolation for arbitrary signature pushouts. It would be interesting to check how such extra requirements on the signature morphisms involved interact with our characterisation theorems.

---

## References

- 1 Egidio Astesiano, Michel Bidoit, Hélène Kirchner, Bernd Krieg-Brückner, Peter D. Mosses, Donald Sannella, and Andrzej Tarlecki. CASL: the Common Algebraic Specification Language. *Theoretical Computer Science*, 286(2):153–196, 2002. doi:10.1016/S0304-3975(01)00368-1.
- 2 Jon Barwise. Axioms for abstract model theory. *Annals of Mathematical Logic*, 7:221–265, 1974. doi:10.1016/0003-4843(74)90016-3.
- 3 Jan A. Bergstra, Jan Heering, and Paul Klint. Module algebra. *Journal of the Association for Computing Machinery*, 37(2):335–372, 1990. doi:10.1145/77600.77621.
- 4 Evert W. Beth. On Padoa’s method in the theory of definition. *Indagationes Mathematicae (Proceedings)*, 56:330–339, 1953. doi:10.1016/S1385-7258(53)50042-3.
- 5 Tomasz Borzyszkowski. Logical systems for structured specifications. *Theoretical Computer Science*, 286(2):197–245, 2002. doi:10.1016/S0304-3975(01)00317-6.
- 6 Tomasz Borzyszkowski. Generalized interpolation in first-order logic. *Fundamenta Informaticae*, 66(3):199–219, 2005. URL: <http://content.iospress.com/articles/fundamenta-informaticae/fi66-3-01>.
- 7 Carlos Caleiro, Paula Gouveia, and Jaime Ramos. Completeness results for fibred parchments: Beyond the propositional base. In Martin Wirsing, Dirk Pattinson, and Rolf Hennicker, editors, *Recent Trends in Algebraic Development Techniques. Selected Papers from the 16th International Workshop on Algebraic Development Techniques*, volume 2755 of *Lecture Notes in Computer Science*, pages 185–200. Springer, 2003. doi:10.1007/978-3-540-40020-2\_10.
- 8 Carlos Caleiro, Paulo Mateus, Jaime Ramos, and Amílcar Sernadas. Combining logics: Parchments revisited. In Maura Cerioli and Gianna Reggio, editors, *Recent Trends in Algebraic Development Techniques. Selected Papers from the 15th Workshop on Algebraic Development Techniques Joint with the CoFI WG Meeting*, volume 2267 of *Lecture Notes in Computer Science*, pages 48–70. Springer, 2001. doi:10.1007/3-540-45645-7\_3.
- 9 Carlos Caleiro, Amílcar Sernadas, and Cristina Sernadas. Fibring logics: Past, present and future. In Sergei N. Artëmov, Howard Barringer, Artur S. d’Avila Garcez, Luís C. Lamb, and John Woods, editors, *We Will Show Them! Essays in Honour of Dov Gabbay, Volume One*, pages 363–388. College Publications, 2005.
- 10 María Victoria Cengarle. *Formal Specifications with Higher-Order Parameterization*. PhD thesis, Ludwig-Maximilians-Universität München, Institut für Informatik, 1994.
- 11 Chen-Chung Chang and H. Jerome Keisler. *Model Theory*. North-Holland, third edition, 1990.
- 12 William Craig. Linear reasoning. A new form of the Herbrand-Gentzen theorem. *The Journal of Symbolic Logic*, 22(3):250–268, 1957. doi:10.2307/2963593.
- 13 Răzvan Diaconescu. *Institution-Independent Model Theory*. Birkhäuser, 2008. doi:10.1007/978-3-7643-8708-2.

- 14 Răzvan Diaconescu. An institution-independent proof of Craig interpolation theorem. *Studia Logica*, 77(1):59–79, 2004. doi:10.1023/B:STUD.0000034185.62660.d6.
- 15 Răzvan Diaconescu. Interpolation for predefined types. *Mathematical Structures in Computer Science*, 22(1):1–24, 2012. doi:10.1017/S0960129511000430.
- 16 Răzvan Diaconescu. Three decades of institution theory. In Jean-Yves Béziau, editor, *Universal Logic: An Anthology*, pages 309–322. Birkhäuser, 2012.
- 17 Răzvan Diaconescu. Generalised graded interpolation. *International Journal of Approximate Reasoning*, 152:236–261, 2023. doi:10.1016/j.ijar.2022.10.018.
- 18 Răzvan Diaconescu. Borrowing interpolation. *J. Logic and Computation*, 22(3):561–586, 2011. doi:10.1093/logcom/exr007.
- 19 Theodosis Dimitrakos and Thomas S.E. Maibaum. On a generalised modularization theorem. *Information Processing Letters*, 74(1–2):65–71, 2000. doi:10.1016/S0020-0190(00)00037-5.
- 20 Hartmut Ehrig, Hans-Jörg Kreowski, James W. Thatcher, Eric G. Wagner, and Jesse B. Wright. Parameter passing in algebraic specification languages. *Theoretical Computer Science*, 28(1–2):45–81, 1984. doi:10.1016/0304-3975(83)90065-8.
- 21 Dov M. Gabbay and Larisa Maksimova. *Interpolation and Definability: Modal and Intuitionistic Logics*. Oxford University Press, 2005. doi:10.1093/acprof:oso/9780198511748.001.0001.
- 22 Daniel Găină. Interpolation in logics with constructors. *Theoretical Computer Science*, 474:46–59, 2013. doi:10.1016/j.tcs.2012.12.002.
- 23 Daniel Găină. Downward Löwenheim–Skolem theorem and interpolation in logics with constructors. *Journal of Logic and Computation*, 27(6):1717–1752, 2015. doi:10.1093/logcom/exv018.
- 24 Daniel Găină and Andrei Popescu. An institution-independent proof of the Robinson consistency theorem. *Studia Logica*, 85:41–73, 2007. doi:10.1007/s11225-007-9022-4.
- 25 Joseph A. Goguen and Rodney M. Burstall. Institutions: Abstract model theory for specification and programming. *Journal of the Association for Computing Machinery*, 39(1):95–146, 1992. doi:10.1145/147508.147524.
- 26 Joseph A. Goguen and Grigore Roşu. Institution morphisms. *Formal Aspects of Computing*, 13(3–5):274–307, 2002. doi:10.1007/s001650200013.
- 27 Thomas S.E. Maibaum, Martin R. Sadler, and Paolo A.S. Veloso. Logical specification and implementation. In Mathai Joseph and Rudrapatna Shyamasundar, editors, *Foundations of Software Technology and Theoretical Computer Science*, pages 13–30. Springer, 1984. doi:10.1007/3-540-13883-8\_62.
- 28 Till Mossakowski, Wiesław Pawłowski, Donald Sannella, and Andrzej Tarlecki. Parchments for CafeOBJ logics. In Shusaku Iida, José Meseguer, and Kazuhiro Ogata, editors, *Specification, Algebra, and Software - Essays Dedicated to Kokichi Futatsugi*, volume 8373 of *Lecture Notes in Computer Science*, pages 66–91. Springer, 2014. doi:10.1007/978-3-642-54624-2\_4.
- 29 Till Mossakowski, Andrzej Tarlecki, and Wiesław Pawłowski. Combining and representing logical systems using model-theoretic parchments. In Francesco Parisi-Presicce, editor, *Recent Trends in Data Type Specification. Selected Papers from the 12th International Workshop on Specification of Abstract Data Types*, volume 1376 of *Lecture Notes in Computer Science*, pages 349–364. Springer, 1998. doi:10.1007/3-540-64299-4\_44.
- 30 Andrei Popescu, Traian Florin Şerbănuţă, and Grigore Roşu. A semantic approach to interpolation. *Theoretical Computer Science*, 410(12–13):1109–1128, 2009. doi:10.1016/j.tcs.2008.09.038.
- 31 Gerard R. Renardel de Lavalette. Interpolation in computing science: The semantics of modularization. *Synthese*, 164(3):437–450, 2008. doi:10.1007/s11229-008-9358-y.
- 32 Abraham Robinson. A result on consistency and its application to the theory of definition. *Indagationes Mathematicae (Proceedings)*, 59:47–58, 1956. doi:10.1016/S1385-7258(56)50008-X.
- 33 Pieter Hendrik Rodenburg. A simple algebraic proof of the equational interpolation theorem. *Algebra Universalis*, 28:48–51, 1991. doi:10.1007/BF01190411.



- 34 Grigore Roşu and Joseph A. Goguen. On equational Craig interpolation. *Journal of Universal Computer Science*, 6(1):194–200, 2000. doi:10.3217/jucs-006-01-0194.
- 35 Donald Sannella and Andrzej Tarlecki. *Foundations of Algebraic Specification and Formal Software Development*. Monographs in Theoretical Computer Science. An EATCS Series. Springer, 2012. doi:10.1007/978-3-642-17336-3.
- 36 Donald Sannella and Andrzej Tarlecki. Property-oriented semantics of structured specifications. *Mathematical Structures in Computer Science*, 24(2):e240205, 2014. doi:10.1017/S0960129513000212.
- 37 Andrzej Tarlecki. Bits and pieces of the theory of institutions. In David H. Pitt, Samson Abramsky, Axel Poigné, and David E. Rydeheard, editors, *Proceedings of the Tutorial and Workshop on Category Theory and Computer Programming*, volume 240 of *Lecture Notes in Computer Science*, pages 334–360. Springer, 1986. doi:10.1007/3-540-17162-2\_132.
- 38 Andrzej Tarlecki. Towards heterogeneous specifications. In Dov Gabbay and Maarten de Rijke, editors, *Frontiers of Combining Systems 2*, volume 7 of *Studies in Logic and Computation*, pages 337–360. Research Studies Press, 2000.
- 39 Andrzej Tarlecki. Some nuances of many-sorted universal algebra: A review. *Bulletin of the European Association for Theoretical Computer Science*, 104:89–111, 2011. URL: <http://eatcs.org/beatcs/index.php/beatcs/article/view/121>.
- 40 Jouko Väänänen. The Craig interpolation theorem in abstract model theory. *Synthese*, 164:401–420, 2008. doi:10.1007/s11229-008-9357-z.
- 41 Paulo A.S. Veloso. On pushout consistency, modularity and interpolation for logical specifications. *Information Processing Letters*, 60(2):59–66, 1996. doi:10.1016/S0020-0190(96)00146-9.
- 42 Paulo A.S. Veloso and Thomas S.E. Maibaum. On the modularization theorem for logical specifications. *Information Processing Letters*, 53(5):287–293, 1995. doi:10.1016/0020-0190(94)00203-B.



# String Diagram Rewriting Modulo Commutative (Co)Monoid Structure

Aleksandar Milosavljević ✉

University College London, UK

Robin Piedeleu ✉ 

University College London, UK

Fabio Zanasi ✉ 

University College London, UK

University of Bologna, Italy

---

## Abstract

String diagrams constitute an intuitive and expressive graphical syntax that has found application in a very diverse range of fields including concurrency theory, quantum computing, control theory, machine learning, linguistics, and digital circuits. Rewriting theory for string diagrams relies on a combinatorial interpretation as double-pushout rewriting of certain hypergraphs. As previously studied, there is a “tension” in this interpretation: in order to make it sound and complete, we either need to add structure on string diagrams (in particular, Frobenius algebra structure) or pose restrictions on double-pushout rewriting (resulting in “convex” rewriting). From the string diagram viewpoint, imposing a full Frobenius structure may not always be natural or desirable in applications, which motivates our study of a weaker requirement: commutative monoid structure. In this work we characterise string diagram rewriting modulo commutative monoid equations, via a sound and complete interpretation in a suitable notion of double-pushout rewriting of hypergraphs.

**2012 ACM Subject Classification** Theory of computation → Rewrite systems

**Keywords and phrases** String diagrams, Double-pushout rewriting, Commutative monoid

**Digital Object Identifier** 10.4230/LIPIcs.CALCO.2023.9

**Related Version** *Full Version:* <https://arxiv.org/abs/2204.04274>

**Funding** RP and FZ acknowledge support from EPSRC grant EP/V002376/1.

**Acknowledgements** We thank Tobias Fritz for helpful discussion and the anonymous reviewers of CALCO for their suggestions.

## 1 Introduction

String diagrams [28] are a diagrammatic syntax for reasoning algebraically about component-based systems, which in the last few years have found application across diverse fields, including quantum computation [23], digital [20] and electrical circuits [3, 10], machine learning [15], concurrency theory [9], control theory [1, 12], and linguistics [29] amongst others. Compared to traditional syntax, the use of string diagrams allows to neatly visualise resource exchange and message passing between different parts of a system, which is pivotal in studying subtle interactions such as those arising in concurrent processes and quantum computation. Moreover, we can reason with string diagrams both combinatorially and as syntactic, inductively defined objects, which enables forms of compositional analysis typical of programming language semantics.

A cornerstone of string diagrammatic approaches is the possibility of performing *diagrammatic reasoning*: transforming a string diagram according to a certain rewrite rule, which replaces a sub-diagram with a different one. A set of such rules, which typically preserve the semantics of the model, may represent for instance a compilation procedure [27], the realisation of a specification [12], or a refinement of system behaviour [8].



© Aleksandar Milosavljević, Robin Piedeleu, and Fabio Zanasi;  
licensed under Creative Commons License CC-BY 4.0

10th Conference on Algebra and Coalgebra in Computer Science (CALCO 2023).

Editors: Paolo Baldan and Valeria de Paiva; Article No. 9; pp. 9:1–9:17

Leibniz International Proceedings in Informatics



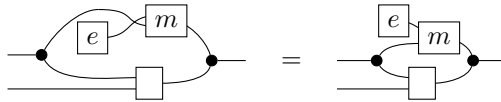
LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

## 9:2 String Diagram Rewriting Modulo Commutative (Co)Monoid Structure

Compared to traditional term rewriting, a mathematical theory of string diagram rewriting poses new challenges. Formally, string diagrams are graphical representations of morphisms in a category, typically assumed in applications to be a symmetric monoidal category (SMC). In order to perform a rewrite step, we need to match the left-hand side of a rewrite rule to a sub-diagram of a given string diagram. For instance, consider the rewrite rule as on the left below, and the string diagram on the right.



Morally, there is a match for the rule in the string diagram. The issue is that, strictly speaking, such a match does not happen on the nose: we need first to apply the laws of SMCs in order to transform the string diagram into an equivalent one, with the wires into  $m$  uncrossed. At this point we have clearly isolated the sub-diagram and are able to perform the rewrite step.



As seen in this example, string diagram rewriting is performed modulo certain structural laws, which reflect the categorical structure in which the string diagrams live. However, from a practical viewpoint, this form of rewriting is not really feasible, as each rewrite step would require us to inspect all string diagrams equivalent to a given one looking for redexes.

This issue can be solved via an interpretation of string diagrams as certain hypergraphs, and of string diagram rewriting as *double-pushout rewriting* (DPO) [14] of such hypergraphs. We refer to [5, 6, 7] for a systematic introduction to this approach. In a nutshell, the benefit of working with such an interpretation is that an equivalence class of string diagrams corresponds to just one hypergraph, meaning that our search for redexes is drastically simplified. However, there is a mismatch: if we want to rewrite string diagrams in a SMC, then soundness is only ensured by adopting a restricted notion of DPO rewriting, called *convex* DPO rewriting [4]. Conversely, if we want to work with arbitrary DPO rewriting steps, then the corresponding notion of string diagram rewriting does not rewrite only modulo the laws of SMCs, but requires a special commutative *Frobenius algebra* on each object of the category. Recall that a Frobenius algebra consists of a commutative monoid and a commutative comonoid, interacting with each other via the so-called Frobenius law [13].

When modelling a certain class of systems with string diagrams, assuming that such Frobenius structure exists is not always reasonable, or desirable. A first class of such examples are matrix-like semantic structures, which are axiomatised by bialgebra equations – see *e.g.* [30] for a survey. It is known that if the monoid and the comonoid both obey the Frobenius and the bialgebra laws, then the equational theory trivialises, *cf.* [17, Ex. 4.3]. A second important class are semantic structures for probability theory, which usually feature a commutative comonoid structure, but no Frobenius equations – introducing Frobenius structure amounts to allowing unnormalised probabilities, *cf.* [22, 18]. These categories, sometimes called *CD-categories*, also play a special role in the study of algebraic theories, because they model the cartesian handling of variables [11].

All these models motivate the study of rewriting for structures intermediate between plain symmetric monoidal and equipped with Frobenius algebras. More specifically, we focus on string diagrams in categories where each object comes with a *commutative monoid* structure.

$$\begin{aligned}
(s; t); u &\equiv s; (t; u), \quad id_n; s \equiv s \equiv s; id_m, \quad (s \oplus t) \oplus u \equiv s \oplus (t \oplus u), \quad id_0 \oplus s \equiv s \equiv s \oplus id_0, \\
id_m \oplus id_n &\equiv id_{m+n}, \quad \sigma_{m,n}; \sigma_{n,m} \equiv id_{m+n}, \quad (s \oplus id_m); \sigma_{m,n} \equiv \sigma_{m,o}; (id_m \oplus s) \\
(s; u) \oplus (t; v) &\equiv (s \oplus t); (u \oplus v), \quad (\sigma_{m,n} \oplus id_o); (id_n \oplus \sigma_{m,o}) \equiv \sigma_{m,n+o},
\end{aligned}$$

■ **Figure 1** Laws of symmetric monoidal categories in a prop, where  $n, m, o$  range over  $\mathbb{N}$ .

From a rewriting viewpoint, this case is particularly significant because symmetries in a SMC may always create redexes for the commutativity axiom of the monoid multiplication, yielding a non-terminating rewrite system:



Therefore, rather than taking commutativity as a rewrite rule, we need to find an alternative representation of string diagrams (and of string diagram rewriting) that is invariant modulo the axioms of commutative monoids (and the laws of SMCs), which is the focus of this paper. Our contribution is two-fold:

- we identify which class of hypergraphs provides an adequate interpretation of string diagrams in a SMC with commutative monoid structure, and organise them into a SMC. This characterisation will take the form of an isomorphism between the SMC of string diagrams and the SMC of hypergraphs.<sup>1</sup>
- We identify which notion of double-pushout hypergraph rewriting interprets string diagram rewriting modulo the axioms of commutative monoids in a sound and complete way.

Note that all of the theory developed in this work can be easily dualised to obtain a framework for rewriting modulo commutative *comonoid* structure, which justifies the title and makes it relevant also for the aforementioned CD categories.

*Synopsis.* Section 2 recalls background on string diagrams and hypergraphs. Section 3 shows the hypergraph characterisation of string diagrams with a chosen commutative monoid structure. Section 4 shows how string diagram rewriting may be characterised in terms of DPO hypergraph rewriting. We summarise our work and suggest future directions in Section 5. Additional details and missing proofs can be found in [26].

## 2 Preliminaries

We recall some basic definitions, using the same terminology as [5].

► **Definition 1** (Theories and Props). *A symmetric monoidal theory is a pair  $(\Sigma, \varepsilon)$ , where  $\Sigma$  is a monoidal signature i.e. a set of operations  $o : m \rightarrow n$  with a fixed arity  $m$  and coarity  $n$ , and  $\varepsilon$  is a set of equations, i.e. pairs  $\langle l, r \rangle$  of  $\Sigma$ -terms  $l, r : v \rightarrow w$  with the same arity and coarity.  $\Sigma$ -terms are constructed by combining the operations in  $\Sigma$ , identities  $id_n : n \rightarrow n$  and symmetries  $\sigma_{m,n} : m + n \rightarrow n + m$  for each  $m, n \in \mathbb{N}$ , by sequential  $(;)$  and parallel  $(\oplus)$  composition. A prop is a symmetric strict monoidal category  $(\mathcal{C}, +, 0)$  for which  $\text{Ob}(\mathcal{C}) = \mathbb{N}$ , the monoidal unit is  $0 \in \mathbb{N}$ , and the monoidal product on objects is given by addition. The prop freely generated from a symmetric monoidal theory  $(\Sigma, \varepsilon)$  has, as*

<sup>1</sup> Simultaneously to a preprint of our work (arXiv:2204.04274), a preprint showing a result closely related to this first contribution also appeared on ArXiv. We comment on their relation in Section 5.

morphisms, the  $\Sigma$ -terms modulo  $\varepsilon$  and the laws of symmetric monoidal categories recalled in Fig. 1. Given two props  $\mathcal{C}$  and  $\mathcal{D}$ , a functor  $F : \mathcal{C} \rightarrow \mathcal{D}$  is called a prop-morphism from  $\mathcal{C}$  to  $\mathcal{D}$  if it is an identity-on-objects strict symmetric monoidal functor. Props and prop-morphisms form a category we call PROP.

► **Example 2** (Monoids and Functions). Particularly relevant to our development are the prop  $\mathbb{F}$  of functions and the prop **CMon** of commutative monoids.  $\mathbb{F}$  has morphisms  $f : m \rightarrow n$  the functions from the set  $\{0, \dots, m - 1\}$  to  $\{0, \dots, n - 1\}$ , with monoidal product on functions being their disjoint union. The prop **CMon** is freely generated by the signature consisting of  $\mu : 2 \rightarrow 1$  (multiplication) and  $\eta : 0 \rightarrow 1$  (unit), and equations expressing commutativity, unitality and associativity of  $\mu$ .

The theory of commutative monoids presents the prop of functions, in the sense that  $\mathbf{CMon} \cong \mathbb{F}$  – see e.g. [24].

► **Example 3** (Cospans). A second prop important for our purposes is the one of cospans. When interpreting string diagrams as hypergraphs, it is fundamental to record the information of what wires are available for composition on the left and right hand side of the diagram: this is achieved by considering cospans of hypergraphs, with the cospan structure indicating which nodes constitute the left and the right interface of the hypergraph.

Given a category  $\mathbb{C}$  with finite colimits, let  $\mathbf{Csp}(\mathbb{C})$  be the category with the same objects as  $\mathbb{C}$  and morphisms  $X \rightarrow Y$  the cospans from  $X$  to  $Y$ , that is, pairs of arrows  $X \rightarrow A \leftarrow Y$ , for any object  $A$  (called the carrier of the cospan). Composition of cospans  $X \xrightarrow{f} A \xleftarrow{h} Z$  and  $Z \xrightarrow{b} B \xleftarrow{i} Y$  is defined by pushout of the span formed by the middle legs, i.e., it is the<sup>2</sup> cospan  $X \xrightarrow{f:p} Q \xleftarrow{i;q} Y$  where  $A \xrightarrow{p} Q \xleftarrow{q} B$  is the pushout of  $A \xleftarrow{h} Z \xrightarrow{i} B$ .  $\mathbf{Csp}(\mathbb{C})$  is symmetric monoidal with the monoidal unit being the initial object  $0 \in \mathbb{C}$  and the monoidal product given by the coproduct in  $\mathbb{C}$  of the two maps of each cospan.

Hypergraphs [2] generalise graphs by replacing edges with ordered and directed hyperedges, which may have lists of source and target nodes instead of just individual ones. Hypergraphs and hypergraph homomorphisms form a category **Hyp**. As observed in [5], this category may also be defined as a presheaf topos – this is particularly convenient for calculating (co)limits and to ensure that it is adhesive [25], a fundamental property for DPO rewriting. More precisely, **Hyp** is isomorphic to the functor category  $\mathbb{F}^{\mathbf{I}}$ , where  $\mathbf{I}$  has objects the pairs of natural numbers  $(k, l) \in \mathbb{N} \times \mathbb{N}$  and an extra object  $\star$ , with  $k + l$  arrows from  $(k, l)$  to  $\star$ , for all  $k, l \in \mathbb{N}$ . A hypergraph  $G$  is therefore given by a set  $G_\star$  of nodes, and sets  $G_{k,l}$  of hyperedges for each  $(k, l) \in \mathbb{N} \times \mathbb{N}$ , with source maps  $s_i : G_{k,l} \rightarrow G_\star$  for  $1 \leq i \leq k$  and target maps  $t_j : G_{k,l} \rightarrow G_\star$ ,  $1 \leq j \leq l$ . A monoidal signature  $\Sigma$  yields a directed hypergraph  $G_\Sigma$  with only a single node and a hyperedge for every  $\Sigma$ -operation  $o : k \rightarrow l$  with  $k$  sources and  $l$  targets (i.e., in  $G_{k,l}$ ). We can use this observation to define the category of  $\Sigma$ -labelled hypergraphs as follows.

► **Definition 4.** The slice category  $\mathbf{Hyp} \downarrow G_\Sigma$  is called the category of  $\Sigma$ -labelled hypergraphs and denoted by  $\mathbf{Hyp}_\Sigma$ .

As proven in [5], morphisms in a prop freely generated by a signature  $\Sigma$  may be faithfully interpreted as discrete cospans of  $\Sigma$ -labelled hypergraphs, where the cospan structure represents the interfaces (left and right) of the string diagram. Motivated by this characterisation,

<sup>2</sup> Note composition is only defined up-to-isomorphism. Strictly speaking, to obtain a (1-)category, we take as morphisms isomorphism classes of cospans (isomorphisms are invertible maps between the carriers that make the obvious diagram commute). We will gloss over this aspect to keep notation light and because the bicategorical aspects do not feature in our development.

we recall from [5] the faithful, coproduct-preserving functor  $D : \mathbb{F} \rightarrow \mathbf{Hyp}_\Sigma$  mapping every object  $i \in \text{Ob}(\mathbb{F}) = \mathbb{N}$  to the discrete hypergraph with set of nodes  $i = \{0, \dots, i - 1\}$  and mapping each function to the induced hypergraph homomorphism. We can define the category  $\text{Csp}_D(\mathbf{Hyp}_\Sigma)$  of discrete cospans of hypergraphs as the full subcategory of  $\text{Csp}(\mathbf{Hyp}_\Sigma)$  (cf. Example 3) on discrete hypergraphs.

### 3 The Combinatorial Interpretation

In this section we prove that a freely generated prop with a chosen commutative monoid structure is isomorphic to a category of cospans of hypergraphs with certain restrictions (Theorem 21 below).

As shown in [6], the standard interpretation of string diagrams in a prop as cospans of hypergraphs is not full. In order to characterise the image of the interpretation, it is necessary to restrict ourselves to a class of so-called *acyclic* and *monogamous* cospans. In order to prove our result for props with a chosen commutative monoid structure, we may relax this notion to *right-monogamous* cospans, which we now introduce.

► **Definition 5** (Degree of a node [6]). *The in-degree of a node  $v$  in hypergraph  $H$  is the number of pairs  $(h, i)$  where  $h$  is a hyperedge with  $v$  as its  $i^{\text{th}}$  target. Similarly, the out-degree of  $v$  is the number of pairs  $(h, j)$  where  $h$  is a hyperedge with  $v$  as its  $j^{\text{th}}$  source.*

► **Definition 6** (Terminal node). *We say that a node  $v$  of a hypergraph  $H$  is terminal if its out-degree is 0, i.e., if there are no hyperedges of  $H$  with source  $v$ .*

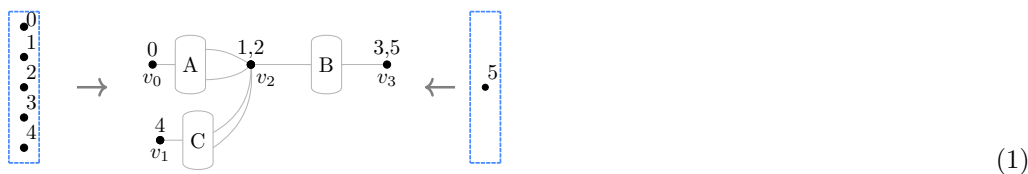
Given  $m \xrightarrow{f} H \xleftarrow{g} n$  in  $\text{Csp}_D(\mathbf{Hyp}_\Sigma)$ , we call inputs of  $H$  the set  $\text{in}(H)$ , defined as the image of  $f$  and outputs, the set  $\text{out}(H)$  defined as the image of  $g$ .

► **Definition 7** (Right-monogamy). *We say that a cospan  $m \xrightarrow{f} H \xleftarrow{g} n$  is right-monogamous if  $g$  is mono and  $\text{out}(H)$  is the set of terminal nodes of  $H$ .*

Compared to monogamy [6], right-monogamy does not impose any requirement on  $f$ , and only constraints the out-degree of nodes (not the in-degree).

Acyclicity is a standard condition which forbids (directed) loops in a hypergraph, cf. [6, Definition 20] Similarly to monogamous cospans [6], one may verify that acyclic right-monogamous cospans in  $\text{Csp}_D(\mathbf{Hyp}_\Sigma)$  form a sub-prop of  $\text{Csp}_D(\mathbf{Hyp}_\Sigma)$ , denoted by  $\text{RMACsp}_D(\mathbf{Hyp}_\Sigma)$ .

► **Example 8.** Let us use blue frames to indicate the left and the right interface of a cospan, natural numbers to indicate how the cospan legs are defined, and rounded rectangles to represent hyperedges. The cospan depicted below is right-monogamous.



The notion of right-monogamy is justified by its connection to commutative monoids, and crystallised in the following result. Below, the empty set in  $\text{RMACsp}_D(\mathbf{Hyp}_\emptyset)$  refers to the empty signature  $\Sigma = \emptyset$ .

► **Proposition 9.** *There is an isomorphism of props  $\mathbf{CMon} \cong \text{RMACsp}_D(\mathbf{Hyp}_\emptyset)$ .*

In particular, the isomorphism interprets the comultiplication and unit as follows:



The fundamental observation leading to Proposition 9 is that the prop of  $\emptyset$ -labelled hypergraphs is isomorphic to  $\mathbb{F}$ . Right-monogamous cospans in this category coincide with cospans of the form  $m \xrightarrow{f} n \xleftarrow{id} n$ , and can thus be thought of as morphisms in  $\mathbb{F}$ . Paired with the fact that  $\mathbf{CMon} \cong \mathbb{F}$  (cf. Example 2), we obtain the above result – see [26] for more details. Following this result, we will sometimes refer abusively to certain functions  $f$  as cospans, assuming implicitly that we mean the cospan  $m \xrightarrow{f} n \xleftarrow{id} n$ .

When referring to “string diagrams with a chosen commutative monoid structure”, we mean morphisms of the prop  $\mathbf{S}_\Sigma + \mathbf{CMon}$ , the coproduct of the free props over signature  $\Sigma$  and  $\mathbf{CMon}$ . Intuitively, such morphisms are obtained by freely combining  $\Sigma$ -terms with terms of the theory of commutative monoids, then quotienting by the laws of symmetric monoidal categories and those of  $\mathbf{CMon}$ . For a formal definition of the coproduct of props, see [30]. Our next goal, and the core result of this section, is extending Proposition 9 to the case where  $\Sigma$  is non-empty, *i.e.*, an isomorphism between  $\mathbf{S}_\Sigma + \mathbf{CMon}$  and  $\mathbf{RMACsp}_D(\mathbf{Hyp}_\Sigma)$ . This will allow us to refer to  $\mathbf{RMACsp}_D(\mathbf{Hyp}_\Sigma)$  as the combinatorial characterisation of string diagrams in  $\mathbf{S}_\Sigma + \mathbf{CMon}$ , and study their rewriting as DPO-rewriting in  $\mathbf{RMACsp}_D(\mathbf{Hyp}_\Sigma)$ .

In order to relate  $\mathbf{S}_\Sigma + \mathbf{CMon}$  and  $\mathbf{RMACsp}_D(\mathbf{Hyp}_\Sigma)$ , we will use a strategy analogous to the one used in [6] for theories with symmetric monoidal structure only. In essence, we want to show that  $\mathbf{RMACsp}_D(\mathbf{Hyp}_\Sigma)$  has the universal property of the coproduct. Consider

$$\mathbf{S}_\Sigma \xrightarrow{[\cdot]} \mathbf{RMACsp}_D(\mathbf{Hyp}_\Sigma) \xleftarrow{|\cdot|} \mathbf{CMon}$$

where  $[\cdot] : \mathbf{S}_\Sigma \rightarrow \mathbf{RMACsp}_D(\mathbf{Hyp}_\Sigma)$  is the faithful prop morphism defined in [6], and  $|\cdot| : \mathbf{CMon} \rightarrow \mathbf{RMACsp}_D(\mathbf{Hyp}_\Sigma)$  is defined by composing the isomorphism of Proposition 9 with the obvious faithful prop morphism  $\mathbf{RMACsp}_D(\mathbf{Hyp}_\emptyset) \rightarrow \mathbf{RMACsp}_D(\mathbf{Hyp}_\Sigma)$ . To show that  $\mathbf{RMACsp}_D(\mathbf{Hyp}_\Sigma)$  has the universal property of the coproduct, the fundamental step is investigating how right-monogamous acyclic cospans can be factorised into a composite cospan  $\mathcal{M}_0; \mathcal{D}_0; \dots; \mathcal{M}_l; \mathcal{D}_l$  that alternates between monogamous acyclic cospans that are in the image  $[\cdot] : \mathbf{S}_\Sigma \rightarrow \mathbf{RMACsp}_D(\mathbf{Hyp}_\Sigma)$  and discrete right-monogamous acyclic cospans that are in the image of  $|\cdot| : \mathbf{CMon} \rightarrow \mathbf{RMACsp}_D(\mathbf{Hyp}_\Sigma)$ .

### 3.1 Weak decomposition

First, we need to show how to decompose right-monogamous cospans of hypergraphs in the same way that we can take sub-diagrams of string diagrams.

Formally, a sub-diagram  $c$  of some larger string diagram  $d$  can be defined as a sub-term (modulo the laws of symmetric monoidal categories) of  $d$ . It is not difficult to show (by induction) that we can always find some  $k \in \mathbb{N}$  and diagrams  $c_1, c_2$  such that  $d = c_1; (id_k \oplus l); c_2$ , that is, such that  $d$  decomposes as


$${}^n \boxed{d} {}^m = {}^n \boxed{c_1} \begin{array}{c} \xrightarrow{k} \\ \boxed{c} \\ \xrightarrow{j} \end{array} \boxed{c_2} {}^m \quad (2)$$

In fact, we could also take this decomposition as a definition of sub-diagrams. We turn to the corresponding notion of sub-structure for cospans of hypergraphs.

In the plain symmetric monoidal case, not all sub-hypergraphs of the cospan representation of a string diagram  $d$  correspond to sub-diagrams of  $d$ . Those that do have the additional properties of being *convex* [6].



► **Definition 10** (Convex sub-hypergraph). *A sub-hypergraph  $H \subseteq G$  is convex if, for any nodes  $v, v'$  in  $H$  and any path  $p$  from  $v$  to  $v'$  in  $G$ , every hyperedge in  $p$  is also in  $H$ .*

► **Example 11.**  is a convex sub-hypergraph of the hypergraph in Example 8.

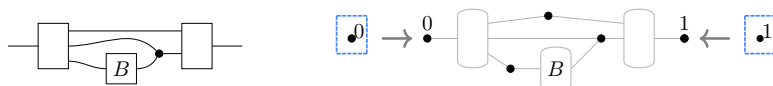
The following lemma shows that convex sub-hypergraphs is the right counterpart of that of sub-diagram, in the hypergraph world. Note the correspondence between (3) below and (2) above.


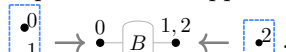
► **Lemma 12** (Weak decomposition). *Let  $\mathcal{G} = m \rightarrow G \leftarrow n$  be a right-monogamous acyclic cospan and  $L$  be a convex sub-hypergraph of  $G$ . We can decompose  $\mathcal{G}$  as*

$$(m \rightarrow \tilde{C}_1 \leftarrow i + k); \left( \begin{array}{c} k \xrightarrow{id} k \xleftarrow{id} k \\ \oplus \\ i \rightarrow L \leftarrow j \end{array} \right); (j + k \rightarrow C_2 \leftarrow n) \quad (3)$$

for some  $k \in \mathbb{N}$  and where all the above cospans are right-monogamous acyclic.

► **Example 13.** Consider the diagram below with its corresponding cospan representation:



For the convex sub-hypergraph  $L := \bullet \text{---} B \text{---} \bullet$ , there are two possible choices of weak decomposition, depending on where we attach the second leg of the monoid multiplication that appears in the corresponding string diagram:  or .

Note that this situation differs from the plain symmetric monoidal case [6], where  $i \rightarrow L \leftarrow j$  is unique, given  $L$ . With commutative monoids, the non-uniqueness comes from having to choose whether we include the monoid structure nodes in the cospan  $i \rightarrow L \leftarrow j$  or in the surrounding two cospans. Of course there is a minimal such cospan, that corresponds to the sub-diagram in the diagrammatic decomposition, but we sometimes need the flexibility to choose another decomposition.

### 3.2 Factorisation into levels

Now we tackle the factorisation of cospans in  $\text{RMACsp}_D(\mathbf{Hyp}_\Sigma)$  into alternating monogamous cospans and right-monogamous discrete cospans, to prove that  $\text{RMACsp}_D(\mathbf{Hyp}_\Sigma) \cong \mathbf{S}_\Sigma + \mathbf{CMon}$ , our main characterisation theorem (Theorem 21).

As we saw, the hypergraphs that correspond to plain string diagrams (in symmetric monoidal categories) are monogamous: nodes are precisely the target and source of one hyperedge. The commutative monoid structure relaxes this requirement for targets. For our last decomposition, we would like to identify nodes that can only appear in the hypergraph representation of diagrams that contain some occurrence of the commutative monoid structure (multiplication or unit), that is, nodes that do not simply represent plain wires. The following definition formalises this idea.

► **Definition 14.** *Let  $m \rightarrow G \leftarrow n$  be a right-monogamous acyclic cospan. We say that the node  $v$  in  $G$  is left-amonogamous if:*

- it is in  $\text{in}(G)$  and its in-degree is not equal to 0, or
- it is not in  $\text{in}(G)$  and its in-degree is not equal to 1

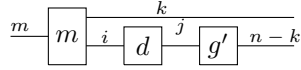
► **Example 15.** In the cospan (1),  $v_2$  and  $v_3$  are left-amonogamous, while  $v_0$  and  $v_1$  are not.

► **Definition 16** (Order of nodes and level of hyperedges). *Let  $m \rightarrow G \leftarrow n$  be a right-monogamous acyclic cospan. We define the order of a node  $v$  to be the largest number of left-amonogamous nodes preceding it (including itself) on a path leading to  $v$ . The level of a hyperedge is the largest number of left-amonogamous nodes on a path ending with  $v$ .*

► **Example 17.** In the cospan (1), hyperedges  $A$  and  $C$  are level-0 hyperedges, and hyperedge  $B$  is a level-1 hyperedge.

Recall that we want to obtain a factorisation of any cospan into an alternating composition of discrete right-monogamous cospans – corresponding to diagrams with no generating boxes from the chosen signature – and monogamous cospans – corresponding to plain string diagrams over the chosen signature. We will do this by induction on the maximum level of hyperedges, effectively stripping the necessary cospans (discrete right-monogamous and monogamous) at each level as we move from left to right.

The following lemma will be used at each induction step: here, we require the decomposition to not only alternate between monogamous and discrete right-monogamous cospans, but to also keep track of the order of terminal nodes. Diagrammatically, we want a decomposition of a right-monogamous cospan into the following form:



with  $m$  corresponding to a monogamous cospan, and  $d$  to a discrete right-monogamous one, and  $g'$  is the rest of the decomposition. Here, the first  $k$  wires are the terminal order-0 nodes of the overall composite. Keeping track of where terminal nodes of each order are located is an important technical complication that will be needed to prove that the map that we construct out of  $\text{RMACsp}_D(\mathbf{Hyp}_\Sigma)$  to show it satisfies the universal property of the coproduct  $\mathbf{S}_\Sigma + \mathbf{CMon}$ , is a monoidal functor. Recall that, following Proposition 9, we refer abusively to permutations  $\pi : n \rightarrow n$  below as cospans, assuming implicitly that we mean the cospan  $n \xrightarrow{\pi} n \xleftarrow{id} n$ .

► **Lemma 18** (Level-0 decomposition). *Let  $m \rightarrow G \leftarrow n$  be a right-monogamous acyclic cospan whose order-0 terminal nodes are the first  $k$  nodes of  $n$ . Then there exists a unique decomposition of  $G$  as  $\mathcal{M}; (id_k \oplus (\mathcal{D}; \mathcal{G}'))$ , where (A)  $\mathcal{M}$  is monogamous acyclic and contains precisely all the level-0 hyperedges; (B)  $\mathcal{D}$  is discrete right-monogamous and contains precisely all order-1 left-amonogamous nodes; (C)  $\mathcal{G}'$  is right-monogamous acyclic and has no left-amonogamous nodes without any in-connections.*

Moreover, any two such factorisations differ only by permutations of the terminal nodes of the factors, i.e., if  $\mathcal{G} = \mathcal{M}; (id_k \oplus (\mathcal{D}; \mathcal{G}')) = \mathcal{M}'; (id_k \oplus (\mathcal{D}'; \mathcal{G}'))$ , then there exists permutations  $\pi, \theta$  such that  $\mathcal{M}' = \mathcal{M}; \pi$ ,  $\pi; (id_k \oplus (\mathcal{D}'; \mathcal{G}')) = id_k \oplus (\mathcal{D}; \mathcal{G}')$ , and  $\mathcal{D}' = \mathcal{D}; \theta$ ,  $\mathcal{G}'' = \mathcal{G}'$ .

Note that the non-uniqueness of the decomposition comes from two distinct sources: 1) arbitrary ordering of nodes on the boundaries of cospans, and 2) the commutativity of the monoid multiplication which can absorb any permutation of the wires that it merges.

► **Lemma 19** (Factorisation into levels). *Any right-monogamous acyclic cospan  $\mathcal{G} = m \rightarrow G \leftarrow n$  can be factored into  $\mathcal{M}_0; (id_{k_0} \oplus (\mathcal{D}_0; \dots; \mathcal{M}_l; (id_{k_l} \oplus \mathcal{D}_l) \dots)); \pi$  for some permutation  $\pi$  and where, for each  $i$ , (A)  $\mathcal{M}_i$  is monogamous acyclic and contains precisely the level  $i$  hyperedges of  $\mathcal{G}$ , and (B)  $\mathcal{D}_i$  is discrete right-monogamous, contains all order  $i + 1$  left-amonogamous nodes and all order- $i$  terminal nodes of  $\mathcal{G}$ .*

**Proof.** First, let  $k_i$  be the set of order- $i$  terminal nodes of  $\mathcal{G}$ . We then define the permutation  $\pi$  to be the reordering of  $n$  such that the order-0 terminal nodes are the first  $k_0$  nodes, the order-1 are the next  $k_1$  terminal nodes and, more generally, the order- $i + 1$  nodes are the first  $k_{i+1}$  terminal nodes after the first  $\sum_{j=0}^i k_j$  nodes.

We can now prove the lemma using induction on the highest order of left-amonogamous nodes in  $\mathcal{G}; \pi^{-1}$ , using Lemma 18.

For the base case we note that any right-monogamous acyclic cospan without any left-amonogamous nodes is simply monogamous acyclic.

For the induction hypothesis, we assume that the statement holds for all the right-monogamous acyclic cospans with the maximum order of left-amonogamous nodes strictly less than  $r$ , where  $r$  is a positive integer. For the inductive case, suppose that  $\mathcal{G}; \pi^{-1}$  is a cospan whose highest order of left-amonogamous nodes is  $r$ . Then, by Lemma 18 (Level-0 decomposition), it can be factored into  $\mathcal{G}; \pi^{-1} = \mathcal{M}_0; (id_{k_0} \oplus (\mathcal{D}_0; \mathcal{G}'))$  where the first cospan is monogamous acyclic and contains all level 0 hyperedges of  $\mathcal{G}$ , and  $\mathcal{D}_0$  is discrete right monogamous with all order 1 left-amonogamous nodes of  $\mathcal{G}; \pi^{-1}$  and  $k_0$  all order 0 terminal nodes of  $\mathcal{G}; \pi^{-1}$ . Now, every node in  $\mathcal{G}'$  corresponding to an order  $i$  left-amonogamous node in  $\mathcal{G}; \pi^{-1}$ , is now an order  $i - 1$  left-amonogamous. Thus, the highest order of left-amonogamous nodes in  $\mathcal{G}'$  is  $r - 1$  and, by the induction hypothesis,  $\mathcal{G}'$  can be factored into  $\mathcal{M}_1; (id_{k_1} \oplus (\mathcal{D}_1; \dots; \mathcal{M}_l; (id_{k_l} \oplus \mathcal{D}_l) \dots))$  as in the statement of the lemma. Therefore, the composite  $\mathcal{M}_0; (id_{k_0} \oplus (\mathcal{D}_0; \dots; \mathcal{M}_l; (id_{k_l} \oplus \mathcal{D}_l) \dots)) = \mathcal{M}_0; (id_{k_0} \oplus (\mathcal{D}_0; \mathcal{G}')) = \mathcal{G}; \pi^{-1}$  satisfies conditions (A) and (B) of the lemma and  $\mathcal{M}_0; (id_{k_0} \oplus (\mathcal{D}_0; \dots; \mathcal{M}_l; (id_{k_l} \oplus \mathcal{D}_l) \dots)); \pi$  is the factorisation we are looking for. ◀

We will also need the following simpler form of the factorisation into levels, which matches closely the leading intuition of a factorisation into an alternating composition of monogamous and discrete right-monogamous cospans.

► **Corollary 20.** *Any right-monogamous acyclic cospan  $\mathcal{G} = m \rightarrow G \leftarrow n$  can be factorised into an alternating sequence of monogamous cospans and discrete right-monogamous cospans, i.e., as  $\mathcal{M}_0; \mathcal{D}_0; \dots; \mathcal{M}_l; \mathcal{D}_l$ .*

*Moreover any two such factorisations differ only by permutations of the terminal nodes of each factor, i.e., if  $\mathcal{M}_0; \mathcal{D}_0; \dots; \mathcal{M}_l; \mathcal{D}_l = \mathcal{M}'_0; \mathcal{D}'_0; \dots; \mathcal{M}'_l; \mathcal{D}'_l$ , there exists permutations  $\pi_i, \theta_i$  such that  $\mathcal{M}'_i = \mathcal{M}_i; \pi_i$ ,  $\pi \mathcal{D}'_i = \mathcal{D}_i$  and  $\mathcal{D}'_i = \mathcal{D}_i; \theta_i$ ,  $\theta_i \mathcal{M}'_{i+1} = \mathcal{M}_{i+1}$ .*

**Proof.** Since identities can be seen as monogamous cospans or discrete right-monogamous, and a permutation can be seen as discrete right-monogamous cospan, if we can get a factorisation of  $\mathcal{G}$  into levels as in Lemma 19, we also obtain a factorisation as in the statement of this lemma.

Finally, we can prove by induction, using from the second part of the statement of Lemma 18 that any two such factorisations differ only by some permutation of the factors. ◀

We are now able to conclude with our characterisation theorem.

► **Theorem 21.** *There exists an isomorphism  $\langle \cdot \rangle : \mathbf{S}_\Sigma + \mathbf{CMon} \rightarrow \mathbf{RMACsp}_D(\mathbf{Hyp}_\Sigma)$ .*

**Proof.** Let us define  $\langle \cdot \rangle$  as a copairing (in PROP) of the faithful functors  $\llbracket \cdot \rrbracket : \mathbf{S}_\Sigma \rightarrow \mathbf{RMACsp}_D(\mathbf{Hyp}_\Sigma)$  and  $|\cdot| : \mathbf{CMon} \rightarrow \mathbf{RMACsp}_D(\mathbf{Hyp}_\Sigma)$ . It suffices to show that the prop  $\mathbf{RMACsp}_D(\mathbf{Hyp}_\Sigma)$  satisfies the universal property of the coproduct  $\mathbf{S}_\Sigma + \mathbf{CMon}$  in PROP:

$$\begin{array}{ccccc}
 \mathbf{S}_\Sigma & \xrightarrow{\llbracket \cdot \rrbracket} & \mathbf{RMACsp}_D(\mathbf{Hyp}_\Sigma) & \xleftarrow{|\cdot|} & \mathbf{CMon} \\
 & \searrow \alpha & \downarrow \exists! \gamma & \swarrow \beta & \\
 & & \mathbb{A} & & 
 \end{array}$$

Given a prop  $\mathbb{A}$  and prop-morphisms  $\alpha : \mathbf{S}_\Sigma \rightarrow \mathbb{A}$ ,  $\beta : \mathbf{CMon} \rightarrow \mathbb{A}$ , we need to prove there exists a unique prop-morphism  $\gamma : \mathbf{RMACsp}_D(\mathbf{Hyp}_\Sigma) \rightarrow \mathbb{A}$ , such that the diagram above commutes. Now, since prop-morphisms are identity-on-objects functors, it is sufficient to consider what happens to the arrows of the above props. Since the diagram needs to commute, for any arrow  $s$  in  $\mathbf{S}_\Sigma$  and for any arrows  $c$  in  $\mathbf{CMon}$  we want  $\gamma(\llbracket s \rrbracket) = \alpha(s)$  and  $\gamma(|c|) = \beta(c)$ . But, by Corollary 20, any cospan  $\mathcal{G}$  in  $\mathbf{RMACsp}_D(\mathbf{Hyp}_\Sigma)$  can be factorised as an alternating sequence of monogamous cospans and discrete right-monogamous cospans, *i.e.*, as  $\mathcal{G} = \mathcal{M}_0; \mathcal{D}_0 \dots; \mathcal{M}_l; \mathcal{D}_l$  where  $\mathcal{M}_i = \llbracket s_i \rrbracket$  for some  $s_i$  in  $\mathbf{S}_\Sigma$  and  $\mathcal{D}_i = |c_i|$  for some  $c_i$  in  $\mathbf{CMon}$ . Then  $\gamma(h_i)$  is uniquely defined by the conditions  $\gamma(\llbracket s_i \rrbracket) = \alpha(s_i)$  and  $\gamma(|c_i|) = \beta(c_i)$ : let  $\gamma(\mathcal{G}) = \gamma(\mathcal{M}_0; \mathcal{D}_0; \dots; \mathcal{M}_l; \mathcal{D}_l) = \alpha(s_0)\beta(c_0) \dots \alpha(s_l)\beta(c_l)$ . We now verify that  $\gamma$  is well-defined and functorial.

*Well-definedness.* Since the factorisation of  $\mathcal{G}$  into levels is not unique, we need to show that  $\gamma$  is well-defined, *i.e.*, that any two such factorisations will define the same value of  $\gamma(\mathcal{G})$ . Consider another factorisation  $\mathcal{G} = \mathcal{M}'_0; \mathcal{D}'_0; \dots; \mathcal{M}'_l; \mathcal{D}'_l$  obtained from Corollary 20. Then there exists permutations  $\pi_i, \theta_i$  such that  $\mathcal{M}'_i = \mathcal{M}_i; \pi_i$ ,  $\pi_i \mathcal{D}'_i = \mathcal{D}_i$  and  $\mathcal{D}'_i = \mathcal{D}_i; \theta_i$ ,  $\theta_i \mathcal{M}'_{i+1} = \mathcal{M}_{i+1}$ . In addition,  $\mathcal{M}'_i = \llbracket s'_i \rrbracket$  for some  $s'_i$  in  $\mathbf{S}_\Sigma$ ,  $\mathcal{D}'_i = |c'_i|$  for some  $c'_i$  in  $\mathbf{CMon}$ . To show well-definedness of  $\gamma$ , we will use the following facts:

1. since  $\mathbf{S}_\Sigma$ ,  $\mathbf{CMon}$  and  $\mathbb{A}$  are props, they all contain a copy of the prop of permutations so we will abuse notation slightly and use the same names to refer to the same permutation in all of them;
2. prop morphisms preserve permutations so that  $\alpha(\pi) = \beta(\pi) = \gamma(\pi) = \llbracket \cdot \rrbracket \pi = | \cdot | \pi = \pi$  for any permutation  $\pi$ ;
3. by definition of  $\gamma$  it is clear that  $\gamma(\mathcal{G}; \pi) = \gamma(\mathcal{G}); \pi$  and  $\gamma(\pi; \mathcal{G}) = \pi; \gamma(\mathcal{G})$ .

Now, we have

$$\begin{aligned}
 \gamma(\mathcal{M}_0; \mathcal{D}_0; \dots; \mathcal{M}_l; \mathcal{D}_l) &= \gamma(\mathcal{M}_0); \gamma(\mathcal{D}_0); \dots; \gamma(\mathcal{M}_l); \gamma(\mathcal{D}_l) \\
 &= \gamma(\mathcal{M}_0); \gamma(\pi_0; \mathcal{D}'_0); \dots; \gamma(\mathcal{M}_l); \gamma(\pi_l; \mathcal{D}'_l) \\
 &= \gamma(\mathcal{M}_0); \pi_0; \gamma(\mathcal{D}'_0); \dots; \gamma(\mathcal{M}_l); \pi_l; \gamma(\mathcal{D}'_l) \\
 &= \gamma(\mathcal{M}_0; \pi_0); \gamma(\mathcal{D}'_0); \dots; \gamma(\mathcal{M}_l; \pi_l); \gamma(\mathcal{D}'_l) \\
 &= \gamma(\mathcal{M}'_0); \gamma(\mathcal{D}'_0); \dots; \gamma(\mathcal{M}'_l); \gamma(\mathcal{D}'_l) = \gamma(\mathcal{M}'_0; \mathcal{D}'_0; \dots; \mathcal{M}'_l; \mathcal{D}'_l)
 \end{aligned}$$

*Monoidal functoriality.* First,  $\gamma$  preserves monoidal products, as the decomposition of a monoidal product is obtained by taking a monoidal product of monogamous acyclic cospans, and a monoidal product of discrete right monogamous cospans, for each level separately. Second, consider two cospans  $\mathcal{G} = m \rightarrow G \leftarrow n$  and  $\mathcal{H} = n \rightarrow G \leftarrow o$ . We can factorise  $\mathcal{H}$  as  $\mathcal{M}_0; \mathcal{D}_0; \dots; \mathcal{M}_l; \mathcal{D}_l$ . Hence, if we can show that  $\gamma(\mathcal{G}; \mathcal{M}; \mathcal{D}) = \gamma(\mathcal{G}); \gamma(\mathcal{M}; \mathcal{D})$ , for  $\mathcal{M}$  monogamous and  $\mathcal{D}$  discrete right-monogamous, a simple induction will allow us to conclude that  $\gamma(\mathcal{G}; \mathcal{H}) = \gamma(\mathcal{G}); \gamma(\mathcal{H})$ . In fact, to show the induction step, it is enough to show that  $\gamma(\mathcal{G}; (\mathcal{M} \oplus id_{n-k+l})) = \gamma(\mathcal{G}); \gamma(\mathcal{M} \oplus id_{n-k+l})$  where  $\mathcal{M}$  consists of a single hyperedge  $h$ , with  $k$  source nodes and  $l \leq n$  target nodes – we can recover the general case of all monogamous cospans by performing another induction on the number of hyperedges in  $\mathcal{M}$ .

Now, we need to understand to what level in  $\mathcal{G}; (\mathcal{M} \oplus id_{n-k+l})$  the single hyperedge  $h$  of  $\mathcal{M}$  belongs. By the definition of the level of hyperedges (Definition 16),  $h$  will belong to level  $i$  of in  $\mathcal{G}; (\mathcal{M} \oplus id_{n-k+l})$  if the node with the largest order in the first  $k$  terminal nodes of  $\mathcal{G}$  is  $i$ . If we assume without loss of generality (as we can always post-compose with a permutation to achieve this), that the terminal nodes of  $\mathcal{G}$  are ordered by order size, this implies that the factorisation of  $\mathcal{G}; (\mathcal{M} \oplus id_{n-k+l})$  into levels is  $\mathcal{G}_{\leq i}; (\mathcal{M} \oplus \mathcal{G}_{> i})$

where  $\mathcal{G}_{>i}$  and  $\mathcal{G}_{\leq i}$  are obtained from the factorisation  $\mathcal{M}_0; (id_{k_0} \oplus (\mathcal{D}_0; \dots; \mathcal{M}_l; (id_{k_l} \oplus \mathcal{D}_l) \dots))$  of  $\mathcal{G}$  (from Lemma 19) as follows:  $\mathcal{G}_{\leq i} := \mathcal{M}_0; (id_{k_0} \oplus (\mathcal{D}_0; \dots; \mathcal{M}_i) \dots)$  and  $\mathcal{G}_{>i} := \mathcal{D}_i; \mathcal{M}_{i+1}; (id_{k_{i+1}} \oplus (\mathcal{D}_{i+1}; \dots; \mathcal{M}_l; (id_{k_l} \oplus \mathcal{D}_l) \dots))$ . Note that, by construction, we have  $\mathcal{G} = \mathcal{G}_{\leq i}; (id_{k_i} \oplus \mathcal{G}_{>i})$ . Thus

$$\begin{aligned} \mathcal{G}; (\mathcal{M} \oplus id_{n-k+l}) &= \mathcal{G}_{\leq i}; (id_{k_i} \oplus \mathcal{G}_{>i}); (\mathcal{M} \oplus id_{n-k+l}) \\ &= \mathcal{G}_{\leq i}; ((id_{k_i}; \mathcal{M}) \oplus (\mathcal{G}_{>i}; id_{n-k+l})) \\ &= \mathcal{G}_{\leq i}; (\mathcal{M} \oplus \mathcal{G}_{>i}) \end{aligned}$$

by the interchange and unitality axioms of symmetric monoidal categories (see Fig. 1).

The intuition now is that we are able to slide the hyperedge  $h$  back to level  $i$  into the decomposition of  $\mathcal{G}$  and that the operation of sliding back – which only uses the monoidal product and composition with identities – is preserved by  $\gamma$ . This will be sufficient to prove functoriality of  $\gamma$ . We have

$$\begin{aligned} \gamma(\mathcal{G}; (\mathcal{M} \oplus id_{n-k+l})) &= \gamma(\mathcal{G}_{\leq i}; (\mathcal{M} \oplus \mathcal{G}_{>i})) \\ &= \gamma(\mathcal{G}_{\leq i}); \gamma(\mathcal{M} \oplus \mathcal{G}_{>i}) \\ &= \gamma(\mathcal{G}_{\leq i}); (\gamma(\mathcal{M}) \oplus \gamma(\mathcal{G}_{>i})) \\ &= \gamma(\mathcal{G}_{\leq i}); (\gamma(id_{k_i}) \oplus \gamma(\mathcal{G}_{>i})); (\gamma(\mathcal{M}) \oplus \gamma(id_{n-k+l})) \end{aligned}$$

where the second equality holds because  $\mathcal{G}_{\leq i}; (\mathcal{M} \oplus \mathcal{G}_{>i})$  is the factorisation of  $\mathcal{G}; \mathcal{M}$  through which we define  $\gamma$ ; the third equality holds because  $\gamma$  preserves monoidal products and the remaining equalities use the interchange and unitality laws of symmetric monoidal categories as above. Finally, by definition of  $\gamma$ ,  $\gamma(\mathcal{G}_{\leq i}); (\gamma(id_{k_i}) \oplus \gamma(\mathcal{G}_{>i})) = \gamma(\mathcal{G})$  and, since  $\gamma$  also preserves monoidal products, we can conclude that  $\gamma(\mathcal{G}; (\mathcal{M} \oplus id_{n-k+l})) = \gamma(\mathcal{G}); \gamma(\mathcal{M} \oplus id_{n-k+l})$  as we wanted to show. ◀

## 4 Characterisation of String Diagram Rewriting

Now that we have a characterisation theorem for  $\mathbf{S}_\Sigma + \mathbf{CMon}$ , we are ready to interpret rewriting modulo commutative monoid structure as DPO rewriting, and to show that such a correspondence is sound and complete. We first recall formally the former notion of rewriting.

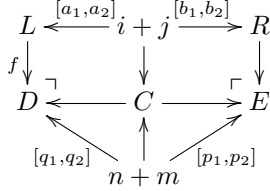
► **Definition 22** (String Diagram Rewriting Modulo  $\mathbf{CMon}$ ). *Let  $d, e: n \rightarrow m$  and  $l, r: i \rightarrow j$  be pairs of morphisms in  $\mathbf{S}_\Sigma + \mathbf{CMon}$ . We say that  $d$  rewrites into  $e$  modulo commutative monoid structure according to the rewrite rule  $\mathcal{R} = \langle l, r \rangle$ , notation  $d \Rightarrow_{\mathcal{R}} e$ , if, in  $\mathbf{S}_\Sigma + \mathbf{CMon}$ , we have:*

$$\begin{array}{c} n \\ \downarrow \\ \boxed{d} \\ \downarrow \\ m \end{array} = \begin{array}{c} n \\ \downarrow \\ \boxed{c_1} \begin{array}{c} \xrightarrow{k} \\ \boxed{i} \quad \boxed{j} \\ \downarrow \quad \downarrow \\ \boxed{l} \quad \boxed{j} \\ \downarrow \quad \downarrow \\ \boxed{c_2} \\ \downarrow \\ m \end{array} \end{array} \quad \begin{array}{c} n \\ \downarrow \\ \boxed{e} \\ \downarrow \\ m \end{array} = \begin{array}{c} n \\ \downarrow \\ \boxed{c_1} \begin{array}{c} \xrightarrow{k} \\ \boxed{i} \quad \boxed{j} \\ \downarrow \quad \downarrow \\ \boxed{r} \quad \boxed{j} \\ \downarrow \quad \downarrow \\ \boxed{c_2} \\ \downarrow \\ m \end{array} \end{array} \quad (4)$$

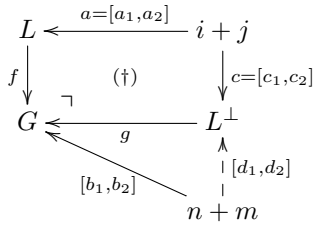
As studied in [5, 6], rewriting of string diagrams may be interpreted as DPO rewriting of the corresponding hypergraphs. The relevant notion is the one of DPO rewriting “with interfaces” (originally used for a single interface in [16], and adapted for two interfaces in [4]), which ensures preservation of the interfaces described by the cospan structure.

9:12 String Diagram Rewriting Modulo Commutative (Co)Monoid Structure

► **Definition 23** (DPO Rewriting (with interfaces)). Consider a DPO rewrite rule  $\mathcal{R} = L \xleftarrow{[a_1, a_2]} i + j \xrightarrow{[b_1, b_2]} R$  given by cospans  $i \xrightarrow{a_1} L \xleftarrow{a_2}$  and  $i \xrightarrow{b_1} R \xleftarrow{b_2}$  in  $\mathbf{Hyp}_\Sigma$ . We say that cospan  $n \xrightarrow{q_1} D \xleftarrow{q_2} m$  rewrites into  $n \xrightarrow{p_1} E \xleftarrow{p_2} m$  with rule  $\mathcal{R}$ , written  $(n \rightarrow D \leftarrow m) \xrightarrow{\mathcal{R}} (n \rightarrow E \leftarrow m)$ , if there is a cospan  $i + j \rightarrow C \leftarrow n + m$  (called the pushout complement) making the diagram below commutes with the two squares being pushouts.

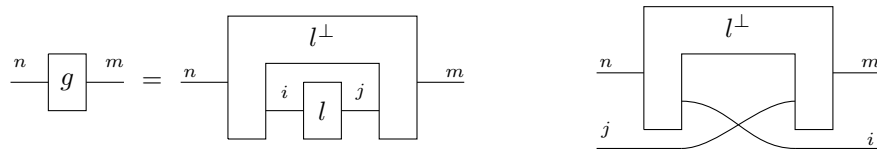


However, unless string diagram rewriting happens modulo the laws of Frobenius algebras, not all DPO rewrites are sound for string diagram rewriting: some pushout complements may yield as outcome of the rewriting hypergraphs that are not in the image of any string diagram [6]. To avoid these situations, [6] introduced the notion of boundary complements and convex matching. The former guarantees that inputs can only be connected to outputs and vice-versa, while the latter are matches that do not contain directed paths from outputs to inputs, *i.e.*, monomorphisms whose image is convex. However, these notions were designed for monogamous hypergraphs, and string diagram rewriting modulo symmetric monoidal structure. In order to capture the correct notion of DPO rewriting for right-monogamous hypergraphs, and rewriting modulo commutative monoid structure, we need to relax the first slightly to that of *weak boundary complements*.



► **Definition 24** (Weak boundary complement). For right-monogamous acyclic cospans  $i \xrightarrow{a_1} L \xleftarrow{a_2} j$  and  $n \xrightarrow{b_1} G \xleftarrow{b_2} m$  and a morphism  $f : L \rightarrow G$ , a pushout complement as on the right above is called a *weak boundary complement* if: (A) given two nodes in  $L$  that are mapped to the same node in  $G$  by  $f$ , they must be in the image of  $a_2$ ; (B)  $c_1$  is mono; (C) no two nodes are both in the image of  $c_1$  and  $c_2$ ; (D) there exist  $d_1 : n \rightarrow L^\perp$  and  $d_2 : m \rightarrow L^\perp$  making the above diagram commute and such that  $n + j \xrightarrow{[c_2, d_1]} L^\perp \xleftarrow{[c_1, d_2]} m + i$  is right-monogamous.

Intuitively, the complement  $L^\perp$  is  $G$  with an  $L$ -shaped hole:



where  $g : n \rightarrow m$ ,  $l : i \rightarrow j$ , and  $l^\perp : n + j \rightarrow m + i$  are diagrams for the cospans  $n \rightarrow G \leftarrow m, i \rightarrow L \leftarrow j$ , and  $n + j \rightarrow L^\perp \leftarrow m + i$  respectively, *i.e.* such that  $\langle\langle g \rangle\rangle = (n \rightarrow G \leftarrow m), \langle\langle l \rangle\rangle = i \rightarrow L \leftarrow j$ , and  $\langle\langle l^\perp \rangle\rangle = n + j \rightarrow L^\perp \leftarrow m + i$ . (Recall that

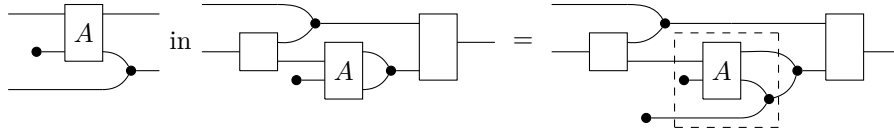
$\langle\langle \cdot \rangle\rangle : \mathbf{S}_\Sigma + \mathbf{CMon} \rightarrow \mathbf{RMAC}_{\text{sp}_D}(\mathbf{Hyp}_\Sigma)$  is the isomorphism established by Theorem 21; we will use it quite liberally from now on in order to manipulate cospans as string diagrams when convenient). Boundary complements restrict the shape that these can take. Let us explain the conditions of Definition 24 in plainer language.

- Condition (A) allows matches to occur in a diagram  $G$  that contains the sub-diagram  $L$  potentially with some nodes identified, *i.e.* wires connected by the monoid multiplication (see Example 26 below. However, these can only occur as terminal nodes, that is, in the image of  $a_2$ , the right boundary of the subdiagram  $L$ ).
- Plain boundary complements [6] require  $c_1, c_2$  to be jointly monic. This enforces two distinct properties: it prevents nodes from the left and right boundaries of the match to be identified, and it prevents nodes from within each of the two boundary sides to be identified. Here, we need to relax the second condition to allow nodes in the right boundary of the match to be identified. This is what conditions (B) and (C) give us.
- Condition (D) forces the boundary of the complement, both with the subdiagram  $L$  and those of the larger diagram  $G$ , to be right-monogamous. In other words, we want the cospan  $n + j \xrightarrow{[c_2, d_1]} L^\perp \xleftarrow{[c_1, d_2]} m + i$  depicted above to be right-monogamous.

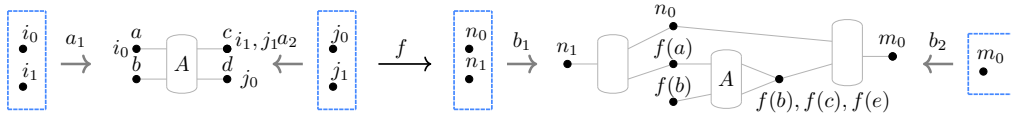
The last ingredient we require is the same as in [6]: we require the match to be *convex*.

► **Definition 25** (Convex matching [6]).  $f : L \rightarrow G$  in  $\mathbf{Hyp}_\Sigma$  is a convex match if its image is a convex sub-hypergraph of  $G$ .

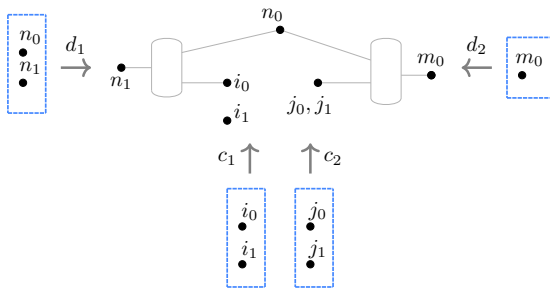
► **Example 26.** Consider the diagram below



As cospans of hypergraphs, this corresponds to the convex matching below



with the following weak boundary complement:



► **Definition 27** (Weakly Convex DPO Rewriting). We call a DPO rewriting step as in Definition 23 weakly convex if  $f : L \rightarrow D$  is a convex matching and  $i + j \rightarrow C$  is a weak boundary complement in the leftmost pushout square.

Note that, contrary to boundary complements in the symmetric monoidal case [6], weak boundary complements are not necessarily unique if they exist. We can now conclude the soundness and completeness of weakly convex DPO rewriting for string diagrams with commutative monoid structure.

## 9:14 String Diagram Rewriting Modulo Commutative (Co)Monoid Structure

► **Theorem 28.** *Let  $\mathcal{R} = \langle l, r \rangle$  be a rewrite rule on  $\mathbf{S}_\Sigma + \mathbf{CMon}$ . Then,*

$$d \Rightarrow_{\mathcal{R}} e \text{ iff } \langle\langle d \rangle\rangle \hookrightarrow_{\langle\mathcal{R}\rangle} \langle\langle e \rangle\rangle \quad (5)$$

**Proof.** For the direction from left to right we proceed as follows. From the definition of rewriting, and given the assumption  $d \Rightarrow_{\mathcal{R}} e$ , we have equalities as in (4). We now interpret the string diagrams involved, obtaining right-monogamous cospans:

$$\begin{aligned} \left( n \xrightarrow{q_1} D \xleftarrow{q_2} m \right) &:= \langle\langle d \rangle\rangle & \left( n \xrightarrow{p_1} E \xleftarrow{p_2} m \right) &:= \langle\langle e \rangle\rangle \\ \left( i \xrightarrow{a_1} L \xleftarrow{a_2} j \right) &:= \langle\langle l \rangle\rangle & \left( i \xrightarrow{b_1} R \xleftarrow{b_2} j \right) &:= \langle\langle r \rangle\rangle \\ \left( n \xrightarrow{x_1} C_1 \xleftarrow{x_2} k + i \right) &:= \langle\langle c_1 \rangle\rangle & \left( k + j \xrightarrow{y_1} C_2 \xleftarrow{y_2} m \right) &:= \langle\langle c_2 \rangle\rangle \end{aligned} \quad (6)$$

From the last two cospans above, by simply rearranging nodes on the interface from the left to the right and viceversa, we obtain:

$$i + k \xrightarrow{\tilde{x}_1} \tilde{C}_1 \xleftarrow{\tilde{x}_2} n \quad k + j \xrightarrow{\tilde{y}_1} \tilde{C}_2 \xleftarrow{\tilde{y}_2} m$$

We now define a cospan  $i + j \rightarrow C \leftarrow n + m$  as:

$$\left( i + j \xrightarrow{z_1} i + k + j \xleftarrow{z_2} i + k + k + j \right); \left( \begin{array}{c} i + k \xrightarrow{\tilde{x}_1} \tilde{C}_1 \xleftarrow{\tilde{x}_2} n \\ \oplus \\ k + j \xrightarrow{\tilde{y}_1} \tilde{C}_2 \xleftarrow{\tilde{y}_2} m \end{array} \right) \quad (7)$$

where  $z_1: (i + j) \rightarrow (i + k + j)$  is the inclusion map,  $z_2: (i + k + k + j) \rightarrow (i + k + j)$  is defined as  $id_i + \mu_k + id_j$ , with  $\mu_k: k + k \rightarrow k$  mapping both copies of node  $i \in \{0, \dots, k - 1\}$  to  $i$ . Intuitively,  $i + j \rightarrow C \leftarrow n + m$  represents the string diagram where we have rearranged interface nodes  $n$  and  $i$  on the opposite interface. One may verify that:

$$\begin{aligned} \left( 0 \rightarrow D \xleftarrow{[q_1, q_2]} n + m \right) &= \left( 0 \rightarrow L \xleftarrow{[a_1, a_2]} i + j \right); (i + j \rightarrow C \leftarrow n + m) \\ \left( 0 \rightarrow E \xleftarrow{[p_1, p_2]} n + m \right) &= \left( 0 \rightarrow R \xleftarrow{[b_1, b_2]} i + j \right); (i + j \rightarrow C \leftarrow n + m) \end{aligned} \quad (8)$$

Recall that composition of cospans is obtained via pushouts, hence the two equalities of (8) yield a DPO rewriting step  $\langle\langle d \rangle\rangle \hookrightarrow_{\langle\mathcal{R}\rangle} \langle\langle e \rangle\rangle$  as in Definition 23. Since  $l$  is simply a sub-string diagram of  $d$ , the mapping from  $L$  to  $D$  is a convex match. Furthermore, note that no two nodes from  $i + k$  can be identified with each other, hence  $i \rightarrow C$  is mono, and no node from  $i$  can be identified with any node in  $j$  or  $m$ . As  $m \rightarrow C$  is trivially mono, we have that  $C$  is indeed a weak boundary complement.

Now we deal with the converse implication. Assume  $\langle\langle d \rangle\rangle \hookrightarrow_{\langle\mathcal{R}\rangle} \langle\langle e \rangle\rangle$ , where  $\langle\langle d \rangle\rangle$ ,  $\langle\langle e \rangle\rangle$ ,  $\langle\langle l \rangle\rangle$ , and  $\langle\langle r \rangle\rangle$  are defined as the cospans in (6). By assumption, and since composition of cospans is performed via pushouts, there exists a weak boundary complement  $i + j \xrightarrow{[c_1, c_2]} L^\perp \xleftarrow{[d_1, d_2]} n + m$  such that

$$\begin{aligned} \langle\langle d \rangle\rangle &= (0 \rightarrow i \xleftarrow{\mu_i} i + i); (id_i \oplus \langle\langle l \rangle\rangle); (i + j \xrightarrow{[c_1, c_2]} L^\perp \xleftarrow{[d_1, d_2]} n + m) \\ \langle\langle e \rangle\rangle &= (0 \rightarrow i \xleftarrow{\mu_i} i + i); (id_i \oplus \langle\langle r \rangle\rangle); (i + j \xrightarrow{[c_1, c_2]} L^\perp \xleftarrow{[d_1, d_2]} n + m) \end{aligned}$$



We can now apply Lemma 12 (weak decomposition) to  $n + j \xrightarrow{[c_2, d_1]} L^\perp \xleftarrow{[c_1, d_2]} m + i$ , with the convex sub-hypergraph of  $L^\perp$  given by the image of  $i$  and  $j$ , to obtain a decomposition of  $n + j \xrightarrow{[c_2, d_1]} L^\perp \xleftarrow{[c_1, d_2]} m + i$  as

$$(n + j \rightarrow C_1 \leftarrow k + i + j); (k + i + j \xrightarrow{id_k \oplus \sigma_i^j} k + j + i \leftarrow k + j + i); (k + j + i \rightarrow C_2 \leftarrow m + i)$$

for some  $k \in \mathbb{N}$ , right monogamous cospans  $n \rightarrow C_1 \leftarrow i + k$  and  $j + k \rightarrow C_2 \leftarrow m$ , and where  $\sigma_i^j : i + j \rightarrow j + i$  the map that swaps the two components  $i$  and  $j$ . By fullness of  $\langle\langle \cdot \rangle\rangle$  we have  $c_1, c_2$  such that  $\langle\langle c_1 \rangle\rangle = n \rightarrow C_1 \leftarrow i + k$  and  $\langle\langle c_2 \rangle\rangle = j + k \rightarrow C_2 \leftarrow m$ ; moreover we have, by construction:

$$\begin{aligned} \langle\langle d \rangle\rangle &= (id_n \oplus (0 \rightarrow i \xleftarrow{\mu_i} i + i)); \langle\langle \begin{array}{c} n \\ \xrightarrow{\quad} \boxed{l} \xrightarrow{j} \\ \xleftarrow{i} \end{array} \rangle\rangle; \langle\langle \begin{array}{c} k \\ \xrightarrow{\quad} \boxed{c_1} \xrightarrow{\quad} \boxed{c_2} \xrightarrow{m} \\ \xleftarrow{i} \end{array} \rangle\rangle; (id_m \oplus i + i \xrightarrow{\mu_i} i \leftarrow 0) \\ \langle\langle e \rangle\rangle &= (id_n \oplus (0 \rightarrow i \xleftarrow{\mu_i} i + i)); \langle\langle \begin{array}{c} n \\ \xrightarrow{\quad} \boxed{r} \xrightarrow{j} \\ \xleftarrow{i} \end{array} \rangle\rangle; \langle\langle \begin{array}{c} k \\ \xrightarrow{\quad} \boxed{c_1} \xrightarrow{\quad} \boxed{c_2} \xrightarrow{m} \\ \xleftarrow{i} \end{array} \rangle\rangle; (id_m \oplus (i + i \xrightarrow{\mu_i} i \leftarrow 0)) \end{aligned}$$

Computing these cospans, we obtain  $\langle\langle d \rangle\rangle = \langle\langle c_1 \rangle\rangle; (\langle\langle id_k \rangle\rangle \oplus \langle\langle l \rangle\rangle); \langle\langle c_2 \rangle\rangle$  and  $\langle\langle e \rangle\rangle = \langle\langle c_1 \rangle\rangle; (\langle\langle id_k \rangle\rangle \oplus \langle\langle r \rangle\rangle); \langle\langle c_2 \rangle\rangle$ . By monoidal functoriality of  $\langle\langle \cdot \rangle\rangle$ , we thus have  $\langle\langle d \rangle\rangle = \langle\langle (c_1; (id_k \oplus l); c_2) \rangle\rangle$  and  $\langle\langle e \rangle\rangle = \langle\langle (c_1; (id_k \oplus r); c_2) \rangle\rangle$ . Finally, since  $\langle\langle \cdot \rangle\rangle$  is faithful, we can conclude that  $d = c_1; (id_k \oplus l); c_2$  and  $e = c_1; (id_k \oplus r); c_2$ . This is precisely what it means to apply the rule  $\langle l, r \rangle$  to  $d$ , so that  $d \Rightarrow_{\emptyset} e$  as we wanted to prove.  $\blacktriangleleft$

## 5 Conclusions and Future Work

The main contribution of this work is twofold. First, with Theorem 21, we identified a combinatorial representation of string diagrams modulo commutative monoid structure. This correspondence relies on introducing a notion of right monogamous cospans, which is intermediate between the “vanilla” cospans characterising string diagrams modulo Frobenius structure, and the monogamous cospans characterising string diagrams modulo symmetric monoidal structure. The characterisation result relies on a factorisation result for right monogamous cospans, which requires some ingenuity: compared with similar theorems in [5, 6] the increased sophistication is due to the fact that there is additional structure to consider both on the side of string diagrams (in contrast with [6, Theorem 25], which only accounts for symmetric monoidal structure) and of hypergraphs (in contrast with [5, Theorem 4.1], which accounts for generic hypergraph without monogamicity conditions). Note the work [19], which appeared at the same time as a preprint of our work [26], provides a result dual to Theorem 21: instead of monoids, they consider props with a chosen commutative *comonoid* structure – called “CD-categories” or “gs-categories”. On the side of hypergraphs, instead of restricting monogamicity to right-monogamicity, they consider *left-monogamicity*, which is essentially the dual notion.

The second contribution of our paper, Theorem 28, showed a correspondence between string diagrams rewriting modulo commutative monoid structure and a certain variant of DPO hypergraph rewriting. In order to ensure soundness and completeness, we introduced a suitable restriction of DPO rewriting, called *weakly convex* to echo the convex rewriting characterising string diagrams in a symmetric monoidal category [6]. A subtlety in this result was identifying a notion of boundary complement which, even though could not be unique “on-the-nose” as the one considered in the more restrictive convex rewriting, it was sufficiently well-behaved for the purpose of showing the correspondence with string diagram rewriting.

Going forward, we believe our approach could be extended to coloured props in a rather straightforward way, following the analogous developments in [31, 6]. Other interesting directions to pursue are the study of confluence [7] and the characterisations of notions of rewriting modulo structures intermediate between commutative monoid and Frobenius algebra – comparison with the very recent work on rewriting in traced comonoid structure [21] seems particularly promising in this regard. In terms of case studies, as mentioned in the introduction, our work paves the way for studying rewriting of theories which do not host a Frobenius structure, but at the same time include commutative (co)monoid equations, which would immediately lead to non-termination if taken as rewrite rules. Categories of matrix-like structures, based on Hopf algebras which would become degenerate if added Frobenius equations (see *eg.* [30] for an overview), seem a particularly fitting candidate for investigation.

---

### References

- 1 John Baez and Jason Erbele. Categories in control. *Theory and Applications of Categories*, 30:836–881, 2015. URL: <http://arxiv.org/abs/1405.6881>.
- 2 C. Berge. *Graphs and Hypergraphs*. Elsevier Science Ltd., GBR, 1985.
- 3 Guillaume Boisseau and Pawel Sobocinski. String diagrammatic electrical circuit theory. In *ACT*, volume 372 of *EPTCS*, pages 178–191, 2021.
- 4 Filippo Bonchi, Fabio Gadducci, Aleks Kissinger, Pawel Sobocinski, and Fabio Zanasi. Rewriting modulo symmetric monoidal structure. In Martin Grohe, Eric Koskinen, and Natarajan Shankar, editors, *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016*, pages 710–719. ACM, 2016. doi:10.1145/2933575.2935316.
- 5 Filippo Bonchi, Fabio Gadducci, Aleks Kissinger, Pawel Sobocinski, and Fabio Zanasi. String diagram rewrite theory I: Rewriting with Frobenius structure, 2020. doi:10.48550/ARXIV.2012.01847.
- 6 Filippo Bonchi, Fabio Gadducci, Aleks Kissinger, Pawel Sobocinski, and Fabio Zanasi. String diagram rewrite theory II: Rewriting with symmetric monoidal structure, 2021. doi:10.48550/ARXIV.2104.14686.
- 7 Filippo Bonchi, Fabio Gadducci, Aleks Kissinger, Pawel Sobociński, and Fabio Zanasi. String diagram rewrite theory III: Confluence with and without Frobenius, 2021. doi:10.48550/ARXIV.2109.06049.
- 8 Filippo Bonchi, Joshua Holland, Dusko Pavlovic, and Pawel Sobocinski. Refinement for signal flow graphs. In *CONCUR*, volume 85 of *LIPICs*, pages 24:1–24:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.
- 9 Filippo Bonchi, Joshua Holland, Robin Piedeleu, Pawel Sobocinski, and Fabio Zanasi. Diagrammatic algebra: from linear to concurrent systems. *Proc. ACM Program. Lang.*, 3(POPL):25:1–25:28, 2019. doi:10.1145/3290338.
- 10 Filippo Bonchi, Robin Piedeleu, Pawel Sobocinski, and Fabio Zanasi. Graphical affine algebra. In *34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2019, Vancouver, BC, Canada, June 24-27, 2019*, pages 1–12. IEEE, 2019. doi:10.1109/LICS.2019.8785877.
- 11 Filippo Bonchi, Pawel Sobocinski, and Fabio Zanasi. Deconstructing Lwvere with distributive laws. *J. Log. Algebraic Methods Program.*, 95:128–146, 2018.
- 12 Filippo Bonchi, Pawel Sobocinski, and Fabio Zanasi. A survey of compositional signal flow theory. In *IFIP's Exciting First 60+ Years*, volume 600 of *IFIP Advances in Information and Communication Technology*, pages 29–56. Springer, 2021.
- 13 Aurelio Carboni and R. F. C. Walters. Cartesian bicategories I. *J Pure Appl Algebra*, 49:11–32, 1987.

- 14 A. Corradini, Ugo Montanari, Francesca Rossi, H. Ehrig, Reiko Heckel, and Michael Löwe. Basic concepts and double pushout approach. *Algebraic Approaches to Graph Transformation*, pages 163–246, January 1997.
- 15 Geoffrey S. H. Cruttwell, Bruno Gavranovic, Neil Ghani, Paul W. Wilson, and Fabio Zanasi. Categorical foundations of gradient-based learning. In *ESOP*, volume 13240 of *Lecture Notes in Computer Science*, pages 1–28. Springer, 2022.
- 16 Hartmut Ehrig and Barbara König. Deriving bisimulation congruences in the DPO approach to graph rewriting. In Igor Walukiewicz, editor, *Foundations of Software Science and Computation Structures*, pages 151–166, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.
- 17 Brendan Fong and Fabio Zanasi. Universal constructions for (co)relations: categories, monoidal categories, and props. *Log. Methods Comput. Sci.*, 14(3), 2018.
- 18 Tobias Fritz. A synthetic approach to markov kernels, conditional independence and theorems on sufficient statistics. *Advances in Mathematics*, 370:107239, August 2020. doi:10.1016/j.aim.2020.107239.
- 19 Tobias Fritz and Wendong Liang. Free gs-monoidal categories and free Markov categories, 2022. doi:10.48550/ARXIV.2204.02284.
- 20 Dan R. Ghica, Achim Jung, and Aliaume Lopez. Diagrammatic semantics for digital circuits. In Valentin Goranko and Mads Dam, editors, *26th EACSL Annual Conference on Computer Science Logic, CSL 2017, August 20-24, 2017, Stockholm, Sweden*, volume 82 of *LIPICs*, pages 24:1–24:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPICs.CSL.2017.24.
- 21 Dan R. Ghica and George Kaye. Rewriting modulo traced comonoid structure. *CoRR*, abs/2302.09631, 2023.
- 22 Bart Jacobs, Aleks Kissinger, and Fabio Zanasi. Causal inference via string diagram surgery: A diagrammatic approach to interventions and counterfactuals. *Math. Struct. Comput. Sci.*, 31(5):553–574, 2021.
- 23 Aleks Kissinger, John van de Wetering, and Renaud Vilmart. Classical simulation of quantum circuits with partial and graphical stabiliser decompositions, 2022. doi:10.48550/ARXIV.2202.09202.
- 24 Stephen Lack. Composing props. *Theory and Applications of Categories*, 13(9):147–163, 2004.
- 25 Stephen Lack and Paweł Sobociński. Adhesive and quasiadhesive categories. *RAIRO - Theoretical Informatics and Applications - Informatique Théorique et Applications*, 39(3):511–545, 2005. doi:10.1051/ita:2005028.
- 26 Aleksandar Milosavljevic, Robin Piedeleu, and Fabio Zanasi. String diagram rewriting modulo commutative (co)monoid structure. *arXiv:2204.04274*, 2023.
- 27 Koko Muroya and Dan R. Ghica. The dynamic geometry of interaction machine: A token-guided graph rewriter. *Log. Methods Comput. Sci.*, 15(4), 2019.
- 28 Robin Piedeleu and Fabio Zanasi. An introduction to string diagrams for computer scientists. *arXiv:2305.08768*, 2023.
- 29 Mehrnoosh Sadrzadeh, Stephen Clark, and Bob Coecke. The Frobenius anatomy of word meanings I: subject and object relative pronouns. *J. Log. Comput.*, 23(6):1293–1317, 2013.
- 30 Fabio Zanasi. *Interacting Hopf Algebras- the Theory of Linear Systems. (Interacting Hopf Algebras - la théorie des systèmes linéaires)*. PhD thesis, École normale supérieure de Lyon, France, 2015. URL: <https://tel.archives-ouvertes.fr/tel-01218015>.
- 31 Fabio Zanasi. Rewriting in free hypergraph categories. In *GaM@ETAPS*, volume 263 of *EPTCS*, pages 16–30, 2017.



# Strongly Finitary Monads for Varieties of Quantitative Algebras

Jiří Adámek ✉

Department of Mathematics, Faculty of Electrical Engineering,  
Czech Technical University in Prague, Czech Republic  
Institute for Theoretical Computer Science, Technische Universität Braunschweig, Germany

Matěj Dostál ✉ 

Department of Mathematics, Faculty of Electrical Engineering,  
Czech Technical University in Prague, Czech Republic

Jiří Velebil ✉

Department of Mathematics, Faculty of Electrical Engineering,  
Czech Technical University in Prague, Czech Republic

---

## Abstract

Quantitative algebras are algebras enriched in the category  $\mathbf{Met}$  of metric spaces or  $\mathbf{UMet}$  of ultrametric spaces so that all operations are nonexpanding. Mardare, Plotkin and Panangaden introduced varieties (aka 1-basic varieties) as classes of quantitative algebras presented by quantitative equations. We prove that, when restricted to ultrametrics, varieties bijectively correspond to strongly finitary monads  $T$  on  $\mathbf{UMet}$ . This means that  $T$  is the left Kan extension of its restriction to finite discrete spaces. An analogous result holds in the category  $\mathbf{CUMet}$  of complete ultrametric spaces.

**2012 ACM Subject Classification** Theory of computation → Equational logic and rewriting

**Keywords and phrases** quantitative algebras, ultra-quantitative algebras, strongly finitary monads, varieties

**Digital Object Identifier** 10.4230/LIPIcs.CALCO.2023.10

**Related Version** *Full Version*: <https://arxiv.org/abs/2210.01565>

**Funding** *Jiří Adámek*: J. Adámek acknowledges the support by the Grant Agency of the Czech Republic under the grant 22-02964S.

*Matěj Dostál*: M. Dostál acknowledges the support by the Grant Agency of the Czech Republic under the grant 22-02964S.

## 1 Introduction

Quantitative algebraic reasoning was formalized in a series of articles of Bacci, Mardare, Panangaden and Plotkin [5, 15, 16, 6] as a tool for studying computational effects in probabilistic computation. Those papers work with algebras in the category  $\mathbf{Met}$  of metric spaces or  $\mathbf{CMet}$  of complete metric spaces. *Quantitative algebras* are algebras acting on a (complete) metric space  $A$  so that every  $n$ -ary operation is a nonexpanding map from  $A^n$ , with the maximum metric, to  $A$ . If the underlying metric is an ultrametric, we speak about *ultra-quantitative algebras*. Mardare et al. introduced quantitative equations, which are formal expressions  $t =_\varepsilon t'$  where  $t$  and  $t'$  are terms and  $\varepsilon \geq 0$  is a rational number. A quantitative algebra  $A$  satisfies this equation iff for every interpretation of the variables the elements of  $A$  corresponding to  $t$  and  $t'$  have distance at most  $\varepsilon$ . A *variety* (called 1-basic variety in [15]) is a class of quantitative algebras presented by a set of quantitative equations. Classical varieties of algebras are well known to correspond bijectively to *finitary monads*  $\mathbf{T}$  on  $\mathbf{Set}$  (preserving directed colimits): every variety is isomorphic to the category  $\mathbf{Set}^{\mathbf{T}}$  of



© Jiří Adámek, Matěj Dostál, and Jiří Velebil;  
licensed under Creative Commons License CC-BY 4.0

10th Conference on Algebra and Coalgebra in Computer Science (CALCO 2023).

Editors: Paolo Baldan and Valeria de Paiva; Article No. 10; pp. 10:1–10:14

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

algebras for  $\mathbf{T}$ , and vice versa. The question whether an analogous correspondence holds for quantitative algebras has been posed in [1] and [17]. For ultra-quantitative algebras we answer this by working with enriched (i.e. locally nonexpanding) monads on the category  $\mathbf{Met}$  of metric spaces, and its full subcategories  $\mathbf{UMet}$  of ultrametric spaces and  $\mathbf{CUMet}$  of complete ultrametric spaces. An enriched monad is *strongly finitary* if it is a left Kan extension of its restriction to finite discrete spaces. We characterize these monads as the enriched finitary monads preserving precongruences. Every strongly finitary monad on  $\mathbf{Met}$ ,  $\mathbf{UMet}$  or  $\mathbf{CUMet}$  is proved to be the free-algebra monad of a variety of quantitative algebras (Theorem 52 and Theorem 56).

For  $\mathbf{UMet}$  and  $\mathbf{CUMet}$  we also prove the converse: for every variety of ultra-quantitative algebras the free-algebra monad is strongly finitary (Theorem 47). We conclude that varieties bijectively correspond to strongly finitary monads on  $\mathbf{UMet}$  or  $\mathbf{CUMet}$ . It is an open problem whether this also holds for  $\mathbf{Met}$ .

## Related Work

A closely related result holds for partially ordered algebras (with nonexpanding operations). Here varieties are presented by inequations between terms. Kurz and Velebil [13] proved that they bijectively correspond to strongly finitary monads on the category  $\mathbf{Pos}$  of posets.

The main tool of Mardare et al. ([15, 16]) are  $\omega$ -basic equations: for a finite set of expressions  $x_i =_{\delta_i} y_i$  (where  $x_i, y_i$  are variables and  $\delta_i \geq 0$ ) and for terms  $t$  and  $t'$  one writes  $x_i =_{\delta_i} y_i \vdash t =_{\varepsilon} t'$ . An algebra  $A$  satisfies this equation if, for every interpretation  $f$  of the variables satisfying  $d(f(x_i), f(y_i)) \leq \delta_i$  for all  $i$ , the elements corresponding to  $t$  and  $t'$  have distance at most  $\varepsilon$ . A class of quantitative algebras presented by such equations is called an  $\omega$ -basic variety. Unfortunately, the free-algebra monad of an  $\omega$ -basic variety need not be finitary ([1], Example 4.1). Monads on  $\mathbf{UMet}$  corresponding to  $\omega$ -basic varieties were characterized in [1], Corollary 4.15.

Full proofs of the results presented in this extended abstract can be found in [3].

## 2 Strongly Finitary Functors

In this section we introduce strongly finitary functors, and present some of their properties. Later we prove a bijective correspondence of varieties and strongly finitary monads for  $\mathbf{UMet}$  and  $\mathbf{CUMet}$ .

► **Assumption 1.** Throughout our paper we work with categories and functors enriched over a symmetric monoidal closed category  $(\mathcal{V}, \otimes, I)$ . We recall these concepts shortly. Our leading examples of  $\mathcal{V}$  are metric spaces, ultrametric spaces and partially ordered sets.

► **Definition 2** ([8], 6.12). A symmetric monoidal closed category is given by a category  $\mathcal{V}$ , a bifunctor  $\otimes : \mathcal{V} \times \mathcal{V} \rightarrow \mathcal{V}$  and an object  $I$ . Moreover, natural isomorphisms are given expressing that  $\otimes$  is commutative and associative, and has the unit  $I$  (all up to coherent natural isomorphisms). Finally, for every object  $Y$  a right adjoint of the functor  $- \otimes Y : \mathcal{V} \rightarrow \mathcal{V}$  is given. We denote it by  $[Y, -]$  and denote the morphism corresponding to  $f : X \otimes Y \rightarrow Z$  by  $\widehat{f} : Y \rightarrow [X, Z]$ .

Often  $\otimes$  is the categorical product and  $I$  the terminal object; then  $\mathcal{V}$  is *cartesian closed*.

► **Example 3.**

(1)  $\mathcal{V} = \mathbf{Pos}$ , the category of posets, is cartesian closed,  $[X, Y]$  is the poset of all monotone maps  $f : X \rightarrow Y$  ordered pointwise. Here  $\widehat{f} = \text{curry} f$  is the curried form of  $f$ .

- (2)  $\mathcal{V} = \mathbf{Met}$ , the category of (*extended*) *metric spaces* and nonexpanding maps. Objects are metric spaces defined as usual, except that the distance  $\infty$  is allowed. Nonexpanding maps are those maps  $f : X \rightarrow Y$  with  $d(x, x') \geq d(f(x), f(x'))$  for all  $x, x' \in X$ . A product of metric spaces  $X \times Y$  is the metric space on the cartesian product with the *maximum metric*

$$d((x, y), (x', y')) = \max\{d(x, x'), d(y, y')\}.$$

This category is not cartesian closed: curryfication is not bijective. However,  $\mathbf{Met}$  is symmetric closed monoidal w.r.t. the *tensor product*  $X \otimes Y$  which is the cartesian product with the *addition metric*

$$d((x, y), (x', y')) = d(x, x') + d(y, y').$$

Here  $[X, Y]$  is the metric space  $\mathbf{Met}(X, Y)$  of all morphisms  $f : X \rightarrow Y$  with the *supremum metric*: the distance of  $f, g : X \rightarrow Y$  is

$$d(f, g) = \sup_{x \in X} d(f(x), g(x)).$$

And  $I$  is the singleton space.

- (3) The cartesian closed category  $\mathbf{UMet}$  of (extended) ultrametric spaces is the full subcategory of  $\mathbf{Met}$  on spaces satisfying the following stricter triangle inequality:

$$d(x, y) \leq \max\{d(x, z), d(z, y)\}.$$

Here the curryfication of morphisms  $f : X \times Y \rightarrow Z$  to  $\hat{f} : Y \rightarrow [X, Z]$  is bijective.

- (4) The category  $\mathbf{CMet}$  of complete metric spaces is the full subcategory of  $\mathbf{Met}$  on spaces with limits of all Cauchy sequences. It has the same symmetric closed monoidal structure as above: if  $X$  and  $Y$  are complete spaces, then so are  $X \otimes Y$  and  $[X, Y]$ . Analogously to (3) the category  $\mathbf{CUMet}$  of complete ultrametric spaces is cartesian closed.

► **Convention 4.** By a *category*  $\mathcal{C}$  we always mean a category enriched over  $\mathcal{V}$ . It is given by

- (1) a class  $\text{ob}\mathcal{C}$  of objects,
- (2) an object  $\mathcal{C}(X, Y)$  of  $\mathcal{V}$  (called the hom-object) for every pair  $X, Y$  in  $\text{ob}\mathcal{C}$ ,
- (3) a 'unit' morphism  $u_X : I \rightarrow \mathcal{C}(X, X)$  in  $\mathcal{V}$  for every object  $X \in \text{ob}\mathcal{C}$ , and
- (4) 'composition' morphisms

$$c_{X, Y, Z} : \mathcal{C}(X, Y) \otimes \mathcal{C}(Y, Z) \rightarrow \mathcal{C}(X, Z)$$

for all  $X, Y, Z \in \text{ob}\mathcal{C}$ , subject to commutative diagrams expressing the associativity of composition and the fact that  $u_X$  are units of composition. For details see [8], 6.2.1.

► **Example 5.**

- (1) If  $\mathcal{V} = \mathbf{Met}$  then  $\mathcal{C}$  is an ordinary category in which every hom-set  $\mathcal{C}(X, Y)$  carries a metric such that composition is nonexpanding. Analogously for  $\mathcal{V} = \mathbf{CMet}$  or  $\mathbf{UMet}$ .
- (2) If  $\mathcal{V} = \mathbf{Pos}$  then each hom-set  $\mathcal{C}(X, Y)$  carries a partial order such that composition is monotone.

Let us recall the concept of an *enriched functor*  $F : \mathcal{C} \rightarrow \mathcal{C}'$  for (enriched) categories  $\mathcal{C}$  and  $\mathcal{C}'$ . It assigns

- (1) an object  $FX \in \text{ob}\mathcal{C}'$  to every object  $X \in \text{ob}\mathcal{C}$ , and
- (2) a morphism  $F_{X, Y} : \mathcal{C}(X, Y) \rightarrow \mathcal{C}'(FX, FY)$  of  $\mathcal{V}$  to every pair  $X, Y \in \text{ob}\mathcal{C}$  so that the expected diagrams expressing that  $F$  preserves composition and identity morphisms commute.

## 10:4 Strongly Finitary Monads for Varieties of Quantitative Algebras

► **Convention 6.** By a *functor* we always mean an enriched functor. We use 'ordinary functor' in the few cases where a non-enriched functor is meant.

► **Example 7.**

- (1) For categories enriched over  $\mathbf{Met}$  a functor  $F : \mathcal{C} \rightarrow \mathcal{C}'$  is an ordinary functor which is *locally nonexpanding*: given  $f, g \in \mathcal{C}(X, Y)$  we have  $d(f, g) \geq d(Ff, Fg)$ . Analogously for  $\mathbf{CMet}$  or  $\mathbf{UMet}$ .
- (2) For categories enriched over  $\mathbf{Pos}$  functors  $F$  are the *locally monotone* ordinary functors: given  $f \leq g$  in  $\mathcal{C}(X, Y)$ , we get  $Ff \leq Fg$  in  $\mathcal{C}(FX, FY)$ .

► **Remark 8.**

- (1) In general one also needs the concept of an enriched natural transformation between parallel (enriched) functors. However, if  $\mathcal{V}$  is one of the categories of Example 3, this concept is just that of an ordinary natural transformation between the underlying ordinary functors.
- (2) Given two categories  $\mathcal{D}, \mathcal{C}$ , we denote by  $[\mathcal{D}, \mathcal{C}]$  the category of all functors  $F : \mathcal{D} \rightarrow \mathcal{C}$  enriched by assigning to every pair of functors  $F, G : \mathcal{D} \rightarrow \mathcal{C}$  an appropriate object  $[F, G]$  of  $\mathcal{V}$  of all natural transformations.  
In case  $\mathcal{V} = \mathbf{Met}, \mathbf{UMet}$  or  $\mathbf{CMet}$  the distance of  $\tau, \tau' : F \rightarrow G$  in  $[F, G]$  is  $\sup_{X \in \text{ob } \mathcal{D}} d(\tau_X, \tau'_X)$ .

► **Notation 9.**

- (1) Every set  $X$  is considered as a *discrete poset*:  $x \sqsubseteq x'$  iff  $x = x'$ . This is the coproduct  $\coprod_X I$  in  $\mathbf{Pos}$ . Analogously,  $X$  is considered as a *discrete metric space*: all distances of  $x \neq x'$  are  $\infty$ . This is the coproduct  $\coprod_X I$  in  $\mathbf{Met}$  (and also in  $\mathbf{UMet}$  and  $\mathbf{CMet}$ ).
- (2) For the category  $\mathbf{Set}_f$  of finite sets and mappings we define a functor

$$K : \mathbf{Set}_f \rightarrow \mathcal{V}, \quad X \mapsto \coprod_X I.$$

Thus for  $\mathcal{V} = \mathbf{Met}, \mathbf{CMet}, \mathbf{UMet}$  or  $\mathbf{Pos}$  it assigns to every finite set the corresponding discrete object.

- (3) Let us recall the concept of the (enriched) *left Kan extension* of a functor  $F : \mathcal{A} \rightarrow \mathcal{C}$  along a functor  $K : \mathcal{A} \rightarrow \mathcal{C}$  [11]: this is an endofunctor  $\text{Lan}_K F : \mathcal{C} \rightarrow \mathcal{C}$  endowed with a universal natural transformation  $\tau : F \rightarrow (\text{Lan}_K F) \cdot K$ . The universal property states that given a natural transformation  $\sigma : F \rightarrow G \cdot K$  for any endofunctor  $G : \mathcal{C} \rightarrow \mathcal{C}$ , there exists a unique natural transformation  $\bar{\sigma} : \text{Lan}_K F \rightarrow G$  with  $\sigma = \bar{\sigma} K \cdot \tau$ . The functor  $\text{Lan}_K F$  is unique up to a natural isomorphism.

► **Definition 10** (Kelly and Lack [12]). *An endofunctor  $F$  of  $\mathcal{V}$  is strongly finitary if it is a left Kan extension of its restriction  $F \cdot K$  to  $\mathbf{Set}_f$ . Shortly:  $F = \text{Lan}_K(F \cdot K)$ .*

► **Example 11.**

1. For every natural number  $n$  the endofunctor  $(-)^n$  of the  $n$ -th power is strongly finitary on  $\mathbf{Met}, \mathbf{UMet}$  and  $\mathbf{CMet}$ .
2. A coproduct of strongly finitary functors is strongly finitary.

► **Theorem 12** ([12]). *If  $\mathcal{V}$  is cartesian closed, then strongly finitary endofunctors are closed under composition.*

► **Open Problem 13.** Are all strongly finitary endofunctors on  $\mathbf{Met}$  closed under composition?



In order to characterize strong finitariness for endofunctors on  $\mathcal{V} = \text{Met}$ ,  $\text{UMet}$  and  $\text{CMet}$ , we apply Kelly's concept of density presentation that we now recall. For that we first shortly recall weighted colimits.

► **Definition 14** ([8, 11]).

- (1) A weighted diagram in a category  $\mathcal{C}$  is given by a functor  $D : \mathcal{D} \rightarrow \mathcal{C}$  together with a weight  $W : \mathcal{D}^{\text{op}} \rightarrow \mathcal{V}$ . A weighted colimit is an object  $C = \text{colim}_W D$  of  $\mathcal{C}$  together with isomorphisms in  $\mathcal{V}$ :

$$\psi_X : \mathcal{C}(C, X) \rightarrow [\mathcal{D}^{\text{op}}, \mathcal{C}](W, \mathcal{C}(D-, X))$$

natural in  $X \in \text{ob } \mathcal{C}$ .

- (2) The unit of this colimit is the natural transformation  $\nu = \psi_C(\text{id}_C) : W \rightarrow \mathcal{C}(D-, C)$ .  
 (3) A functor  $F : \mathcal{C} \rightarrow \mathcal{C}'$  preserves this colimit if  $\text{colim}_W (F \cdot D) = FC$  with the unit having components  $F\nu_d$  for  $d \in \mathcal{D}$ .

In all categories of Example 3 weighted colimits (for all  $\mathcal{D}$  small) exist.

► **Example 15.** (Conical) directed colimits are the special case where  $\mathcal{D}$  is a directed poset (every finite subset has an upper bound), and the weight  $W$  is trivial: the constant functor with value  $\mathbf{1}$  (the terminal object).

- (1) In  $\text{Pos}$  directed colimits are formed on the level of the underlying sets. They commute with finite products.  
 (2) Directed colimits in  $\text{Met}$ ,  $\text{UMet}$  and  $\text{CMet}$  also exist, but they are not formed on the level of the underlying sets. For example, consider the diagram of metric space  $A_n = \{0, 1\}$  with  $d_n(0, 1) = 2^{-n}$ , where the connecting maps are  $\text{id} : A_n \rightarrow A_{n+1}$  ( $n < \omega$ ). The colimit is a singleton space.

► **Lemma 16.** In  $\text{Met}$ ,  $\text{UMet}$  and  $\text{CUMet}$  every space is a directed colimit of all of its finite subspaces.

► **Theorem 17.** Directed colimits in  $\text{Met}$ ,  $\text{UMet}$  or  $\text{CMet}$  commute with finite products.

**Proof sketch.**

- (1) For a directed diagram  $(D_i)_{i \in I}$  in  $\text{Met}$ , cocones  $c_i : D_i \rightarrow C$  forming a colimit were characterized in [4], Lemma 2.4, by the following properties: (a)  $C = \bigcup_{i \in I} c_i[D_i]$ , and (b) for every  $i \in I$ , given  $y, y' \in D_i$  we have  $d(c_i(y), c_i(y')) = \inf_{j \geq i} d(f_j(y), f_j(y'))$ , where  $f_j : D_i \rightarrow D_j$  denotes the connecting map.  
 Given another directed diagram  $(D'_i)_{i \in I}$  with a cocone  $c'_i : D'_i \rightarrow C'$  satisfying (a) and (b), it is our task to prove that the cocone  $c_i \times c'_i : D_i \times D'_i \rightarrow C \times C'$  satisfies (a), (b), too. Since  $I$  is directed, (a) is clear, and (b) needs just a short computation.  
 (2) The argument for  $\text{UMet}$  is the same.  
 (3) For directed colimits in  $\text{CMet}$  the characterization of colimit cocones is analogous: (b) is unchanged, and in (a) one states that  $\bigcup_{i \in I} c_i[D_i]$  is dense in  $C$ . The proof is then analogous to (1). ◀

► **Definition 18.** A functor is finitary if it preserves directed colimits.

► **Example 19.**

- (1) An endofunctor of  $\text{Set}$  is strongly finitary iff it is finitary.  
 (2) An endofunctor of  $\text{Pos}$  is strongly finitary iff it is finitary and preserves reflexive coinserters, see [2].

## 10:6 Strongly Finitary Monads for Varieties of Quantitative Algebras

► **Notation 20.** Let  $K : \mathcal{A} \rightarrow \mathcal{C}$  be a functor. We denote by  $\tilde{K} : \mathcal{C} \rightarrow [\mathcal{A}^{op}, \mathcal{V}]$  the functor with  $\tilde{K}C = \mathcal{C}(K-, C)$ .

For example, the functor  $K : \mathbf{Set}_f \rightarrow \mathbf{Met}$  yields  $\tilde{K} : \mathbf{Met} \rightarrow [\mathbf{Set}_f^{op}, \mathbf{Met}]$  taking a metric space  $M$  to the functor  $M^{(-)} : \mathbf{Set}_f^{op} \rightarrow \mathbf{Met}$  of finite powers of  $M$ .

► **Definition 21** ([11]). A density presentation of a functor  $K : \mathcal{A} \rightarrow \mathcal{C}$  is a collection of weighted colimits in  $\mathcal{C}$  such that

- (a)  $\tilde{K}$  preserves those colimits, and
- (b)  $\mathcal{C}$  is the (iterated) closure of the image  $K[\mathcal{A}]$  under those colimits.

► **Example 22.** A density presentation of the functor  $K : \mathbf{Set}_f \rightarrow \mathbf{Met}$  (Notation 9) is given by all directed colimits and all *precongruences* (a name borrowed from [9]) which we now present. They express every metric space as a colimit of discrete spaces. (The weight used for precongruence is, however, not discrete.)

► **Notation 23.** For every metric space  $M$  let  $|M|$  denote its underlying set (a discrete metric space).

► **Definition 24.**

- (1) We define the basic weight  $W_0 : \mathcal{D}_0^{op} \rightarrow \mathbf{Met}$  as follows. The category  $\mathcal{D}_0$  consists of
  - a. the linearly ordered set of all rational numbers  $\varepsilon \geq 0$ ,
  - b. two parallel cocones of it  $\lambda_\varepsilon, \rho_\varepsilon : \varepsilon \rightarrow a$ , and
  - c. a morphism  $\sigma_\varepsilon : a \rightarrow \varepsilon$  splitting that pair:  $\lambda_\varepsilon \cdot \sigma_\varepsilon = id = \rho_\varepsilon \cdot \sigma_\varepsilon$  (for all  $\varepsilon$ ). The posets  $\mathcal{D}_0(\lambda_\varepsilon, \rho_\varepsilon)$  are all discrete.
 The values of  $W_0$  are  $W_0a = \{0\}$  and  $W_0\varepsilon = \{l, r\}$  with  $d(l, r) = \varepsilon$ . The morphisms  $W_0\lambda_\varepsilon, W_0\rho_\varepsilon : \{0\} \rightarrow \{l, r\}$  are given by  $0 \mapsto l, 0 \mapsto r$ , respectively, and  $W_0\sigma_\varepsilon$  is clear.
- (2) For every metric space  $M$  we define its precongruence as the weighted diagram  $D_M : \mathcal{D}_0 \rightarrow \mathbf{Met}$  with the basic weight  $W_0$ , where  $D_Ma = |M|$  and  $D_M\varepsilon \subseteq |M| \times |M|$  is the discrete space of all pairs of distance at most  $\varepsilon$ . Here  $D_M\lambda_\varepsilon, D_M\rho_\varepsilon : D_M\varepsilon \rightarrow |M|$  are the projections  $\pi_l$  and  $\pi_r$ , respectively, and  $D_M\sigma_\varepsilon : |M| \rightarrow D_M\varepsilon$  is the diagonal. The diagram  $D_M$  assigns to the morphism  $\varepsilon \leq \varepsilon'$  the inclusion map of the subset  $D_M\varepsilon$  of  $D_M\varepsilon'$ .

► **Proposition 25.** Every metric space  $M$  is the weighted colimit of its precongruence in  $\mathbf{Met}$ .

**Proof.** For every space  $X$ , to give a natural transformation  $\tau : W_0 \rightarrow [\mathcal{D}_0^{op}, \mathbf{Met}](D_M-, X)$  means to specify a map  $f = \tau_a(0) : |M| \rightarrow X$  together with maps  $\tau_\varepsilon(l), \tau_\varepsilon(r) : D_M\varepsilon \rightarrow X$  such that  $\tau_\varepsilon(l) = f \cdot \pi_l$  and  $\tau_\varepsilon(r) = f \cdot \pi_r$ . Thus  $\tau$  is determined by  $f$ , and the last equations are equivalent to  $f : M \rightarrow X$  being nonexpanding. The desired isomorphism  $\psi_X$  of Definition 14 is given by  $\psi_X(\tau) = f$ . ◀

► **Remark 26.** To define precongruences in  $\mathbf{UMet}$ , we just use the codomain restrictions  $W_0 : \mathcal{D}_0^{op} \rightarrow \mathbf{UMet}$  and  $D_M : \mathcal{D}_0 \rightarrow \mathbf{UMet}$ . Again, every ultrametric space is the weighted colimit of its precongruence in  $\mathbf{UMet}$ . Analogously for  $\mathbf{CUMet}$ .

► **Example 27.** The categories  $\mathbf{Met}$ ,  $\mathbf{UMet}$ ,  $\mathbf{CMet}$  and  $\mathbf{CUMet}$  have a density presentation of  $K$  (Notation 9) consisting of all directed diagrams and precongruences of finite spaces. Indeed, in Definition 21 Condition (a) follows from Example 15. For Condition (b) observe that finite metric spaces are obtained from  $\mathbf{Set}_f$  as colimits of precongruences by Proposition 25, and every metric space is a directed colimit of all of its finite subspaces in  $\mathbf{Met}$ . Analogously for the three subcategories of  $\mathbf{Met}$ .

The importance of the concept of density presentation for our paper stems from the following result of Kelly:

► **Theorem 28** ([11], Theorem 5.29). *Given a density presentation of a functor  $K : \mathcal{A} \rightarrow \mathcal{C}$ , an endofunctor  $T$  of  $\mathcal{C}$  fulfils  $T = \text{Lan}_K(T \cdot K)$  iff it preserves the colimits of that presentation.*

► **Corollary 29.** *An endofunctor of  $\text{Met}$ ,  $\text{UMet}$ ,  $\text{CMet}$  or  $\text{CUMet}$  is strongly finitary iff it preserves directed colimits and colimits of precongruences.*

This follows from Theorem 28 and the example above.

### 3 Varieties of Quantitative Algebras

We now prove that varieties of ultra-quantitative algebras bijectively correspond to *strongly finitary monads* on  $\text{UMet}$ . These are monads carried by a strongly finitary endofunctor. Throughout this section  $\Sigma = (\Sigma_n)_{n \in \mathbb{N}}$  denotes a signature, and  $V$  is a specified countable set of variables.

#### ► Notation 30.

- (1) Following Mardare, Panangaden and Plotkin [15], a *quantitative algebra* is a metric space  $A$  endowed with a nonexpanding operation  $\sigma_A : A^n \rightarrow A$  for every  $\sigma \in \Sigma_n$  (w.r.t. the maximum metric (Example 3)). We denote by  $\Sigma\text{-Met}$  the category of quantitative algebras and nonexpanding homomorphisms. Its forgetful functor is denoted by  $U_\Sigma : \Sigma\text{-Met} \rightarrow \text{Met}$ .
- (2) If  $A$  is an ultrametric space we speak about an *ultra-quantitative algebra* and denote the corresponding category by  $\Sigma\text{-UMet}$ .
- (3) Analogously, a *complete ultra-quantitative algebra* is an ultra-quantitative algebra carried by a complete metric space. The category  $\Sigma\text{-CUMet}$  is the corresponding full subcategory of  $\Sigma\text{-UMet}$ . We again use  $U_\Sigma : \Sigma\text{-CUMet} \rightarrow \text{CUMet}$  for the forgetful functor.

#### ► Example 31.

- (1) A free quantitative algebra on a metric space  $M$  is the usual algebra  $T_\Sigma M$  of *terms* on variables from  $|M|$ . That is, the smallest set containing  $|M|$  and such that for every  $n$ -ary symbol  $\sigma$  and every  $n$ -tuple of terms  $t_i$  ( $i < n$ ) we obtain a composite term  $\sigma(t_i)_{i < n}$ . To describe the metric, let us introduce the following equivalence  $\sim$  on  $T_\Sigma M$  (*similarity* of terms): it is the smallest equivalence turning all variables of  $|M|$  into one class, and such that  $\sigma(t_i)_{i < n} \sim \sigma'(t'_i)_{i < n'}$  holds iff  $\sigma = \sigma'$  and  $t_i \sim t'_i$  for all  $i < n$ . The metric of  $T_\Sigma M$  extends that of  $M$  as follows:  $d(t, t') = \infty$  if  $t$  is not similar to  $t'$ . For similar terms  $t = \sigma(t_i)$  and  $t' = \sigma(t'_i)$  we put  $d(t, t') = \max_{i < n} d(t_i, t'_i)$ .
- (2) If  $M$  is an ultrametric space, the space  $T_\Sigma M$  is clearly ultrametric, too. This is the free quantitative algebra in  $\Sigma\text{-UMet}$ .
- (3) If  $M$  is a complete space,  $T_\Sigma M$  is also complete, and this is the free quantitative algebra on  $M$  in  $\Sigma\text{-CUMet}$ .

In particular, if we consider the specified set  $V$  of variables as a discrete metric space, then  $T_\Sigma V$  is the discrete algebra of usual terms. For every algebra  $A$  and every interpretation of variables  $f : V \rightarrow A$  (in  $\text{Met}$ ,  $\text{UMet}$  or  $\text{CUMet}$ ) we denote by  $f^\# : T_\Sigma V \rightarrow A$  the corresponding homomorphism: it interprets terms in  $A$ .

## 10:8 Strongly Finitary Monads for Varieties of Quantitative Algebras

► **Definition 32** ([15]). *By a quantitative equation (aka 1-basic quantitative equation) is meant a formal expression  $t =_\varepsilon t'$  where  $t, t'$  are terms in  $T_\Sigma V$  and  $\varepsilon \geq 0$  is a rational number. An algebra  $A$  in  $\Sigma\text{-Met}$  ( $\Sigma\text{-UMet}$  or  $\Sigma\text{-CUMet}$ ) satisfies that equation if for every interpretation  $f : V \rightarrow A$  we have  $d(f^\sharp(t), f^\sharp(t')) \leq \varepsilon$ . We write  $t = t'$  in case  $\varepsilon = 0$ .*

*By a variety, aka 1-basic variety, of quantitative (or ultra-quantitative or complete ultra-quantitative) algebras is meant a full subcategory of  $\Sigma\text{-Met}$  (or  $\Sigma\text{-UMet}$  or  $\Sigma\text{-CUMet}$ , resp.) specified by a set of quantitative equations.*

► **Example 33.**

- (1) *Quantitative monoids* are given by the usual signature: a binary symbol  $\cdot$  and a constant  $e$ , and by the usual equations:  $(x \cdot y) \cdot z = x \cdot (y \cdot z)$ ,  $e \cdot x = x$ , and  $x \cdot e = x$ .
- (2) *Almost commutative monoids* are quantitative monoids in which the distance of  $ab$  and  $ba$  is always at most 1. They are presented by the quantitative equation  $x \cdot y =_1 y \cdot x$ .
- (3) *Quantitative semilattices* are commutative, idempotent quantitative monoids, see [15], Section 9.1.

► **Proposition 34** (See [15]). *Every variety  $\mathcal{V}$  of quantitative algebras has free algebras: the forgetful functor  $U_\mathcal{V} : \mathcal{V} \rightarrow \text{Met}$  has a left adjoint  $F_\mathcal{V} : \text{Met} \rightarrow \mathcal{V}$ .*

► **Notation 35.** We denote by  $\mathbf{T}_\mathcal{V}$  the free-algebra monad of a variety  $\mathcal{V}$  on  $\text{Met}$ . Its underlying functor is  $T_\mathcal{V} = U_\mathcal{V} \cdot F_\mathcal{V}$ . As usual,  $\text{Met}^{\mathbf{T}_\mathcal{V}}$  denotes the Eilenberg-Moore category of algebras for  $\mathbf{T}_\mathcal{V}$ .

► **Example 36.** For  $\mathcal{V} = \Sigma\text{-Met}$  we have seen the monad  $T_\Sigma$  in Example 31:  $T_\Sigma M$  is the metric space of all terms over  $M$ . Observe that  $T_\Sigma$  is a coproduct of endofunctors  $(-)^n$ , one summand for each similarity class of terms on  $n$  variables over  $M$  (which is independent of the choice  $M$ ). Thus  $\mathbf{T}_\Sigma$  is a strongly finitary monad: see Example 11.

► **Remark 37.**

- (1) Recall the comparison functor  $K_\mathcal{V} : \mathcal{V} \rightarrow \text{Met}^{\mathbf{T}_\mathcal{V}}$ : it assigns to every algebra  $A$  of  $\mathcal{V}$  the algebra on  $U_\mathcal{V}A$  for  $\mathbf{T}_\mathcal{V}$  given by the unique homomorphism  $\alpha : F_\mathcal{V}U_\mathcal{V}A \rightarrow A$  extending  $id_{U_\mathcal{V}A}$ . More precisely:  $K_\mathcal{V}A = (U_\mathcal{V}A, U_\mathcal{V}\alpha)$ .
- (2) By a *concrete category* over  $\text{Met}$  is meant a category  $\mathcal{V}$  together with a faithful 'forgetful' functor  $U_\mathcal{V} : \mathcal{V} \rightarrow \text{Met}$ . For example a variety, or  $\text{Met}^{\mathbf{T}}$  for every monad  $\mathbf{T}$ . A *concrete functor* is a functor  $F : \mathcal{V} \rightarrow \mathcal{W}$  with  $U_\mathcal{V} = U_\mathcal{W}F$ . For example, the comparison functor  $K_\mathcal{V}$ .

► **Proposition 38.** *Every variety  $\mathcal{V}$  of quantitative algebras is concretely isomorphic to the category  $\text{Met}^{\mathbf{T}_\mathcal{V}}$ : the comparison functor  $K_\mathcal{V} : \mathcal{V} \rightarrow \text{Met}^{\mathbf{T}_\mathcal{V}}$  is a concrete isomorphism. Analogously for  $\text{UMet}$  and  $\text{CUMet}$ .*

**Proof.** For classical varieties (over  $\text{Set}$ ) this is proved in [14], Theorem VI.8.1. The proof for  $\text{Met}$  in place of  $\text{Set}$  is analogous. ◀

► **Example 39** ([15], Theorem 9.3). For the variety  $\mathcal{V}$  of quantitative semilattices (Example 3.4 (3)) the monad  $\mathbf{T}_\mathcal{V}$  assigns to a metric space  $M$  the space of all finite subsets of  $M$  with the Hausdorff metric:

$$d(A, B) = \max\left\{\sup_{a \in A} d(a, B), \sup_{b \in B} d(b, A)\right\}.$$

Here,  $d(a, B) = \inf_{b \in B} d(a, b)$ . In particular,  $d(A, \emptyset) = \infty$  for all  $A \neq \emptyset$ .

► **Notation 40.**

- (1) Given a natural number  $n$  denote by  $[n]$  the signature of one  $n$ -ary symbol  $\delta$ . If a term  $t \in T_\Sigma V$  contains at most  $n$  variables (say, all variables of  $t$  are among  $x_0, \dots, x_{n-1}$ ), we obtain a monad morphism  $\bar{t} : \mathbf{T}_{[n]} \rightarrow \mathbf{T}_\Sigma$  as follows. For every space  $M$  the function  $\bar{t}_M$  takes a term  $s$  using the single symbol  $\delta$  and substitutes each occurrence of  $\delta$  by  $t(x_0, \dots, x_{n-1})$ . More precisely:  $\bar{t}_M : T_{[n]}M \rightarrow T_\Sigma M$  is defined by  $x_i \mapsto x_i$  ( $i < n$ ) and  $\delta(s_0, \dots, s_{n-1}) \mapsto t(\bar{t}_M(s_0), \dots, \bar{t}_M(s_{n-1}))$ .
- (2) Every metric space  $A$  defines the continuation monad  $\langle A, A \rangle$  on  $\mathbf{Met}$  assigning to  $X \in \mathbf{Met}$  the space  $\langle A, A \rangle X = [[X, A], A]$ . More precisely: the functor  $[-, A] : \mathbf{Met} \rightarrow \mathbf{Met}^{op}$  is self-adjoint, and  $\langle A, A \rangle$  is the monad corresponding to that adjunction.
- (3) Let  $\mathbf{T}$  be a monad on  $\mathbf{Met}$  and  $\alpha : TA \rightarrow A$  an algebra for it. We denote by  $\hat{\alpha}_X : TX \rightarrow \langle A, A \rangle X$  the morphism which is adjoint to the following composite

$$[X, A] \otimes TX \xrightarrow{T(-) \otimes TX} [TX, TA] \otimes TX \xrightarrow{ev} TA \xrightarrow{\alpha} A.$$

► **Theorem 41** ([10]). *Given an algebra  $\alpha : TA \rightarrow A$  for a monad  $\mathbf{T}$  on  $\mathbf{Met}$ ,  $\mathbf{UMet}$  or  $\mathbf{CUMet}$ , the morphisms  $\hat{\alpha}_X$  above form a monad morphism  $\hat{\alpha} : \mathbf{T} \rightarrow \langle A, A \rangle$ . Moreover, every monad morphism from  $\mathbf{T}$  to  $\langle A, A \rangle$  has that form for a unique algebra  $(A, \alpha)$ .*

► **Lemma 42.** *Let  $A$  be a  $\Sigma$ -algebra expressed as a monad algebra  $\alpha : T_\Sigma A \rightarrow A$ . It satisfies a quantitative equation  $l =_\varepsilon r$  iff the distance of  $\hat{\alpha} \cdot \bar{l}, \hat{\alpha} \cdot \bar{r} : \mathbf{T}_{[n]} \rightarrow \langle A, A \rangle$  is at most  $\varepsilon$ .*

► **Notation 43.**

1. The category of finitary monads on  $\mathbf{Met}$  (and monad morphisms) is denoted by  $\mathbf{Mnd}_f(\mathbf{Met})$ . It is enriched via the supremum metric: the distance of morphisms  $\sigma, \tau : \mathbf{T} \rightarrow \mathbf{T}'$  in  $\mathbf{Mnd}_f(\mathbf{Met})$  is  $\sup_{X \in \mathbf{Met}} d(\sigma_X, \tau_X)$ . We use the same enrichment for its full subcategory of strongly finitary monads, denoted by  $\mathbf{Mnd}_{sf}(\mathbf{Met})$ .
2. Analogously for monads on  $\mathbf{UMet}$  we use  $\mathbf{Mnd}_f(\mathbf{UMet})$  and  $\mathbf{Mnd}_{sf}(\mathbf{UMet})$ . Again for  $\mathbf{CUMet}$  we use  $\mathbf{Mnd}_f(\mathbf{CUMet})$  and  $\mathbf{Mnd}_{sf}(\mathbf{CUMet})$ .

► **Lemma 44.** *The category  $\mathbf{Mnd}_f(\mathbf{UMet})$  has weighted colimits, and  $\mathbf{Mnd}_{sf}(\mathbf{UMet})$  is closed under them.*

**Proof sketch.**

- (1) The category  $\mathbf{Mnd}_c(\mathbf{UMet})$  of countably accessible monads, i.e., monads preserving countably directed colimits (enriched again by the supremum metric), is locally countably presentable as an enriched category, thus it has weighted colimits.
- (2) Both  $\mathbf{Mnd}_f(\mathbf{UMet})$  and  $\mathbf{Mnd}_{sf}(\mathbf{UMet})$  are coreflective subcategories of  $\mathbf{Mnd}_c(\mathbf{UMet})$ . The coreflection of a countably accessible monad  $\mathbf{T}$  in  $\mathbf{Mnd}_{sf}(\mathbf{UMet})$  is given by the left Kan extension  $\tilde{T} = \mathbf{Lan}_K(T \cdot K)$ . Analogously for  $\mathbf{Mnd}_f(\mathbf{UMet})$ : let  $\bar{K} : \mathbf{UMet}_f \rightarrow \mathbf{UMet}$  be the full embedding of all finite metric spaces. The coreflection is  $\tilde{T} = \mathbf{Lan}_{\bar{K}}(T \cdot \bar{K})$ . ◀

► **Remark 45.**

1. The same result holds for the base category  $\mathbf{CUMet}$ .
2. Unfortunately, we do not know whether the above result holds for  $\mathbf{Met}$ . The problem is that for the coreflection of a monad  $\mathbf{T}$  in  $\mathbf{Mnd}_{sf}(\mathbf{Met})$  to be given by  $\tilde{T} = \mathbf{Lan}_K T \cdot K$ , we need to know that  $\tilde{T} \cdot \tilde{T}$  is strongly finitary. Whereas this holds in every cartesian closed category by [12], thus in  $\mathbf{UMet}$  and  $\mathbf{CUMet}$ , we do not know whether it also holds for monads on  $\mathbf{Met}$ .
3. The categories  $\mathbf{Met}$  and  $\mathbf{UMet}$  have a factorization system  $(\mathcal{E}, \mathcal{M})$  where  $\mathcal{E}$  consists of surjective morphisms and  $\mathcal{M}$  of isometric embeddings, i.e., morphisms preserving distances.

## 10:10 Strongly Finitary Monads for Varieties of Quantitative Algebras

► **Lemma 46.** *Every monad morphism  $\alpha : \mathbf{T}_\Sigma \rightarrow \mathbf{S}$  in the category  $\mathbf{Mnd}_f(\mathbf{UMet})$  factorizes as a morphism  $\mathbf{T}_\Sigma \rightarrow \overline{\mathbf{S}}$  with surjective components followed by a morphism  $\overline{\mathbf{S}} \rightarrow \mathbf{S}$  whose components are isometric embeddings.*

► **Theorem 47.** *For every variety  $\mathcal{V}$  of ultra-quantitative algebras the free-algebra monad  $\mathbf{T}_\mathcal{V}$  is strongly finitary on  $\mathbf{UMet}$ .*

**Proof sketch.**

(1) Let  $\mathcal{V}$  be given by a signature  $\Sigma$  and quantitative equations  $l_i =_{\varepsilon_i} r_i$  ( $i \in I$ ), each containing  $n_i$  variables. For every  $i \in I$  we consider the signature  $[n(i)]$  of one symbol  $\delta_i$  of arity  $n(i)$ . Then the terms  $l_i, r_i$  yield the corresponding monad morphisms  $\bar{l}_i, \bar{r}_i : \mathbf{T}_{[n(i)]} \rightarrow \mathbf{T}_\Sigma$  of Notation 40. An algebra  $\alpha : T_\Sigma A \rightarrow A$  lies in  $\mathcal{V}$  iff the distance of  $\hat{\alpha} \cdot \bar{l}_i, \hat{\alpha} \cdot \bar{r}_i : \mathbf{T}_{[n(i)]} \rightarrow \langle A, A \rangle$  is at most  $\varepsilon_i$  for each  $i$  (Lemma 42).

(2) We verify that  $\mathbf{T}_\mathcal{V}$  is a weighted colimit of strongly finitary monads in  $\mathbf{Mnd}_f(\mathbf{UMet})$ . Then  $\mathbf{T}_\mathcal{V}$  is strongly finitary by Lemma 44. The domain  $\mathcal{D}$  of the weighted diagram  $D : \mathcal{D} \rightarrow \mathbf{Mnd}_f(\mathbf{UMet})$  is the discrete category  $I$  (indexing the equations) enlarged by a new object  $a$ , and by morphisms  $\lambda_i, \rho_i : i \rightarrow a$  (for every  $i \in I$ ) of distance  $\varepsilon_i$ . Then put  $Di = \mathbf{T}_{[n(i)]}$  and  $Da = \mathbf{T}_\Sigma$ ; further  $D\lambda_i = \bar{l}_i$  and  $D\rho_i = \bar{r}_i$ . The weight  $W : \mathcal{D}^{op} \rightarrow \mathbf{Met}$  takes  $i$  to the space  $\{l, r\}$  with  $d(l, r) = \varepsilon_i$  and  $a$  to  $\{0\}$ . We define  $W\lambda_i(0) = l$  and  $W\rho_i(0) = r$ . The monads  $\mathbf{T}_\Sigma$  and  $\mathbf{T}_{[n(i)]}$  are strongly finitary by Example 36. Proving that  $\mathbf{T}_\mathcal{V} = \text{colim}_W D$  will finish the proof by Lemma 44.

We denote by  $\mathbf{T}$  the weighted colimit  $\mathbf{T} = \text{colim}_W D$  in  $\mathbf{Mnd}_f(\mathbf{UMet})$ . The proof is concluded by proving that  $\mathcal{V}$  is isomorphic, as a concrete category, to the category  $\mathbf{UMet}^{\mathbf{T}}$  of algebras for  $\mathbf{T}$ . Then  $\mathbf{T}$  is the free-algebra monad of  $\mathcal{V}$ . For  $\mathbf{T}$  we have the unit  $\nu : W \rightarrow [\mathcal{D}^{op}, \mathbf{Mnd}_f(\mathbf{UMet})](D-, \mathbf{T})$  (Definition 14). Its component  $\nu_a$  assigns to 0 a monad morphism  $\gamma = \nu_a(0) : \mathbf{T}_\Sigma \rightarrow \mathbf{T}$ , whereas for  $i \in I$  the component  $\nu_i$  is given by  $l \mapsto \gamma \cdot \bar{l}_i$  and  $r \mapsto \gamma \cdot \bar{r}_i$ . Since  $\nu_i$  is nonexpanding, we conclude that  $\gamma \cdot \bar{l}_i, \gamma \cdot \bar{r}_i : \mathbf{T}_{[n(i)]} \rightarrow \mathbf{T}$  have distance at most  $\varepsilon_i$ . We thus obtain a functor  $E : \mathbf{UMet}^{\mathbf{T}} \rightarrow \mathcal{V}$  assigning to every algebra  $\alpha : TA \rightarrow A$  the  $\Sigma$ -algebra corresponding to  $\alpha \cdot \gamma_A : T_\Sigma A \rightarrow A$ : it satisfies  $l_i =_{\varepsilon_i} r_i$  due to Lemma 42. Moreover,  $\gamma$  has surjective components, which can be derived from Lemma 46. Therefore,  $E$  is a concrete isomorphism, which concludes the proof. ◀

► **Remark 48.** The same result holds for varieties of quantitative algebras in  $\mathbf{CUMet}$ .

► **Open Problem 49.** Is the free-algebra monad of every variety of quantitative algebras strongly finitary on  $\mathbf{Met}$ ?

► **Construction 50.** In the reverse direction we assign to every strongly finitary monad  $\mathbf{T} = (T, \mu, \eta)$  on  $\mathbf{Met}$ ,  $\mathbf{UMet}$  or  $\mathbf{CUMet}$  a variety  $\mathcal{V}_\mathbf{T}$ , and prove that  $\mathbf{T}$  is its free-algebra monad.

For every morphism  $k : X \rightarrow A$  in  $\mathbf{Met}$  together with an algebra  $\alpha : TA \rightarrow A$ , let us denote by

$$k^* = \alpha \cdot Tk : TX \rightarrow A$$

the corresponding homomorphism in  $\mathbf{Met}^{\mathbf{T}}$ . Recall our fixed set  $V = \{x_i \mid i \in \mathbb{N}\}$  of variables, and form, for each  $n \in \mathbb{N}$ , the finite discrete space  $V_n = \{x_i \mid i < n\}$ . The signature we use has as  $n$ -ary symbols the elements of the space  $TV_n$ :

$$\Sigma_n = |TV_n| \text{ for } n \in \mathbb{N}.$$

The variety  $\mathcal{V}_\mathbf{T}$  is given by the following quantitative equations, where each symbol  $\sigma \in \Sigma_n$  is considered as the term  $\sigma(x_0, \dots, x_{n-1})$ , and  $n, m$  range over  $\mathbb{N}$ :

- (1)  $\sigma =_\varepsilon \sigma'$  for all  $\sigma, \sigma' \in \Sigma_n$  with  $d(\sigma, \sigma') \leq \varepsilon$  in  $TV_n$ .
- (2)  $k^*(\sigma) = \sigma(k(x_i))_{i < n}$  for all  $\sigma \in \Sigma_n$  and all maps  $k : V_n \rightarrow \Sigma_m$  in  $\text{Set}$ .
- (3)  $\eta_{V_n}(x_i) = x_i$  for all  $i < n$ .

► **Lemma 51.** *Every algebra  $\alpha : TA \rightarrow A$  in  $\text{Met}^{\mathbf{T}}$  yields an algebra  $A$  in  $\mathcal{V}_{\mathbf{T}}$  with operations  $\sigma_A : A^n \rightarrow A$  defined by*

$$\sigma_A(a(x_i)) = a^*(\sigma) \text{ for all } \sigma \in \Sigma_n \text{ and } a : V_n \rightarrow A.$$

Moreover, every homomorphism in  $\text{Met}^{\mathbf{T}}$  is also a  $\Sigma$ -homomorphism between the corresponding algebras in  $\mathcal{V}_{\mathbf{T}}$ .

**Proof sketch.**

- (a) The mapping  $\sigma_A$  is nonexpanding: given  $d((a_i)_{i < n}, (b_i)_{i < n}) = \varepsilon$  in  $A^n$ , the corresponding maps  $a, b : V_n \rightarrow A$  fulfil  $d(a, b) = \varepsilon$ . Since  $T$  is enriched, this yields  $d(Ta, Tb) \leq \varepsilon$ . Finally  $\alpha$  is nonexpanding and  $a^* = \alpha \cdot Ta$ ,  $b^* = \alpha \cdot Tb$ , thus  $d(a^*, b^*) \leq \varepsilon$ . In particular  $d(a^*(\sigma), b^*(\sigma)) \leq \varepsilon$ .
- (b) The quantitative equations (1)-(3) hold:
  - Ad (1)** Given  $l, r \in TV_n$  with  $d(l, r) \leq \varepsilon$ , then for every map  $a : V_n \rightarrow A$  we have  $d(a^*(l), a^*(r)) \leq \varepsilon$ . Thus  $d(l_A(a_i), r_A(a_i)) \leq \varepsilon$  for all  $(a_i) \in A^n$ .
  - Ad (2)** Given  $a : V_n \rightarrow A$  we prove  $(k^*(\sigma))_A(a_j) = \sigma_A(k(x_i))(a_j)$ . The left-hand side is  $a^*(k^*(\sigma)) = (a^*k)^*(\sigma)$  since  $a^* \cdot k^* = (a^* \cdot k)^*$  holds in general. The right-hand one is  $a^*(\sigma_A(k(x_i))) = (a^*k)^*(\sigma)$ , too.
  - Ad (3)** Recall that  $\alpha \cdot \eta_A = id$  and  $Ta \cdot \eta_{V_n} = \eta_A \cdot a$  for every map  $a : V_n \rightarrow A$ . Therefore

$$\begin{aligned} (\eta_{V_n}(x_i))_A(a_j) &= a^*(\eta_{V_n}(x_i)) \\ &= \alpha \cdot Ta \cdot \eta_{V_n}(x_i) \\ &= a(x_i) = a_i. \end{aligned}$$

- (c) Given a morphism  $h : (A, \alpha) \rightarrow (B, \beta)$  in  $\text{Met}^{\mathbf{T}}$  (i.e.,  $h \cdot \alpha = \beta \cdot Th$ ) we are to prove that  $h \cdot \sigma_A = \sigma_B \cdot h^n$  for all  $\sigma \in TV_n$ . This follows easily from  $h \cdot a^* = (h \cdot a)^*$  for each  $a : V_n \rightarrow A$ . ◀

► **Theorem 52.** *Every strongly finitary monad  $\mathbf{T}$  on  $\text{UMet}$  is the free-algebra monad of the variety  $\mathcal{V}_{\mathbf{T}}$ .*

**Proof.** For every ultrametric space  $M$  we need to prove that the  $\Sigma$ -algebra associated with  $(TM, \mu_M)$  in Lemma 51 is free in  $\mathcal{V}_{\mathbf{T}}$  w.r.t. the universal map  $\eta_M$ . Then the theorem follows from Proposition 38.

We have two strongly finitary monads,  $\mathbf{T}$  and the free-algebra monad of  $\mathcal{V}_{\mathbf{T}}$  (Theorem 47). Thus, it is sufficient to prove the above for finite discrete spaces  $M$ . Then this extends to all finite spaces because we have  $M = \text{colim}_{W_0} D_M$  (Lemma 25) and both monads preserve this colimit by Theorem 28. Since they coincide on all finite discrete spaces, they coincide on all finite spaces. Finally, the above extends to all spaces  $M$ : by Lemma 16 we have a directed colimit  $M = \text{colim}_{i \in I} M_i$  of the diagram of all finite subspaces  $M_i$  ( $i \in I$ ) which both monads preserve.

Given a finite discrete space  $M$ , we can assume without loss of generality  $M = V_n$  for some  $n \in \mathbb{N}$ . For every algebra  $A$  in  $\mathcal{V}_{\mathbf{T}}$  and an interpretation  $f : V_n \rightarrow A$ , we prove that there exists a unique  $\Sigma$ -homomorphism  $\bar{f} : TV_n \rightarrow A$  with  $f = \bar{f} \cdot \eta_{V_n}$ .

## 10:12 Strongly Finitary Monads for Varieties of Quantitative Algebras

**Existence.** Define  $\bar{f}(\sigma) = \sigma_A(f(x_i))_{i < n}$  for every  $\sigma \in TV_n$ . The equality  $f = \bar{f} \cdot \eta_{V_n}$  follows since  $A$  satisfies the equations (3) above:  $\eta_{V_n}(x_i) = x_i$ , thus the operation of  $A$  corresponding to  $\eta_{V_n}(x_i)$  is the  $i$ -th projection. The map  $\bar{f}$  is nonexpanding: given  $d(l, r) \leq \varepsilon$  in  $TV_n$ , the algebra  $A$  satisfies the equation (1) above:  $l =_\varepsilon r$ . Therefore given an  $n$ -tuple  $f : V_n \rightarrow A$  we have

$$d(l_A(f(x_i)), r_A(f(x_i))) \leq \varepsilon.$$

To prove that  $\bar{f}$  is a  $\Sigma$ -homomorphism, take an  $m$ -ary operation symbol  $\tau \in TV_m$ . We prove  $\bar{f} \cdot \tau_{V_m} = \tau_A \cdot \bar{f}^m$ . This means that every  $k : V_m \rightarrow TV_n$  fulfils

$$\bar{f} \cdot \tau_{V_m}(k(x_j))_{j < m} = \tau_A \cdot \bar{f}^m(k(x_j))_{j < m}.$$

The definition of  $\bar{f}$  yields that the right-hand side is  $\tau_A(k(x_j)_A(f(x_i)))$ . Due to equation (2) in Construction 50 with  $\tau$  in place of  $\sigma$ , this is  $k^*(\tau)_A(f(x_i))$ . The left-hand side yields the same result since

$$\tau_A \cdot \bar{f}^m(k(x_j)) = \tau_A(k(x_j))_A(f(x_i)) = k^*(\tau)_A(f(x_i)).$$

**Uniqueness.** Let  $\bar{f}$  be a nonexpanding  $\Sigma$ -homomorphism with  $f = \bar{f} \cdot \eta_{V_n}$ . In  $TV_n$  the operation  $\sigma$  assigns to  $\eta_{V_n}(x_i)$  the value  $\sigma$ . (Indeed, for every  $a : n \rightarrow |TV_n|$  we have  $\sigma_{TV_n}(a_i) = a^*(\sigma) = \mu_{V_n} \cdot Ta(\sigma)$ . Thus due to  $\mu \cdot T\eta = id$  we get  $\sigma_{TV_n}(\eta_{V_n}(x_i)) = \mu_{V_n} \cdot T\eta_{V_n}(\sigma) = \sigma$ .) Since  $\bar{f}$  is a homomorphism, we conclude

$$f(\sigma) = \sigma_A(\bar{f} \cdot \eta_{V_n}(x_i)) = \sigma_A(f(x_i))$$

which is the above formula. ◀

► **Corollary 53.** *Varieties of ultra-quantitative algebras correspond bijectively, up to isomorphism, to strongly finitary monads on  $\mathbf{UMet}$ .*

Indeed, a stronger result can be deduced from Theorems 47 and 52: let  $\mathbf{Var}(\mathbf{UMet})$  denote the category of varieties of quantitative algebras and concrete functors (Remark 37 (2)). Recall that  $\mathbf{Mnd}_{\text{sf}}(\mathbf{UMet})$  denotes the category of strongly finitary monads.

► **Theorem 54.** *The category  $\mathbf{Var}(\mathbf{UMet})$  of varieties of ultra-quantitative algebras is equivalent to the dual of the category  $\mathbf{Mnd}_{\text{sf}}(\mathbf{UMet})$  of strongly finitary monads on  $\mathbf{UMet}$ .*

**Proof.** Morphisms  $\varphi : \mathbf{S} \rightarrow \mathbf{T}$  between monads in  $\mathbf{Mnd}_{\text{sf}}(\mathbf{UMet})$  bijectively correspond to concrete functors  $\bar{\varphi} : \mathbf{UMet}^{\mathbf{T}} \rightarrow \mathbf{UMet}^{\mathbf{S}}$  ([7], Theorem 3.3):  $\bar{\varphi}$  assigns to an algebra  $\alpha : TA \rightarrow A$  of  $\mathbf{UMet}^{\mathbf{T}}$  the algebra  $\alpha \cdot \varphi_A : SA \rightarrow A$  in  $\mathbf{UMet}^{\mathbf{S}}$ . We know that for every variety  $\mathcal{V}$  the comparison functor  $K_{\mathcal{V}}$  is invertible (Proposition 38). This yields a functor  $\Phi : \mathbf{Var}(\mathbf{UMet})^{\text{op}} \rightarrow \mathbf{Mnd}_{\text{sf}}(\mathbf{UMet})$  assigning to a variety  $\mathcal{V}$  the monad  $\mathbf{T}_{\mathcal{V}}$  (Theorem 47). Given a concrete functor  $F : \mathcal{V} \rightarrow \mathcal{W}$  between varieties, there is a unique monad morphism  $\varphi : \mathbf{T}_{\mathcal{W}} \rightarrow \mathbf{T}_{\mathcal{V}}$  such that  $\bar{\varphi} = K_{\mathcal{W}} \cdot F \cdot K_{\mathcal{V}}^{-1} : \mathbf{UMet}^{\mathbf{T}_{\mathcal{V}}} \rightarrow \mathbf{UMet}^{\mathbf{T}_{\mathcal{W}}}$ . We define  $\Phi F = \varphi$  and get a functor which is clearly full and faithful. Thus Theorem 52 implies that  $\Phi$  is an equivalence of categories. ◀

## 4 Varieties of Complete Quantitative Algebras

If we take  $\mathbf{CUMet}$  as our base category, the development of Section 3 works for  $\Sigma$ - $\mathbf{CUMet}$  as well. The main difference is in Lemma 46: instead of the factorization system in  $\mathbf{UMet}$  of Remark 45, we use the factorization system in  $\mathbf{CUMet}$  where  $\mathcal{E} =$  dense morphisms  $f : A \rightarrow B$



( $f[A]$  is a dense subset of  $B$ ) and  $\mathcal{M}$  = isometric embeddings of closed subspaces. Another difference is that for the enrichment of the category  $\mathbf{Mnd}_f(\mathbf{CUMet})$  of finitary monads (cf. Notation 43) we must verify that the metric space of monad morphisms (with the supremum metric) is complete; this is easy.

By Example 31 (2) for every complete space  $M$  the space  $T_\Sigma M$  is complete. The resulting monad  $\mathbf{T}_\Sigma$  on the category  $\mathbf{CUMet}$  is strongly finitary (as in Example 36).

► **Example 55.** We describe the monad  $\mathbf{T}$  of free complete ultra-quantitative semilattices. It assigns to every complete ultrametric space  $M$  the space  $TM$  of all compact subsets with the Hausdorff metric (Example 39).

This holds for separable complete spaces: see [15], Theorem 9.6. To extend this result to all complete spaces, first observe that the subset  $Z$  of  $TM$  of all finite sets is dense. Indeed, every compact set  $K \subseteq M$  lies in the closure of  $Z$ : given  $\varepsilon > 0$ , let  $K_0 \subseteq K$  be a finite set such that  $\varepsilon$ -balls with centers in  $K_0$  cover  $K$ . Then  $K_0 \in Z$  and the Hausdorff distance of  $K_0$  and  $K$  is at most  $\varepsilon$ .

Given a complete ultrametric space  $M$ , let  $X_i$  ( $i \in I$ ) be the collection of all countable subsets. Each closure  $\overline{X}_i$  is a complete separable space, and  $M = \bigcup_{i \in I} \overline{X}_i$  is a directed colimit preserved by  $T$ . Since  $T\overline{X}_i$  is the space of all compact subsets of  $\overline{X}_i$ , and since finite subsets of  $M$  form a dense set, we conclude that  $TM$  is the space of all compact subsets of  $M$ .

Every variety  $\mathcal{V}$  of complete ultrametric quantitative algebras yields a monad  $\mathbf{T}_\mathcal{V}$  on  $\mathbf{CUMet}$  which is strongly finitary, and  $\mathcal{V}$  is isomorphic to  $\mathbf{UMet}^{\mathbf{T}_\mathcal{V}}$ . The proof is analogous to that of Theorem 47, just at the end we use the above factorization system of  $\mathbf{CUMet}$ . The proof that every strongly finitary monad on  $\mathbf{CUMet}$  is the free-algebra monad of a variety is completely analogous to that of Theorem 52. We thus obtain

► **Theorem 56.** *The category  $\mathbf{Var}(\mathbf{CUMet})$  of varieties of complete ultra-quantitative algebras is equivalent to the dual of the category  $\mathbf{Mnd}_{\mathbf{sf}}(\mathbf{CUMet})$  of strongly finitary monads on  $\mathbf{CUMet}$ .*

## 5 Conclusions and Open Problems

Varieties (aka 1-basic varieties) of quantitative algebras of Mardare et al. [15, 16], restricted to ultrametrics, correspond bijectively to strongly finitary monads on the category  $\mathbf{UMet}$ . This is the main result of our paper. It is in surprising contrast to the fact that  $\omega$ -varieties in op. cit. (where distance restrictions on finitely many variables in equations are imposed) do not even yield finitary monads in general, as demonstrated in [1].

For varieties in  $\mathbf{Met}$  we do not whether the same is true.

► **Open Problem 57.** Is the free-algebra monad of every variety of quantitative algebras strongly finitary?

Our proof would show this is the case provided that strongly finitary endofunctors on  $\mathbf{Met}$  are closed under composition.

For varieties of complete ultra-quantitative algebras the same result holds: they correspond bijectively to strongly finitary monads on  $\mathbf{CUMet}$ . This relates the quantitative algebraic reasoning of Mardare et al. closely to the classical equational reasoning of universal algebra where varieties are known to correspond to finitary (= strongly finitary) monads on  $\mathbf{Set}$  [14].

► **Open Problem 58.** Characterize monads on  $\mathbf{Met}$  or  $\mathbf{CMet}$  corresponding to  $\omega$ -varieties of quantitative algebras.

In [1] a partial answer has been given: enriched monads on  $\mathbf{UMet}$  corresponding to  $\omega$ -varieties of ultra-quantitative algebras are precisely the enriched monads preserving

- (1) directed colimits of split monomorphisms and
- (2) surjective morphisms.

---

### References

- 1 J. Adámek. Varieties of quantitative algebras and their monads. In *Proceedings of the 37th Annual Symposium on Logic in Computer Science (LICS 2022)*, pages 1–12, 2022.
- 2 J. Adámek, M. Dostál, and J. Velebil. A categorical view of varieties of ordered algebras. *Math. Struct. Comput. Sci.*, 32(4):349–373, 2022.
- 3 J. Adámek, M. Dostál, and J. Velebil. Quantitative algebras and a classification of metric monads, 2023. arXiv:2210.01565.
- 4 J. Adámek and J. Rosický. Approximate injectivity and smallness in metric-enriched categories. *J. Pure Appl. Algebra*, 226:1–30, 2022.
- 5 G. Bacci, R. Mardare, P. Panaganden, and G. D. Plotkin. An algebraic theory of Markov processes. In *Proceedings of Logic in Computer Science (LICS 2018)*, pages 679–688. ACM, 2018.
- 6 G. Bacci, R. Mardare, P. Panaganden, and G. D. Plotkin. Tensors of quantitative equational theories. In *Proceedings of Coalgebraic and Algebraic Methods in Computer Science (CALCO 2021)*, 2021.
- 7 M. Barr and Ch. Wells. *Toposes, triples and theories*. Springer-Verlag, New York, 1985.
- 8 F. Borceux. *Handbook of Categorical Algebra: Volume 2, Categories and Structures*. Cambridge Univ. Press, 1994.
- 9 J. Bourke and R. Garner. Monads and theories. *Adv. Math.*, 351:1024–1071, 2019.
- 10 E. J. Dubuc. *Kan Extensions in Enriched Category Theory*, volume 145 of *Lecture Notes in Mathematics*. Springer-Verlag, 1970.
- 11 G. M. Kelly. *Basic concepts of enriched category theory*. Number 64 in London Math. Soc. Lecture Notes Series. Cambridge Univ. Press, 1982. Also available as Repr. Theory Appl. Categ. 10 (2005).
- 12 G. M. Kelly and S. Lack. Finite-product-preserving functors, Kan extensions and strongly-finitary 2-monads. *Appl. Categ. Structures*, 1:85–94, 1993.
- 13 A. Kurz and J. Velebil. Quasivarieties and varieties of ordered algebras: regularity and exactness. *Math. Structures Comput. Sci.*, pages 1–42, 2016.
- 14 S. Mac Lane. *Categories for the working mathematician*. Springer, 2nd edition, 1998.
- 15 R. Mardare, P. Panangaden, and G. D. Plotkin. Quantitative algebraic reasoning. In *Proceedings of Logic in Computer Science (LICS 2016)*, pages 700–709. IEEE Computer Science, 2016.
- 16 R. Mardare, P. Panangaden, and G. D. Plotkin. On the axiomatizability of quantitative algebras. In *Proceedings of Logic in Computer Science (LICS 2017)*, pages 1–12. IEEE Computer Science, 2017.
- 17 M. Mio, R. Sarkis, and V. Vignudelli. Beyond nonexpansive operations in quantitative algebraic reasoning. In *Proceedings of Logic in Computer Science (LICS 2022)*, pages 1–13, 2022. Article 52.

# Generators and Bases for Monadic Closures

Stefan Zetsche<sup>1</sup> ✉ 🏠

Amazon Web Services, London, UK  
University College London, UK

Alexandra Silva ✉ 🏠

Cornell University, Ithaca, NY, USA  
University College London, UK

Matteo Sammartino ✉ 🏠

Royal Holloway, University of London, UK  
University College London, UK

---

## Abstract

It is well-known that every regular language admits a unique minimal deterministic acceptor. Establishing an analogous result for non-deterministic acceptors is significantly more difficult, but nonetheless of great practical importance. To tackle this issue, a number of sub-classes of non-deterministic automata have been identified, all admitting canonical minimal representatives. In previous work, we have shown that such representatives can be recovered categorically in two steps. First, one constructs the minimal bialgebra accepting a given regular language, by closing the minimal coalgebra with additional algebraic structure over a monad. Second, one identifies canonical generators for the algebraic part of the bialgebra, to derive an equivalent coalgebra with side effects in a monad. In this paper, we further develop the general theory underlying these two steps. On the one hand, we show that deriving a minimal bialgebra from a minimal coalgebra can be realized by applying a monad on an appropriate category of subobjects. On the other hand, we explore the abstract theory of generators and bases for algebras over a monad.

**2012 ACM Subject Classification** Theory of computation → Abstract machines

**Keywords and phrases** Monads, Category Theory, Generators, Automata, Coalgebras, Bialgebras

**Digital Object Identifier** 10.4230/LIPIcs.CALCO.2023.11

**Funding** *Stefan Zetsche*: Prior to their affiliation with Amazon Web Services, supported by GCHQ via the VeTSS grant *Automated Black-Box Verification of Networking Systems* (4207703/RFA 15845) and by the ERC via the Consolidator Grant *AutoProbe* (101002697).

*Alexandra Silva*: Supported by the ERC via the Consolidator Grant *AutoProbe* (101002697) and by a Royal Society Wolfson Fellowship.

*Matteo Sammartino*: Supported by the EPSRC Standard Grant *CLeVer* (EP/S028641/1).

## 1 Introduction

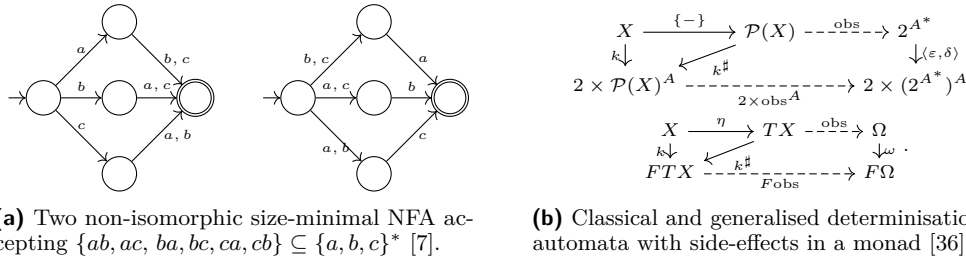
The existence of a unique minimal *deterministic* finite automaton is an important property of regular languages [30]. Establishing a similar result for *non-deterministic* finite automata is of great importance, as non-deterministic automata can be exponentially more succinct than deterministic ones, but turns out to be surprisingly difficult. The main problem is that a regular language can be accepted by several size-minimal NFAs that are not isomorphic. An example illustrating the situation is displayed in Figure 1a.

To tackle the issue, a number of sub-classes of non-deterministic automata admitting canonical representatives have been identified [15, 16, 44, 29]. One such example is the *canonical residual finite state automaton* (short *canonical RFSA*, also known as *jiromaton*),

---

<sup>1</sup> This paper is the result of work done prior to the author's affiliation with Amazon Web Services.





■ **Figure 1** Non-isomorphic NFAs and generalised determinisation.

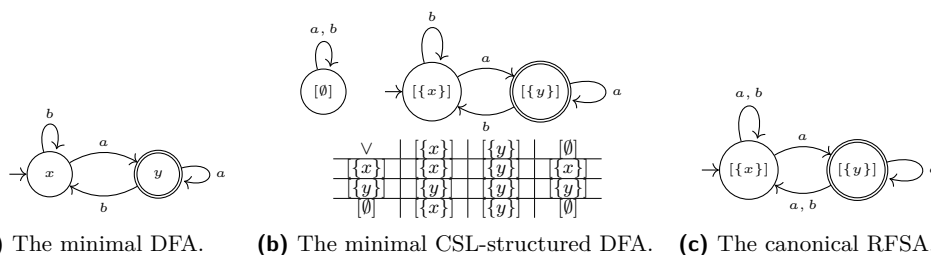
which is minimal among non-deterministic automata accepting joins of residual languages [16]. In previous work [46], we have presented a categorical framework that unifies constructions and correctness proofs of canonical non-deterministic automata and unveils new ones.

The framework adopts the well-known representation of automata as coalgebras [20, 34, 33] and side-effects like non-determinism as monads [26, 27, 28]. For instance, an NFA (without initial states) is represented as a coalgebra  $(X, k)$  with side-effects in the powerset monad  $(\mathcal{P}, \{-\}, \mu)$ , where  $X$  is the set of states,  $k: X \rightarrow 2 \times \mathcal{P}(X)^A$  combines the function classifying each state as accepting or rejecting with the function giving the set of next states for each input,  $\{-\}$  creates singleton sets, and  $\mu$  takes the union of a set of sets.

To derive canonical non-deterministic acceptors, the framework suggests a procedure that is closely related to the so-called *powerset construction*. As depicted at the top of Figure 1b, the latter converts a non-deterministic finite automaton  $(X, k)$  into an equivalent deterministic finite automaton  $(\mathcal{P}X, k^\sharp)$ , where  $k^\sharp$  is obtained by lifting  $k$  to the subsets of  $X$ , the tuple  $\langle \varepsilon, \delta \rangle$  is the automaton of languages, and the morphism  $\text{obs}$  assigns language semantics to each set of states. As seen at the bottom of Figure 1b, the construction can be generalised by replacing the functor  $2 \times (-)^A$  with any (suitable) functor  $F$  describing the automaton structure, and  $\mathcal{P}$  with a monad  $T$  describing the automaton side-effects, to transform a coalgebra  $k: X \rightarrow FTX$  with side-effects in  $T$  into an equivalent coalgebra  $k^\sharp: TX \rightarrow FTX$  [36]. Under this perspective,  $\Omega \xrightarrow{\omega} F\Omega$  is the so-called *final coalgebra*, providing a semantic universe that generalises the automaton of languages. The deterministic automata resulting from such determinisation constructions have *additional algebraic structure*: the state space  $\mathcal{P}(X)$  defines a free complete join-semilattice (CSL) over  $X$  and  $k^\sharp$  is a CSL homomorphism. More generally,  $TX$  defines a (free) algebra for the monad  $T$ , and  $k^\sharp$  is a  $T$ -algebra homomorphism, thus constituting a so-called *bialgebra* over a *distributive law* relating  $F$  and  $T$  [9, 38].

Using the powerset construction, a canonical succinct acceptor for a regular language  $L \subseteq A^*$  over an alphabet  $A$  can be obtained in two steps:

1. One constructs the minimal (pointed) coalgebra  $M_L$  for the functor  $F = 2 \times (-)^A$  accepting  $L$ . For the case  $A = \{a, b\}$  and  $L = (a + b)^*a$ , the coalgebra  $M_L$  is depicted in Figure 2a. Generally, it can be obtained via the Myhill-Nerode construction [30]. One then equips the former with additional algebraic structure in a monad  $T$  (which is related to  $F$  via a typically canonically<sup>3</sup> induced distributive law). This can be done by applying the generalised determinisation procedure to  $M_L$ , when seen as coalgebra with trivial side-effects in  $T$ . By identifying semantically equivalent states one consequently derives the minimal (pointed) bialgebra for  $L$ . If  $T = \mathcal{P}$  is the powerset monad, the minimal bialgebra for the language  $L = (a + b)^*a$  is depicted in Figure 2b.
2. One exploits the algebraic structure underlying the minimal bialgebra for  $L$  to “reverse” the generalised determinisation procedure. That is, one identifies a minimal set of *generators* that spans the full algebraic structure, to derive an equivalent succinct automaton with



■ **Figure 2** Three automata accepting the language  $(a + b)^*a \subseteq \{a, b\}^*$ .

side-effects in  $T$ . For example, by choosing the *join-irreducibles*<sup>2</sup> for the CSL underlying the minimal bialgebra in Figure 2b as generators (in this case, the join-irreducibles are given by all non-zero states), one recovers the canonical acceptor in Figure 2c.

In this paper, we further develop the general theory underlying these two steps. First, we generalise the closure of a subset of an algebraic structure as a functor between categories of subobjects relative to a factorisation system. We then equip the functor with the structure of a monad. We investigate the closure of a particular subclass of subobjects: the ones that arise from the image of a morphism. We show that deriving a minimal bialgebra from a minimal coalgebra can be realized by applying the monad to a subobject in this class. Second, we define a category of algebras with generators, which is in adjunction with the category of Eilenberg-Moore algebras, and, under certain assumptions, monoidal. We generalise the matrix representation theory of vector spaces and discuss bases for bialgebras. We compare our ideas with an approach that generalises bases as coalgebras [19]. We find that a basis in our sense induces a basis in the sense of [19], and identify assumptions under which the reverse is true, too. We characterise generators for finitary varieties in the sense of universal algebra and relate our work to the theory of locally finitely presentable categories.

## 2 Preliminaries

We assume basic knowledge of category theory (including functors, natural transformations, adjunctions), for an overview see e.g. [8].

We briefly recall the definitions of coalgebras, monads, and Eilenberg-Moore algebras. A *coalgebra* for an endofunctor  $F$  on a category  $\mathcal{C}$  is a tuple  $(X, k)$  consisting of an object  $X$  in  $\mathcal{C}$  and a morphism  $k: X \rightarrow FX$ . A homomorphism  $f: (X, k_X) \rightarrow (Y, k_Y)$  between coalgebras for  $F$  is a morphism  $f: X \rightarrow Y$  in  $\mathcal{C}$  satisfying  $k_Y \circ f = Ff \circ k_X$ . The category of coalgebras for  $F$  and homomorphisms is denoted by  $\text{Coalg}(F)$ .

A *monad* on a category  $\mathcal{C}$  is a tuple  $(T, \eta, \mu)$  consisting of an endofunctor  $T: \mathcal{C} \rightarrow \mathcal{C}$  and natural transformations  $\eta: \text{id}_{\mathcal{C}} \Rightarrow T$  and  $\mu: T^2 \Rightarrow T$  satisfying  $\mu \circ T\mu = \mu \circ \mu_T$  and  $\mu \circ \eta_T = \text{id}_T = \mu \circ T\eta$ . A morphism  $(F, \alpha): (\mathcal{C}, S) \rightarrow (\mathcal{D}, T)$  between a monad  $S$  on a category  $\mathcal{C}$  and a monad  $T$  on a category  $\mathcal{D}$  consists of a functor  $F: \mathcal{C} \rightarrow \mathcal{D}$  and a natural transformation  $\alpha: TF \Rightarrow FS$  satisfying  $\alpha \circ \eta^T F = F\eta^S$  and  $F\mu^S \circ \alpha S \circ T\alpha = \alpha \circ \mu^T F$  [37]. The composition of two monad morphisms  $(F, \alpha): (\mathcal{C}, S) \rightarrow (\mathcal{D}, T)$  and  $(G, \beta): (\mathcal{D}, T) \rightarrow (\mathcal{E}, U)$  is the monad morphism  $(GF, G\alpha \circ \beta F): (\mathcal{C}, S) \rightarrow (\mathcal{E}, U)$  [37]. Two well-known monads on the category of sets and functions are the *powerset monad*  $\mathcal{P}$  and the *free  $\mathbb{K}$ -vector space monad*  $\mathcal{V}_{\mathbb{K}}$  [46].

<sup>2</sup> A *join-irreducible* is a non-zero element  $a$  satisfying, for all  $y, z \in L$  with  $a = y \vee z$ , that  $a = y$  or  $a = z$ .

## 11:4 Generators and Bases for Monadic Closures

An *Eilenberg-Moore algebra* over a monad  $T$  on  $\mathcal{C}$  is a tuple  $(X, h)$  consisting of an object  $X$  in  $\mathcal{C}$  and a morphism  $h : TX \rightarrow X$  satisfying  $h \circ \mu_X = h \circ Th$  and  $h \circ \eta_X = \text{id}_X$ . A homomorphism  $f : (X, h_X) \rightarrow (Y, h_Y)$  between Eilenberg-Moore algebras over  $T$  is a morphism  $f : X \rightarrow Y$  in  $\mathcal{C}$  satisfying  $h_Y \circ Tf = f \circ h_X$ . The category of Eilenberg-Moore algebras over  $T$  is denoted by  $\mathcal{C}^T$ . One can show that the category of algebras over  $\mathcal{P}$  is isomorphic to the category of complete join-semi lattices, and the category of algebras over  $\mathcal{V}_{\mathbb{K}}$  is isomorphic to the category of  $\mathbb{K}$ -vector spaces.

We now introduce other notions that are necessary to follow our technical development: distributive laws, bialgebras, and generators and bases for algebras over a monad.

Distributive laws have originally occurred as a way to compose monads [9], but now also exist in a wide range of other forms [38]. For our case it is sufficient to consider distributive laws between a monad and an endofunctor, sometimes called *Eilenberg-Moore laws* [21].

► **Definition 1** (Distributive Law). *A distributive law between a monad  $T$  and an endofunctor  $F$  on  $\mathcal{C}$  is a natural transformation  $\lambda : TF \Rightarrow FT$  satisfying  $F\eta_X = \lambda_X \circ \eta_{FX}$  and  $\lambda_X \circ \mu_{FX} = F\mu_X \circ \lambda_{TX} \circ T\lambda_X$ .*

Given a distributive law, one can model the determinisation of a system with dynamics in  $F$  and side-effects in  $T$  by lifting a  $FT$ -coalgebra  $(X, k)$  to the  $F$ -coalgebra  $(TX, k^\sharp)$ , where  $k^\sharp := (F\mu_X \circ \lambda_{TX}) \circ Tk$ . As one verifies,  $k^\sharp$  is a  $T$ -algebra homomorphism of type  $(TX, \mu_X) \rightarrow (FTX, F\mu_X \circ \lambda_{TX})$ . There exists a distributive law for which the lifting  $k^\sharp$  is the DFA in CSL obtained from an NFA  $k$  via the classical powerset construction [36].

The example illustrates the concept of a bialgebra: the algebraic part  $(TX, \mu_X)$  and the coalgebraic part  $(TX, k^\sharp)$  of a lifted automaton are compatible along the distributive law  $\lambda$ .

► **Definition 2** (Bialgebra). *A  $\lambda$ -bialgebra is a tuple  $(X, h, k)$  consisting of a  $T$ -algebra  $(X, h)$  and an  $F$ -coalgebra  $(X, k)$  satisfying  $Fh \circ \lambda_X \circ Tk = k \circ h$ .*

A homomorphism between  $\lambda$ -bialgebras is a morphism between the underlying objects that is simultaneously a  $T$ -algebra homomorphism and an  $F$ -coalgebra homomorphism. The category of  $\lambda$ -bialgebras and homomorphisms is denoted by  $\text{Bialg}(\lambda)$ .

The generalised determinisation can be rephrased as a functor  $\text{exp}_T$  that *expands* a  $F$ -coalgebra with side-effects in  $T$  into a  $\lambda$ -bialgebra. We will also refer to the functor  $\text{free}_T$  that arises from  $\text{exp}_T$  by pre-composition with the canonical embedding of  $F$ -coalgebras into  $FT$ -coalgebras, therefore assigning to a  $F$ -coalgebra the  $\lambda$ -bialgebra it *freely* generates.

► **Lemma 3** ([21]).

- Defining  $\text{exp}_T(X, k) := (TX, \mu_X, F\mu_X \circ \lambda_{TX} \circ Tk)$  and  $\text{exp}_T(f) := Tf$  yields a functor  $\text{exp}_T : \text{Coalg}(F) \rightarrow \text{Bialg}(\lambda)$ .
- Defining  $\text{free}_T(X, k) := (TX, \mu_X, \lambda_X \circ Tk)$  and  $\text{free}_T(f) := Tf$  yields a functor  $\text{free}_T : \text{Coalg}(F) \rightarrow \text{Bialg}(\lambda)$  satisfying  $\text{free}_T(X, k) = \text{exp}_T(X, F\eta_X \circ k)$ .

The last ingredient is a generalisation of generators for structures such as vector spaces.

► **Definition 4** (Generator and Basis [46]). *A generator for a  $T$ -algebra  $(X, h)$  is a tuple  $(Y, i, d)$  consisting of an object  $Y$ , a morphism  $i : Y \rightarrow X$ , and a morphism  $d : X \rightarrow TY$  such that  $(h \circ Ti) \circ d = \text{id}_X$ . A generator is called a *basis* if it additionally satisfies  $d \circ (h \circ Ti) = \text{id}_{TY}$ .*

A generator for a  $T$ -algebra is called a *scoop* by Arbib and Manes [6]. Intuitively, a set  $Y$  embedded into an algebraic structure  $X$  via  $i$  is a generator for the latter if every element  $x \in X$  admits a decomposition into a *formal* combination  $d(x) \in TY$  of elements of  $Y$  that evaluates to  $x$  via the interpretation  $h \circ Ti$ . If the decomposition is moreover *unique*, that is,

$h \circ Ti$  is not only a *surjection* with right-inverse  $d$ , but a *bijection* with two-sided inverse  $d$ , then a generator is called a basis. Every  $T$ -algebra  $(X, h)$  is generated by  $(X, \text{id}_X, \eta_X)$  and admits a basis iff it is isomorphic to a free algebra.

► **Example 5.**

- A tuple  $(Y, i, d)$  is a generator for a  $\mathcal{P}$ -algebra  $L = (X, h) \simeq (X, \vee^h)$  iff  $x = \vee_{y \in d(x)}^h i(y)$  for all  $x \in X$ , where we write  $\vee^h$  for the complete join-semilattice structure induced by  $h$ . In the case that  $Y \subseteq X$  is a subset, one typically defines  $i(y) = y$  for all  $y \in Y$ . If  $L$  satisfies the descending chain condition, which is in particular the case if  $X$  is finite, then defining  $i(y) = y$  and  $d(x) = \{y \in J(L) \mid y \leq x\}$  turns the set of join-irreducibles  $J(L)$  into a size-minimal generator  $(J(L), i, d)$  for  $L$  [46].
- A tuple  $(Y, i, d)$  is a generator for a  $\mathcal{V}_{\mathbb{K}}$ -algebra  $V = (X, h) \simeq (X, +^h, \cdot^h)$  iff  $x = \sum_{y \in Y}^h d(x)(y) \cdot^h i(y)$  for all  $x \in X$ , where we write  $+^h$  and  $\cdot^h$  for the  $\mathbb{K}$ -vector space structure induced by  $h$ . As every vector space can be equipped with a basis, every  $\mathcal{V}_{\mathbb{K}}$ -algebra  $V$  admits a basis. One can show that a basis is a size-minimal generator [46].

A central result in [46] shows that it is enough to find generators for the underlying algebra of a bialgebra to derive an equivalent free bialgebra. This is because the algebraic and coalgebraic components are tightly intertwined via a distributive law.

► **Proposition 6** ([46]). *Let  $(X, h, k)$  be a  $\lambda$ -bialgebra and let  $(Y, i, d)$  be a generator for the  $T$ -algebra  $(X, h)$ . Then  $h \circ Ti: \exp_T(Y, Fd \circ k \circ i) \rightarrow (X, h, k)$  is a  $\lambda$ -bialgebra homomorphism.*

### 3 Step 1: Closure

In this section, we further explore the categorical construction of minimal canonical acceptors given in [46]. In particular, we show that deriving a minimal bialgebra from a minimal coalgebra by closing the latter with additional algebraic structure has a direct analogue in universal algebra: taking the closure of a subset of an algebra.

#### 3.1 Factorisation Systems and Subobjects

In the category of sets and functions, every morphism can be factored into a surjection onto its image followed by an injection into the codomain of the morphism. In this section we recall a convenient abstraction of this phenomenon for arbitrary categories. The ideas are well established [13, 32, 25]. We choose to adapt the formalism of [2].

► **Definition 7** (Factorisation System). *Let  $\mathcal{E}$  and  $\mathcal{M}$  be classes of morphisms in a category  $\mathcal{C}$ . We call the tuple  $(\mathcal{E}, \mathcal{M})$  a factorisation system for  $\mathcal{C}$  if the following three conditions hold:*

- (F1) *Each of  $\mathcal{E}$  and  $\mathcal{M}$  is closed under composition with isomorphisms.*
- (F2) *Each morphism  $f$  in  $\mathcal{C}$  can be factored as  $f = m \circ e$ , with  $e \in \mathcal{E}$  and  $m \in \mathcal{M}$ .*
- (F3) *Whenever  $g \circ e = m \circ f$  with  $e \in \mathcal{E}$  and  $m \in \mathcal{M}$ , there exists a unique diagonal  $d$ , such that  $f = d \circ e$  and  $g = m \circ d$ .*

We use double headed ( $\rightrightarrows$ ) and hooked ( $\hookrightarrow$ ) arrows to indicate that a morphism is in  $\mathcal{E}$  or  $\mathcal{M}$ , respectively. If  $f$  factors into  $e$  and  $m$ , we call the codomain of  $e$ , or equivalently, the domain of  $m$ , the *image* of  $f$  and denote it by  $\text{im}(f)$ .

One can show that each of  $\mathcal{E}$  and  $\mathcal{M}$  contains all isomorphisms and is closed under composition [2, Prop. 14.6]. From the uniqueness of the diagonal one can deduce that factorisations are unique up to unique isomorphism [2, Prop. 14.4]. It further follows that  $\mathcal{E}$  has the *right cancellation property*, that is  $g \circ f \in \mathcal{E}$  and  $f \in \mathcal{E}$  implies  $g \in \mathcal{E}$ . Dually,  $\mathcal{M}$  has the *left cancellation property*, that is,  $g \circ f \in \mathcal{M}$  and  $g \in \mathcal{M}$  implies  $f \in \mathcal{M}$  [2, Prop. 14.9].

$$\begin{array}{ccc}
 TX \xrightarrow{Tf} TY & X \xrightarrow{e} \text{im}(f) & TX \xrightarrow{Te} T\text{im}(f) \\
 h_X \downarrow & \searrow f & \swarrow e \circ h_X \downarrow \\
 X \xrightarrow{f} Y & \downarrow m & \text{im}(f) \xrightarrow{m} Y \\
 & & \downarrow h_Y \circ Tm \\
 & & \swarrow h_{\text{im}(f)} \dashrightarrow
 \end{array}$$

■ **Figure 3** Factorising a  $T$ -algebra homomorphism via the factorisation system of a base category.

As intended, in the category of sets and functions, surjective and injective functions, or equivalently, epi- and monomorphisms, constitute a factorisation system [2, Ex. 14.2]. More involved examples can be constructed for e.g. the category of topological spaces or the category of categories [2, Ex. 14.2]. We are particularly interested in factorisation systems for the categories of algebras over a monad and coalgebras over an endofunctor.

The naive categorification of a subset  $Y \subseteq X$  is a monomorphism  $Y \rightarrow X$ . Since in the category of sets epi- and monomorphism constitute a factorisation system, we may generalise subsets to arbitrary categories  $\mathcal{C}$  with a factorisation system  $(\mathcal{E}, \mathcal{M})$  in the following way:

► **Definition 8 (Subobjects).** A subobject of an object  $X \in \mathcal{C}$  is a morphism  $m_Y : Y \hookrightarrow X \in \mathcal{M}$ . A morphism  $f : m_{Y_1} \rightarrow m_{Y_2}$  between subobjects of  $X$  consists of a morphism  $f : Y_1 \rightarrow Y_2$  such that  $m_{Y_2} \circ f = m_{Y_1}$ .

The category of (isomorphism classes of) subobjects of  $X$  is denoted by  $\text{Sub}(X)$ .

As  $\mathcal{M}$  has the left cancellation property, every morphism between subobjects in fact lies in  $\mathcal{M}$ . We work with isomorphism classes of subobjects since factorisations of morphisms are only defined up to unique isomorphism. For epi-mono factorizations, there is at most one morphism between any two subobjects, that is,  $\text{Sub}(X)$  is simply a partially ordered set.

### 3.2 Factorising Algebra Homomorphisms

In this section, we recall that if one is given a category  $\mathcal{C}$  with a factorisation system  $(\mathcal{E}, \mathcal{M})$  and a monad  $T$  on  $\mathcal{C}$  that preserves  $\mathcal{E}$  (that is, satisfies  $T(e) \in \mathcal{E}$  for all  $e \in \mathcal{E}$ ), it is possible to lift the factorisation system of the base category  $\mathcal{C}$  to a factorisation system on the category of Eilenberg-Moore algebras  $\mathcal{C}^T$ .

The result appears in e.g. [45] and may be extended to algebras over an endofunctor. It can also be stated in its dual version: if an endofunctor on  $\mathcal{C}$  preserves  $\mathcal{M}$ , it is possible to lift the factorisation system of  $\mathcal{C}$  to the category of coalgebras [22, 45].

The induced factorisation system for  $\mathcal{C}^T$  consists of those algebra homomorphisms, whose underlying morphism lies in  $\mathcal{E}$  or  $\mathcal{M}$ , respectively. Clearly in such a system condition (F1) holds. The next result shows that it also satisfies (F3).

► **Lemma 9** ([45, Lem. 3.6]). *Whenever  $g \circ e = m \circ f$  for  $T$ -algebra homomorphisms  $f, g, e, m$ , with  $e \in \mathcal{E}$  and  $m \in \mathcal{M}$ , there exists a unique diagonal  $T$ -algebra homomorphism  $d$ , such that  $f = d \circ e$  and  $g = m \circ d$ .*

Let us now show that the proposed factorisation system satisfies (F2). Assume we are given a homomorphism  $f$  as on the left of Figure 3. Using the factorisation system of the base category  $\mathcal{C}$ , we can factorise it, as ordinary morphism, into  $e \in \mathcal{E}$  and  $m \in \mathcal{M}$ . In consequence the outer square of the diagram on the right of Figure 3 commutes. Since by assumption the morphism  $Te$  is again in  $\mathcal{E}$ , we thus find a unique diagonal  $h_{\text{im}(f)}$  in  $\mathcal{C}$  that makes the triangles on the right of Figure 3 commute. The result below shows that  $h_{\text{im}(f)}$  equips  $\text{im}(f)$  with the structure of a  $T$ -algebra.

► **Lemma 10** ([45, Prop. 3.7]).  *$(\text{im}(f), h_{\text{im}(f)})$  is an Eilenberg-Moore  $T$ -algebra.*



$$\begin{array}{ccc}
\mathcal{C}/X & \longrightarrow & \mathcal{C}^T/\mathbb{X} \\
\uparrow & & \downarrow \\
\text{Sub}(X) & \xrightarrow{(\cdot)^{\mathbb{X}}} & \text{Sub}(\mathbb{X})
\end{array}
\qquad
\begin{array}{ccc}
\mathcal{C}/X & \longrightarrow & \mathcal{C}^T/\mathbb{X} \\
\downarrow & & \downarrow \\
\text{Sub}(X) & \xrightarrow{(\cdot)^{\mathbb{X}}} & \text{Sub}(\mathbb{X})
\end{array}$$

(a) Decomposition. (b) Commutativity.

■ **Figure 4** A high-level perspective on the subobject closure functor defined in Proposition 11.

We thus obtain a factorisation of  $f : (X, h_X) \rightarrow (Y, h_Y)$  into Eilenberg-Moore  $T$ -algebra homomorphisms  $e : (X, h_X) \rightarrow (\text{im}(f), h_{\text{im}(f)})$  and  $m : (\text{im}(f), h_{\text{im}(f)}) \hookrightarrow (Y, h_Y)$ .

### 3.3 The Subobject Closure Functor

While subobjects in the category of sets generalise subsets, subobjects in the category of algebras generalise subalgebras. By taking the algebraic closure of a subset of an algebra one can thus transition from one category of subobjects to the other.

In this section, we generalise this phenomenon from the category of sets to more general categories. As before, we assume a base category  $\mathcal{C}$  with a factorisation system  $(\mathcal{E}, \mathcal{M})$  and a monad  $T$  on  $\mathcal{C}$  that preserves  $\mathcal{E}$ . Our aim is to construct, for any  $T$ -algebra  $\mathbb{X}$  with carrier  $X$ , a functor from the subobjects  $\text{Sub}(X)$  in  $\mathcal{C}$  to the subobjects  $\text{Sub}(\mathbb{X})$  in  $\mathcal{C}^T$  that assigns to a subobject of  $X$  its *closure*, that is, the least  $T$ -subalgebra of  $\mathbb{X}$  containing it.

Recall the free Eilenberg-Moore algebra adjunction. For any object  $Y$  in  $\mathcal{C}$  and  $T$ -algebra  $\mathbb{X} = (X, h)$ , it maps a morphism  $\varphi : Y \rightarrow X$  to the  $T$ -algebra homomorphism  $\varphi^\sharp := h \circ T\varphi : (TY, \mu_Y) \rightarrow \mathbb{X}$ . In Section 3.1 we have seen that the factorisation system of  $\mathcal{C}$  naturally lifts to a factorisation system on the category of  $T$ -algebras. In particular, we know that up to isomorphism the homomorphism  $\varphi^\sharp$  admits a factorisation into algebra homomorphisms of the form  $\varphi^\sharp = m_{\text{im}(\varphi^\sharp)} \circ e_{\text{im}(\varphi^\sharp)}$ . If the morphism  $\varphi$  is given by a subobject  $m_Y$ , let  $\bar{Y} := (\text{im}(m_Y^\sharp), h_{\text{im}(m_Y^\sharp)})$ , then above construction yields a second subobject  $m_{\bar{Y}}$ :

$$m_Y : Y \rightarrow X \in \mathcal{M} \qquad m_{\bar{Y}} : \bar{Y} \rightarrow \mathbb{X} \in \mathcal{M}.$$

Since for any morphism  $f : m_{Y_1} \rightarrow m_{Y_2}$  between subobjects of  $X$  one has  $m_{\bar{Y}_1} \circ e_{\bar{Y}_1} = m_{\bar{Y}_2} \circ (e_{\bar{Y}_2} \circ Tf)$ , there exists a unique homomorphism  $\bar{f} : m_{\bar{Y}_1} \rightarrow m_{\bar{Y}_2}$  satisfying  $\bar{f} \circ e_{\bar{Y}_1} = e_{\bar{Y}_2} \circ Tf$ .

The following result shows that above constructions are compositional.

► **Proposition 11.** *Assigning  $m_Y \mapsto m_{\bar{Y}}$  and  $f \mapsto \bar{f}$  yields a functor  $\overline{(\cdot)}^{\mathbb{X}} : \text{Sub}(X) \rightarrow \text{Sub}(\mathbb{X})$ .*

Mapping an algebra homomorphism with codomain  $\mathbb{X}$  to the  $\mathcal{M}$ -part of its factorisation extends to a functor from the slice category over  $\mathbb{X}$  (in which we here and in the following identify isomorphic objects) to the category of subobjects of  $\mathbb{X}$ . Similarly, one observes that the free Eilenberg-Moore algebra adjunction gives rise to a functor from the slice category over  $X$  to the slice category over  $\mathbb{X}$ . Finally, it is clear that there exists a functor from the category of subobjects of  $X$  to the slice category over  $X$ . The functor defined in Proposition 11 can thus be recognised as the composition in Figure 4a.

### 3.4 The Subobject Closure Monad

In this section, we show that the functor in Proposition 11 induces a monad on the category of subobjects  $\text{Sub}(X)$ . As before, we assume a base category  $\mathcal{C}$  with a factorisation system  $(\mathcal{E}, \mathcal{M})$  and a monad  $T = (T, \eta, \mu)$  on  $\mathcal{C}$  that preserves  $\mathcal{E}$ .

$$\begin{array}{ccc}
 \begin{array}{ccc}
 Y & \xrightarrow{1} & Y \\
 e_{\overline{Y}} \circ \eta_Y \downarrow & \swarrow \eta_{m_Y}^{\mathbb{X}} & \downarrow m_Y \\
 \overline{Y} & \xrightarrow{m_{\overline{Y}}} & X
 \end{array} &
 \begin{array}{ccc}
 T^2 Y & \xrightarrow{e_{\overline{Y}} \circ T e_{\overline{Y}}} & \overline{Y} \\
 e_{\overline{Y}} \circ \mu_Y \downarrow & \swarrow \mu_{m_Y}^{\mathbb{X}} & \downarrow m_{\overline{Y}} \\
 \overline{Y} & \xrightarrow{m_{\overline{Y}}} & X
 \end{array} &
 \begin{array}{ccc}
 (\text{Sub}(A), \overline{(\cdot)}^A) & \xrightarrow{(f_*, \alpha_f)} & (\text{Sub}(B), \overline{(\cdot)}^B) \\
 (U_A, \alpha_A) \searrow & & \swarrow (U_B, \alpha_B) \\
 & (\mathcal{C}, T) &
 \end{array}
 \end{array}$$

(a) Induced unit  $\eta^{\mathbb{X}}$  and multiplication  $\mu^{\mathbb{X}}$  of the monad in Theorem 13.

(b) Commutativity of the monad morphisms in Lemma 14 and Lemma 15.

■ **Figure 5** Structure and properties of the monad in Theorem 13.

We begin by establishing the following two technical identities, which assume a  $T$ -algebra  $\mathbb{X} = (X, h)$  and a subobject  $m_Y : Y \rightarrow X \in \mathcal{M}$ .

► **Lemma 12.**  $m_{\overline{Y}} \circ e_{\overline{Y}} \circ \eta_Y = m_Y$  and  $m_{\overline{Y}} \circ e_{\overline{Y}} \circ \mu_Y = m_{\overline{Y}} \circ e_{\overline{Y}} \circ T e_{\overline{Y}}$ .

In consequence, we can define candidates for the monad unit  $\eta^{\mathbb{X}}$  and the monad multiplication  $\mu^{\mathbb{X}}$ , respectively, as the unique diagonals in Figure 5a. By construction both morphisms are homomorphisms of subobjects:  $\eta_{m_Y}^{\mathbb{X}} : m_Y \rightarrow m_{\overline{Y}}$  and  $\mu_{m_Y}^{\mathbb{X}} : m_{\overline{Y}} \rightarrow m_{\overline{Y}}$ . The remaining proofs of naturality and the monad laws are covered below. By a slight abuse of notation, we write  $\overline{(\cdot)}^{\mathbb{X}}$  for the endofunctor on  $\text{Sub}(X)$  that arises by post-composition of the functor in Proposition 11 with the canonical forgetful functor from  $\text{Sub}(\mathbb{X})$  to  $\text{Sub}(X)$ .

► **Theorem 13.**  $(\overline{(\cdot)}^{\mathbb{X}}, \eta^{\mathbb{X}}, \mu^{\mathbb{X}})$  is a monad on  $\text{Sub}(X)$ .

We will now show that the mapping of an algebra  $\mathbb{X}$  to the monad  $\overline{(\cdot)}^{\mathbb{X}}$  in Theorem 13 extends to algebra homomorphisms. To this end, for any algebra homomorphism  $f : \mathbb{A} \rightarrow \mathbb{B}$  in  $\mathcal{M}$ , let  $f_* : \text{Sub}(A) \rightarrow \text{Sub}(B)$  be the induced functor defined by  $f_*(m_X) = f \circ m_X$  and  $f_*(g) = g$ . The result below shows that  $f_*$  can be extended to a morphism between monads.

► **Lemma 14.** For any  $f : \mathbb{A} \rightarrow \mathbb{B} \in \mathcal{M}$ , there exists a monad morphism  $(f_*, \alpha) : (\text{Sub}(A), \overline{(\cdot)}^A) \rightarrow (\text{Sub}(B), \overline{(\cdot)}^B)$ .

The next statement establishes that the canonical forgetful functor  $U : \text{Sub}(X) \rightarrow \mathcal{C}$  defined by  $U(m_Y) = Y$  and  $U(f) = f$  extends to a morphism between monads.

► **Lemma 15.** There exists a monad morphism  $(U, \alpha) : (\text{Sub}(X), \overline{(\cdot)}^{\mathbb{X}}) \rightarrow (\mathcal{C}, T)$ .

We conclude with the observation that the monad morphism defined in Lemma 14 commutes with the monad morphisms defined in Lemma 15.

► **Lemma 16.** Figure 5b commutes for any algebra homomorphism  $f : \mathbb{A} \rightarrow \mathbb{B} \in \mathcal{M}$ .

### 3.5 Closing an Image

In this section we investigate the closure of a particular class of subobjects: the ones that arise by taking the image of a morphism. We then show that deriving a minimal bialgebra from a minimal coalgebra by equipping the latter with additional algebraic structure can be realized as the closure of a subobject in this class.

As before, we assume a category  $\mathcal{C}$  with a factorisation system  $(\mathcal{E}, \mathcal{M})$  and a monad  $T$  on  $\mathcal{C}$  that preserves  $\mathcal{E}$ . Suppose that  $\mathbb{X} = (X, h_X)$  is a  $T$ -algebra and  $f : Y \rightarrow X$  a morphism in  $\mathcal{C}$ . On the one hand, there exists a factorisation of  $f$  in  $\mathcal{C}$ :

$$f = Y \xrightarrow{e_{\text{im}(f)}} \text{im}(f) \xrightarrow{m_{\text{im}(f)}} X.$$

On the other hand, there exists a factorisation of the lifting  $f^\sharp = h_X \circ Tf$  in the category of Eilenberg-Moore algebras  $\mathcal{C}^T$ :

$$f^\sharp = (TY, \mu_Y) \xrightarrow{e_{\text{im}(f^\sharp)}} (\text{im}(f^\sharp), h_{\text{im}(f^\sharp)}) \xrightarrow{m_{\text{im}(f^\sharp)}} (X, h_X).$$

The next result shows that, up to isomorphism, the closure of the subobject  $m_{\text{im}(f)}$  with respect to the algebra  $\mathbb{X}$  is given by the subobject  $m_{\text{im}(f^\sharp)}$ .

► **Lemma 17.**  $\overline{m_{\text{im}(f)}}^{\mathbb{X}} = m_{\text{im}(f^\sharp)}$  in  $\text{Sub}(\mathbb{X})$ .

The following example shows that closing a minimal Moore automaton with additional algebraic structure can be realised by applying a monad of the type in Theorem 13.

► **Example 18 (Closure of Minimal Moore Automata).** Let  $F$  be the set endofunctor with  $FX = B \times X^A$ , for fixed sets  $A$  and  $B$ . As  $F$  preserves monomorphisms, the canonical epi-mono factorisation system of the category of sets lifts to the category  $\text{Coalg}(F)$ , which consists of unpointed Moore automata with input  $A$  and output  $B$ .

For any language  $L : A^* \rightarrow B$ , there exists a size-minimal Moore automaton  $M_L$  that accepts  $L$ . It can be recovered as the epi-mono factorisation of the final  $F$ -coalgebra homomorphism  $\text{obs} : A^* \rightarrow \Omega$ , that is,  $M_L = m_{\text{im}(\text{obs})}$ . In more detail,  $\Omega$  is carried by  $B^{A^*}$ ,  $\text{obs}$  satisfies  $\text{obs}(w)(v) = L(wv)$ , and  $A^*$  is equipped with the  $F$ -coalgebra structure  $\langle \varepsilon, \delta \rangle : A^* \rightarrow B \times (A^*)^A$  defined by  $\varepsilon(w) = L(w)$  and  $\delta(w)(a) = wa$  [41].

Any algebra structure  $h : TB \rightarrow B$  over a set monad  $T$  induces a canonical<sup>3</sup> distributive law  $\lambda$  between  $T$  and  $F$  with  $FX = B \times X^A$ . It is well-known that  $\lambda$ -bialgebras are algebras over the monad  $T_\lambda$  on  $\text{Coalg}(F)$  defined by  $T_\lambda(X, k) = (TX, \lambda_X \circ Tk)$  and  $T_\lambda f = Tf$  [40]. One such  $T_\lambda$ -algebra is the final  $F$ -coalgebra  $\Omega$ , when equipped with a canonical  $T$ -algebra structure induced by finality [21, Prop. 3].

The functor  $T_\lambda$  preserves epimorphisms in the category  $\text{Coalg}(F)$ , if  $T$  preserves epimorphisms in the category of sets. The latter is the case for every set functor. By Theorem 13, there thus exists a well-defined monad  $\overline{(\cdot)}$  on  $\text{Sub}(\Omega)$ .

By construction, the minimal Moore automaton  $M_L$  lives in  $\text{Sub}(\Omega)$ . Reviewing the constructions in [46] shows that the minimal  $\lambda$ -bialgebra  $\mathbb{M}_L$  for  $L$  is given by the image of the lifting of  $\text{obs}$ , that is,  $\mathbb{M}_L = m_{\text{im}(\text{obs}^\sharp)}$ . From Lemma 17 it thus follows  $\mathbb{M}_L = \overline{M}_L$ . In other words, the minimal  $\lambda$ -bialgebra for  $L$  can be obtained from the minimal  $F$ -coalgebra for  $L$  by closing the latter with respect to the  $T_\lambda$ -algebra structure of  $\Omega$ .

For an example of the monad unit, observe how the minimal coalgebra in Figure 2a embeds into the minimal bialgebra in Figure 2b.

The situation can be further generalised. We assume that i)  $\mathcal{C}$  is a category with an  $(\mathcal{E}, \mathcal{M})$ -factorisation system; ii)  $\lambda$  is a distributive law between a monad  $T$  on  $\mathcal{C}$  that preserves  $\mathcal{E}$  and an endofunctor  $F$  on  $\mathcal{C}$  that preserves  $\mathcal{M}$ ; iii)  $(\Omega, h_\Omega, k_\Omega)$  is a final  $\lambda$ -bialgebra.

► **Theorem 19.** *There exists a functor  $\overline{(\cdot)} : \text{Sub}(\Omega, k_\Omega) \rightarrow \text{Sub}(\Omega, h_\Omega, k_\Omega)$  that yields a monad on  $\text{Sub}(\Omega, k_\Omega)$  and satisfies  $\overline{m_{\text{im}(\text{obs}_{(X,k)}})} \cong m_{\text{im}(\text{obs}_{\text{free}_T(X,k)}}$  in  $\text{Sub}(\Omega, h_\Omega, k_\Omega)$ , for any  $F$ -coalgebra  $(X, k)$ .*

<sup>3</sup> Given an algebra  $h : TB \rightarrow B$  for a set monad  $T$ , one can define a distributive law  $\lambda$  between  $T$  and  $F$  with  $FX = B \times X^A$  by  $\lambda_X := (h \times \text{st}) \circ \langle T\pi_1, T\pi_2 \rangle : TFX \rightarrow FTX$  [18]. (We write  $\text{st}$  for the usual strength function  $\text{st} : T(X^A) \rightarrow (TX)^A$  defined by  $\text{st}(U)(a) = T(\text{ev}_a)(U)$ , where  $\text{ev}_a(f) = f(a)$ .)

## 11:10 Generators and Bases for Monadic Closures

To recover Example 18 as a special case of Theorem 19, one instantiates the latter for the set endofunctor  $F$  with  $FX = B \times X^A$  and the canonical  $F$ -coalgebra with carrier  $A^*$ .

Finally, using analogous functors to the ones present in Figure 4a, we observe that, as a consequence of Lemma 17, the diagram in Figure 4b commutes.

### 4 Step 2: Generators and Bases

One of the central notions of linear algebra is the *basis*: a subset of a vector space is called basis, if every vector can be uniquely written as a linear combination of basis elements.

Part of the importance of bases stems from the convenient consequences that follow from their existence. For example, linear transformations between vector spaces admit matrix representations relative to pairs of bases [23], which can be used for efficient calculations. The idea of a basis however is not restricted to the theory of vector spaces: other algebraic theories have analogous notions of bases – and generators, by waiving the uniqueness constraint –, for instance modules, semi-lattices, Boolean algebras, convex sets, and many more. In fact, the theory of bases for vector spaces is special only in the sense that every vector space admits a basis, which is not the case for e.g. modules.

In this section, we use the abstraction of generators and bases given in Definition 4 to lift results from one theory to the others. For example, one may wonder if there exists a matrix representation theory for convex sets that is analogous to the one of vector spaces.

#### 4.1 Categorification

This section introduces a notion of morphism between algebras with a generator or a basis.

► **Definition 20.** *The category  $\text{GAlg}(T)$  of algebras with a generator over a monad  $T$  is defined as: objects are pairs  $(\mathbb{X}_\alpha, \alpha)$ , where  $\mathbb{X}_\alpha = (X_\alpha, h_\alpha)$  is a  $T$ -algebra with generator  $\alpha = (Y_\alpha, i_\alpha, d_\alpha)$ ; a morphism  $(f, p) : (\mathbb{X}_\alpha, \alpha) \rightarrow (\mathbb{X}_\beta, \beta)$  consists of a  $T$ -algebra homomorphism  $f : \mathbb{X}_\alpha \rightarrow \mathbb{X}_\beta$  and a Kleisli-morphism  $p : Y_\alpha \rightarrow TY_\beta$ , such that the diagram below commutes:*

$$\begin{array}{ccccc} X_\alpha & \xrightarrow{d_\alpha} & TY_\alpha & \xrightarrow{i_\alpha^\#} & X_\alpha \\ f \downarrow & & \downarrow p^\# & & \downarrow f \\ X_\beta & \xrightarrow{d_\beta} & TY_\beta & \xrightarrow{i_\beta^\#} & X_\beta \end{array} \quad (1)$$

Given  $(f, p) : (\mathbb{X}_\alpha, \alpha) \rightarrow (\mathbb{X}_\beta, \beta)$  and  $(g, q) : (\mathbb{X}_\beta, \beta) \rightarrow (\mathbb{X}_\gamma, \gamma)$ , their composition is defined componentwise as  $(g, q) \circ (f, p) := (g \circ f, q \cdot p)$ , where  $q \cdot p := \mu_{Y_\gamma} \circ Tq \circ p$  denotes the usual Kleisli-composition.

The category  $\text{BAlg}(T)$  of algebras with a basis is defined as full subcategory of  $\text{GAlg}(T)$ .

Let  $F : \mathcal{C}^T \rightarrow \text{GAlg}(T)$  be the functor with  $F(\mathbb{X}) := (\mathbb{X}, (X, \text{id}_X, \eta_X))$  and  $F(f : \mathbb{X} \rightarrow \mathbb{Y}) := (f, \eta_Y \circ f)$ , and  $U : \text{GAlg}(T) \rightarrow \mathcal{C}^T$  the forgetful functor defined as the projection on the first component. Then  $F$  and  $U$  are in an adjoint relation:

► **Lemma 21.**  $F \dashv U : \text{GAlg}(T) \rightleftarrows \mathcal{C}^T$ .

#### 4.2 Products

In this section we show that, under certain assumptions, the monoidal product of a category naturally extends to a monoidal product of algebras with bases within that category. As a natural example we obtain the tensor-product of vector spaces with fixed bases.

We assume basic familiarity with monoidal categories. A monoidal monad  $T$  on a monoidal category  $(\mathcal{C}, \otimes, I)$  is a monad which is equipped with natural transformations  $T_{X,Y} : TX \otimes TY \rightarrow T(X \otimes Y)$  and  $T_0 : I \rightarrow TI$ , satisfying certain coherence conditions (see e.g. [35]). One can show that, given such additional data, the monoidal structure of  $\mathcal{C}$  induces a monoidal category  $(\mathcal{C}^T, \boxtimes, (TI, \mu_I))$ , if two appropriately defined<sup>4</sup> assumptions (A1) and (A2) are satisfied [35, Cor. 2.5.6]. The two monoidal products  $\otimes$  and  $\boxtimes$  are related via the natural embedding  $q_{\mathbb{X}_\alpha, \mathbb{X}_\beta} \circ \eta_{X_\alpha \otimes X_\beta}$ , in the following referred to as  $\iota_{\mathbb{X}_\alpha, \mathbb{X}_\beta}$ . One can prove that the product  $TY_\alpha \boxtimes TY_\beta$  is given by  $T(Y_\alpha \otimes Y_\beta)$  and the coequaliser  $q_{TY_\alpha, TY_\beta}$  by  $\mu_{Y_\alpha \otimes Y_\beta} \circ T(T_{Y_\alpha, Y_\beta})$ , where we abbreviate the free algebra  $(TY, \mu_Y)$  as  $TY$  [35].

With the previous remarks in mind, we are able to claim the following.

► **Lemma 22.** *Let  $T$  be a monoidal monad on  $(\mathcal{C}, \otimes, I)$  satisfying (A1) and (A2). Let  $\alpha = (Y_\alpha, i_\alpha, d_\alpha)$  and  $\beta = (Y_\beta, i_\beta, d_\beta)$  be generators (bases) for  $T$ -algebras  $\mathbb{X}_\alpha$  and  $\mathbb{X}_\beta$ . Then  $\alpha \boxtimes \beta = (Y_\alpha \otimes Y_\beta, \iota_{\mathbb{X}_\alpha, \mathbb{X}_\beta} \circ (i_\alpha \otimes i_\beta), (d_\alpha \boxtimes d_\beta))$  is a generator (basis) for the  $T$ -algebra  $\mathbb{X}_\alpha \boxtimes \mathbb{X}_\beta$ .*

► **Corollary 23.** *Let  $T$  be a monoidal monad on  $(\mathcal{C}, \otimes, I)$  such that (A1) and (A2) are satisfied. The definitions  $(\mathbb{X}_\alpha, \alpha) \boxtimes (\mathbb{X}_\beta, \beta) := (\mathbb{X}_\alpha \boxtimes \mathbb{X}_\beta, \alpha \boxtimes \beta)$  and  $(f, p) \boxtimes (g, q) := (f \boxtimes g, T_{Y_{\alpha'}, Y_{\beta'}} \circ (p \boxtimes q))$  yield monoidal structures with unit  $((TI, \mu_I), (I, \eta_I, \text{id}_{TI}))$  on  $\text{GAlg}(T)$  and  $\text{BAlg}(T)$ .*

We conclude by instantiating above construction to the setting of vector spaces.

► **Example 24 (Tensor Product of Vector Spaces).** Recall the free  $\mathbb{K}$ -vector space monad  $\mathcal{V}_{\mathbb{K}}$  defined by  $\mathcal{V}_{\mathbb{K}}(X) = X \rightarrow \mathbb{K}$  and  $\mathcal{V}_{\mathbb{K}}(\varphi)(y) = \sum_{x \in f^{-1}(y)} \varphi(x)$ . Its unit is given by  $\eta_X(x)(y) = [x = y]$  and its multiplication by  $\mu_X(\Phi)(x) = \sum_{\varphi \in \mathbb{K}^X} \Phi(\varphi) \cdot \varphi(x)$ .

The category of sets is monoidal (in fact, cartesian) with respect to the cartesian product  $\times$  and the singleton set  $\{*\}$ . The monad  $\mathcal{V}_{\mathbb{K}}$  is monoidal when equipped with  $(\mathcal{V}_{\mathbb{K}})_{X,Y}(\varphi, \psi)(x, y) := \varphi(x) \cdot \psi(y)$  and  $(\mathcal{V}_{\mathbb{K}})_0(*)(*) := 1_{\mathbb{K}}$  [31]. The category of  $\mathcal{V}_{\mathbb{K}}$ -algebras is isomorphic to the category of  $\mathbb{K}$ -vector spaces, and satisfies (A1) and (A2). The monoidal structure induced by  $\mathcal{V}_{\mathbb{K}}$  is the usual tensor product  $\otimes$  with the unit field  $\mathcal{V}_{\mathbb{K}}(\{*\}) \simeq \mathbb{K}$ .

Lemma 22 captures the well-known fact that the dimension of the tensor product of two vector spaces is the product of the respective dimensions. The structure maps of the product generator map  $(y_\alpha, y_\beta)$  to the vector  $i(y_\alpha) \otimes i(y_\beta)$ , and  $x$  to  $(d_\alpha \otimes d_\beta)(x)$ , where

$$d_\alpha \otimes d_\beta = \overline{d_\alpha \times d_\beta} : \mathbb{X}_\alpha \otimes \mathbb{X}_\beta \rightarrow (\mathcal{V}_{\mathbb{K}}(Y_\alpha), \mu_{Y_\alpha}) \otimes (\mathcal{V}_{\mathbb{K}}(Y_\beta), \mu_{Y_\beta}) \simeq (\mathcal{V}_{\mathbb{K}}(Y_\alpha \times Y_\beta), \mu_{Y_\alpha \otimes Y_\beta})$$

is the unique linear extension of the bilinear map defined by

$$(d_\alpha \times d_\beta)(x_\alpha, x_\beta)(y_\alpha, y_\beta) := d_\alpha(x_\alpha)(y_\alpha) \cdot d_\beta(x_\beta)(y_\beta).$$

### 4.3 Kleisli Representation Theory

In this section we use our category-theoretical definition of a basis to derive a representation theory for homomorphisms between algebras over monads that is analogous to the matrix representation theory for linear transformations between vector spaces.

In more detail, recall that a linear transformation  $L : V \rightarrow W$  between  $k$ -vector spaces with finite bases  $\alpha = \{v_1, \dots, v_n\}$  and  $\beta = \{w_1, \dots, w_m\}$ , respectively, admits a matrix representation  $L_{\alpha\beta} \in \text{Mat}_k(m, n)$  with  $L(v_j) = \sum_i (L_{\alpha\beta})_{i,j} w_i$ , such that for any vector  $v$  in

<sup>4</sup> (A1) For any two algebras  $\mathbb{X}_\alpha = (X_\alpha, h_\alpha)$  and  $\mathbb{X}_\beta = (X_\beta, h_\beta)$  the coequaliser  $q_{\mathbb{X}_\alpha, \mathbb{X}_\beta}$  of the algebra homomorphisms  $T(h_\alpha \otimes h_\beta)$  and  $\mu_{X_\alpha \otimes X_\beta} \circ T(T_{X_\alpha, X_\beta})$  of type  $(T(TX_\alpha \otimes TX_\beta), \mu_{TX_\alpha \otimes TX_\beta}) \rightarrow (T(X_\alpha \otimes X_\beta), \mu_{X_\alpha \otimes X_\beta})$  exists (we denote its codomain by  $\mathbb{X}_\alpha \boxtimes \mathbb{X}_\beta := (X_\alpha \boxtimes X_\beta, h_{\alpha \boxtimes \beta})$ ). (A2) Left and right-tensoring with the induced functor  $\boxtimes$  preserves reflexive coequalisers.

## 11:12 Generators and Bases for Monadic Closures

$$A = L_{\alpha'\alpha'} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \quad L_{\alpha\alpha} = \begin{pmatrix} 3 & 2 \\ -5 & -3 \end{pmatrix}, \quad P = \begin{pmatrix} -1 & 1 \\ 2 & -1 \end{pmatrix}, \quad P^{-1} = \begin{pmatrix} 1 & 1 \\ 2 & 1 \end{pmatrix}$$

■ **Figure 6** The basis representation of the counter-clockwise rotation by 90 degree  $L : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ ,  $L(v) = Av$  with respect to  $\alpha = \{(1, 2), (1, 1)\}$  and  $\alpha' = \{(1, 0), (0, 1)\}$  satisfies  $L_{\alpha'\alpha'} = P^{-1}L_{\alpha\alpha}P$ .

$V$  the coordinate vectors  $L(v)_\beta \in k^m$  and  $v_\alpha \in k^n$  satisfy the equality  $L(v)_\beta = L_{\alpha\beta}v_\alpha$ . A great amount of linear algebra is concerned with finding bases such that the corresponding matrix representation is in an efficient shape, for instance diagonalised. The following definitions generalise the situation by substituting Kleisli morphisms for matrices.

► **Definition 25.** Let  $\alpha = (Y_\alpha, i_\alpha, d_\alpha)$  and  $\beta = (Y_\beta, i_\beta, d_\beta)$  be bases for  $T$ -algebras  $\mathbb{X}_\alpha = (X_\alpha, h_\alpha)$  and  $\mathbb{X}_\beta = (X_\beta, h_\beta)$ , respectively. The basis representation  $f_{\alpha\beta}$  of a  $T$ -algebra homomorphism  $f : \mathbb{X}_\alpha \rightarrow \mathbb{X}_\beta$  with respect to  $\alpha$  and  $\beta$  is defined by

$$f_{\alpha\beta} := Y_\alpha \xrightarrow{i_\alpha} X_\alpha \xrightarrow{f} X_\beta \xrightarrow{d_\beta} TY_\beta. \quad (2)$$

Conversely, the morphism  $p^{\alpha\beta}$  associated with a Kleisli morphism  $p : Y_\alpha \rightarrow TY_\beta$  with respect to  $\alpha$  and  $\beta$  is defined by

$$p^{\alpha\beta} := X_\alpha \xrightarrow{d_\alpha} TY_\alpha \xrightarrow{Tp} T^2Y_\beta \xrightarrow{\mu_{Y_\beta}} TY_\beta \xrightarrow{Ti_\beta} TX_\beta \xrightarrow{h_\beta} X_\beta. \quad (3)$$

The associated morphism is the linear transformation between vector spaces induced by some matrix of the right type. The following result confirms this intuition.

► **Lemma 26.** The function (3) is a  $T$ -algebra homomorphism  $p^{\alpha\beta} : \mathbb{X}_\alpha \rightarrow \mathbb{X}_\beta$ .

The next result establishes a generalisation of the observation that for fixed bases, constructing a matrix representation of a linear transformation and associating a linear transformation to a matrix of the right type are mutually inverse operations.

► **Lemma 27.** The operations (2) and (3) are mutually inverse.

At the beginning of this section we recalled the soundness identity  $L(v)_\beta = L_{\alpha\beta}v_\alpha$  for the matrix representation  $L_{\alpha\beta}$  of a linear transformation  $L$ . The next result is a natural generalisation of this statement.

► **Lemma 28.**  $f_{\alpha\beta}$  is the unique Kleisli-morphism such that  $f_{\alpha\beta} \cdot d_\alpha = d_\beta \circ f$ . Conversely,  $p^{\alpha\beta}$  is the unique  $T$ -algebra homomorphism such that  $p \cdot d_\alpha = d_\beta \circ p^{\alpha\beta}$ .

The next result establishes the compositionality of the operations (2) and (3). For example, the matrix representation of the composition of two linear transformations is given by the multiplication of the matrix representations of the individual linear transformations.

► **Lemma 29.**  $g_{\beta\gamma} \cdot f_{\alpha\beta} = (g \circ f)_{\alpha\gamma}$  and  $q^{\beta\gamma} \circ p^{\alpha\beta} = (q \cdot p)^{\alpha\gamma}$ .

The previous statements may be summarised as functors between appropriately defined<sup>5</sup> categories  $\text{Alg}_B(T)$  and  $\text{Kl}_B(T)$ .

<sup>5</sup> Let  $\text{Alg}_B(T)$  be the category in which objects are given by pairs  $(\mathbb{X}_\alpha, \alpha)$ , where  $\mathbb{X}_\alpha$  is a  $T$ -algebra with basis  $\alpha = (Y_\alpha, i_\alpha, d_\alpha)$ , and a morphism  $f : (\mathbb{X}_\alpha, \alpha) \rightarrow (\mathbb{X}_\beta, \beta)$  consists of a  $T$ -algebra homomorphism  $f : \mathbb{X}_\alpha \rightarrow \mathbb{X}_\beta$ . Let  $\text{Kl}_B(T)$  be the category in which objects are the same ones as for  $\text{Alg}_B(T)$ , and a morphism  $p : (\mathbb{X}_\alpha, \alpha) \rightarrow (\mathbb{X}_\beta, \beta)$  consists of a Kleisli-morphism  $p : Y_\alpha \rightarrow TY_\beta$ .

► **Corollary 30.** *There exist isomorphisms of categories  $\text{BAlg}(T) \simeq \text{Alg}_B(T) \simeq \text{Kl}_B(T)$ .*

Assume we are given bases  $\alpha, \alpha'$  and  $\beta, \beta'$  for  $T$ -algebras  $(X_\alpha, h_\alpha)$  and  $(X_\beta, h_\beta)$ , respectively. The following result clarifies how the representations  $f_{\alpha\beta}$  and  $f_{\alpha'\beta'}$  are related.

► **Proposition 31.** *There exist Kleisli isomorphisms  $p$  and  $q$  such that  $f_{\alpha'\beta'} = q \cdot f_{\alpha\beta} \cdot p$ .*

Above result simplifies if one restricts to an endomorphism: the basis representations are *similar*. This generalises the situation for vector spaces, cf. Figure 6.

► **Proposition 32.** *There exists a Kleisli isomorphism  $p$  with Kleisli inverse  $p^{-1}$  such that  $f_{\alpha'\alpha'} = p^{-1} \cdot f_{\alpha\alpha} \cdot p$ .*

#### 4.4 Bases for Bialgebras

This section is concerned with generators and bases for *bialgebras*. It is well-known [40] that an Eilenberg-Moore law  $\lambda$  between a monad  $T$  and an endofunctor  $F$  induces simultaneously i) a monad  $T_\lambda = (T_\lambda, \mu, \eta)$  on  $\text{Coalg}(F)$  by  $T_\lambda(X, k) = (TX, \lambda_X \circ Tk)$  and  $T_\lambda f = Tf$ ; and ii) an endofunctor  $F_\lambda$  on  $\mathcal{C}^T$  by  $F_\lambda(X, h) = (FX, Fh \circ \lambda_X)$  and  $F_\lambda f = Ff$ , such that the algebras over  $T_\lambda$ , the coalgebras of  $F_\lambda$ , and  $\lambda$ -bialgebras coincide. We will consider generators and bases for  $T_\lambda$ -algebras, or equivalently,  $\lambda$ -bialgebras.

By Definition 4, a generator for a  $\lambda$ -bialgebra  $(X, h, k)$  consists of a  $F$ -coalgebra  $(Y, k_Y)$  and morphisms  $i : Y \rightarrow X$ ,  $d : X \rightarrow TY$ , such that the three squares on the left of (4) commute:

$$\begin{array}{ccccc}
 Y & \xrightarrow{i} & X & \xrightarrow{d} & TY & \xrightarrow{Ti} & TX & \xrightarrow{h} & X \\
 k_Y \downarrow & & k \downarrow & & \downarrow \lambda_Y \circ Tk_Y & & d \uparrow & & \downarrow h \\
 FY & \xrightarrow{Fi} & FX & \xrightarrow{Fd} & FTY & & X & \xrightarrow{\text{id}_X} & X \\
 & & & & & & & & TY & \xrightarrow{\text{id}_{TY}} & TY
 \end{array} \quad (4)$$

A basis for a bialgebra is a generator such that the diagram on the right of (4) commutes.

By forgetting the  $F$ -coalgebra structure, every generator for a bialgebra is in particular a generator for its underlying  $T$ -algebra. By Proposition 6 there exists a  $\lambda$ -bialgebra homomorphism  $i^\# := h \circ Ti : \text{exp}_T(Y, Fd \circ k \circ i) \rightarrow (X, h, k)$ . The next result establishes that there exists a second equivalent free bialgebra with a different coalgebra structure.

► **Lemma 33.** *Let  $(Y, k_Y, i, d)$  be a generator for  $(X, h, k)$ . Then  $i^\# : TY \rightarrow X$  is a  $\lambda$ -bialgebra homomorphism  $i^\# : \text{free}_T(Y, k_Y) \rightarrow (X, h, k)$ .*

If one moves from generators to bases for bialgebras, both structures coincide.

► **Lemma 34.** *Let  $(Y, k_Y, i, d)$  be a basis for  $(X, h, k)$ , then  $\text{free}_T(Y, k_Y) = \text{exp}_T(Y, Fd \circ k \circ i)$ .*

► **Example 35 (Canonical RFSA).** Recall the minimal pointed bialgebra  $(X, h, k)$  for the language  $L = (a+b)^*a$  depicted in Figure 2b. Let  $(J(\mathbb{X}), i, d)$  be the generator for  $\mathbb{X} = (X, h)$  defined as follows: the carrier  $J(\mathbb{X})$  consists of the join-irreducibles for  $\mathbb{X}$ , the embedding satisfies  $i(y) = y$ , and the decomposition is given by  $d(x) = \{y \in J(\mathbb{X}) \mid y \leq x\}$ . We used  $(J(\mathbb{X}), i, d)$  to recover the canonical RFSA for  $L$  depicted in Figure 2c as the coalgebra  $(J(\mathbb{X}), Fd \circ k \circ i)$ . Examining the graphs shows that  $k$  restricts to the join-irreducibles  $J(\mathbb{X})$ , suggesting  $\alpha = (J(\mathbb{X}), k, i, d)$  as a possible generator for the full bialgebra. However, the  $a$ -action on  $[\{y\}]$  implies the non-commutativity of the second diagram on the left of (4). The issue can be fixed by modifying  $d$  via  $d([\{y\}]) := \{[\{y\}]\}$ . In consequence  $\text{free}(J(\mathbb{X}), k)$  and  $\text{exp}(J(\mathbb{X}), Fd \circ k \circ i)$  coincide (even though the assumptions of Lemma 34 are not satisfied).

## 11:14 Generators and Bases for Monadic Closures

We close this section by observing that a basis for the underlying algebra of a bialgebra is sufficient for constructing a generator for the full bialgebra.

► **Lemma 36.** *Let  $(X, h, k)$  be a  $\lambda$ -bialgebra and  $(Y, i, d)$  a basis for the  $T$ -algebra  $(X, h)$ . Then  $(TY, F\mu_Y \circ \lambda_{TY} \circ T(Fd \circ k \circ i), h \circ Ti, \eta_{TY} \circ d)$  is a generator for  $(X, h, k)$ .*

### 4.5 Bases as Coalgebras

In this section, we compare our approach to an alternative perspective on the generalisation of bases. More specifically, we are interested in the work of Jacobs [19], where a basis is defined as a coalgebra for the comonad on the category of Eilenberg-Moore algebras induced by the free algebra adjunction. Explicitly, a basis for a  $T$ -algebra  $(X, h)$ , in the sense of [19], consists of a  $T$ -coalgebra  $(X, k)$  such that the following three diagrams commute:

$$\begin{array}{ccc} TX \xrightarrow{Tk} T^2X & X \xrightarrow{k} TX & X \xrightarrow{k} TX \\ h \downarrow & \text{id}_X \searrow & k \downarrow \\ X \xrightarrow{k} TX & & X \xrightarrow{k} TX \\ & & \downarrow T\eta_X \\ & & TX \xrightarrow{Tk} T^2X \end{array} \quad (5)$$

The next result shows that a basis as in Definition 4 induces a basis in the sense of [19].

► **Lemma 37.** *Let  $(Y, i, d)$  be a basis for a  $T$ -algebra  $(X, h)$ , then (5) commutes for  $k := Ti \circ d$ .*

Conversely, assume  $(X, k)$  is a  $T$ -coalgebra structure satisfying (5) and  $i_k : Y_k \rightarrow X$  an equaliser of  $k$  and  $\eta_X$ . If the underlying category is the usual category of sets, the equaliser of any two functions exists. If  $Y_k$  non-empty, one can show that the equaliser is preserved under  $T$ , that is,  $Ti_k$  is an equaliser of  $Tk$  and  $T\eta_X$  [19]. By (5) we have  $Tk \circ k = T\eta_X \circ k$ . Thus there exists a unique morphism  $d_k : X \rightarrow TY_k$  such that  $Ti_k \circ d_k = k$ , which can be shown to be the inverse of  $h \circ Ti_k$  [19]. In other words,  $G(X, k) := (Y_k, i_k, d_k)$  is a basis for  $(X, h)$  in the sense of Definition 4. In the following let  $F(Y, i, d) := (X, Ti \circ d)$  for an arbitrary basis of  $(X, h)$ .

► **Lemma 38.** *Let  $(Y, i, d)$  be a basis for a  $T$ -algebra  $(X, h)$  and  $k := Ti \circ d$ . Then  $\eta_X \circ i = k \circ i$  and  $Tk \circ (\eta_X \circ i) = T\eta_X \circ (\eta_X \circ i)$ .*

► **Corollary 39.** *Let  $\alpha := (Y, i, d)$  be a basis for a set-based  $T$ -algebra  $(X, h)$  and  $k := Ti \circ d$ . Let  $i_k : Y_k \rightarrow X$  be an equaliser of  $k$  and  $\eta_X$ , and  $Y_k$  non-empty, then  $(\text{id}_{(X, h)})_{\alpha, GF\alpha} : Y \rightarrow TY_k$  is the unique morphism  $\psi$  making the diagram below commute:*

$$\begin{array}{ccccc} & \eta_X \circ i & & & \\ & \curvearrowright & & & \\ Y & \xrightarrow{\psi} & TY_k & \xrightarrow{Ti_k} & TX \\ & & & & \downarrow T\eta_X \\ & & & & T^2X \end{array} \quad \begin{array}{c} \xrightarrow{Tk} \\ \xrightarrow{T\eta_X} \end{array}$$

### 4.6 Signatures, Equations, and Finitary Monads

Most of the algebras over set monads one usually considers generators for constitute finitary varieties in the sense of universal algebra. In this section, we will briefly explore the consequences for generators that arise from this observation. The constructions are well-known; we include them for completeness.

Let  $\Sigma$  be a set, whose elements we think of as *operations*, and  $\text{ar} : \Sigma \rightarrow \mathbb{N}$  a function that assigns to an operation its *arity*. Any such *signature* induces a set endofunctor  $H_\Sigma$  defined on a set as the coproduct  $H_\Sigma X = \coprod_{\sigma \in \Sigma} X^{\text{ar}(\sigma)}$ , and consequently, a set monad  $\mathbb{S}_\Sigma$  that assigns to a set  $V$  of variables the initial algebra  $S_\Sigma V = \mu X.(V + H_\Sigma X)$ , i.e. the set of



$\Sigma$ -terms generated by  $V$  (see e.g. [39]). One can show that the categories of  $H_\Sigma$ -algebras and  $\mathbb{S}_\Sigma$ -algebras are isomorphic. A  $\mathbb{S}_\Sigma$ -algebra  $\mathbb{X}$  satisfies a set of equations  $E \subseteq S_\Sigma V \times S_\Sigma V$ , if for all  $(s, t) \in E$  and valuations  $v : V \rightarrow X$  it holds  $v^\sharp(s) = v^\sharp(t)$ , where  $v^\sharp : (S_\Sigma V, \mu_V) \rightarrow \mathbb{X}$  is the unique extension of  $v$  to a  $\mathbb{S}_\Sigma$ -algebra homomorphism [4]. The set of  $\mathbb{S}_\Sigma$ -algebras that satisfy  $E$  is denoted by  $\text{Alg}(\Sigma, E)$ . As one verifies, the forgetful functor  $U : \text{Alg}(\Sigma, E) \rightarrow \text{Set}$  admits a left-adjoint  $F : \text{Set} \rightarrow \text{Alg}(\Sigma, E)$ , thus resulting in a set monad  $T_{\Sigma, E}$  with underlying endofunctor  $U \circ F$  that preserves directed colimits. The functor  $U$  can be shown to be monadic, that is, the comparison functor  $K : \text{Alg}(\Sigma, E) \rightarrow \text{Set}^{T_{\Sigma, E}}$  is an isomorphism [24]. In other words, the category of Eilenberg-Moore algebras over  $T_{\Sigma, E}$  and the finitary variety of algebras over  $\Sigma$  and  $E$  coincide. In fact, set monads preserving directed colimits (sometimes called *finitary* monads [4]) and finitary varieties are in *bijection*.

The following result characterises generators for algebras over  $T_{\Sigma, E}$ . It can be seen as a unifying proof for observations analogous to the one in Example 5.

► **Lemma 40.** *A morphism  $i : Y \rightarrow X$  is part of a generator for a  $T_{\Sigma, E}$ -algebra  $\mathbb{X}$  iff every element  $x \in X$  can be expressed as a  $\Sigma$ -term in  $i[Y]$  modulo  $E$ , that is, there is a term  $d(x) \in S_\Sigma Y$  such that  $i^\sharp(\llbracket d(x) \rrbracket_E) = x$ .*

## 4.7 Finitely Generated Objects

In this section, we relate our abstract definition of a generator to the theory of *locally finitely presentable* categories, in particular, to the notions of *finitely generated* and *finitely presentable* objects, which are categorical abstractions of finitely generated algebraic structures.

For intuition, recall that an element  $x \in X$  of a partially ordered set is *compact*, if for each directed set  $D \subseteq X$  with  $x \leq \bigvee D$ , there exists some  $d \in D$  satisfying  $x \leq d$ . An *algebraic lattice* is a partially ordered set that has all joins, and every element is a join of compact elements. The naive categorification of compact elements is equivalent to the following definition: a object  $Y$  in  $\mathcal{C}$  is *finitely presentable (generated)*, if  $\text{Hom}_{\mathcal{C}}(Y, -) : \mathcal{C} \rightarrow \text{Set}$  preserves filtered colimits (of monomorphisms). Consequently, one can categorify algebraic lattices as *locally finitely presentable (lfp)* categories, which are cocomplete and admit a set of finitely presentable objects, such that every object is a filtered colimit of that set [4].

In [3, Theor. 3.5] it is shown that an algebra  $\mathbb{X}$  over a finitary monad  $T$  on an lfp category  $\mathcal{C}$  is a finitely generated object of  $\mathcal{C}^T$  iff there exists a finitely presentable object  $Y$  of  $\mathcal{C}$  and a morphism  $i : Y \rightarrow X$ , such that  $i^\sharp : (TY, \mu_Y) \rightarrow \mathbb{X}$  is a strong<sup>6</sup> epimorphism in  $\mathcal{C}^T$ . Below, we give a variant of this statement where instead the carrier of  $i^\sharp$  is a split<sup>7</sup> epimorphism in  $\mathcal{C}$ , which is the case iff  $\mathbb{X}$  admits a generator in the sense of Definition 4.

► **Proposition 41.** *Let  $\mathcal{C}$  be a lfp category in which strong epimorphisms split and  $T$  a finitary monad on  $\mathcal{C}$  preserving epimorphisms. Then an algebra  $\mathbb{X}$  over  $T$  is a finitely generated object of  $\mathcal{C}^T$  iff it is generated by a finitely presentable object  $Y$  in  $\mathcal{C}$  in the sense of Definition 4.*

<sup>6</sup> An epimorphism  $e : A \rightarrow B$  is said to be *strong*, if for any monomorphism  $m : C \rightarrow D$  and any morphisms  $f : A \rightarrow C$  and  $g : B \rightarrow D$  such that  $g \circ e = m \circ f$ , there exists a diagonal monomorphism  $d : B \rightarrow C$  such that  $f = d \circ e$  and  $g = m \circ d$ .

<sup>7</sup> A morphism  $e : A \rightarrow B$  is called *split*, if there exists a morphism  $s : B \rightarrow A$  such that  $e \circ s = \text{id}_B$ . Any morphism that is split is necessarily a strong epimorphism.

## 5 Related Work

A central motivation for this paper has been our broad interest in active learning algorithms for state-based models [5]. One of the challenges in learning non-deterministic models is the common lack of a unique minimal acceptor for a given language [16]. The problem has been independently approached for different variants of non-determinism, often with the common idea of finding a subclass admitting a unique representative [17, 10]. Unifying perspectives were given by van Heerdt [43, 41, 42] and Myers et al. [29]. One of the central notions in the work of van Heerdt is the concept of a scoop, originally introduced by Arbib and Manes [6].

In [46] we have presented a categorical framework that recovers minimal non-deterministic representatives in two steps. The framework is based on ideas closely related to the ones in [29], adopts scoops under the name generators, and strengthens the former to the notion of a basis. In a first step, it constructs a minimal bialgebra by closing a minimal coalgebra with additional algebraic structure over a monad. In a second step, it identifies generators for the algebraic part of the bialgebra, to derive an equivalent coalgebra with side effects in a monad. In this paper, we generalise the first step as application of a monad on an appropriate category of subobjects with respect to a  $(\mathcal{E}, \mathcal{M})$ -factorisation system, and explore the second step by further developing the abstract theory of generators and bases.

Categorical factorisation systems are well-established [13, 32, 25]. Among others, they have been used for a general view on the minimisation and determinisation of state-based systems [2, 1, 45]. In Section 3 we use the formalism of [2]. In Section 3.1 we have shown that under certain assumptions factorisation systems can be lifted to the categories of algebras and coalgebras. We later realised that the constructions had recently been published in [45].

The notion of a basis for an algebra over an arbitrary monad has been subject of previous interest. Jacobs, for instance, defines a basis as a coalgebra for the comonad on the category of algebras induced by the free algebra adjunction [19]. In Section 4.5 we show that a basis in our sense always induces a basis in their sense, and, conversely, it is possible to recover a basis in our sense from a basis in their sense, if certain assumptions about the existence and preservation of equalisers are given. As equalisers do not necessarily exist and are not necessarily preserved, our approach carries additional data and thus can be seen as finer.

## 6 Discussion and Future Work

We have generalised the closure of a subset of an algebraic structure as a monad between categories of subobjects relative to a factorisation system. We have identified the closure of a minimal coalgebra as an instance of the closure of subobjects that arise by taking the image of a morphism. We have extended the notion of a generator to a category of algebras with generators, and explored its characteristics. We have generalised the matrix representation theory of vector spaces and discussed bases for bialgebras. We compared our ideas with a coalgebraic generalisation of bases, explored the case in which a monad is induced by a variety, and related our notion to finitely generated objects in finitely presentable categories.

In [46] we have shown that generators and bases in our sense are central ingredients in the definitions of minimal canonical acceptors. Many such acceptors admit double-reversal characterisations [14, 15, 29, 44]. Duality based characterisations as the former have been shown to be closely related to minimisation procedures with respect to factorisation systems [12, 11, 45]. In the future, it would be interesting to further explore the connection between the minimality of generators on the one side, and the minimality of an acceptor with respect to a factorisation system on the other side.

Another interesting question is whether the construction that underlies our definition of a monad in Theorem 13 could be introduced at a more general level of an arbitrary adjunction between categories with suitable factorisation systems, such that the adjunction between the base category  $\mathcal{C}$  and the category of Eilenberg-Moore algebras  $\mathcal{C}^T$  is a special case.

---

## References

- 1 Jiri Adamek, Filippo Bonchi, Mathias Hülsbusch, Barbara König, Stefan Milius, and Alexandra Silva. A coalgebraic perspective on minimization and determinization. In *International Conference on Foundations of Software Science and Computational Structures*, pages 58–73. Springer, 2012. doi:10.1007/978-3-642-28729-9\_4.
- 2 Jiri Adamek, Horst Herrlich, and George E Strecker. Abstract and concrete categories: The joy of cats. *Reprints in Theory and Applications of Categories*, 2009.
- 3 Jiri Adamek, Stefan Milius, Lurdes Sousa, and Thorsten Wißmann. Finitely presentable algebras for finitary monads. *Theory and Applications of Categories*, 34(37):1179–1195, 2019.
- 4 Jiri Adamek and Jiri Rosicky. *Locally Presentable and Accessible Categories*, volume 189. Cambridge University Press, 1994. doi:10.1017/CB09780511600579.
- 5 Dana Angluin. Learning regular sets from queries and counterexamples. *Information and Computation*, 75(2):87–106, 1987. doi:10.1016/0890-5401(87)90052-6.
- 6 Michael A Arbib and Ernest G Manes. Fuzzy machines in a category. *Bulletin of the Australian Mathematical Society*, 13(2):169–210, 1975. doi:10.1017/S0004972700024412.
- 7 André Arnold, Anne Dicky, and Maurice Nivat. A note about minimal non-deterministic automata. *Bulletin of the EATCS*, 47:166–169, 1992.
- 8 Steve Awodey. *Category Theory*. Oxford University Press, Inc., 2010.
- 9 Jon Beck. Distributive laws. In *Seminar on Triples and Categorical Homology Theory*, pages 119–140. Springer, 1969. doi:10.1007/BFb0083084.
- 10 Sebastian Berndt, Maciej Liškiewicz, Matthias Lutter, and Rüdiger Reischuk. Learning residual alternating automata. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017. doi:10.1609/aaai.v31i1.10891.
- 11 Filippo Bonchi, Marcello M Bonsangue, Helle H Hansen, Prakash Panangaden, Jan Rutten, and Alexandra Silva. Algebra-coalgebra duality in brzozowski’s minimization algorithm. *ACM Transactions on Computational Logic (TOCL)*, 15(1):1–29, 2014. doi:10.1145/2490818.
- 12 Filippo Bonchi, Marcello M Bonsangue, Jan Rutten, and Alexandra Silva. Brzozowski’s algorithm (co)algebraically. In *Logic and Program Semantics*, pages 12–23. Springer, 2012. doi:10.1007/978-3-642-29485-3\_2.
- 13 Aldridge K Bousfield. Constructions of factorization systems in categories. *Journal of Pure and Applied Algebra*, 9(2-3):207–220, 1977. doi:10.1016/0022-4049(77)90067-6.
- 14 Janusz A Brzozowski. Canonical regular expressions and minimal state graphs for definite events. In *Proc. Symposium of Mathematical Theory of Automata*, volume 12, pages 529–561, 1962.
- 15 Janusz A. Brzozowski and Hellis Tamm. Theory of átomata. *Theor. Comput. Sci.*, 539:13–27, 2014. doi:10.1016/j.tcs.2014.04.016.
- 16 François Denis, Aurélien Lemay, and Alain Terlutte. Residual finite state automata. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 144–157. Springer, 2001. doi:10.1007/3-540-44693-1\_13.
- 17 Yann Esposito, Aurélien Lemay, François Denis, and Pierre Dupont. Learning probabilistic residual finite state automata. In *International Colloquium on Grammatical Inference*, pages 77–91. Springer, 2002. doi:10.1007/3-540-45790-9\_7.
- 18 Bart Jacobs. A bialgebraic review of deterministic automata, regular expressions and languages. In *Algebra, Meaning, and Computation*, pages 375–404. Springer, 2006. doi:10.1007/11780274\_20.

- 19 Bart Jacobs. Bases as coalgebras. In *Algebra and Coalgebra in Computer Science*, pages 237–252. Springer, 2011. doi:10.1007/978-3-642-22944-2\_17.
- 20 Bart Jacobs. *Introduction to Coalgebra: Towards Mathematics of States and Observation*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2016. doi:10.1017/CB09781316823187.
- 21 Bart Jacobs, Alexandra Silva, and Ana Sokolova. Trace semantics via determinization. In *International Workshop on Coalgebraic Methods in Computer Science*, pages 109–129. Springer, 2012. doi:10.1007/978-3-642-32784-1\_7.
- 22 Alexander Kurz. *Logics for Coalgebras and Applications to Computer Science*. PhD thesis, Ludwig-Maximilians-Universität München, 2000.
- 23 Serge Lang. Algebra. *Graduate Texts in Mathematics*, 2002. doi:10.1007/978-1-4613-0041-0.
- 24 Saunders Mac Lane. *Categories for the Working Mathematician*, volume 5. Springer, 2013. doi:10.1007/978-1-4757-4721-8.
- 25 Saunders MacLane. Duality for groups. *Bulletin of the American Mathematical Society*, 56(6):485–516, 1950.
- 26 Eugenio Moggi. *Computational Lambda-Calculus and Monads*. University of Edinburgh, Department of Computer Science, Laboratory for Foundations of Computer Science, 1988.
- 27 Eugenio Moggi. *An Abstract View of Programming Languages*. University of Edinburgh, Department of Computer Science, Laboratory for Foundations of Computer Science, 1990.
- 28 Eugenio Moggi. Notions of computation and monads. *Information and Computation*, 93(1):55–92, 1991. doi:10.1016/0890-5401(91)90052-4.
- 29 Robert S. R. Myers, Jiri Adamek, Stefan Milius, and Henning Urbat. Coalgebraic constructions of canonical nondeterministic automata. *Theoretical Computer Science*, 604:81–101, 2015. doi:10.1016/j.tcs.2015.03.035.
- 30 Anil Nerode. Linear automaton transformations. *Proceedings of the American Mathematical Society*, 9(4):541–544, 1958. doi:10.2307/2033204.
- 31 Louis Parlant, Jurriaan Rot, Alexandra Silva, and Bas Westerbaan. Preservation of Equations by Monoidal Monads. In *45th International Symposium on Mathematical Foundations of Computer Science (MFCS 2020)*, volume 170 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 77:1–77:14. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.MFCS.2020.77.
- 32 Emily Riehl. Factorization systems, 2008. URL: <https://math.jhu.edu/~eriehl/factorization.pdf>.
- 33 Jan Rutten. Universal coalgebra: A theory of systems. *Theoretical Computer Science*, 249(1):3–80, 2000. doi:10.1016/S0304-3975(00)00056-6.
- 34 Jan Rutten. The method of coalgebra: Exercises in coinduction, 2019.
- 35 Gavin J Seal. Tensors, monads and actions. *Theory and Applications of Categories*, 28(15):403–433, 2013.
- 36 Alexandra Silva, Filippo Bonchi, Marcello M Bonsangue, and Jan Rutten. Generalizing the powerset construction, coalgebraically. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2010)*, volume 8, pages 272–283. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2010. doi:10.4230/LIPIcs.FSTTCS.2010.272.
- 37 Ross Street. The formal theory of monads. *Journal of Pure and Applied Algebra*, 2(2):149–168, 1972. doi:10.1016/0022-4049(72)90019-9.
- 38 Ross Street. Weak distributive laws. *Theory and Applications of Categories*, 22:313–320, 2009.
- 39 Daniele Turi. *Functorial Operational Semantics*. PhD thesis, Vrije Universiteit Amsterdam, 1996.
- 40 Daniele Turi and Gordon Plotkin. Towards a mathematical operational semantics. In *Proceedings of Twelfth Annual IEEE Symposium on Logic in Computer Science*, pages 280–291. IEEE, 1997. doi:10.1109/LICS.1997.614955.

- 41 Gerco van Heerdt. An abstract automata learning framework. Master's thesis, Radboud University Nijmegen, 2016.
- 42 Gerco van Heerdt. *CALF: Categorical Automata Learning Framework*. PhD thesis, University College London, 2020.
- 43 Gerco van Heerdt, Matteo Sammartino, and Alexandra Silva. Learning automata with side-effects. In *Coalgebraic Methods in Computer Science*, pages 68–89. Springer, 2020. doi:10.1007/978-3-030-57201-3\_5.
- 44 Jean Vuillemin and Nicolas Gama. Efficient equivalence and minimization for non deterministic xor automata. Technical report, Ecole Normale Supérieure, 2010.
- 45 Thorsten Wißmann. Minimality notions via factorization systems and examples. *Logical Methods in Computer Science*, 18(3), 2022. doi:10.46298/lmcs-18(3:31)2022.
- 46 Stefan Zetsche, Gerco van Heerdt, Matteo Sammartino, and Alexandra Silva. Canonical automata via distributive law homomorphisms. *Electronic Proceedings in Theoretical Computer Science*, 351:296–313, 2021. doi:10.4204/eptcs.351.18.



# Bisimilar States in Uncertain Structures

Jurriaan Rot   

Radboud University, Nijmegen, The Netherlands

Thorsten Wißmann   

Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

Radboud University, Nijmegen, The Netherlands

---

## Abstract

We provide a categorical notion called uncertain bisimilarity, which allows to reason about bisimilarity in combination with a lack of knowledge about the involved systems. Such uncertainty arises naturally in automata learning algorithms, where one investigates whether two observed behaviours come from the same internal state of a black-box system that can not be transparently inspected. We model this uncertainty as a set functor equipped with a partial order which describes possible future developments of the learning game. On such a functor, we provide a lifting-based definition of uncertain bisimilarity and verify basic properties. Beside its applications to Mealy machines, a natural model for automata learning, our framework also instantiates to an existing compatibility relation on suspension automata, which are used in model-based testing. We show that uncertain bisimilarity is a necessary but not sufficient condition for two states being implementable by the same state in the black-box system. We remedy the lack of sufficiency by a characterization of uncertain bisimilarity in terms of coalgebraic simulations.

**2012 ACM Subject Classification** Theory of computation

**Keywords and phrases** Coalgebra, Relation Lifting, Bisimilarity, Mealy Machines, ioco

**Digital Object Identifier** 10.4230/LIPIcs.CALCO.2023.12

**Related Version** *Full Version:* <https://arxiv.org/abs/2303.15279>

**Funding** *Jurriaan Rot:* Partially supported by the NWO grant OCENW.M20.053.

*Thorsten Wißmann:* Supported by the NWO TOP project 612.001.852 (until 2022) and the DFG project 419850228 (since 2023)

## 1 Introduction

Inspired from constructive mathematics, Geuvers and Jacobs [6] introduced apartness relations on coalgebraic systems, complementing bisimilarity. While bisimilarity is a coinductive characterization of behavioural equivalence, apartness is inductive, and allows constructing *finite* proofs of difference in behaviour.

Although apartness and bisimilarity are just different sides of the same coin, the angle of ‘apartness’ turned out to be fruitful in the recent  $L^\#$  automata learning algorithm [21]. This algorithm works in the active learning setting of Angluin [1], where a *learner* tries to reconstruct the implementation of an automaton (or concretely a Mealy machine in [21]) from only its black-box behaviour. In  $L^\#$ , a crucial task of the learner is to determine whether two input words  $w, v$  lead to the identical or to distinct states in the black box. Throughout the learning game, the learner makes more and more observations. If at some point the learner finds out that the states  $q_w, q_v$  reached by  $w$  and  $v$  respectively have different behaviours, then  $q_w$  and  $q_v$  are provably different – called *apart*. For that, it is not required that we know the entire semantics of  $q_w$  and  $q_v$ ; instead, it suffices to observe one aspect of their behaviour in which they differ in incompatible ways.



© Jurriaan Rot and Thorsten Wißmann;

licensed under Creative Commons License CC-BY 4.0

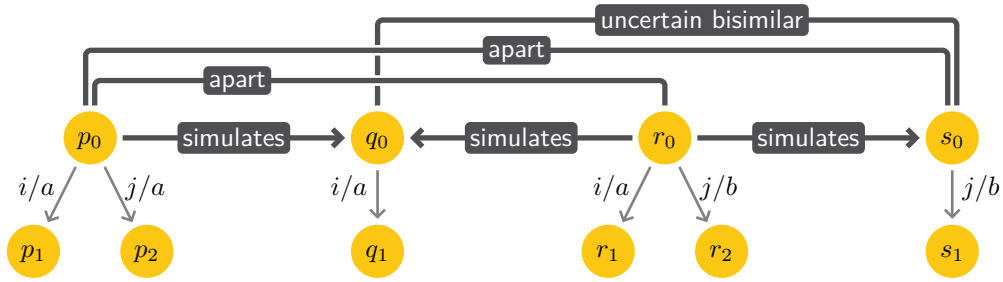
10th Conference on Algebra and Coalgebra in Computer Science (CALCO 2023).

Editors: Paolo Baldan and Valeria de Paiva; Article No. 12; pp. 12:1–12:17

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



■ **Figure 1** Partial Mealy machine for the input alphabet  $\{i, j\}$  and output alphabet  $\{a, b\}$ .

Once states turn out to be apart, they stay so throughout the entire remaining learning game, no matter which further observations of the black box are made. Thus, the apartness relation grows monotonically as the learning game progresses. This beauty of monotonicity breaks if we consider *bisimilarity*: as long as states  $q_w, q_v$  have not been proven different yet, should they be considered bisimilar? Or do we just have insufficiently much information at hand? If we do not know the number of states in the black box, we can never consider states  $q_w, q_v$  bisimilar with 100% certainty during the learning process.

In the present work, we close this gap by introducing the notion of *uncertain bisimilarity*, which expresses that two states might be bisimilar – but we are not certain about it, because we simply did not observe any reason yet that would disprove bisimilarity. The main idea is exemplified by the Mealy machine in Figure 1: states  $p_0$  and  $r_0$  are apart, because  $p_0$  has output  $a$  on input  $j$  whereas  $r_0$  yields a different output  $b$  on input  $j$ . By the same input  $j$ , we can tell that  $p_0$  and  $s_0$  are apart, even though we do not yet know the behaviour of  $s_0$  on input  $i$ . Furthermore, states  $q_0$  and  $s_0$  can either turn out to be apart or to be bisimilar, depending on their outputs on input  $j$ . Thus, we call  $q_0$  and  $s_0$  *uncertain bisimilar*. If we for instance try to explore the output of  $q_0$  on input  $j$ , then depending on the output,  $q_0$  will be identical to  $p_0$  or  $r_0$ . Until we know this,  $q_0$  is simulated by both  $p_0$  and by  $r_0$ . Simulation is a special case of uncertain bisimilarity, because it not only says that the behaviours are compatible, but that one behaviour is even included in the other.

Our aim is to establish a theory of uncertain bisimilarity at the level of coalgebras, including the motivating example of Mealy machines. Working with bisimulation relations and bisimilarity benefits from a rich categorical theory. In particular, they are themselves coalgebras in the category of relations [8, 11]. Here, the coalgebraic type functor considered on relations is a lifting of the original coalgebraic type functor for the systems of interest.

In the present paper, we incorporate the explicit treatment of the lack of knowledge which is omnipresent in the learning setting. Formally, we do this by equipping the type functor with a partial order. This order  $s \sqsubseteq t$  represents that the behaviour  $s$  observed so far might be extended to behaviour  $t$  after additional observations. This order immediately induces two further notions of lax coalgebra morphisms.

**Contributions.** With such a partial order on the type functor, we will:

- Define a generic system equality notion, called *uncertain bisimilarity*, derived from the relation lifting of the type functor.
- We show basic properties of the relation such as reflexivity and symmetry. It is immediate that uncertain bisimilarity is not transitive, and thus, no equivalence relation.
- As instances, we discuss (partial) Mealy machines as the running example. Moreover, we cover suspension automata, for which uncertain bisimilarity instantiates to an existing compatibility notion, used in the *ioco* conformance relation from model-based testing [22].



- It is a standard result that standard coalgebraic bisimilarity coincides with being identifiable by coalgebra morphisms (under often-met assumptions on the type functor, see e.g. [18]). We show that uncertain bisimilarity is not characterized via identifiability in lax coalgebra morphisms – for the running example of Mealy machines.
- Instead, we show that uncertain bisimilarity is characterized via coalgebraic simulations – for two definitions of coalgebraic simulation. Concretely, two states are uncertain bisimilar if there is a state in another coalgebra that simulates both (e.g.  $r_0$  simulates both  $q_0$  and  $s_0$  in Figure 1).

## 2 Preliminaries

We first establish the basic coalgebraic notions used in the technical development later, see e.g. [11]. We assume that the reader is familiar with basic concepts from category theory.

► **Definition 2.1.** For a functor  $F: \mathcal{C} \rightarrow \mathcal{C}$ , an  $F$ -coalgebra  $(C, c)$  is an object  $C \in \mathcal{C}$  (the carrier) together with a morphism  $c: C \rightarrow FC$  in  $\mathcal{C}$  (the structure). For two  $F$ -coalgebras, a coalgebra morphism  $h: (C, c) \rightarrow (D, d)$  is a morphism  $h: C \rightarrow D$  satisfying  $Fh \cdot c = d \cdot h$ . We denote the category of  $F$ -coalgebras by  $\text{Coalg}(F)$ .

Most of our coalgebras will live in the category  $\text{Set}$  of sets and maps and our leading example is the functor modelling Mealy machines:

► **Example 2.2.**

1. For fixed sets  $I$  and  $O$  of input and output symbols, consider the functor

$$\mathcal{M}_T: \text{Set} \rightarrow \text{Set} \quad \mathcal{M}_T X = (O \times X)^I.$$

An  $\mathcal{M}_T$ -coalgebra is then a set  $C$  together with a map  $c: C \rightarrow (O \times C)^I$  which sends each state  $q \in C$  and input symbol  $i \in I$  to a pair of an output symbol and a successor state to which the Mealy machine transitions:  $c(q)(i) \in O \times C$ . We write

$$q \xrightarrow{i/o} q'$$

to specify that  $c(q)(i) = (o, q')$ . In the name of the functor, the index  $T$  shall indicate that the Mealy machine is *total*, in the sense that it is defined for every input  $i \in I$ .

2. The finitary powerset functor  $\mathcal{P}_f$  sends each set  $X$  to the set of its finite subsets  $\mathcal{P}_f X = \{S \subseteq X \mid S \text{ finite}\}$  and maps  $f: X \rightarrow Y$  to direct images:  $\mathcal{P}_f f: \mathcal{P}_f X \rightarrow \mathcal{P}_f Y$ ,  $\mathcal{P}_f f(S) := \{f(x) \mid x \in S\}$ .

A canonical domain for the semantics of coalgebras is the *final* coalgebra:

► **Definition 2.3.** The final  $F$ -coalgebra is the final object in  $\text{Coalg}(F)$ . Concretely, a coalgebra  $(D, d)$  is final if for every  $(C, c)$  in  $\text{Coalg}(F)$  there is a unique coalgebra morphism  $h: (C, c) \rightarrow (D, d)$ . If it exists, we denote the final coalgebra for  $F$  by  $(\nu F, \tau)$  and the induced unique morphism for  $(C, c)$  by  $\llbracket - \rrbracket: (C, c) \rightarrow (\nu F, \tau)$ .

► **Example 2.4.** The final  $\mathcal{M}_T$ -coalgebra is carried by the set  $\nu \mathcal{M}_T = O^{I^+}$  – the set of all maps  $I^+ \rightarrow O$  from non-empty words  $I^+$  to  $O$ .

Equivalently, we can characterize the semantics  $\nu \mathcal{M}_T$  in terms of maps  $I^* \rightarrow O^*$  that interact nicely with the prefix-order on words:

► **Notation 2.5.** For words  $v, w \in I^*$  (in particular also for non-empty words  $I^+ \subseteq I^*$ ), we write  $v \leq w$  to denote that  $v$  is a prefix of  $w$ . The length of a word  $w$  is denoted by  $|w| \in \mathbb{N}$ .

Then, we can characterize  $\nu\mathcal{M}_T$  as maps  $I^* \rightarrow O^*$  that preserve length and prefixes of words:

$$\nu\mathcal{M}_T \cong \{f: I^* \rightarrow O^* \mid \text{for all } w \in I^*: |f(w)| = |w| \text{ and for all } v \leq w: f(v) \leq f(w)\}$$

### 3 A Lax Coalgebra Morphism Lacks Knowledge

In the learning game for Mealy machines, the learner tries to reconstruct the internal implementation of a Mealy machine

$$c: C \rightarrow \mathcal{M}_T C = (O \times C)^I$$

by only its black-box behaviour. For that, one assumes a distinguished initial state  $q_0 \in C$  and it is the task of the learner to construct a Mealy machine with the same behaviour  $\llbracket q_0 \rrbracket$  as that of  $q_0$ . Being in a block-box setting means that the learner knows neither  $C$  or  $c$ . Instead, the learner can enter a word  $i_1, \dots, i_n \in I$  of input symbols from the input alphabet  $I$  to the black box, referred to as a *query*, and then observe the output symbols  $o_1, \dots, o_n \in O$ . More precisely, the learner observes the output symbols

$$o_k = \llbracket q_0 \rrbracket \underbrace{(i_1 \cdots i_k)}_{\in I^+} \in O \quad \text{for every } 1 \leq k \leq n \quad (\text{with } \llbracket - \rrbracket \text{ as in Example 2.4}).$$

On this query, the black box reveals the output  $o_1 \in O$  of the initial state for input  $i_1$ . But after performing only this query, we still don't know the output for all the other input symbols  $i'_1 \in I$ ,  $i'_1 \neq i_1$  with which we could have started the input word.

After such a query, the black box returns to the initial state  $q_0$  in order to be ready for the next query. In concrete learning scenarios this reset to initial state is for example realized by resetting the actual hardware of a system that is learned. When learning network protocol implementations, this reset-behaviour is realized by opening a separate network connection (or session) for each new input query.

The  $L^\#$  algorithm (for Mealy machines) gathers all the information from the performed input queries in an *observation tree*. This bundles the observations of the experiments so far in a single data structure. However, this structure is not an  $\mathcal{M}_T$ -coalgebra itself, because the knowledge about the outputs for some inputs  $i \in I$  in some states in the tree will be lacking.

We can model this lack of knowledge by the following functor

$$\mathcal{M}: \text{Set} \rightarrow \text{Set} \quad \mathcal{M}X = (\{?\} + O \times X)^I. \quad (1)$$

The element '?' models that we do not know the transition yet.

► **Notation 3.1.** We abbreviate partial functions via  $(X \rightarrow Y) := (\{?\} + Y)^X$ .

So we can also write  $\mathcal{M}X = (I \rightarrow O \times X)$ . Compared to  $\mathcal{M}_T$ , a state  $q$  in an  $\mathcal{M}$ -coalgebra  $d: D \rightarrow \mathcal{M}D$  is either undefined for an input  $i \in I$ , i.e.  $d(q)(i) = ?$ , or has a transition defined, i.e. we have both an output  $o \in O$  and a successor state  $q' \in D$  with

$$d(q)(i) = (o, q') \quad \text{or using notation:} \quad q \xrightarrow{i/o} q'.$$

This kind of partiality in  $\mathcal{M}$  models that whenever the learner sends a word  $w \in I^+$ , the black box reveals the output symbols of all transitions along the way of processing  $w$ . Thus, the semantics of states  $q \in D$  in such a partial Mealy machine, i.e. an  $\mathcal{M}$ -coalgebra, can be characterized by:

$$\nu\mathcal{M} := \{f: I^+ \rightarrow O \mid \text{for all } v \leq w \text{ if } f(w) \in O \text{ then } f(v) \in O\}. \quad (2)$$

This monotonicity condition describes that whenever a learner has observed the behaviour for an input word  $w \in I^+$ , then we also have observed the outputs of all the prefixes  $v \leq w$ .

$$\begin{array}{ccc} C & \xrightarrow{h} & D \\ c \downarrow & \sqsubseteq & \downarrow d \\ FC & \xrightarrow{Fh} & FD \end{array}$$

■ **Figure 2** Diagrammatic notation of a lax  $F$ -coalgebra morphism  $h: (C, c) -\sqsubseteq\rightarrow (D, d)$ .

$$\begin{array}{ccc} C & \xrightarrow{h} & D \\ c \downarrow & \supseteq & \downarrow d \\ FC & \xrightarrow{Fh} & FD \end{array}$$

■ **Figure 3** Diagrammatic notation of an oplax  $F$ -coalgebra morphism  $h: (C, c) -\supseteq\rightarrow (D, d)$ .

► **Proposition 3.2.** *The final  $\mathcal{M}$ -coalgebra  $(\nu\mathcal{M}, \tau)$  is characterized by (2) and the map*

$$\tau: \nu\mathcal{M} \rightarrow (I \rightarrow O \times \nu\mathcal{M}) \quad \tau(f) = i \mapsto \begin{cases} ? & \text{if } f(i) = ? \\ (o, w \mapsto f(iw)) & \text{if } f(i) \in O \end{cases}$$

The structure sends every  $f \in \nu\mathcal{M}$  to a successor structure of type  $\mathcal{M}(\nu\mathcal{M})$ . For  $i \in I$ , this successor structure yields  $\tau(f)(i) \in \{?\} + O \times \nu\mathcal{M}$ .

During the learning process, the learner might be able to modify the coalgebra after the output of state  $q$  on input  $i$  has been observed. In this sense, we increase knowledge, and this can be modelled by the usual order on partial functions:

► **Definition 3.3.** *For partial functions  $t, s: A \rightarrow B$ , we fix the partial order*

$$t \sqsubseteq_{A \rightarrow B} s \stackrel{\text{def}}{\iff} \forall i \in I: t(i) \in \{s(i), ?\}.$$

The functor  $\mathcal{M}X = (I \rightarrow O \times X)$  inherits the poset structure  $(\mathcal{M}X, \sqsubseteq)$  from partial maps.

The equivalence means that for every input  $i \in I$ , the value of  $t(i)$  is either undefined ( $t(i) = ?$ ) or agrees with  $i$ th entry in the other successor structure ( $t(i) = s(i)$ ). The partial order itself represents how the behaviour can possibly be completed if we found out more information about the full Mealy machines. That is, the partial order shows possible options in the future learning process.

This principle also works for other system types, so we generally assume (e.g. [9]):

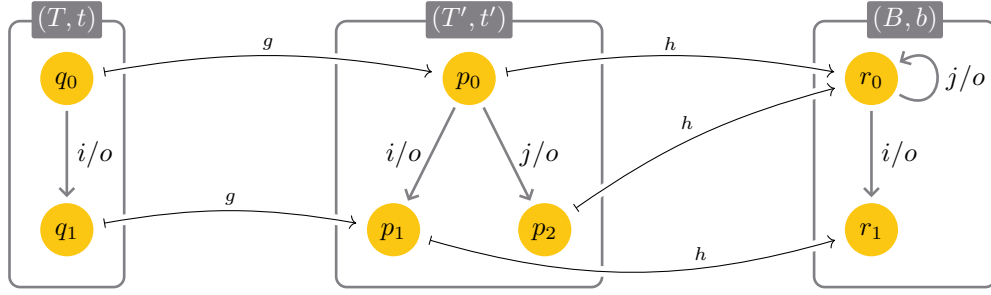
► **Assumption 3.4.** *Fix a functor  $F_{\text{Pos}}: \text{Set} \rightarrow \text{Pos}$  and define:*

- $F := U \cdot F_{\text{Pos}}$ , where  $U: \text{Pos} \rightarrow \text{Set}$  is the usual forgetful functor.
- Let  $\sqsubseteq_{FX}$  be the order on  $F_{\text{Pos}}X$ , i.e. we have  $F_{\text{Pos}}X = (FX, \sqsubseteq_{FX})$ .

The functoriality of  $F_{\text{Pos}}$  means that for every  $f: X \rightarrow Y$ , the map  $Ff: FX \rightarrow FY$  is monotone. This partial order gives rise to a lax notion of coalgebra morphisms:

► **Definition 3.5.** *A lax  $F$ -coalgebra morphism  $h: (C, c) -\sqsubseteq\rightarrow (D, d)$  between  $F$ -coalgebras is a map  $h: C \rightarrow D$  such that for all  $x \in C$  we have  $Fh(c(x)) \sqsubseteq_{FD} d(h(x))$ . We write  $\sqsubseteq$  in squares to indicate lax commutativity as shown in Figure 2. Dually, an oplax  $F$ -coalgebra morphism  $h: (C, c) -\supseteq\rightarrow (D, d)$  is a map  $h: C \rightarrow D$  such that for all  $x \in C$ , we have  $Fh(c(x)) \supseteq_{FD} d(h(x))$ , and denoted in diagrams as shown in Figure 3. In contrast, we write  $\circlearrowright$  to emphasize proper commutativity.*

Lax coalgebra morphisms could also be called functional simulations, because they are simulations (a special kind of relation) and they are functional (a property on relations). Intuitively,  $h: (C, c) -\sqsubseteq\rightarrow (D, d)$  means that  $(D, d)$  has at least as many transitions as  $(C, c)$ . Conversely,  $h: (C, c) -\supseteq\rightarrow (D, d)$  means that  $(D, d)$  has possibly fewer transitions than  $(C, c)$ .



■ **Figure 4** Two lax  $\mathcal{M}$ -coalgebra morphism  $g: (T, t) \dashv\vdash (T', t')$  and  $h: (T', t') \dashv\vdash (B, b)$  for  $\mathcal{M}X = (\{?\} + O \times X)^I$  with  $I = \{i, j\}$ ,  $O = \{o\}$ .

In the learning game, lax coalgebra homomorphisms arise naturally, because there, all observations are collected in an observation tree  $(T, t)$ . This observation tree is an  $F$ -coalgebra that admits a lax  $F$ -coalgebra morphism  $h: (T, t) \rightarrow (B, b)$  to the black-box  $(B, b)$  that needs to be learned. An example of lax morphisms for Mealy machines is visualized in Figure 4.

Of course, the learner only sees the observation tree  $(T, t)$  but neither  $(B, b)$  nor  $h$ . But, the learner can make use of the fact that there is *some* lax coalgebra morphism, and can use it to deduce properties of  $(B, b)$ . The correctness proof of the  $L^\#$  learning algorithm [21] in fact relies on the existence of such a lax coalgebra morphism.

**Suspension automata.** Related to automata learning is the application of *conformance testing* of state-based systems. In particular, the *ioco* (input output conformance) relation from testing theory [19] nicely fits into the coalgebraic theory too, while using a non-trivial order on the functor. Specifically, we will focus on *suspension automata*, and later recover the notion of *ioco*-compatibility from [22], see Definition 5.10. Suspension automata are a subclass of deterministic labelled transition systems. They are coalgebras for the following functor:

► **Definition 3.6.** For partial maps which are defined on at least one input we write

$$(A \xrightarrow{\text{ne}} B) := \{f: A \rightarrow B \mid \exists a \in A: f(a) \neq ?\}.$$

For a fixed set of inputs  $I$  and outputs  $O$ , define the suspension automaton functor

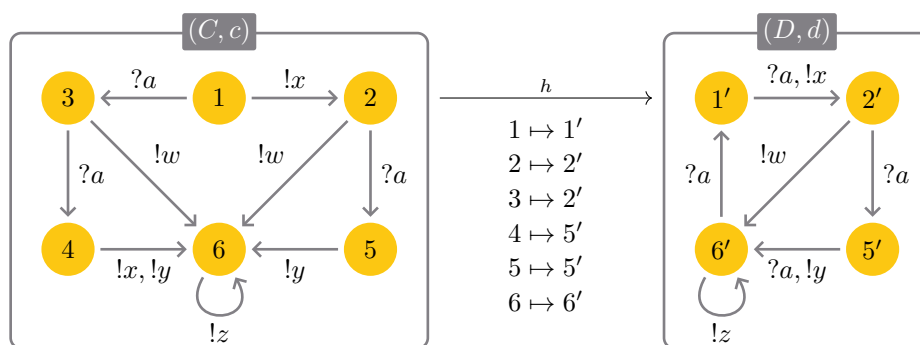
$$SX := (I \rightarrow X) \times (O \xrightarrow{\text{ne}} X).$$

► **Notation 3.7.** We denote the projections from (subsets of) the cartesian product  $X \times Y$  by  $\pi_1: X \times Y \rightarrow X$  and  $\pi_2: X \times Y \rightarrow Y$ .

Following existing presentations [22, 19], it is not hard to see that suspension automata are coalgebras for this functor:

► **Definition 3.8.** A suspension automaton is a finite  $\mathcal{S}$ -coalgebra, i.e. a finite set of states  $C$  and a map  $c: C \rightarrow \mathcal{S}C$ . For a given coalgebra  $(C, c)$ , we write  $?$  for input transitions and  $!$  for output transitions:

$$x \xrightarrow{?a} y \quad \stackrel{\text{def}}{\iff} \quad \pi_1(c(x))(a) = y \quad \text{and} \quad x \xrightarrow{!a} y \quad \stackrel{\text{def}}{\iff} \quad \pi_2(c(x))(a) = y$$



■ **Figure 5** Examples of suspension automata and a lax  $\mathcal{S}$ -coalgebra morphism between them.

In other words, a suspension automaton is a deterministic LTS where the set of labels is partitioned into inputs and outputs. For some of the inputs and for some of the outputs, a suspension automaton in some state  $x$  can make a transition to another state. But every state is non-blocking in the sense that for every state  $x$  there is at least one output  $o \in O$  such that  $x$  can make a transition  $x \xrightarrow{!o} y$ .

The *ioco* compatibility relation (recalled in Definition 5.10) is characterized by a bisimulation game with an alternating flavour, which can be captured by reversing the order in the output part of the functor  $\mathcal{S}$ :

► **Definition 3.9.** For  $(s_i, s_o) \in \mathcal{S}X$  and  $(t_i, t_o) \in \mathcal{S}X$  we put:

$$(s_i, s_o) \sqsubseteq_{\mathcal{S}X} (t_i, t_o) \quad \stackrel{\text{def}}{\iff} \quad \underbrace{s_i \sqsubseteq t_i}_{\text{in } I \rightarrow X} \quad \text{and} \quad \underbrace{t_o \sqsubseteq s_o}_{\text{in } O \rightarrow X}$$

In other words, when ascending in the order of  $\mathcal{S}$ , input transitions can be added and output transitions can be removed if there is still at least one output transition afterwards.

► **Example 3.10.** We recall two examples of suspension automata from van den Bos et al. [22, Fig. 1] in Figure 5. With the order on  $\mathcal{S}$  (Definition 3.9), there is a lax coalgebra morphism  $h: (C, c) \rightarrow (D, d)$  between them that identifies some of the states:  $h(3) = h(2)$  and  $h(4) = h(5)$ . The map  $h$  is only a lax coalgebra morphism because there is no input transition for  $a$  from 5 (or 4) to 6 in  $(C, c)$ , but we have  $5' \xrightarrow{?a} 6'$  in  $(D, d)$ . Conversely, there is an output transition  $4 \xrightarrow{!x} 6$  in  $(C, c)$  but there is no transition  $h(4) = 5' \xrightarrow{!x} 6' = h(6)$  in  $(D, d)$ . Summarizing the above two points, we have:

$$\pi_1(\mathcal{S}h(c(5))) \not\sqsubseteq \pi_1(d(h(5))) \quad \text{and} \quad \pi_2(\mathcal{S}h(c(4))) \not\sqsupseteq \pi_2(d(h(4)))$$

The partial order on the functor does not only relax the notion of morphism, but also gives rise to a new coalgebraic bisimulation notion, which we introduce in Section 5.

## 4 Bisimulation Notions are Liftings

In this section we recall coalgebraic bisimulation through the lens of relation liftings; for an extensive introduction see [11]. We start by fixing some notation regarding relations.

► **Notation 4.1.** Given relations  $R \subseteq X \times Y$  and  $S \subseteq Y \times Z$ , the composition  $R \circ S$  is given by:  $R \circ S := \{(x, z) \in X \times Z \mid \exists y \in Y: (x, y) \in R \text{ and } (y, z) \in S\}$ . We denote the converse of  $R$  by  $R^{\text{op}} = \{(y, x) \mid (x, y) \in R\}$ . The equality relation (also called the diagonal) on a set

## 12:8 Bisimilar States in Uncertain Structures

$X$  is denoted by  $\text{Eq}_X = \{(x, x) \mid x \in X\}$ . Given a map  $f: X \rightarrow Y$  and a relation  $U \subseteq Y \times Y$ , inverse image is denoted by  $(f \times f)^{-1}(U) = \{(x_1, x_2) \mid (f(x_1), f(x_2)) \in U\}$ . The kernel relation of  $f$  is given by  $\ker f = \{(x_1, x_2) \mid f(x_1) = f(x_2)\}$ . Note that  $\ker f = (f \times f)^{-1}(\text{Eq}_Y)$ .

For a structural study of (bi)simulation notions on coalgebras, we consider the fibred category of relations:

► **Definition 4.2.** The category  $\text{Rel}$  has objects  $(X, R)$ , where  $X$  is a set and  $R \subseteq X \times X$ , i.e.  $R$  is a relation on  $X$ . The morphisms  $f: (X, R) \rightarrow (Y, S)$  in  $\text{Rel}$  are maps  $f: X \rightarrow Y$  that preserve the relation, i.e.  $(x_1, x_2) \in R$  implies  $(f(x_1), f(x_2)) \in S$ . The obvious forgetful functor is  $p: \text{Rel} \rightarrow \text{Set}$ , given by  $p(X, R) = X$ .

The forgetful functor  $p$  is a fibration; for a thorough introduction, see the first chapter of Jacobs' book [10]. We can express the preservation property of the morphisms in  $\text{Rel}$  in a point-free way:  $f: X \rightarrow Y$  is a map from  $(X, R)$  to  $(Y, S)$  in  $\text{Rel}$  if and only if

$$R \subseteq (f \times f)^{-1}(S).$$

Equality extends to a functor  $\text{Eq}: \text{Set} \rightarrow \text{Rel}$ , given by  $\text{Eq}(X) = (X, \text{Eq}_X)$  and  $\text{Eq}(f) = f$ . This is well-defined since we have  $\text{Eq}_X \subseteq (f \times f)^{-1}(\text{Eq}_Y)$  for every map  $f: X \rightarrow Y$ .

To study relations on  $F$ -coalgebras – most notably notions of behavioural equivalence and inclusion – we lift the type functor  $F$  from  $\text{Set}$  to an endofunctor  $\hat{F}$  on  $\text{Rel}$ .

$$\begin{array}{ccc} \text{Rel} & \xrightarrow{\hat{F}} & \text{Rel} \\ p \downarrow & & \downarrow p \\ \text{Set} & \xrightarrow{F} & \text{Set} \end{array}$$

For all set functors, such a lifting exists in a canonical way:

► **Definition 4.3.** For a functor  $F: \text{Set} \rightarrow \text{Set}$ , the relation lifting  $\hat{F}: \text{Rel} \rightarrow \text{Rel}$  is given by

$$\hat{F}(R \subseteq X \times X) = \{(x, y) \in FX \times FX \mid \exists t \in FR: F\pi_1(t) = x \text{ and } F\pi_2(t) = y\}.$$

Hence,  $\hat{F}$  transforms relations on  $X$  into relations on  $FX$ . Before we proceed, we list several standard properties of the relation lifting:

► **Lemma 4.4** [11]. For every functor  $F: \text{Set} \rightarrow \text{Set}$ , we have

1. *Monotonicity:* if  $R \subseteq S$  then  $\hat{F}(R) \subseteq \hat{F}(S)$ .
2. *Preservation of equality:*  $\text{Eq} \circ F \subseteq \hat{F} \circ \text{Eq}$ .
3. *Preservation of converse:*  $\hat{F}(R^{\text{op}}) = (\hat{F}(R))^{\text{op}}$  for all  $R \subseteq X \times X$ .
4. *Preservation of inverse images:* For a map  $f: X \rightarrow Y$  and relation  $S \subseteq Y \times Y$ , we have

$$\hat{F}((f \times f)^{-1}(S)) \subseteq (Ff \times Ff)^{-1}(\hat{F}(S)).$$

Moreover, if  $F$  preserves weak pullbacks, then this is an equality.

As a consequence of monotonicity and preservation of inverse images,  $\hat{F}$  indeed extends to a lifting of  $F$ , given on morphisms by  $\hat{F}(f) = F(f)$ .

► **Example 4.5.** In our example of (partial) Mealy machines as coalgebras for  $\mathcal{M}X = (\{?\} + O \times X)^I$ , a relation  $R \subseteq X \times X$  is lifted to the relation  $\hat{\mathcal{M}}R \subseteq \mathcal{M}X \times \mathcal{M}X$  given by

$$(s, t) \in \hat{\mathcal{M}}R \quad \text{iff} \quad \text{for all } i \in I: \quad (s(i) = ? \text{ and } t(i) = ?) \text{ or} \\ (s(i), t(i)) \in \{(o, x), (o, y) \mid o \in O, (x, y) \in R\}$$

Thus, successor structures  $s, t \in \mathcal{M}R$  are related by  $\hat{\mathcal{M}}R$  if  $s$  and  $t$  have transitions defined for the same inputs  $i \in I$ , and for all inputs  $i \in I$  for which  $s(i) = (o, x)$  and  $t(i) = (o', y)$  are defined, both have the same output  $o = o'$ , and the successor states are related  $(x, y) \in R$ .

The relation lifting is reminiscent of the criterion of a relation  $R \subseteq C \times C$  being a bisimulation on Mealy machines. However, in relation liftings, we can distinguish between the relation  $R$  on the successor states on the one hand and the relation on the predecessor states on the other hand. If we let the relation on predecessor and successor states coincide, then the relation lifting gives rise to bisimilarity as follows [8, 11].

► **Definition 4.6.** *A relation  $R \subseteq C \times C$  on the state space of a coalgebra  $c: C \rightarrow FC$  is a bisimulation if  $R \subseteq (c \times c)^{-1}(\hat{F}(R))$ . States  $x, y \in C$  are called bisimilar if there is a bisimulation relating them.*

Note that  $(c \times c)^{-1}(\hat{F}(-)): \text{Rel}_C \rightarrow \text{Rel}_C$  is a monotone map on the complete lattice  $\text{Rel}_C = \mathcal{P}(C \times C)$  of relations on  $C$ , ordered by inclusion. A bisimulation is thus a post-fixed point for this map, and bisimilarity is the greatest post-fixed point, which is also the greatest fixed point by the Knaster-Tarski theorem. Characterizing bisimilarity as the greatest fixed point of a monotone map is standard in the classical theory of coinduction [16].

► **Remark 4.7 (Disjoint union of coalgebras).** In the definition of bisimulation, we consider a relation  $R$  on the state space of a single coalgebra  $c: C \rightarrow FC$ . This bisimulation notion straightforwardly generalizes to states of different  $F$ -coalgebras  $x \in C$ , and  $y \in D \xrightarrow{d} FD$ , because we can consider the bisimulation notion on the disjoint union (i.e. coproduct) of the coalgebras  $(C, c)$  and  $(D, d)$ :

$$C + D \xrightarrow{c+d} FC + FD \xrightarrow{[F\text{inl}, F\text{inr}]} F(C + D)$$

where  $\text{inl}: C \rightarrow C + D$  and  $\text{inr}: D \rightarrow C + D$  are the coproduct injections and  $[-, -]$  is case distinction (i.e. the universal mapping property of the coproduct). So by the bisimilarity of  $x$  and  $y$  we mean the bisimilarity of  $\text{inl}(x)$  and  $\text{inr}(y)$  in the above combined coalgebra. One can easily see that this generalization is well-defined: in the special case where  $(D, d) := (C, c)$ , states  $x, y$  in  $C$  are bisimilar iff  $\text{inl}(x), \text{inr}(y)$  are bisimilar in the coalgebra on  $C + C$ .

► **Example 4.8.** The relation lifting for  $\mathcal{M}$  (Example 4.5) thus gives rise to the following: a bisimulation on a coalgebra  $c: C \rightarrow \mathcal{M}C$  is a relation  $R \subseteq C \times C$  such that

$$R \subseteq (c \times c)^{-1}(\hat{\mathcal{M}}R).$$

Spelling out the inclusion yields that  $R$  is a bisimulation iff for all  $(x, y) \in R$  and  $i \in I$ :

1.  $c(x)(i) = ?$  iff  $c(y)(i) = ?$ ,
2. if  $c(x)(i) = (o, x') \in O \times C$ , then  $c(y)(i) = (o, y')$  for some  $y' \in C$  with  $(x', y') \in R$ , and
3. if  $c(y)(i) = (o, y') \in O \times C$ , then  $c(x)(i) = (o, x')$  for some  $x' \in C$  with  $(x', y') \in R$ .

For example, the leaf states  $q_1, p_1, p_2, r_1$  in Figure 4 are all pairwise bisimilar. However,  $q_0$  and  $p_0$  are not bisimilar:  $q_0$  can not mimic the  $j$ -transition of  $p_0$ . Similarly,  $q_0$  and  $r_0$  are not bisimilar (and also  $p_0$  and  $r_0$  are not bisimilar).

In order to still express the compatibility of  $q_0$  and  $p_0$ , we relax the notion of coalgebraic bisimilarity in the next section.

## 5 Uncertain Bisimilarity

So far, we have not considered the order  $\sqsubseteq$  when discussing bisimulations on coalgebras for a functor  $F$  satisfying Assumption 3.4. By taking the order into account, we introduce the notion of *uncertain bisimilarity*. In particular, in the example of Mealy machines it captures a notion of equivalence where ‘unknown’ transitions are ignored. Since we stick to the principle that bisimulation notions are coalgebras in  $\mathbf{Rel}$ , we only need to make use of the order  $\sqsubseteq$  when defining a functor on  $\mathbf{Rel}$ . The desired bisimulations will then be coalgebras for this functor:

► **Definition 5.1.** *The uncertain relation lifting of  $F$  is defined on a relation  $R \subseteq X \times X$  by*

$$\hat{F}_{\sqsubseteq}(R) := \sqsubseteq_{FX} \circ \hat{F}(R) \circ \sqsupseteq_{FX}$$

► **Remark 5.2.** Definition 5.1 is inspired by the notion of *simulation* on coalgebras by Hughes and Jacobs [9]. In their work, a simulation on a coalgebra  $(C, c)$  is a relation  $R$  such that

$$R \subseteq (c \times c)^{-1}(\sqsubseteq_{FC} \circ \hat{F}(R) \circ \sqsubseteq_{FC})$$

The lifting  $\sqsubseteq \circ \hat{F}(-) \circ \sqsupseteq$  of  $F$  is referred to in *op. cit.* as the *lax relation lifting*.

► **Definition 5.3.** *An uncertain bisimulation  $R$  on a coalgebra  $c: C \rightarrow FC$  is a relation  $R \subseteq C \times C$  with  $R \subseteq (c \times c)^{-1}(\hat{F}_{\sqsubseteq}(R))$ . States  $x, y \in C$  are called uncertain bisimilar if there is an uncertain bisimulation relating them. Complementarily,  $x, y \in C$  are called *apart* if there is no uncertain bisimulation relating them.*

The uncertainty here expresses that in the learning setting, we are not entirely certain that the two states are bisimilar. With an extension of the system by a future observation, they might turn out to be non-bisimilar. With this intuition, the opposite property is simply called *apart*: whenever two states are separated, they will stay so no matter how the system might be extended by further transitions.

When unfolding the definitions, we obtain the following explicit characterization:

► **Lemma 5.4.** *A relation  $R \subseteq C \times C$  on  $c: C \rightarrow FC$  is an uncertain bisimulation if and only if for every  $(x, y) \in R$  there exists  $t \in FR$  such that*

$$c(x) \sqsubseteq F\pi_1(t) \quad \text{and} \quad c(y) \sqsubseteq F\pi_2(t).$$

When representing the witnesses  $t$  as a choice function, then we equivalently have a map  $r: R \rightarrow FR$  making the projections  $\pi_1, \pi_2$  oplax coalgebra morphisms  $(R, r) \multimap (C, c)$ :

$$\begin{array}{ccccc} C & \xleftarrow{\pi_1} & R & \xrightarrow{\pi_2} & C \\ \downarrow c & \sqsubseteq & \downarrow r & \sqsupseteq & \downarrow c \\ FC & \xleftarrow{F\pi_1} & FR & \xrightarrow{F\pi_2} & FC \end{array}$$

This characterization instantiates to partial Mealy machines as:

► **Lemma 5.5.** *For partial Mealy machines  $c: C \rightarrow MC$  a reflexive relation  $R \subseteq C \times C$  is an uncertain bisimulation if and only if for all  $(x, y) \in R$  and  $i \in I$  we have:*

$$x \xrightarrow{i/o} x' \text{ and } y \xrightarrow{i/o'} y' \quad \text{imply} \quad o = o' \text{ and } (x', y') \in R. \quad (3)$$



This condition is vacuously satisfied for all  $(x, y) \in R$  and  $i \in I$  whenever  $x$  or  $y$  have no  $i$ -transition defined. In this characterization, we use the mild assumption of  $R$  being reflexive in order to be able to define the coalgebra structure  $r: R \rightarrow FR$  of Lemma 5.4 in the case where only one of the related states has an  $i$ -transition defined. Even without  $R$  being reflexive, every uncertain bisimulation  $R$  satisfies (3). Conversely, for every relation  $R$  satisfying (3), the relation  $\text{Eq} \cup R$  is an uncertain bisimulation (i.e. we implicitly work with reflexive closure as an up-to technique [3]).

The characterization in Lemma 5.5 leads to the following coinduction principle:

► **Proposition 5.6.** *States  $x, y$  in a partial Mealy machine  $c: C \rightarrow \mathcal{M}C$  are uncertain bisimilar iff*

$$\text{for all } w \in I^+ : \quad \llbracket x \rrbracket(w) \in O \text{ and } \llbracket y \rrbracket(w) \in O \quad \Longrightarrow \quad \llbracket x \rrbracket(w) = \llbracket y \rrbracket(w)$$

Dually,  $x$  and  $y$  are apart iff there is some  $w \in I^+$  for which both are defined but differ:  $? \neq \llbracket x \rrbracket \neq \llbracket y \rrbracket \neq ?$ . Thus, this instance matches the explicit definition of *apart states* in the context of the  $L^\#$  learning algorithm [21, Def. 2.6].

Recall that the final coalgebra semantics of a state  $x \in C$  is a partial map  $\llbracket x \rrbracket: I^+ \rightarrow O$  (in  $\nu\mathcal{M}$ , Proposition 3.2). This map sends each input word  $w \in I^+$  to the output symbol of the last transition of the run of  $w$ , if such a run exists. If not all required transitions exist, then the partial map is undefined (i.e.  $\llbracket x \rrbracket(w) = ?$ ). The characterization in Proposition 5.6 states that two states are uncertain bisimilar if for all input words  $w \in I^+$ , whenever both behaviours  $\llbracket x \rrbracket, \llbracket y \rrbracket$  are defined, then they must agree.

► **Example 5.7.** If there is a natural transformation  $\top: 1 \rightarrow F$  such that  $\top_X$  is the greatest element of  $FX$ , then all states in any  $F$ -coalgebra are uncertain bisimilar.

► **Example 5.8.** For the inclusion order  $\subseteq$  on the finitary powerset functor  $\mathcal{P}_f$ , any two states  $x, y$  in any  $\mathcal{P}_f$ -coalgebra  $c: C \rightarrow \mathcal{P}_f C$  are uncertain bisimilar. Essentially, the issue is that any pair of elements  $s, t \in \mathcal{P}_f C$  has an upper bound in  $(\mathcal{P}_f C, \subseteq)$ .

► **Example 5.9.** We re-obtain ordinary bisimilarity as the instance where the order  $\sqsubseteq$  on  $FX$  is the discrete poset structure:  $\sqsubseteq_{FC} := \text{Eq}_{FC}$ .

The instance for suspension automata has explicitly been studied in the literature [22]:

► **Definition 5.10** [22, Def. 15]. *A relation  $R \subseteq C \times C$  on a suspension automaton  $c: C \rightarrow \mathcal{S}C$  is an (ioco) compatibility relation if for all  $(x, y) \in R$  we have:*

1. *for all  $x \xrightarrow{?a} x'$  and  $y \xrightarrow{?a} y'$  we have  $(x', y') \in R$*
2. *there exists  $o \in O$  such that  $x \xrightarrow{!o} x', y \xrightarrow{!o} y'$ , and  $(x', y') \in R$ .*

► **Proposition 5.11.** *A reflexive relation on a suspension automaton is a ioco compatibility relation iff it is an uncertain bisimulation (for  $\mathcal{S}$  with the order from Definition 3.9).*

In the proof it is relevant that the output transitions of suspension automata are non-empty partial maps  $C \rightarrow (O \xrightarrow{\text{ne}} C)$ . Non-emptiness means that whenever there is a coalgebra structure  $r: R \rightarrow \mathcal{S}R$  on a relation  $R \subseteq C \times C$ , then all related states  $(x, y) \in R$  have at least one common output  $o \in O$ . This is reflected by the existentially quantified condition in the definition of ioco compatibility.

## 5.1 Properties

Having discussed instances, we now uniformly establish general properties of uncertain bisimilarity. We start by listing properties of uncertain relation lifting, analogous to Lemma 4.4.

► **Lemma 5.12.** *For any functor  $F_{\text{Pos}}: \text{Set} \rightarrow \text{Pos}$ , we have the following properties of uncertain relation lifting:*

1. *Monotonicity: if  $R \subseteq S$  then  $\hat{F}_{\sqsubseteq}(R) \subseteq \hat{F}_{\sqsubseteq}(S)$ .*
2. *Preservation of equality:  $\text{Eq} \circ F \subseteq \hat{F}_{\sqsubseteq} \circ \text{Eq}$ .*
3. *Preservation of converse:  $\hat{F}_{\sqsubseteq}(R^{\text{op}}) = (\hat{F}_{\sqsubseteq}(R))^{\text{op}}$  for all  $R \subseteq X \times X$ .*
4. *Preservation of inverse images: For a map  $f: X \rightarrow Y$  and relation  $S \subseteq Y \times Y$ , we have*

$$\hat{F}_{\sqsubseteq}((f \times f)^{-1}(S)) \subseteq (Ff \times Ff)^{-1}(\hat{F}_{\sqsubseteq}(S)).$$

Similar to the case of  $\hat{F}$ , by monotonicity and preservation of inverse images,  $\hat{F}_{\sqsubseteq}$  extends to a lifting of  $F$ . Uncertain bisimilarity is reflexive and symmetric:

► **Lemma 5.13.** *The equality relation  $\text{Eq}_C$  on any coalgebra  $(C, c)$  is an uncertain bisimulation, and if  $R \subseteq C \times C$  is an uncertain bisimulation then so is  $R^{\text{op}}$ .*

Unsurprisingly, uncertain bisimilarity is not transitive: even though two states  $x$  and  $z$  are certainly non-bisimilar (i.e. not uncertain bisimilar), there can be a state  $y$  that is uncertain bisimilar to both  $x$  and  $z$  (e.g.  $p_0, q_0, s_0$  in Figure 1). Similarly, ioco compatibility is known to not be transitive in general [22, Ex. 17].

This lack of transitivity makes it non-trivial to characterize uncertain bisimilarity in terms of being identifiable by a morphism, in the way it holds for normal bisimilarity. Still, we can show some preservation results that match the intuition that the order  $\sqsubseteq$  adds transitions (or other information). Since uncertain bisimilarity of two states means that there is no conflict in their existing transition behaviour, they stay uncertain bisimilar if we omit transitions:

► **Lemma 5.14.** *Uncertain bisimilarity is preserved by oplax morphisms: whenever states  $x, y$  in  $(C, c)$  are uncertain bisimilar, then for every oplax coalgebra morphism  $h: (C, c) \rightarrow (D, d)$ , the states  $h(x)$  and  $h(y)$  are uncertain bisimilar in  $(D, d)$ .*

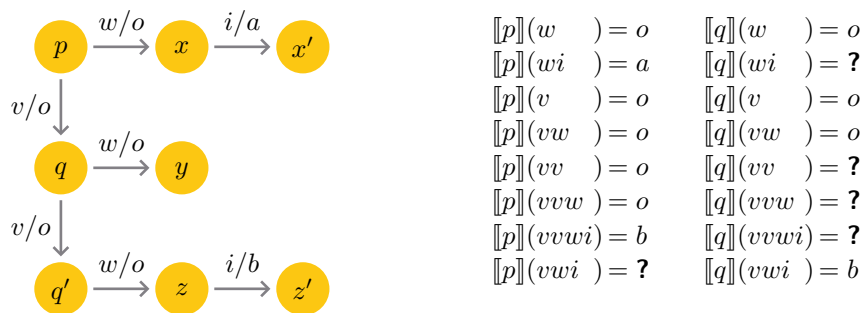
Conversely, we can show that if two states can be identified by a lax coalgebra morphism, then they are uncertain bisimilar. For the corresponding proof for (canonical) relation liftings  $\hat{F}$ , one uses weak pullback preservation as a sufficient condition for preservation of inverse images. For uncertain bisimilarity we will simply make preservation of inverse images an assumption, referred to as *stability*. This terminology follows Hughes and Jacobs [9], who define a similar condition for their lax relation lifting.

► **Definition 5.15.** *The functor  $F_{\text{Pos}}$  is called stable if  $\hat{F}_{\sqsubseteq}$  commutes with inverse images on reflexive relations, i.e. the inclusion in Item 4 of Lemma 5.12 is an equality if  $S$  is reflexive.*

► **Remark 5.16.** Contrary to the variant in [9], we require the converse of Lemma 5.12.4 only for *reflexive* relations. The reason is that even for the case of Mealy machines,  $F = \mathcal{M}$ , the converse of Lemma 5.12.4 does not hold if we drop that assumption.

► **Example 5.17.**  $\hat{\mathcal{M}}_{\sqsubseteq}$  is stable.

► **Lemma 5.18.** *Suppose that  $F_{\text{Pos}}$  is stable. Then any lax coalgebra morphism  $h: (C, c) \dashv\vdash \rightarrow (D, d)$  reflects uncertain bisimilarity, that is, if  $R \subseteq D \times D$  is a reflexive uncertain bisimulation relation then so is  $(h \times h)^{-1}(R)$ .*



■ **Figure 6** Mealy machine of Example 5.20 ... ■ **Figure 7** ... and its semantics.

As a consequence, under the assumption of stability, if states  $x, y$  of a coalgebra are identified by a lax homomorphism  $h$  then they are uncertain bisimilar.

► **Corollary 5.19.** *Suppose that  $F_{\text{Pos}}$  is stable. Then the kernel  $\ker h$  of a lax coalgebra morphism  $h: (C, c) \dashv\vdash (D, d)$  is an uncertain bisimulation.*

This gives half a characterization theorem of uncertain bisimilarity (assuming stability):

$$\begin{array}{l} \text{States } x, y \text{ can be identified} \\ \text{by a lax coalgebra morphism} \end{array} \implies \begin{array}{l} \text{States } x, y \text{ are} \\ \text{uncertain bisimilar} \end{array} \quad (4)$$

For standard bisimilarity, the converse direction also holds: whenever states  $x, y$  are bisimilar (in the ordinary sense), then they can be identified by an (ordinary) coalgebra morphism. For uncertain bisimilarity however, the converse direction even fails when restricting to tree-shaped Mealy machines:

► **Example 5.20.** Consider the partial Mealy machine  $(C, c)$  in Figure 6 for  $I = \{v, w, i\}$  and  $O = \{a, b, o\}$ . In this machine,  $p$  and  $q$  are uncertain bisimilar, because their semantics matches on all defined input words, as verified in Figure 7 (using Proposition 5.6). However, there is no lax coalgebra morphism  $f: (C, c) \dashv\vdash (D, d)$  with  $f(p) = f(q)$ . To see this, first observe that for any such  $f$ , we can derive the following equalities:

$$\begin{aligned} f(p) \xrightarrow{w/o} f(x) \quad \text{and} \quad f(q) \xrightarrow{w/o} f(y) \quad \text{implies} \quad f(x) = f(y), \\ f(p) \xrightarrow{v/o} f(q) \xrightarrow{w/o} f(y) \quad \text{and} \quad f(q) \xrightarrow{v/o} f(q') \xrightarrow{w/o} f(z) \quad \text{implies} \quad f(y) = f(z) \end{aligned}$$

and hence  $f(x) = f(z)$ . By Corollary 5.19 this means  $x$  and  $z$  are uncertain bisimilar; but  $i$  witnesses that  $x$  and  $z$  are apart – a contradiction! Thus, there is no  $f: (C, c) \dashv\vdash (D, d)$  with  $f(p) = f(q)$ .

## 5.2 Characterization via Simulations

We can remedy the failure of the converse direction of (4) by going from functional simulations (i.e. lax coalgebra morphisms) to proper simulations in the sense of spans of (lax) morphisms.

There are multiple ways to define simulations between coalgebras for functors  $F: \mathbf{Set} \rightarrow \mathbf{Set}$  equipped with an order  $\sqsubseteq$ . The way that Hughes and Jacobs [9] define simulations (see also Remark 5.2) between coalgebras  $(C, c)$  and  $(D, d)$  corresponds to a relation  $R \subseteq C \times D$  and a structure  $r: R \rightarrow FR$  making the projections oplax and lax morphisms:

## 12:14 Bisimilar States in Uncertain Structures

$$\begin{array}{l} \pi_1: (R, r) -\exists\rightarrow (C, c) \\ \pi_2: (R, r) -\sqsubseteq\rightarrow (D, d) \end{array} \quad \text{that is, diagrammatically:} \quad \begin{array}{ccccc} C & \xleftarrow{\pi_1} & R & \xrightarrow{\pi_2} & D \\ \downarrow c & \sqsubseteq & \downarrow r & \sqsubseteq & \downarrow d \\ FC & \xleftarrow{F\pi_1} & FR & \xrightarrow{F\pi_2} & FD \end{array}$$

Note that due to using both lax and oplax morphisms, such a simulation is subtly different from the diagram in Lemma 5.4. We can now show that this span-based definition of simulation characterizes uncertain bisimilarity:

► **Proposition 5.21.** *Given that  $F_{\text{Pos}}$  is stable, the following are equivalent for all states  $x, y$  in a coalgebra  $c: C \rightarrow FC$ :*

1.  $x$  and  $y$  are uncertain bisimilar.
2. There is a state  $z \in D$  in another coalgebra  $(D, d)$  and a simulation  $S \subseteq C \times D$  in the style of Hughes and Jacobs such that  $(x, z) \in S$  and  $(y, z) \in S$ .

The second item intuitively means that the states  $x$  and  $y$  can be ‘identified’ by a simulation. We obtained a converse to the implication in (4) when replacing ‘lax coalgebra morphism’ with ‘simulation’. In the proof of the first direction (top to bottom), we use that in sets, every surjective function  $e: X \rightarrow Y$  has a right-inverse  $a: Y \rightarrow X$  (i.e. with  $e \circ a = \text{id}_Y$ ), using the axiom of choice. In the second direction (bottom to top), we use the stability of  $F_{\text{Pos}}$ .

Another slightly different notion of simulation on coalgebras arises from the approach to bisimilarity via open maps [12, 23]. Here, a simulation between  $(C, c)$  and  $(D, d)$  is again a relation  $R \subseteq C \times D$  equipped with a coalgebra structure  $r: R \rightarrow FR$  such that

1. the projection  $\pi_1$  is a coalgebra morphism  $\pi_1: (R, r) \rightarrow (C, c)$ , and
2. the projection  $\pi_2$  is a lax coalgebra morphism  $\pi_2: (R, r) -\sqsubseteq\rightarrow (D, d)$ :

$$\begin{array}{ccccc} C & \xleftarrow{\pi_1} & R & \xrightarrow{\pi_2} & D \\ \downarrow c & \circlearrowleft & \downarrow r & \sqsubseteq & \downarrow d \\ FC & \xleftarrow{F\pi_1} & FR & \xrightarrow{F\pi_2} & FD \end{array}$$

Hence, any open-map-style simulation is also a simulation in the style of Hughes and Jacobs. In our leading examples, the converse inclusion also holds, as we show in the following.

► **Remark 5.22.** The above definition of simulation is reminiscent of Fiore’s *ordered categorical bisimulation* [5, Def. 6.1], for which the partial order comes from the base category being Pos-enriched, i.e. a partial order on every hom set  $\text{hom}(A, B)$  is assumed. In contrast, we only require a partial order on  $FX$ , i.e. the partial order is part of the functor data, not the category.

In order to show the equivalence of open-map-style simulations to that by Hughes and Jacobs, we impose another assumption on the functor:

► **Definition 5.23.** *We call the order  $\sqsubseteq$  on the functor  $F$  restricting if for all maps  $f: X \rightarrow Y$  and all  $s \in FX$ ,  $t \in FY$  we have*

$$t \sqsubseteq_{FY} Ff(s) \quad \Longrightarrow \quad \text{there is some } s' \sqsubseteq s \text{ with } t = Ff(s'). \quad (5)$$

The idea behind  $s'$  is that it is the restriction of  $s$  to those transitions that are defined in  $t$ , so that  $Ff: FX \rightarrow FY$  maps  $s'$  to  $t$ :

► **Example 5.24.** The functor  $\mathcal{M}$  for partial Mealy machines is restricting: for  $f: X \rightarrow Y$ ,  $s \in \mathcal{M}X$ , and  $t \sqsubseteq_{\mathcal{M}Y} \mathcal{M}f(s)$ , define

$$s' \in \mathcal{M}X = (\{?\} + O \times X)^I \quad \text{by} \quad s'(i) = \begin{cases} ? & \text{if } t(i) = ? \\ s(i) & \text{otherwise.} \end{cases}$$

This definition makes  $s' \sqsubseteq s$  true because for all  $i \in I$ , whenever  $s'(i)$  is defined (i.e.  $s'(i) \neq ?$ ) then  $s(i)$  is defined, too. The inequality  $t \sqsubseteq_{\mathcal{M}Y} \mathcal{M}f(s)$  implies

$$(s'(i) = ? \iff t(i) = ?) \quad \text{for all } i \in I$$

and moreover, whenever  $s'(i) = (o, x)$  for  $i \in I$ , then  $t(i) = s(i) = s'(i)$ . Hence,  $\mathcal{M}f(s') = t$ .

► **Remark 5.25.** In the definition of *restricting*, we have  $t \sqsubseteq Ff(s)$  as the condition and then construct some  $s'$  with  $s' \sqsubseteq s$ . Thus, one might be tempted to think that there is a Galois connection hidden. Note however, that this is not the case in the example of partial Mealy machines because the construction of  $s'$  does depend on  $s!$  Hence, it is not possible to construct an adjoint map  $FY \rightarrow FX$  in general.

► **Lemma 5.26.** *If  $F$  is restricting, then for every oplax morphism  $h: (C, c) \multimap (D, d)$ , there is a structure  $c': C \rightarrow FC$  such that  $c'(x) \sqsubseteq c(x)$  for all  $x \in C$  making  $h$  a (proper) coalgebra morphism.*

$$\begin{array}{ccc} C & \xrightarrow{h} & D \\ \downarrow c & \sqsubseteq & \downarrow d \\ FC & \xrightarrow{Fh} & FD \end{array} \implies \exists c': \begin{array}{ccc} C & \xrightarrow{h} & D \\ c(\downarrow) c' \circlearrowleft & & \downarrow d \\ FC & \xrightarrow{Fh} & FD \end{array}$$

This lemma turns Hughes/Jacobs simulations into simulations in the style of open maps:

► **Lemma 5.27.** *Given that  $\sqsubseteq$  is restricting, the following are equivalent for any relation  $S \subseteq C \times D$  on coalgebras  $(C, c), (D, d)$ :*

1. *there is a map  $S \rightarrow FS$  making  $S$  a simulation in the style of Hughes and Jacobs.*
2. *there is a map  $S \rightarrow FS$  making  $S$  a simulation in the style used in open maps.*

Thus, we can combine Lemma 5.27 and the previous characterization Proposition 5.21:

► **Theorem 5.28.** *Given that  $F_{\text{Pos}}$  is stable and that  $\sqsubseteq$  is restricting, the following are equivalent for all states  $x, y$  in a coalgebra  $c: C \rightarrow FC$ :*

1.  *$x$  and  $y$  are uncertain bisimilar.*
2. *There is a state  $z \in D$  in another coalgebra  $(D, d)$  and a simulation  $S \subseteq C \times D$  in the style of Hughes and Jacobs such that  $(x, z) \in S$  and  $(y, z) \in S$ .*
3. *There is a state  $z \in D$  in another coalgebra  $(D, d)$  and an open-map-style simulation  $S \subseteq C \times D$  such that  $(x, z) \in S$  and  $(y, z) \in S$ .*

► **Example 5.29.** For partial Mealy machines, the abstract definitions of simulation instantiate to the usual notion of simulation between  $(C, c)$  and  $(D, d)$  when considering the Mealy machines as deterministic LTSs for the alphabet  $I \times O$ : a simulation is a relation  $R \subseteq C \times D$  such that for all  $(x, z) \in R$  and  $x \xrightarrow{i/o} x'$  there is some  $z' \in D$  such that  $z \xrightarrow{i/o} z'$  and  $(x', z') \in R$ . The characterization in Proposition 5.21 shows that for all states  $x, y \in C$  in  $c: C \rightarrow \mathcal{M}C$  we have

$$\begin{array}{ccc} x \text{ and } y \text{ are} & & \text{There is a state } z \text{ in some } d: D \rightarrow \mathcal{M}D \\ \text{uncertain bisimilar} & \iff & \text{such that } z \text{ simulates } x \text{ and } y \end{array}$$

► **Example 5.30.** For the compatibility relation on suspension automata, a similar equivalence holds. In the specific simulation notion (called *coinductive ioco relation* [22, Def. 4]), the input transitions are preserved in the usual direction and the output transitions are preserved in the converse direction. Then, it is shown that states  $x, y$  in a suspension automaton are compatible iff there is a state  $z$  in another suspension automaton which conforms (according to the *ioco* relation) to both  $x$  and  $y$  [22, Lem. 24.2].

## 6 Conclusions and Future Work

We introduced *uncertain bisimilarity*, a notion to talk about behavioural compatibility on a coalgebraic level of generality. Instances include both partial Mealy machines and the *ioco* conformance relation from model-based testing. The setting is tailored towards the lack of knowledge in automata learning games. We are optimistic that this generalization provides a step from the  $L^\#$  learning algorithm [21] towards new coalgebra learning algorithms. While previous categorical frameworks [2, 20, 7, 4] generalize Angluin’s classical  $L^*$  algorithm, the development of a variant of  $L^\#$  at a high level of generality could be useful, as the experiments [21] point to a better performance in the case of Mealy machines.

So far, we have shown that uncertain bisimilarity is equivalent to being simulated by a common state. A similar observation might be lifted to the final coalgebra by defining a suitable simulation order on the final coalgebra. In this context it would be interesting to explicitly connect our results to the similarity quotients in [15].

Standard coalgebraic bisimilarity can also be characterized as indistinguishability via formulas of coalgebraic modal logic [17, 13]. We are confident that uncertain bisimilarity can be characterized in similar terms. Since it is a coinductively defined relation involving a non-standard relation lifting, a good starting point may be the framework in [14], although that does not provide a canonical construction of a logic but only the infrastructure for proving expressiveness and adequacy. To obtain such a logic, it should be feasible to transfer modalities from an existing system type functor (e.g.  $\mathcal{M}_T$ ) to the functors involving order and partial behaviours (e.g.  $\mathcal{M}$ ), such that properties like adequacy or even expressiveness are inherited. These distinguishing modal formulas can then serve as witnesses for disproving uncertain bisimilarity, that is, for showing apartness.

The relation lifting based definition of uncertain bisimilarity makes it amenable to the use of coalgebraic up-to techniques as developed in [3]. This could be helpful in the development of efficient algorithms for checking uncertain bisimilarity, which could be particularly interesting to check compatibility of *ioco* specifications as studied in [22].

---

### References

- 1 Dana Angluin. Learning regular sets from queries and counterexamples. *Inf. Comput.*, 75(2):87–106, 1987.
- 2 Simone Barlocco, Clemens Kupke, and Jurriaan Rot. Coalgebra learning via duality. In *FoSSaCS*, volume 11425 of *Lecture Notes in Computer Science*, pages 62–79. Springer, 2019.
- 3 Filippo Bonchi, Daniela Petrisan, Damien Pous, and Jurriaan Rot. A general account of coinduction up-to. *Acta Informatica*, 54(2):127–190, 2017.
- 4 Thomas Colcombet, Daniela Petrisan, and Riccardo Stabile. Learning automata and transducers: A categorical approach. In *CSL*, volume 183 of *LIPICs*, pages 15:1–15:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.
- 5 Marcelo P. Fiore. A coinduction principle for recursive data types based on bisimulation. *Inf. Comput.*, 127(2):186–198, 1996. doi:10.1006/inco.1996.0058.
- 6 Herman Geuvers and Bart Jacobs. Relating apartness and bisimulation. *Logical Methods in Computer Science*, Volume 17, Issue 3, July 2021. doi:10.46298/lmcs-17(3:15)2021.
- 7 Gerco van Heerdt. *CALF: Categorical Automata Learning Framework*. Phd thesis, University College London, October 2020.
- 8 Claudio Hermida and Bart Jacobs. Structural induction and coinduction in a fibrational setting. *Inf. Comput.*, 145(2):107–152, 1998.
- 9 Jesse Hughes and Bart Jacobs. Simulations in coalgebra. *Theor. Comput. Sci.*, 327(1-2):71–108, 2004.

- 10 Bart Jacobs. *Categorical Logic and Type Theory*. Number 141 in Studies in Logic and the Foundations of Mathematics. North Holland, Amsterdam, 1999.
- 11 Bart Jacobs. *Introduction to Coalgebra: Towards Mathematics of States and Observation*, volume 59 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2016.
- 12 André Joyal, Mogens Nielsen, and Glynn Winskel. Bisimulation from Open Maps. *Information and Computation*, 127:164–185, 1996. doi:10.1006/inco.1996.0057.
- 13 Clemens Kupke and Dirk Pattinson. Coalgebraic semantics of modal logics: An overview. *Theor. Comput. Sci.*, 412(38):5070–5094, 2011.
- 14 Clemens Kupke and Jurriaan Rot. Expressive logics for coinductive predicates. *Log. Methods Comput. Sci.*, 17(4), 2021.
- 15 Paul Blain Levy. Similarity quotients as final coalgebras. In *FoSSaCS*, volume 6604 of *Lecture Notes in Computer Science*, pages 27–41. Springer, 2011.
- 16 Davide Sangiorgi. *An introduction to Bisimulation and Coinduction*. Cambridge University Press, 2012.
- 17 Lutz Schröder. Expressivity of coalgebraic modal logic: The limits and beyond. *Theor. Comput. Sci.*, 390(2-3):230–247, 2008.
- 18 Sam Staton. Relating coalgebraic notions of bisimulation. *Log. Methods Comput. Sci.*, 7(1), 2011.
- 19 Jan Tretmans. Model based testing with labelled transition systems. In Robert M. Hierons, Jonathan P. Bowen, and Mark Harman, editors, *Formal Methods and Testing, An Outcome of the FORTEST Network, Revised Selected Papers*, volume 4949 of *Lecture Notes in Computer Science*, pages 1–38. Springer, 2008. doi:10.1007/978-3-540-78917-8\_1.
- 20 Henning Urbat and Lutz Schröder. Automata learning: An algebraic approach. In *LICS*, pages 900–914. ACM, 2020.
- 21 Frits Vaandrager, Bharat Garhewal, Jurriaan Rot, and Thorsten Wißmann. A new approach for active automata learning based on apartness. In *Tools and Algorithms for the Construction and Analysis of Systems - 28th International Conference, TACAS 2022*, Lecture Notes in Computer Science. Springer, April 2022.
- 22 Petra van den Bos, Ramon Janssen, and Joshua Moerman. n-complete test suites for IOCO. *Softw. Qual. J.*, 27(2):563–588, 2019. doi:10.1007/s11219-018-9422-x.
- 23 Thorsten Wißmann, Jérémy Dubut, Shin-ya Katsumata, and Ichiro Hasuo. Path category for free. In Mikołaj Bojańczyk and Alex Simpson, editors, *FoSSaCS 2019*, pages 523–540, Cham, April 2019. Springer International Publishing. doi:10.1007/978-3-030-17127-8\_30.





# A Category for Unifying Gaussian Probability and Nondeterminism

Dario Stein ✉

iHub, Radboud University Nijmegen, The Netherlands

Richard Samuelson ✉

Humming Inc., Tacoma, WA, USA

---

## Abstract

We introduce categories of extended Gaussian maps and Gaussian relations which unify Gaussian probability distributions with relational nondeterminism in the form of linear relations. Both have crucial and well-understood applications in statistics, engineering, and control theory, but combining them in a single formalism is challenging. It enables us to rigorously describe a variety of phenomena like noisy physical laws, Willems' theory of open systems and uninformative priors in Bayesian statistics. The core idea is to formally admit vector subspaces  $D \subseteq X$  as generalized uniform probability distribution. Our formalism represents a first bridge between the literature on categorical systems theory (signal-flow diagrams, linear relations, hypergraph categories) and notions of probability theory.

**2012 ACM Subject Classification** Theory of computation → Probabilistic computation; Theory of computation → Categorical semantics; Mathematics of computing → Probability and statistics

**Keywords and phrases** systems theory, hypergraph categories, Bayesian inference, category theory, Markov categories

**Digital Object Identifier** 10.4230/LIPIcs.CALCO.2023.13

**Related Version** *Previous Version*: <https://arxiv.org/abs/2204.14024v1> [32]

**Acknowledgements** It has been useful to discuss this work with many people. Particular thanks go to Tobias Fritz, Bart Jacobs, Dusko Pavlovic, Sam Staton and Alexander Terenin.

## 1 Introduction

Modelling the behavior of systems under uncertainty is of crucial importance in engineering and computer science. We can distinguish two different kinds of uncertainty:

- Probabilistic uncertainty means we may not know the exact value of some quantity, like a measurement error, but we do know the statistical distribution of such errors. A typical such distribution is the *normal (Gaussian) distribution*  $\mathcal{N}(\mu, \sigma^2)$  of mean  $\mu$  and variance  $\sigma^2$ .
- Nondeterministic uncertainty models complete ignorance of a quantity. We know which values the quantity may feasibly assume but have no statistical information beyond that. Nondeterministic uncertainty can be modelled using subsets  $R \subseteq X$  which identify the feasible values. In practice, such subsets are often characterized by equational constraints such as natural laws.

Systems may be subject to both probabilistic and nondeterministic constraints, but describing such systems mathematically is more challenging. A classical treatment is Willems' theory of *open stochastic systems* [38, 37], where “openness” in his terminology refers to nondeterminism or lack of information. We recall a simple example:



© Dario Stein and Richard Samuelson;

licensed under Creative Commons License CC-BY 4.0

10th Conference on Algebra and Coalgebra in Computer Science (CALCO 2023).

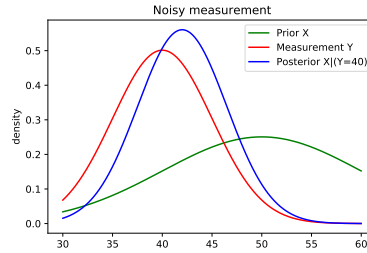
Editors: Paolo Baldan and Valeria de Paiva; Article No. 13; pp. 13:1–13:18

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

## 13:2 Combining Gaussian Probability with Nondeterminism



■ **Figure 1** Gaussian prior and posterior in a noisy measurement example.

► **Example 1 (Noisy resistor).** For a resistor of resistance  $R$ , Ohm’s law constrains pairs  $(V, I)$  of voltage and current to lie in the subspace  $D = \{(V, I) : V = RI\}$ . This is a relational constraint – values must lie in  $D$ , but we have no further statistical information about which values the system takes. In a realistic system, thermal noise is always present; such a noisy system is better modelled by the equation

$$V = RI + \epsilon \tag{1}$$

where  $\epsilon \sim \mathcal{N}(0, \sigma^2)$  is a Gaussian random variable with some small variance  $\sigma^2$ . Willems notices that the variables  $V, I$  are not random variables in the usual sense; we have not associated any distribution to them. On the other hand, the quantity  $V - RI$  is a honest random variable. Furthermore, if we supply a fixed voltage  $V_0$ , we can solve for  $I$  and

$$I = R^{-1}(V_0 - \epsilon) \tag{2}$$

becomes a classical (Gaussian) random variable. Willems calls this “interconnection” of systems.

Willems models the “openness” of the stochastic systems by endowing the outcome space  $\mathbb{R}^2$  with an unusually coarse  $\sigma$ -algebra  $\mathcal{E}$  to formalize the lack of information. Measurable sets are restricted to the form  $\{(V, I) : V - RI \in A\}$  for  $A \subseteq \mathbb{R}$  Borel. The Gaussian probability measure is then only defined on  $\mathcal{E}$ , which essentially makes it a measure on the quotient space  $\mathbb{R}^2/D$ . We purely formally define an *extended Gaussian distributions* on a space  $X$  as a pair  $(D, \psi)$  of a subspace  $D$  and a Gaussian distribution on  $X/D$ . In particular, we can think of any subspace  $D$  as an extended Gaussian distribution  $(D, 0)$ . Operationally, sampling a point  $x \sim D$  means picking it nondeterministically from  $D$ . Every extended Gaussian distribution can be seen as a formal sum  $\psi + D$  of a Gaussian contribution  $\psi$  and a nondeterministic contribution  $D$ .

In our approach, the noisy resistor is described by a single extended Gaussian distribution where  $D$  is the subspace for Ohm’s law, and  $\psi$  is Gaussian noise in a direction orthogonal to  $D$ . The marginals  $V, I$  are themselves extended Gaussian distributions: we find that  $V \sim \mathbb{R}$  and  $I \sim \mathbb{R}$ , that is they are picked nondeterministically from the real line, so in this sense we have no information about them. We also find that  $V - RI \sim \mathcal{N}(0, \sigma^2)$  follows a classical Gaussian distribution without any nondeterministic contribution. The interconnection (2) is obtained as an instance of probabilistic conditioning  $V = V_0$ . We compare our approach to the one of Willems in Section 5.1.

We now describe a completely different situation where it makes sense to admit subspaces as idealized probability distributions, namely uninformative priors in Bayesian inference:

► **Example 2** (Uninformative Priors). Our prior experience tell us that we expect the mass  $X$  of some object to be normally distributed with mean 50 and variance 100. We use a noisy scale to obtain a measurement of  $Y = 40$ . If the scale error has variance of 25, we can compute our *posterior* belief over  $X$ , which turns out to be <sup>1</sup>  $X|(Y = 40) \sim \mathcal{N}(42, 20)$ . Here, the influence of the prior has corrected the predicted value to lie slightly above the measured value, and have smaller overall variance (see Figure 1).

If we had no prior information at all about  $X$ , the posterior should simply reflect the measurement uncertainty  $\mathcal{N}(40, 25)$ . We can model this by putting a larger and larger variance on  $X$ . However, the limit of distributions  $\mathcal{N}(50, \sigma^2)$  for  $\sigma^2 \rightarrow \infty$  does not exist in any measure-theoretic sense, because it would approach the zero measure on every measurable set. There exists no uniform probability distribution over the real line. In practice, one can sometimes pretend (using the method of *improper priors*, e.g. [19, 22]) that  $X$  is sampled from the Lebesgue measure  $\lambda$  (with constant density 1). This measure fails to be normalized, however the resulting density calculations may yield the correct probability measures.

Our theory of extended Gaussians avoids unnormalized measures altogether: The non-deterministic distribution  $X \sim \mathbb{R}$  is used as the uninformative prior on  $X$ , which gives the desired results, and  $\mathbb{R}$  can be seen as the limit of  $\mathcal{N}(50, \sigma^2)$  for  $\sigma^2 \rightarrow \infty$  in an appropriate sense.

## 1.1 Contribution

The paper is devoted to making our manipulations of subspaces as generalized probability distributions rigorous. We introduce a class of mathematical objects called *extended Gaussian distributions* and show that such distributions can be manipulated (combined, pushed forward, marginalized) as if they were ordinary probability distributions. Importantly, extended Gaussians remain closed under taking conditional distributions, which means we can use them in applications such as statistical learning and Kalman filtering. The subspace  $\mathbb{R}$ , seen as a uniform distribution, formalizes the role of an improper prior.

Describing distributions on a space  $X$  is only the first step. In order to build up systems in a compositional way, we need to understand transformations between spaces  $X \rightarrow Y$ . Category theory is a widely used language to study the composition of different kinds of systems. We identify two relevant flavors in the literature

- categorical and diagrammatic methods for engineering and control theory, such as graphical linear algebra (e.g. [28]), cartesian bicategories (e.g. [6]) and signal-flow diagrams ([4, 3, 5, 2, 1]). A central notion is that of a *hypergraph category* [13], and prototypical models are the categories of linear maps or linear relations. Willems' system theory has been explored in these terms [12], but probability is absent from these developments.
- categorical models of probability, such as copy-delete categories [7] and Markov categories [15, 17, 16, 18]. Prototypical models are stochastic matrices, or the category **Gauss** of affine-linear maps with Gaussian noise.

Despite these developments, it has been challenging to combine probability and nondeterminism into a single model – mathematical obstructions to achieving this are described in [39, 21]. Our work is a first successful step in combining these bodies of literature: We define a category **GaussEx** of extended Gaussian maps which can be seen both as extending linear relations with

---

<sup>1</sup> see appendix 6.2 for the calculation

## 13:4 Combining Gaussian Probability with Nondeterminism

probability, or extending Gaussian probability with nondeterminism (or improper priors). Gaussian probability is a very expressive fragment of probability with a variety of useful applications (Gaussian processes, Kalman filters, Bayesian Linear Regression).

Our definition of the Markov category `GaussEx` uses a special case of the widely studied construction of *decorated cospans* [9, 11, 14, 12]. We recall that `GaussEx` has conditionals, which is the categorical formulation of conditional probability used to formalize inference problems like Example 2.

We then define a hypergraph category of *Gaussian relations*, which allows arbitrary decorated cospans to allow the possibility of failure and explicit conditioning in the categorical structure. Hypergraph categories are highly symmetrical categories with an appealing duality theory. To our knowledge, probabilistic models of hypergraph categories are novel. The self-duality of hypergraph categories is reflected in the duality between covariance and precision forms, which takes a particularly canonical form for extended Gaussians. We elaborate this in Section 5.2.

The following table summarizes the relationships between our constructions:

		(adding Gaussian noise)
(adding nondeterminism)	linear maps	Gaussian maps
(adding failure)	total linear relations	extended Gaussian maps
	linear relations	Gaussian Relations

### 1.2 Outline

We assume basic familiarity of the reader with linear algebra, (monoidal) categories and string diagrams; an overview can be found in the appendix Section 6. All categories considered will be symmetric monoidal and have a copy-delete structure. All vector spaces are assumed finite dimensional.

We begin Section 2 with a recap of Gaussian probability and continue to define extended Gaussian distributions as Gaussian distributions on quotient spaces. We extend this definition to a notion of *extended Gaussian map* in Section 3 and establish the structure of a Markov category. We give the construction both in elementary terms and using the formalism of decorated cospans in Section 3.2.

In Section 4, we define a hypergraph category of Gaussian relations, which extends extended Gaussian maps with the possibility of failure and conditioning. This makes use of the discussion of conditionals in Section 4.1.

The idea of extended Gaussian distributions has appeared in several places independently, for different motivations. We conclude the paper with an extended “Related Works” Section 5, which compares these approaches in detail, and gives perspective in terms of measure theory, topology, and program semantics.

## 2 Extended Gaussian Distributions

We begin with a short review of Gaussian probability; we assume basic concepts of linear algebra but have summarize the terminology in the appendix (Section 6.3). For a more detailed introduction to Gaussian probability see e.g. [35, 24].

The normal distribution or Gaussian distribution  $\mathcal{N}(\mu, \sigma^2)$  of mean  $\mu$  and variance  $\sigma^2$  is defined by having the density function

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

with respect to the Lebesgue measure. This is generalized to multivariate normal distributions as follows: Every *Gaussian distribution* on  $\mathbb{R}^n$  can be written uniquely as  $\mathcal{N}(\mu, \Sigma)$  where  $\mu \in \mathbb{R}^n$  is its *mean* and  $\Sigma \in \mathbb{R}^{n \times n}$  is a symmetric positive-semidefinite matrix called its *covariance matrix*. Note that a vanishing covariance matrix is explicitly allowed; in that case the Gaussian reduces to a point-mass  $\delta_x = \mathcal{N}(x, 0)$ . We will sometimes abbreviate the point-mass  $\delta_x$  by  $x$  if the context is clear.

We write  $\text{Gauss}(\mathbb{R}^n)$  for the set of all Gaussian distributions on  $\mathbb{R}^n$ . The *support* of  $\mathcal{N}(\mu, \Sigma)$  is the affine subspace  $\mu + \text{col}(\Sigma)$  where  $\text{col}(\Sigma)$  is the column space (image) of  $\Sigma$ . Gaussian distributions transform as follows under linear maps: If  $A \in \mathbb{R}^{m \times n}$  is a matrix, then the pushforward distribution is given by

$$A_*(\mathcal{N}(\mu, \Sigma)) = \mathcal{N}(A\mu, A\Sigma A^T) \quad (3)$$

Product distributions are formed as follows

$$\mathcal{N}(\mu, \Sigma) \otimes \mathcal{N}(\mu', \Sigma') = \mathcal{N}\left(\begin{pmatrix} \mu \\ \mu' \end{pmatrix}, \begin{pmatrix} \Sigma & 0 \\ 0 & \Sigma' \end{pmatrix}\right) \quad (4)$$

We write addition  $+$  between distributions to indicate the distribution of the sum of two independent variables (convolution). For example, if  $X, Y \sim \mathcal{N}(0, 1)$  are independent, then  $X + Y \sim \mathcal{N}(0, 2)$  because variance is additive for independent variables. We have

$$\mathcal{N}(\mu, \Sigma) + \mathcal{N}(\mu', \Sigma') = \mathcal{N}(\mu + \mu', \Sigma + \Sigma')$$

which can be confirmed by first forming the product distribution (4) and pushing forward under the addition map (3). The set  $\text{Gauss}(\mathbb{R}^n)$  forms a commutative monoid with convolution  $+$  and neutral element 0.

We now wish to combine Gaussian distributions on  $\mathbb{R}^n$  with uninformative (nondeterministic) distributions along a vector subspace  $D$ .

► **Definition 3.** *An extended Gaussian distribution on  $\mathbb{R}^n$  is a pair  $(D, \psi)$  of a subspace  $D \subseteq \mathbb{R}^n$  and a Gaussian distribution  $\mu$  on the quotient  $\mathbb{R}^n/D$ . Following [38], we call the space  $D$  the (nondeterministic) fibre of the extended Gaussian. We write  $\text{GaussEx}(\mathbb{R}^n)$  for the set of all extended Gaussian distributions on  $\mathbb{R}^n$ .*

There are several equivalent ways to formalize the notion of a Gaussian distribution over this quotient space.

1. We identify the quotient space  $\mathbb{R}^n/D$  with a complementary subspace  $K$  of  $D$ , and give a Gaussian distribution on that space. This has the advantage of only involving Euclidean spaces, and we can use matrices to represent linear maps.
2. We develop a coordinate-free definition of Gaussian distributions on arbitrary vector spaces  $X$  so we can then interpret the construction  $\text{Gauss}(\mathbb{R}^n/D)$  directly. This will be useful for the duality results in Section 5.2.
3. Willems keeps the spaces  $\mathbb{R}^n$  but equips them with restricted  $\sigma$ -algebras. This corresponds to a quotient on the level of measurable spaces. We discard this perspective for now but will return to it in Section 5.1.

For now, it doesn't matter which formalization we choose. We will build intuitions with some examples:

## 13:6 Combining Gaussian Probability with Nondeterminism

1. Every Gaussian distribution  $\psi$  becomes an extended Gaussian distribution with  $D = 0$ ; that is the nondeterministic contribution vanishes (is constantly zero).
2. Every subspace  $D$  becomes an extended Gaussian distribution with  $\psi = 0$ ; that is the probabilistic contribution vanishes. By slight abuse of notation, we will simply write  $D$  or  $\psi$  for the embedding of subspaces or distributions into extended Gaussian distributions.
3. If the nondeterministic fibre  $D = \mathbb{R}^n$  is the whole space, then  $\text{Gauss}(\mathbb{R}^n/D) = \{0\}$ . Hence, the only extended Gaussian with fibre  $D$  is the subspace  $D$  itself. This distribution expresses total ignorance.
4. We can easily classify all extended Gaussian distributions on  $\mathbb{R}$ . The fibre  $D$  must be either  $0$  or  $\mathbb{R}$ , so we have  $\text{GaussEx}(\mathbb{R}) = \text{Gauss}(\mathbb{R}) \dot{\cup} \{\mathbb{R}\}$ .
5. The possible pairs  $(V, I)$  satisfying Ohm's law are given by the subspace  $D = \{(V, I) : V = RI\}$ . For noisy Ohm's law, we let  $\epsilon \sim \mathcal{N}(0, \sigma^2)$  and notice that the random vector  $w = (-1, R) \cdot \epsilon$  is orthogonal to  $D$ . Its covariance matrix is

$$\Sigma_w = \begin{pmatrix} 1 & -R \\ -R & R^2 \end{pmatrix}$$

and thus the distribution of the noisy law is given by  $(\mathcal{N}(0, \Sigma_w), D)$ .

We may think of the extended distribution  $(D, \psi)$  as being composed of nondeterministic noise along the space  $D$ , and Gaussian noise  $\psi$ . It is evocative to write the extended Gaussian distribution as a formal sum  $\psi + D$  of distribution and a subspace. The distribution  $\psi$  is not unique because the nondeterministic noise absorbs components of  $\psi$  that are parallel to  $D$ . This is analogous to how we use notation like  $3 + 2\mathbb{Z}$  for elements of quotient groups (cosets). This notation is formally justified by the formula for addition of extended Gaussians, as discussed next.

### 2.1 Transformations of Extended Gaussians

Extended Gaussian distributions support the same basic transformations as ordinary Gaussians. If  $A$  is a matrix, we push forward the Gaussian and nondeterministic contribution separately,

$$A_*(\psi + D) = A_*\psi + A[D]$$

where  $A[D] = \{Ax : x \in D\}$  denotes the image subspace. Tensor and sum are similarly component-wise

$$(\mu + D) \otimes (\psi + E) = (\mu \otimes \nu) + (D \times E) \quad (\mu + D) + (\nu + E) = (\mu + \nu) + (D + E)$$

Well-definedness is a corollary of the next section, because those operations are special cases of the categorical structure of  $\text{GaussEx}$ .

► **Example 4.** The subspace  $\mathbb{R} \in \text{GaussEx}(\mathbb{R})$  absorbs all additive contributions, e.g.  $42 + \mathbb{R} = \mathcal{N}(0, 1) + \mathbb{R} = \mathbb{R}$

## 3 A Category of Extended Gaussian maps

After defining extended Gaussians on Euclidean spaces  $X$ , the next challenge is to develop a notion of *extended Gaussian map*  $X \rightarrow Y$  between spaces. We wish to define a category  $\text{GaussEx}$  such that we recover extended Gaussian distributions as maps out of the unit space  $0$ , i.e.  $\text{GaussEx}(X) \cong \text{GaussEx}(0, X)$ . The operations of pushforward, product and sum of

distributions will be simple instances of categorical and monoidal composition in the category  $\mathbf{GaussEx}$ . For purely Gaussian probability, the appropriate definition of a map is a linear function together with Gaussian noise, informally written  $f(x) = Ax + \mathcal{N}(b, \Sigma)$ . We begin by analyzing this construction before generalizing it to the extended Gaussian case.

### 3.1 Decorated Linear Maps and the Category Gauss

We write  $\mathbf{Vec}$  for the category of finite dimensional vector spaces. The category  $\mathbf{Gauss}$  [15] is defined as follows: Objects are vector spaces  $X$ , and morphisms  $X \rightarrow Y$  are pairs  $(f, \psi)$  of a linear map  $f : X \rightarrow Y$  and a Gaussian distribution  $\psi \in \mathbf{Gauss}(Y)$ . The identity is given by  $(\text{id}_X, 0)$  and composition is given by pushing forward and addition of the noise,  $(f, \xi) \circ (g, \psi) = (fg, \xi + f_*\psi)$ .

It is straightforward to generalize the pattern of this construction: The set of distributions  $\mathbf{Gauss}(X)$  is a commutative monoid  $(\mathbf{Gauss}(X), +, 0)$  and the assignment  $X \mapsto \mathbf{Gauss}(X)$  becomes a lax monoidal functor  $\mathbf{Gauss} : \mathbf{Vec} \rightarrow \mathbf{CMon}$  from vector spaces to commutative monoids. By understanding a commutative monoid as a one-object category, the functor  $\mathbf{Gauss} : \mathbf{Vec} \rightarrow \mathbf{Cat}$  is an indexed category, and the category  $\mathbf{Gauss}$  is the monoidal op-Grothendieck construction associated to this functor [26].

We do not use any special properties of Gaussian distributions, other than that they can be added and pushed forward. In other words, can think of the distribution  $\psi$  as a purely abstract decoration on the codomain of the linear map  $f$ . Any functor  $S : \mathbf{Vec} \rightarrow \mathbf{CMon}$  can be used to supply such a decoration, because it automatically inherits a lax monoidal structure (see below). In concrete terms, the op-Grothendieck construction can be described as decorated linear maps:

► **Definition 5.** *Let  $S : \mathbf{Vec} \rightarrow \mathbf{CMon}$  be a functor. The category  $\mathbf{Lin}_S$  of  $S$ -decorated linear maps is defined as follows*

1. *Objects are vector spaces  $X$*
2. *Morphisms are pairs  $(f, s)$  where  $f : X \rightarrow Y$  is a linear map and  $s \in S(Y)$*
3. *Composition is defined as follows: for  $g : X \rightarrow Y$ ,  $f : Y \rightarrow Z$ ,  $s \in S(Y)$ ,  $t \in S(Z)$  let*

$$(f, t) \circ (g, s) = (fg, t + S(f)(s))$$

*Note that addition takes place in the commutative monoid  $S(Z)$ .*

There is a faithful inclusion  $\mathbf{Vec} \rightarrow \mathbf{Lin}_S$  sending  $f$  to  $(f, 0)$ . We argue that  $\mathbf{Lin}_S$  has the structure of a symmetric monoidal category with the tensor  $X \otimes Y = X \times Y$  on objects. For this, we first observe that  $S$  is automatically lax monoidal: For  $(s, t) \in S(X) \times S(Y)$ , let  $s \oplus t = S(i_X)(s) + S(i_Y)(t)$  where  $i_X : X \rightarrow X \times Y$ ,  $i_Y : Y \rightarrow X \times Y$  are the biproduct inclusions. We can now define the tensor of decorated map as  $(f, s) \otimes (g, t) = (f \times g, s \oplus t)$ . The monoidal category  $\mathbf{Lin}_S$  is in general not cartesian; it does however inherit copy and delete maps from  $\mathbf{Vec}$ . The category  $\mathbf{Lin}_S$  is a Markov category if and only if deleting is natural, i.e.  $S(0) \cong 0$ , where  $0$  denotes the terminal vector space/commutative monoid.

► **Example 6.** The following categories are instances of decorated linear maps:

1. For  $S(X) = 0$ ,  $\mathbf{Lin}_S$  is equivalent to  $\mathbf{Vec}$ .
2. For  $S(X) = X$ ,  $\mathbf{Lin}_S$  is equivalent to the category of affine-linear maps. A map  $X \rightarrow Y$  consists of a pair  $(f, y)$  with  $f : X \rightarrow Y$  linear and  $y \in Y$ .
3. For  $S(X) = \mathbf{Gauss}(X)$ ,  $\mathbf{Lin}_S$  is (by construction) the category  $\mathbf{Gauss}$

### 3.2 Decorated Cospans and Linear Relations

Like for Gauss, we wish to define an extended Gaussian map as a linear map with extended Gaussian noise. The naive approach of considering linear maps decorated by  $S = \mathbf{GaussEx}$  is not fruitful, because the quotient by the nondeterministic fibre is not properly taken into account: For example, for any two linear maps  $f, g : \mathbb{R}^n \rightarrow \mathbb{R}$ , the decorated maps  $f + \mathbb{R}$  and  $g + \mathbb{R}$  should be considered equal (Example 4). We can remedy this by considering maps into the quotient  $X \rightarrow \mathbb{R}/\mathbb{R}$ . This kind of behavior is precisely captured by (total) linear relations.

► **Lemma 7** (Section 6.3). *To give a total linear relation  $R \subseteq X \times Y$  is to give a subspace  $D \subseteq Y$  and a linear map  $X \rightarrow Y/D$ .*

► **Definition 8.** *An extended Gaussian map  $X \rightarrow Y$  is a tuple  $(D, f, \psi)$  where  $D \subseteq Y$  is a subspace,  $f : X \rightarrow Y/D$  and  $\psi \in \mathbf{Gauss}(Y/D)$ .*

In order to describe composition of such maps, it is convenient to use the formalism of decorated cospans, which we recall now:

A cospan in a category  $\mathbb{C}$  with finite colimits is a diagram of the form  $X \xrightarrow{f} P \xleftarrow{g} Y$ . We will identify two cospans if there exists an isomorphism  $P \cong P'$  commuting with the legs. Equivalence classes of cospans can be seen as morphisms between  $X$  and  $Y$  in a category  $\mathbf{Cospan}(\mathbb{C})$ , where composition is given by pushout

$$\begin{array}{ccc}
 & X & \\
 \text{id}_X \nearrow & & \nwarrow \text{id}_X \\
 X & & X
 \end{array}
 \qquad
 \begin{array}{ccccc}
 & & W & & \\
 & & \downarrow & & \\
 & P & & Q & \\
 X \nearrow & & Y & & \nwarrow Z
 \end{array}
 \tag{5}$$

The following classes of cospans deserve special attention:

1. a cospan whose right leg is an isomorphism is the same thing as a map  $X \rightarrow Y$
2. a relation is a span  $X \leftarrow R \rightarrow Y$  which is jointly monic. Dually, a *co-relation* is a cospan  $X \rightarrow P \leftarrow Y$  which is jointly epic [11].
3. a partial map is a span  $X \leftarrow R \rightarrow Y$  whose left leg is monic [8]. Dually we define a *copartial map* to be a cospan  $X \rightarrow P \leftarrow Y$  whose right leg is epic.

Just as partial maps are maps out of subobjects, copartial maps are maps into quotients. It is worth noting that while the pushout of copartial maps is again a copartial map, co-relations are not closed under pushout. Instead, the an image factorization has to be used to compose them [11]. Lemma 7 can be rephrased as follows:

► **Proposition 9.** *To give a copartial map  $X \xrightarrow{f} P \xleftarrow{p} Y$  in  $\mathbf{Vec}$  is to give a total linear relation  $X \rightarrow Y$ . The relation is obtained as  $R = \{(x, y) : f(x) = p(y)\}$ .*

We now use the abstract theory of decorated cospans [9, 10, 14] to add Gaussian probability to the cospans:

► **Definition 10** ([9]). *Given a lax monoidal functor  $S : (\mathbb{C}, +) \rightarrow (\mathbf{Set}, \times)$ , an  $S$ -decorated cospan is a cospan  $X \rightarrow P \leftarrow Y$  together with a decoration  $s \in S(P)$ . Given composable cospans like in (5), the decoration of the composite is computed by the canonical morphism  $S(P) \times S(Q) \rightarrow S(P + Q) \rightarrow S(W)$ . The category of  $S$ -decorated cospans is written  $SCospan(\mathbb{C})$ .*



The category **Gauss** is a special case of the decorated cospan construction, for cospans whose right leg is an identity. We can now define:

► **Definition 11.** *The category **GaussEx** of extended Gaussian maps is defined as the category of copartial maps in **Vec**, decorated by the functor  $\text{Gauss} : \text{Vec} \rightarrow \text{CMon} \rightarrow \text{Set}$ .*

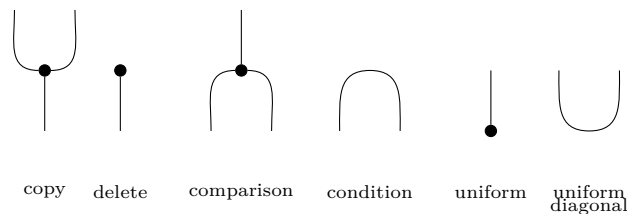
Categories of decorated cospans are hypergraph categories [9, § 2] their monoidal structure is given by the coproduct  $+$ . As the subcategory of decorated copartial maps, extended Gaussians do inherit the symmetric monoidal and copy-delete structure, but are not a hypergraph category. To obtain a useful hypergraph category of Gaussian probability, we must study conditioning.

#### 4 A Hypergraph Category of Gaussian Relations

A hypergraph category extends the structure of a copy-delete category in two important ways

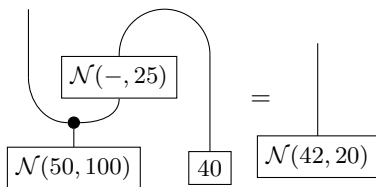
1. there is a multiplication  $\mu_X : X \otimes X \rightarrow X$  on every object, which we think of as a comparison operation. It succeeds if both inputs are equal (and return the input), and fails otherwise. In linear relations, comparison is the relation  $\{(x, x, x) : x \in X\}$ . Multiplication is dual to copying. In a probabilistic setting, we propose to think of the comparison as conditioning on equality. The “cap”  $X \otimes X \rightarrow I$  is denoted as  $\text{=:}$  [25, 33].
2. there is a unit  $u_X : I \rightarrow X$  on every object, dual to deletion. The unit is neutral with respect to the multiplication, i.e. conditioning on the unit has no effect. This suggests we should think of the unit as a uniform distribution, or an improper prior. Both in linear relations and extended Gaussians, the unit is the subspace  $X \subseteq X$ .

We arrive at the following synthetic dictionary for probabilistic inference and constraints in hypergraph categories:



■ **Figure 2** Dictionary for hypergraph categories.

For example, the noisy measurement example Example 2 can be expressed in the following convenient way using hypergraph structure

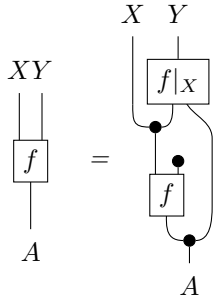


We begin by recalling how conditioning works in the category **Gauss**, and prove that extended Gaussians remain closed under conditioning. We then define a hypergraph category **GaussRel** of Gaussian relations in which conditioning is internalized using a comparison operation.

### 4.1 Conditioning

Gaussian distributions are self-conjugate; that is conditional distributions of Gaussians are themselves Gaussian. More precisely, given a joint distribution  $\psi \in \text{Gauss}(X \times Y)$ , the map which sends  $x \mapsto \psi|_{X=x}$  is a Gaussian map  $X \rightarrow Y$ . This is captured using the following categorical definition:

► **Definition 12** ([15, Definition 11.5]). *A conditional for a morphism  $f : A \rightarrow X \otimes Y$  in a Markov category is a morphism  $f|_X : X \otimes A \rightarrow Y$  which lets us reconstruct  $f$  from its  $X$ -marginal as  $f(x, y|a) = f|_X(y|x, a)f_X(y|a)$ . In string diagrams, it satisfies*



The category **Gauss** has all conditionals. By picking a convenient complement to the fibre  $D$ , we can reduce the problem of conditioning in **GaussEx** to conditioning in **Gauss**.

► **Theorem 13.** *GaussEx has conditionals.*

**Proof.** In the appendix (Section 6.5). ◀

### 4.2 Gaussian Relations

One difficulty of conditioning is that it introduces the possibility of failure. For example, the condition  $0 = 1$  is infeasible. In general, given a joint distribution  $\psi \in \text{Gauss}(X \times Y)$ , we can only condition  $\psi|_{X=x}$  if  $x$  lies in the support of the marginal  $\psi_X$ . The dependence on supports is carefully analyzed in the “Cond” construction of [34].

We define a hypergraph category **GaussRel** of *Gaussian relations* as follows

$$\text{GaussRel}(X, Y) \stackrel{\text{def}}{=} \text{GaussEx}(X \times Y) + \{\perp\}$$

That is, a Gaussian relation is either a joint extended Gaussian distribution, or a special failure symbol  $\perp$  which represents infeasibility. Failure is strict in all categorical operations, i.e. composing or tensoring anything with failure is again failure.

Most of the categorical structure of **GaussRel** is easy to define.

1. any morphism  $f \in \text{GaussEx}(X, Y)$  can be embedded into **GaussRel** as its *name*  $[f]$  given by  $I \xrightarrow{u_X} X \xrightarrow{\text{copy}_X} X \otimes X \xrightarrow{\text{id}_X \otimes f} X \otimes Y$
2. the identity is the diagonal relation  $D = \{(x, x) : x \in X\} \in \text{GaussEx}(X \times X)$
3. copying and comparison are the both given by the relation  $\{(x, x, x) : x \in X\}$

Composition of Gaussian relations requires conditioning: Given  $R \in \text{GaussRel}(X, Y)$  and  $S \in \text{GaussRel}(Y, Z)$ , we compose them as follows: If any of them is  $\perp$ , return  $\perp$ . Otherwise form the tensor  $R \otimes S \in \text{GaussEx}(X \otimes Y \otimes Y \otimes Z)$ , and condition the two copies of  $Y$  to be equal. If that condition is infeasible, return  $\perp$ .

### 4.3 Decorated cospans as generalized statistical models

We can get a clearer view of composition  $\mathbf{GaussRel}$  by using decorated cospans. Recall that decorated copartial functions  $X \rightarrow P \leftarrow Y$  corresponded to extended Gaussian maps  $X \rightarrow Y$ . If we allow arbitrary cospans, we know that the category  $\mathbf{GaussCospan}(\mathbf{Vec})$  is a hypergraph category by construction. We now explain how to view such a cospan as a kind of generalized statistical model, whose “solution” is a Gaussian relation.

► **Theorem 14.** *We have a functor of hypergraph categories*

$$F : \mathbf{GaussCospan}(\mathbf{Vec}) \rightarrow \mathbf{GaussRel}$$

which sends the decorated cospan  $X \xrightarrow{f} P \xleftarrow{g} Y$  with decoration  $\psi \in \mathbf{Gauss}(P)$  to the Gaussian relation described by the solution to the following inference problem: Initialize  $x \sim X$  and  $y \sim Y$  with an uninformative prior. Then condition  $f(x) - g(y) =: \psi$ , and return the posterior distribution in  $\mathbf{GaussEx}(X \times Y)$ , or  $\perp$  if the condition was infeasible.

Decorated cospans thus have an interpretation as a generalized kind of statistical model, and Gaussian relations can be understood as equivalence classes of such cospans which have the same solution. This approach is systematically explored with the  $\mathbf{Cond}$  construction of [34], and indeed we can see  $\mathbf{GaussRel}$  as a concrete representation of  $\mathbf{Cond}(\mathbf{GaussEx})$ .

## 5 Related Work and Applications

### 5.1 Open Linear Systems and $\sigma$ -algebras

Recall that a probability space is a tuple  $(X, \mathcal{E}, P)$  of a set  $X$ , a  $\sigma$ -algebra  $\mathcal{E}$  and a probability measure  $P : \mathcal{E} \rightarrow [0, 1]$ . A random variable is a function  $V : X \rightarrow \mathbb{R}$  which is  $(\mathcal{E}, \mathcal{B}(\mathbb{R}))$ -measurable, where  $\mathcal{B}(\mathbb{R})$  denotes the Borel  $\sigma$ -algebra.

Willems defines an *n-dimensional linear stochastic system* to be a probability space of the form  $(\mathbb{R}^n, \mathcal{E}, P)$  for which there exists a “fibre” subspace  $D \subseteq \mathbb{R}^n$  such that the  $\sigma$ -algebra  $\mathcal{E}$  is given by the Borel subsets of  $\mathbb{R}^n/D$  in the following sense: Pick any complementary subspace  $K$  with  $K \oplus D = \mathbb{R}^n$ . Then, the events  $V \in \mathcal{E}$  are precisely Borel cylinders parallel to  $D$ , i.e. of the form  $V = A + D$  for  $A \in \mathcal{B}(K)$ . As an aside, we might wonder in which sense the algebra  $\mathcal{E}$  is a quotient construction. The measurable projection  $p : (\mathbb{R}^n, \mathcal{E}) \rightarrow (K, \mathcal{B}(K))$  is *not* an isomorphism of measurable spaces; after all, the underlying function is not invertible. It is however an isomorphism in the category of probability kernels, namely the inclusion  $i : K \rightarrow \mathbb{R}^n$  is an inverse when considered as a stochastic map. This is because the Dirac measures  $\delta_x$  and  $\delta_{ipx}$  are equal on  $\mathcal{E}$ . This phenomenon of “weak quotients” is nicely explained in [27, Appendix A].

A linear system is called *Gaussian* if the measure  $P$  on  $K$  is a normal distribution. We notice that this agrees precisely with our definition of an extended Gaussian distribution on  $\mathbb{R}^n$  with fibre  $D$ . A linear system is classical only if  $D = 0$ , in the sense that only in this case the measure  $P$  is defined on the whole algebra  $\mathcal{B}(\mathbb{R})$ . In the case  $D = \mathbb{R}^n$ , the  $\sigma$ -algebra becomes  $\mathcal{E} = \{\emptyset, \mathbb{R}^n\}$  and we cannot answer any nontrivial questions about the system (Example 4).

Willems gives explicit formulas for combining Gaussian linear systems (“tearing, zooming and linking”) [38]. These operations have been treated in categorical form in [12] but not for probabilistic systems. One fundamental operation in Willems’ calculus is the *interconnection* of systems: Two probability systems  $(X, \mathcal{E}_1, P_1), (X, \mathcal{E}_2, P_2)$  on the same state space  $X$  are called *complementary* if for all  $E_1, E'_1 \in \mathcal{E}_1$  and  $E_2, E'_2 \in \mathcal{E}_2$ , we have

$$E_1 \cap E_2 = E'_1 \cap E'_2 \Rightarrow P_1(E_1)P_2(E_2) = P_1(E'_1)P_2(E'_2)$$

## 13:12 Combining Gaussian Probability with Nondeterminism

That is, the product of probabilities  $P_1(E_1)P(E_2)$  depends only on the intersection  $E_1 \cap E_2$ . The two probability measures  $P_1, P_2$  can now be joined together on the larger  $\sigma$ -algebra  $\mathcal{E} = \sigma(\mathcal{E}_1 \cup \mathcal{E}_2)$  by defining  $P(E_1 \cap E_2) = P(E_1)P(E_2)$ . This is what happens in Example 1 when connecting a noisy resistor to a voltage source: The underspecified  $\sigma$ -algebras gets enlarged and nondeterministic relationships become probabilistic ones. It seems to us that interconnection is a special case of the composition of Gaussian relations.

It is furthermore interesting that Willems uses the term *open* for probability systems with an underspecified  $\sigma$ -algebra on  $X$ , while in category theory, we think of open systems as morphisms  $Y \rightarrow X$ . A remarkable feature is that the  $\sigma$ -algebra, which in measure-theoretic probability is considered a property of the objects in question (i.e. measurable spaces), is here part of the morphisms. The cospan perspective unifies this, for in a cospan  $Y \xrightarrow{f} Q \xleftarrow{g} X$ , we can equip  $X$  with the  $\sigma$ -algebra generated by the quotient map  $q$ .

### 5.2 Variance-Precision Duality

We recall the coordinate-free description of Gaussian probability and use it to show that extended Gaussians are highly symmetric objects, which enjoy an improved duality theory over ordinary Gaussians (reflecting the hypergraph structure of Gaussian relations). This also points towards future research to understand **GaussEx** as a topological completion of ordinary Gaussians. Work in this direction is the variance-information manifold of [23]. For simplicity, we will consider only Gaussians of mean zero.

If the covariance matrix  $\Sigma \in \mathbb{R}^{n \times n}$  is invertible, then its inverse  $\Omega = \Sigma^{-1}$  is known as *precision* or *information matrix*. Precision is dual to covariance, in the sense that while covariance is additive for convolution  $+$ , precision is additive for conditioning.

$$\Sigma_{\psi_1 + \psi_2} = \Sigma_{\psi_1} + \Sigma_{\psi_2}, \quad \Omega_{\psi_1 \cap \psi_2} = \Omega_{\psi_1} + \Omega_{\psi_2}$$

The latter equation is reminiscent of logdensities, which add when conditioning. Indeed, the precision matrix appears in the density function of the multivariate Gaussian distribution  $f(x) \propto \exp(-\frac{1}{2}(x - \mu)^T \Omega (x - \mu))$ .

If we allow singular covariance matrices  $\Sigma$ , we still have well-defined Gaussian distributions albeit with non-full support; however the information matrix ceases to exist (and the distribution no longer has a density with respect to the  $n$ -dimensional Lebesgue measure). Not only does this break the duality, but we are left to wonder which kind of distribution corresponds to singular precision matrices: The answer is extended Gaussian distributions with nonvanishing fibre.

In a coordinate-free way, the covariance of a distribution  $\psi \in \mathbf{Gauss}(X)$  is the bilinear form on the dual space  $\Sigma : X^* \times X^* \rightarrow \mathbb{R}$  given by  $\Sigma(f, g) \stackrel{\text{def}}{=} \mathbb{E}[f(U)g(U)] - \mathbb{E}[f(U)]\mathbb{E}[g(U)]$ . This form is symmetric and positive semidefinite. The precision form  $\Omega$  is instead of type  $\Omega : X \times X \rightarrow \mathbb{R}$ . The duality between the two forms can be stated as follows:

► **Theorem 15.** *The following data are equivalent for every f.d.-vector space  $X$*

1. *pairs  $\langle S, \Omega \rangle$  of a subspace  $S \subseteq X$  and a bilinear form  $\Omega : S \times S \rightarrow \mathbb{R}$*
2. *pairs  $\langle F, \Sigma \rangle$  of a subspace  $F \subseteq X^*$  and a bilinear form  $\Sigma : F \times F \rightarrow \mathbb{R}$*

At the core of this duality lies the notion of the *annihilator* of a subspace, here denoted  $(-)^{\perp}$ . In brief, the correspondences are as follows

precision	$S = \ker(\Sigma)^{\perp}$	$\ker(\Omega) = D$
covariance	$F = D^{\perp}$	$\ker(\Sigma) = S^{\perp}$

We give a proof of the duality in the appendix (Section 6.3).

### 5.3 Statistical Learning and Probabilistic Programming

It is unsurprising that notions equivalent to extended Gaussians have appeared in the statistics (e.g. in [22]). A novel perspective on statistical inference which more closely matches the categorical semantics is *probabilistic programming*, a powerful and flexible paradigm which has gained traction in recent years (e.g. [36, 29, 20]). In [34], we argued that the exact conditioning operation (conditioning on equality) described in Section 4.2 is a fundamental primitive in such programs, and enjoys good logical properties. We presented a programming language for Gaussian probability featuring a first-class exact conditioning operator ( $\text{:=}$ ), with Python/F# implementations available under [30]. For example, the noisy measurement example expressed as a probabilistic program reads

```
x = normal(50, 100)
y = normal(x, 25)
y := 40
return x
```

This language uses Gaussian distributions only, but it can effortlessly be extended to use extended Gaussian distributions, which are likewise closed under conditioning (Theorem 13).

The behavior of the conditioning operator ( $\text{:=}$ ) can be quite subtle, and it is difficult to decide when two programs are observationally equivalent. The denotational semantics defined in [34] on the basis of the category  $\text{Cond}(\text{Gauss})$  is fully abstract, but it is still lacking a concrete description of when two different programs fragments have the same behavior in all contexts. This is remedied by passing to the concrete description of  $\text{GaussRel}$ . In terms of Section 4.3, a program denotes a decorated cospan over  $\text{Vec}$ , and contextual equivalence is precisely the equivalence relation Theorem 14.

The correspondence between probabilistic programs and categorical models of probability (with conditioning) is elaborated in detail in [31].

---

#### References

- 1 John C. Baez, Brandon Coya, and Franciscus Rebro. Props in network theory, 2018. [arXiv:1707.08321](#).
- 2 John C. Baez and Jason Erbele. Categories in control. *Theory Appl. Categ.*, 30:836–881, 2015. [arXiv:1405.6881](#).
- 3 Filippo Bonchi, Paweł Sobociński, and Fabio Zanasi. A categorical semantics of signal flow graphs. In *CONCUR 2014—Concurrency Theory: 25th International Conference, CONCUR 2014, Rome, Italy, September 2-5, 2014. Proceedings 25*, pages 435–450. Springer, 2014.
- 4 Filippo Bonchi, Paweł Sobocinski, and Fabio Zanasi. The calculus of signal flow diagrams I: linear relations on streams. *Inform. Comput.*, 252, 2017.
- 5 Filippo Bonchi, Paweł Sobociński, and Fabio Zanasi. Interacting Hopf algebras. *Journal of Pure and Applied Algebra*, 221(1):144–184, 2017.
- 6 A. Carboni and R.F.C. Walters. Cartesian bicategories i. *Journal of Pure and Applied Algebra*, 49(1):11–32, 1987. [doi:10.1016/0022-4049\(87\)90121-6](#).
- 7 Kenta Cho and B. Jacobs. Disintegration and Bayesian inversion via string diagrams. *Mathematical Structures in Computer Science*, 29:938–971, 2019.
- 8 J Robin B Cockett and Stephen Lack. Restriction categories i: categories of partial maps. *Theoretical computer science*, 270(1-2):223–259, 2002.
- 9 Brendan Fong. Decorated cospans. *arXiv preprint arXiv:1502.00872*, 2015.

- 10 Brendan Fong. *The Algebra of Open and Interconnected Systems*. PhD thesis, University of Oxford, September 2016.
- 11 Brendan Fong. Decorated corelations, 2017. [arXiv:1703.09888](https://arxiv.org/abs/1703.09888).
- 12 Brendan Fong, Paweł Sobociński, and Paolo Rapisarda. A categorical approach to open and interconnected dynamical systems. In *Proceedings of the 31st annual ACM/IEEE symposium on Logic in Computer Science*, pages 495–504, 2016.
- 13 Brendan Fong and David I. Spivak. Hypergraph categories. *ArXiv*, abs/1806.08304, 2019.
- 14 Brendan Fong and David I Spivak. *An invitation to applied category theory: seven sketches in compositionality*. Cambridge University Press, 2019.
- 15 T. Fritz. A synthetic approach to Markov kernels, conditional independence and theorems on sufficient statistics. *Adv. Math.*, 370, 2020.
- 16 T. Fritz and Paolo Perrone. A probability monad as the colimit of spaces of finite samples. *arXiv: Probability*, 2017.
- 17 Tobias Fritz, Tomáš Gonda, and Paolo Perrone. De Finetti’s theorem in categorical probability, 2021. [arXiv:2105.02639](https://arxiv.org/abs/2105.02639).
- 18 Tobias Fritz and Eigil Fjeldgren Rischel. Infinite products and zero-one laws in categorical probability. *Compositionality*, 2, August 2020. doi:10.32408/compositionality-2-3.
- 19 Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, 2nd ed. edition, 2004.
- 20 Noah D Goodman, Joshua B. Tenenbaum, and The ProbMods Contributors. Probabilistic Models of Cognition. <http://probmods.org>, 2016. Accessed: 2021-3-26.
- 21 Alexandre Goy and Daniela Petrişan. Combining probabilistic and non-deterministic choice via weak distributive laws. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS ’20*, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3373718.3394795.
- 22 Jakob Nebeling Hedegaard. Gaussian random fields – infinite, improper and intrinsic. Master’s thesis, Aalborg University, 2019.
- 23 A.T. JAMES. The variance information manifold and the functions on it. In *Multivariate Analysis-III*, pages 157–169. Academic Press, 1973. doi:10.1016/B978-0-12-426653-7.50016-8.
- 24 Steffen Lauritzen and Frank Jensen. Stable local computation with conditional Gaussian distributions. *Statistics and Computing*, 11, November 1999. doi:10.1023/A:1008935617754.
- 25 Elena Di Lavore and Mario Román. Evidential decision theory via partial markov categories, 2023. [arXiv:2301.12989](https://arxiv.org/abs/2301.12989).
- 26 Joe Moeller and Christina Vasilakopoulou. Monoidal Grothendieck construction. *arXiv preprint arXiv:1809.00727*, 2018.
- 27 SEAN MOSS and PAOLO PERRONE. A category-theoretic proof of the ergodic decomposition theorem. *Ergodic Theory and Dynamical Systems*, pages 1–27, 2023. doi:10.1017/etds.2023.6.
- 28 João Paixão, Lucas Rufino, and Paweł Sobociński. High-level axioms for graphical linear algebra. *Science of Computer Programming*, 218:102791, 2022. doi:10.1016/j.scico.2022.102791.
- 29 Sam Staton. Commutative semantics for probabilistic programming. In Hongseok Yang, editor, *Programming Languages and Systems*, pages 855–879, Berlin, Heidelberg, 2017. Springer Berlin Heidelberg.
- 30 Dario Stein. GaussianInfer. <https://github.com/damast93/GaussianInfer>, 2021.
- 31 Dario Stein. *Structural Foundations for Probabilistic Programming Languages*. PhD thesis, University of Oxford, 2021.
- 32 Dario Stein. Decorated linear relations: Extending gaussian probability with uninformative priors, 2022. [arXiv:2204.14024](https://arxiv.org/abs/2204.14024).
- 33 Dario Stein and Sam Staton. Compositional semantics for probabilistic programs with exact conditioning (long version). In *Proceedings of Thirty-Sixth Annual ACM/IEEE Conference on Logic in Computer Science (LICS 2021)*, 2021.

- 34 Dario Stein and Sam Staton. Compositional semantics for probabilistic programs with exact conditioning (long version), 2021. [arXiv:2101.11351](#).
- 35 Alexander Terenin. *Gaussian Processes and Statistical Decision-making in Non-Euclidean spaces*. PhD thesis, Imperial College London, 2022.
- 36 Jan-Willem van de Meent, Brooks Paige, Hongseok Yang, and Frank Wood. An introduction to probabilistic programming, 2018. [arXiv:1809.10756](#).
- 37 Jan C. Willems. Constrained probability. In *2012 IEEE International Symposium on Information Theory Proceedings*, pages 1049–1053, 2012. [doi:10.1109/ISIT.2012.6283011](#).
- 38 Jan C. Willems. Open stochastic systems. *IEEE Transactions on Automatic Control*, 58(2):406–421, 2013. [doi:10.1109/TAC.2012.2210836](#).
- 39 Maaïke Zwart and Dan Marsden. No-go theorems for distributive laws. In *Proceedings of the 34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '19*. IEEE Press, 2019.

## 6 Appendix

### 6.1 Glossary: Category Theory

We assume basic familiarity of the reader with monoidal category theory and string diagrams. All relevant categories in this article are symmetric monoidal.

A copy-delete category [7] (or gs-monoidal category) is a symmetric monoidal category  $(\mathcal{C}, \otimes, I)$  where every object  $X$  is coherently equipped with the structure of a commutative comonoid, which is used to model copying ( $\text{copy}_X : X \rightarrow X \otimes X$ ) and discarding ( $\text{del}_X : X \rightarrow I$ ) of information. In string diagrams, the comonoid axioms are rendered as

Neither deleting nor copying are assumed to be natural in a copy-delete category. A *Markov category* is a copy-delete category where deleting is natural, or equivalently, the monoidal unit  $I$  is terminal. Markov categories typically model probabilistic or nondeterministic computation without possibility of failure, such as stochastic matrices, Gauss or total (linear) relations.

Copy-delete categories can model unnormalized probabilistic computation, or the potential of failure. The categories of partial functions or (linear) relations are typical examples of copy-delete categories that are not Markov categories.

A *hypergraph category* [13] is a symmetric monoidal category with a particularly powerful self-duality: Every object is equipped with a special commutative Frobenius algebra structure.

### 6.2 Noisy measurement example

► **Example 16.** We elaborate the noisy measurement example from the introduction. Formally, we introduce random variables

$$X \sim \mathcal{N}(50, 100)$$

$$Y \sim \mathcal{N}(X, 25)$$

The vector  $(X, Y)$  is multivariate Gaussian with mean  $(50, 50)$  and covariance matrix

$$\Sigma = \begin{pmatrix} 100 & 100 \\ 100 & 125 \end{pmatrix}$$

### 13:16 Combining Gaussian Probability with Nondeterminism

The conditional distribution  $X|(Y = 40)$  is  $\mathcal{N}(\mu = 42, \sigma^2 = 20)$ .

**Proof.** The random vector  $(X, Y)$  has joint density function

$$f(x, y) = \frac{1}{2\pi \cdot \sqrt{100 \cdot 25}} \exp\left(-\frac{(x-50)^2}{2 \cdot 100}\right) \cdot \exp\left(-\frac{(y-x)^2}{2 \cdot 25}\right)$$

The conditional density of  $x$  given  $y$  has the form

$$f(x|y) = \frac{f(x, y)}{\int f(x, y) dx}$$

By expanding and “completing the square”, it is easy to check that

$$f(x, 40) \propto \exp\left(-\frac{(x-50)^2}{200} - \frac{(40-x)^2}{50}\right) \propto \exp\left(-\frac{(x-42)^2}{2 \cdot 20}\right)$$

is again a Gaussian density, from which we read off  $\mu = 42$  and  $\sigma^2 = 20$ . ◀

### 6.3 Glossary: Linear Algebra

All vector spaces in this paper are assumed finite dimensional. For vector subspaces  $U, V \subseteq X$ , their *Minkowski sum* is the subspace  $U+V = \{u+v : u \in U, v \in V\}$ . If furthermore  $U \cap V = 0$ , we call their sum a *direct sum* and write  $U \oplus V$ . A *complement* of  $U$  is a subspace  $V$  such that  $U \oplus V = X$ . An *affine subspace*  $W \subseteq X$  is a subset of the form  $x + U$  for some  $x \in X$  and a (unique) vector subspace  $U \subseteq X$ . The space  $W$  is called a *coset* of  $U$  and the cosets of  $U$  organize into the quotient vector space  $X/U = \{x + U : x \in X\}$ .

An affine-linear map  $f : X \rightarrow Y$  between vector spaces is a map of the form  $f(x) = g(x) + y$  for some linear function  $g : X \rightarrow Y$  and  $y \in Y$ . Vector spaces and affine-linear maps form a category **Aff**.

A *linear relation*  $R \subseteq X \times Y$  is a relation which is also a vector subspace of  $X \times Y$ . We write  $R(x) \stackrel{\text{def}}{=} \{y \in Y : (x, y) \in R\}$ . A relation  $R \subseteq X \times Y$  is called *total* if  $R(x) \neq \emptyset$  for all  $x \in X$ . Linear relations and total linear relations are closed under the usual composition of relations. We denote by **LinRel** and **LinRel**<sup>+</sup> the categories whose objects are vector spaces, and morphisms are linear relations and total linear relations respectively. **LinRel** is a hypergraph category, while **LinRel**<sup>+</sup> is a Markov category.

The following lemma is crucial for relating linear relations and cospans: Every left-total linear relation can be written as a “linear map with nondeterministic noise”  $x \mapsto f(x) + D$ .

► **Proposition 17.** *Let  $R \subseteq X \times Y$  be a left-total linear relation. Then*

1.  $R(0)$  is a vector subspace of  $Y$
2.  $R(x)$  is a coset of  $R(0)$  for every  $x \in X$
3. the assignment  $x \mapsto R(x)$  is a well-defined linear map  $X \rightarrow Y/R(0)$
4. every linear map  $X \rightarrow Y/D$  is of that form for a unique left-total linear relation.

**Proof.** For 1, consider  $y, y' \in R(0)$  (by assumption nonempty), then by linearity of  $R$

$$(0, y) \in R, (0, y') \in R \Rightarrow (0, \alpha y + \beta y') \in R$$

so  $R(0)$  is a vector subspace. For 2, we can find some  $w \in R(x)$  and wish to show that  $R(x) = w + R(0)$ . Indeed if  $y \in R(x)$  then  $(x, y) - (x, w) = (0, y - w) \in R$  so  $y - w \in R(0)$ , hence  $y \in w + R(0)$ . Conversely for all  $z \in R(0)$  we have  $(x, w + z) = (x, w) + (0, z) \in R$



so  $w + z \in R(x)$ . This completes the proof that  $R(x)$  is a coset. For 3, the previous point shows that the map  $\rho : x \mapsto R(x)$  is a well-defined map  $X \rightarrow Y/R(0)$ . It remains to show it is linear. That is, if  $w \in R(x)$  and  $z \in R(y)$  then  $\alpha w + \beta z \in R(\alpha x + \beta y)$ . This follows immediately from the linearity of  $R$ . For the last point 4, given a linear map  $f : X \rightarrow Y/V$  we construct the relation

$$(x, y) \in R \Leftrightarrow y \in f(x)$$

which is left-total because  $f(x) \neq \emptyset$ . To see that  $R$  is linear, let  $(x, y) \in R, (x', y') \in R$  meaning  $y - z \in V$  and  $y' - z' \in V$  for representatives  $z, z'$  of  $f(x), f(x')$ . Linearity of  $f$  means that  $\alpha z + \beta z'$  is a representative of  $f(\alpha x + \beta x')$ . Thus

$$\alpha y + \beta y' - (\alpha z + \beta z') = \alpha(y - z) + \beta(y' - z') \in V \quad \blacktriangleleft$$

## 6.4 Annihilators

For subspaces  $D \subseteq X$  and  $F \subseteq X^*$ , the subspaces  $D^\perp \subseteq X^*, F^\perp \subseteq X$  are defined as

$$D^\perp \stackrel{\text{def}}{=} \{f \in X^* : f|_D = 0\}, \quad F^\perp \stackrel{\text{def}}{=} \{x \in X : \forall f \in F, f(x) = 0\} \quad (6)$$

### ► Proposition 18.

1. Taking annihilators is order-reversing and involutive
2. If  $D \subseteq S \subseteq X$ , then  $S^\perp \subseteq D^\perp \subseteq X^*$  and we have a canonical isomorphism

$$(S/D)^* \cong D^\perp/S^\perp \quad (7)$$

and similarly for  $K \subseteq F \subseteq X^*$ , we have

$$(F/K)^* \cong K^\perp/F^\perp \quad (8)$$

3. We have

$$(V + W)^\perp = V^\perp \cap W^\perp$$

$$(F \cap W)^\perp = F^\perp + G^\perp$$

If  $D \subseteq X$  and  $f : X \rightarrow Y$ , then

$$(f[D])^\perp = \{g \in Y^* : gf \in D^\perp\}$$

If  $U \subseteq X, V \subseteq Y$ , we have a canonical isomorphism

$$(U \times V)^\perp \cong U^\perp \times V^\perp$$

**Proof.** Standard. An explicit description of the canonical iso (7) is given as follows.

1. We define  $\alpha : D^\perp/S^\perp \rightarrow (S/D)^*$  as follows. If  $f \in D^\perp$ , then  $f$  is a function  $X \rightarrow \mathbb{R}$  such that  $f|_D = 0$ . The restriction  $f|_S : S \rightarrow \mathbb{R}$  thus descends to the quotient  $S/D \rightarrow \mathbb{R}$ , and we let  $\tilde{\alpha}(f) = f|_S$ . To check this is well-defined, notice that the kernel of  $\tilde{\alpha}$  consists of those  $f \in X^*$  such that  $f|_S = 0$ , that is  $S^\perp$ .
2. We define  $\alpha^{-1} : (S/D)^* \rightarrow D^\perp/S^\perp$  as follows. An element  $f \in (S/D)^*$  is a function  $f : S \rightarrow \mathbb{R}$  with  $f|_D = 0$ . Find any extension of  $f$  to a linear function  $\bar{f} : X \rightarrow \mathbb{R}$  (such an extension exists because  $S$  is a retract of  $X$ ). Then still  $\bar{f}|_D = 0$ , so  $\bar{f} \in D^\perp$ . It remains to show that the choice of extension does not matter in the quotient  $D^\perp/S^\perp$ . Indeed if  $\bar{f}_2$  is another extension, then  $(\bar{f} - \bar{f}_2)|_S = f - f = 0$ , hence  $(\bar{f} - \bar{f}_2) \in S^\perp$ .  $\blacktriangleleft$

## 6.5 Conditionals

The existence proof of conditionals in `GaussEx` relies on the ability to pick a convenient complement to a subspace, as constructed by the following lemma:

► **Lemma 19.** *Let  $V \subseteq X \times Y$  be a vector subspace, and let  $V_X \subseteq X$  be its projection. Then there exists a complement  $K \subseteq X \times Y$  of  $V$  such that  $K_X$  is a complement of  $V_X$ .*

**Proof.** We give an explicit construction, where in fact we can choose  $K$  to be a cartesian product of subspaces  $U \times W$ . Let

$$V_X = \{x : (x, y) \in V\} \quad H = \{y : (0, y) \in V\}$$

We argue that if  $U \oplus V_X = X$  and  $W \oplus H = Y$ , then  $(U \times W) \oplus V = X \times Y$ . First we prove that  $(U \times W) \cap V = 0$ : Indeed, if  $(u, w) \in V$  for  $u \in U, w \in W$ , then  $u \in V_X$ , but that implies  $u = 0$ . So we know  $(0, w) \in V$ , i.e.  $w \in H$ . Thus  $w = 0$ .

It remains to show that we can write every  $(x, y)$  as  $(u + v_1, w + v_2)$  with  $u \in U, w \in W$  and  $(v_1, v_2) \in V$ .

1. We can write  $x = u + v_1$  with  $u \in U$  and  $v_1 \in V_X$ .
2. We claim that there exists a  $b \in W$  such that  $(v_1, b) \in V$ . Because  $v_1 \in V_X$ , there exists some  $b' \in Y$  such that  $(v_1, b') \in V$ . We now decompose  $b' = b + h$  for  $b \in W, h \in H$ . By definition of  $H$ , we have  $(0, h) \in V$ , so  $(v_1, b) = (v_1, b') - (0, h) \in V$ .
3. Write  $y = w' + h$  with  $w' \in W, h \in H$  and define  $w = w' - b$  and  $v_2 = h + b$ . Then we have  $w \in W$  and  $(v_1, v_2) = (v_1, b) + (0, h) \in V$ , and as desired

$$(u, w) + (v_1, v_2) = (x, w' - b + h + b) = (x, w' + h) = (x, y). \quad \blacktriangleleft$$

We can now prove the existence of conditionals in `GaussEx`.

**Proof of Theorem 13.** Let  $\varphi \in \text{GaussEx}(A, X \times Y)$  be given by  $(D, \tilde{f}, \tilde{\psi})$  where  $\tilde{f} : A \rightarrow (X \times Y)/D$ ,  $D \subseteq X \times Y$  and  $\tilde{\psi} \in \text{Gauss}((X \times Y)/D)$ . By Lemma 19, we can pick a complement  $K \subseteq X \times Y$  of  $D$  such that  $K_X$  is a complement of  $D_X$  in  $X$ . Under the identification  $(X \times Y)/D \cong K$ , we replace  $\tilde{f}, \tilde{\psi}$  with  $f : A \rightarrow X \times Y$  and  $\psi \in \text{Gauss}(X \times Y)$ .

Now we consider the morphism  $x \mapsto f(x) + \psi$  in  $\text{Gauss}(A, X \times Y)$  and find a conditional  $f|_X \in \text{Gauss}(X \times A, Y)$ . Informally, this means we can obtain  $(X_1, Y_1) \sim f(a) + \psi$  as follows:

$$X_1 \sim f_X(a) + \psi_X, \quad Y_1 \sim g(x, a) + \xi$$

Similarly we can use conditionals in  $\text{LinRel}^+$  to find a linear function  $h : X \rightarrow Y$  and a subspace  $H \subseteq Y$  such that  $(X_2, Y_2) \sim D$  can be obtained as

$$X_2 \sim D_X, \quad Y_2 \sim h(X_2) + H$$

Thus a joint sample  $(X, Y) \sim f(x) + \psi + D$  can be obtained using the following process

$$\begin{array}{lll} X_1 \sim f_X(a) + \psi_X, & X_2 \sim D_X, & X = X_1 + X_2 \\ Y_1 \sim g(X_1, a) + \xi, & Y_2 \sim h(X_2) + H, & Y = Y_1 + Y_2 \end{array}$$

By construction we have  $X_1 \in K_X, X_2 \in D_X$ . Because  $K$  was chosen such that  $K_X \oplus D_X = X$ , we can extract the individual values of  $X_1, X_2$  from  $X$  via the projections  $P_{K_X}, P_{D_X} : X \rightarrow X$ . A conditional for  $\varphi$  is thus given by the formula

$$\varphi|_X(x, a) = g(P_{K_X}(x), a) + h(P_{D_X}(x)) + \xi + H \quad \blacktriangleleft$$




# Fractals from Regular Behaviours

Todd Schmid   

University College, London, UK

Victoria Noquez  

St. Mary's College of California, Moraga, CA, USA

Lawrence S. Moss   

Indiana University, Bloomington, IN, USA

---

## Abstract

---

We are interested in connections between the theory of fractal sets obtained as attractors of iterated function systems and process calculi. To this end, we reinterpret Milner's expressions for processes as contraction operators on a complete metric space. When the space is, for example, the plane, the denotations of fixed point terms correspond to familiar fractal sets. We give a sound and complete axiomatization of fractal equivalence, the congruence on terms consisting of pairs that construct identical self-similar sets in all interpretations. We further make connections to labelled Markov chains and to invariant measures. In all of this work, we use important results from process calculi. For example, we use Rabinovich's completeness theorem for trace equivalence in our own completeness theorem. In addition to our results, we also raise many questions related to both fractals and process calculi.

**2012 ACM Subject Classification** Theory of computation → Process calculi

**Keywords and phrases** fixed-point terms, labelled transition system, fractal, final coalgebra, equational logic, completeness

**Digital Object Identifier** 10.4230/LIPIcs.CALCO.2023.14

**Funding** *Todd Schmid*: Partially funded by ERC Grant Autoprobe (grant agreement 10100269).

*Lawrence S. Moss*: Supported by grant #586136 from the Simons Foundation.

**Acknowledgements** We would like to thank Alexandra Silva and Dylan Thurston for helpful discussions. The images in Figures 1 and 2 were made using SageMath and GIMP.

## 1 Introduction

Hutchinson noticed in [13] that many familiar examples of fractals can be captured as the set-wise fixed-point of a finite family of contraction (i.e., distance shrinking) operators on a metric space. He called these spaces (*strictly*) *self-similar*, since the intuition behind the contraction operators is that they are witnesses for the appearance of the fractal in a proper (smaller) subset of itself. For example, the famous Sierpiński gasket is the unique nonempty compact subset of the plane left fixed by the union of the three operators  $\sigma_a, \sigma_b, \sigma_c : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  in Figure 1. The Sierpiński gasket is a scaled-up version of each of its thirds.

The self-similarity of Hutchinson's fractals hints at an algorithm for constructing them: Each point in a self-similar set is the limit of a sequence of points obtained by applying the contraction operators one after the other to an initial point. In the Sierpiński gasket, the point  $(1/4, \sqrt{3}/4)$  is the limit of the sequence

$$p, \sigma_b(p), \sigma_b\sigma_a(p), \sigma_b\sigma_a\sigma_a(p), \sigma_b\sigma_a\sigma_a\sigma_a(p), \dots \quad (1)$$

where the initial point  $p$  is an arbitrary element of  $\mathbf{R}^2$  (note that  $\sigma_b$  is applied last). Hutchinson showed in [13] that the self-similar set corresponding to a given family of contraction operators is precisely the collection of points obtained in the manner just described. The limit of the



© Todd Schmid, Victoria Noquez, and Lawrence S. Moss;  
licensed under Creative Commons License CC-BY 4.0

10th Conference on Algebra and Coalgebra in Computer Science (CALCO 2023).

Editors: Paolo Baldan and Valeria de Paiva; Article No. 14; pp. 14:1–14:18

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

14:2 Fractals from Regular Behaviours

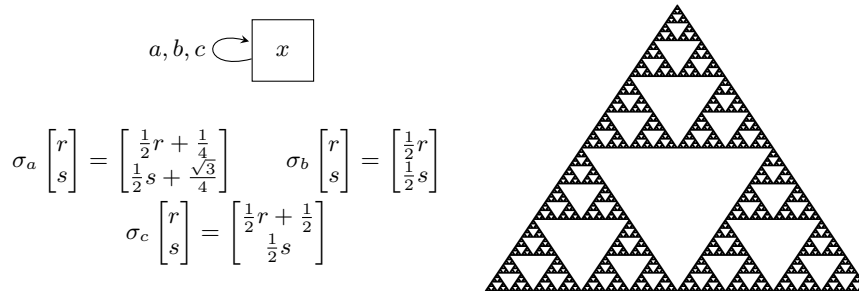
sequence in (1) does not depend on the initial point  $p$  because  $\sigma_a, \sigma_b, \sigma_c$  are contractions. Much like digit expansions to real numbers, every stream of  $a$ 's,  $b$ 's, and  $c$ 's corresponds to a unique point in the Sierpiński gasket. The point  $(1/4, \sqrt{3}/4)$ , for example, corresponds to the stream  $(b, a, a, a, \dots)$  ending in an infinite sequence of  $a$ 's. Conversely, every point in the Sierpiński gasket comes from (in general more than one) corresponding stream.

From a computer science perspective, the languages of streams considered by Hutchinson are the *traces* observed by one-state labelled transition systems, like the one in Figure 1. We investigate whether one could achieve a similar effect with languages of streams obtained from labelled transition systems having more than one state. Observe, for example, Figure 2. This twisted version of the Sierpiński gasket is constructed from a two-state labelled transition system. Each point in the twisted Sierpiński gasket corresponds to a stream of  $a$ 's,  $b$ 's, and  $c$ 's, but not every stream corresponds to a point in the set: The limit corresponding to  $(c, a, b, c, c, c, \dots)$  is  $(3/4, \sqrt{3}/8)$ , for example.

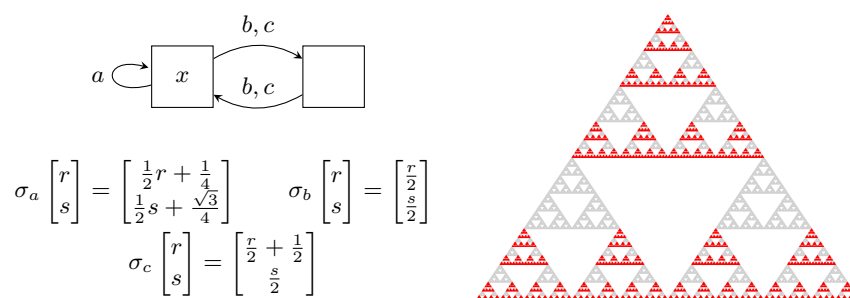
A labelled transition system paired with an interpretation of its labels as contractions on a complete metric space is the same data as a *directed-graph iterated function system* (GIFS), a generalization of iterated function systems introduced by Mauldin and Williams [18]. GIFSs generate their own kind of self-similar set, and much work has been done to understand the geometric properties of fractal sets generated by GIFSs [7–10, 18]. We take this work in a slightly different direction by presenting a coalgebraic perspective on GIFSs, seeing each labelled transition system as a “recipe” for constructing fractal sets.

In analogy with the theory of regular languages, we call the fractals generated by finite labelled transition systems *regular subfractals*, and give a logic for deciding if two labelled transition systems represent the same recipe under all interpretations of the labels. By identifying points in the fractal set generated by a labelled transition system with traces observed by the labelled transition system, it is reasonable to suspect that two labelled transition systems represent equivalent fractal recipes – i.e., they represent the same fractal under every interpretation – if and only if they are trace equivalent. This is the content of Theorem 4.4, which allows us to connect the theory of fractal sets to mainstream topics in computer science.

Labelled transition systems are a staple of theoretical computer science, especially in the area of process algebra [1], where a vast array of different notions of equivalence and axiomatization problems have been studied. We specifically use a syntax introduced by Milner in [22] to express labelled transition systems as terms in an expression language with recursion. This leads us to a fragment of Milner’s calculus consisting of just the terms that constitute recipes for fractal constructions. Using a logic of Rabinovich [25] for deciding trace equivalence in Milner’s calculus, we obtain a complete axiomatization of fractal recipe equivalence.



■ **Figure 1** The Sierpiński gasket is the unique nonempty compact subset  $\mathbf{S}$  of  $\mathbf{R}^2$  such that  $\mathbf{S} = \sigma_a(\mathbf{S}) \cup \sigma_b(\mathbf{S}) \cup \sigma_c(\mathbf{S})$ . Each of its points corresponds to a stream emitted by the state  $x$ .



■ **Figure 2** A twisted Sierpiński gasket, depicted in red. In the construction of this set,  $\sigma_b$  and  $\sigma_c$  are applied twice to a single copy of  $\sigma_a$  applied to the set. This has the effect of systematically removing the “top” part of the Sierpiński gasket from its bottom thirds.

In his study of self-similar sets, Hutchinson also makes use of probability measures supported on self-similar sets, called *invariant measures*. Each invariant measure is specified by a probability distribution on the set of contractions generating its support. In the last technical section of the paper, we adapt the construction of invariant measures to a probabilistic version of labelled transition systems called *labelled Markov chains*, which allows us to give a measure-theoretic semantics to terms in a probabilistic version of Milner’s specification language, the calculus introduced by Stark and Smolka [27]. Our measure-theoretic semantics of probabilistic process terms can be seen as a generalization of the trace measure semantics of Kerstan and König [14]. We offer a sound axiomatization of equivalence under this semantics and pose completeness as an open problem.

In sum, the contributions of this paper are as follows.

- In Section 3, we give a fractal recipe semantics to process terms using a generalization of iterated function systems.
- In Section 4, we show that two process terms agree on all fractal interpretations if and only if they are trace equivalent. This implies that fractal recipe equivalence is decidable for process terms, and it allows us to derive a complete axiomatization of fractal recipe equivalence from Rabinovich’s axiomatization [25] of trace equivalence of process terms.
- Finally, we adapt the fractal semantics of process terms to the probabilistic setting in Section 5 and propose an axiomatization of probabilistic fractal recipe equivalence.

We start with a brief overview of trace semantics in process algebra and Rabinovich’s Theorem (Theorem 2.7) in Section 2.

## 2 Labelled Transition Systems and Trace Semantics

Labelled transition systems are a widely used model of nondeterminism. Given a fixed finite set  $A$  of *action labels*, a *labelled transition system* (LTS) is a pair  $(X, \alpha)$  consisting of a set  $X$  of *states* and a *transition function*  $\alpha : X \rightarrow \mathcal{P}(A \times X)$ . We generally write  $x \xrightarrow{a} y$  if  $(a, y) \in \alpha(x)$ , or simply  $x \xrightarrow{a} y$  if  $\alpha$  is clear from context, and say that  $x$  *emits a and transitions to y*.

Given a state  $x$  of an LTS  $(X, \alpha)$ , we write  $\langle x \rangle_\alpha$  for the LTS obtained by restricting the relations  $\xrightarrow{a}$  to the set of states *reachable* from  $x$ , meaning there exists a *path* of the form  $x \xrightarrow{a_1} x_1 \rightarrow \dots \rightarrow x_{n-1} \xrightarrow{a_n} x_n$ . We refer to  $\langle x \rangle_\alpha$  as either the LTS *generated by x*, or as the *process starting at x*. An LTS  $(X, \alpha)$  is *locally finite* if  $\langle x \rangle_\alpha$  is finite for all states  $x$ .

$$ae \xrightarrow{a} e \qquad \frac{e_1 \xrightarrow{a} f}{e_1 + e_2 \xrightarrow{a} f} \qquad \frac{e_2 \xrightarrow{a} f}{e_1 + e_2 \xrightarrow{a} f} \qquad \frac{e[\mu\nu e/v] \xrightarrow{a} f}{\mu\nu e \xrightarrow{a} f}$$

■ **Figure 3** The relation  $\xrightarrow{a} \subseteq \text{Term} \times \text{Term}$  defining  $(\text{Term}, \gamma)$ .

### Traces

In the context of the current work, nondeterminism occurs when a process branches into multiple threads that execute in parallel. Under this interpretation, to an outside observer (without direct access to the implementation details of an LTS), two processes that emit the same set of sequences of action labels are indistinguishable.

Formally, let  $A^*$  be the set of words formed from the alphabet  $A$ . Given a state  $x$  of an LTS  $(X, \alpha)$ , the set  $\text{tr}_\alpha(x)$  of *traces emitted by  $x$*  is the set of words  $a_1 \dots a_n \in A^*$  such that there is a path of the form  $x \xrightarrow{a_1} x_1 \rightarrow \dots \rightarrow x_{n-1} \xrightarrow{a_n} x_n$  through  $(X, \alpha)$ . Two states  $x$  and  $y$  are called *trace equivalent* if  $\text{tr}(x) = \text{tr}(y)$ . Each trace language  $\text{tr}(x)$  is *prefix-closed*, which for a language  $L$  means that  $w \in L$  whenever  $wa \in L$ .

Trace equivalence is a well-documented notion of equivalence for processes [3, 11], and we shall see it in our work on fractals as well.

► **Definition 2.1.** A *stream* is an infinite sequence  $(a_1, a_2, \dots)$  of letters from  $A$ . A state  $x$  in an LTS  $(X, \alpha)$  emits a stream  $(a_1, \dots)$  if for any  $n > 0$ ,  $a_1 \dots a_n \in \text{tr}(x)$ . We write  $\text{str}(x)$  for the set of streams emitted by  $x$ .

In our construction of fractals from LTSs, points are represented only by (infinite) streams. We therefore focus primarily on LTSs with the property that for all states  $x$ ,  $\text{tr}(x)$  is precisely the set of finite prefixes of streams emitted by  $x$ . We refer to an LTS  $(X, \alpha)$  satisfying this condition as *productive*. Productivity is equivalent to the absence of *deadlock* states, states with no outgoing transitions.

► **Lemma 2.2.** Let  $(X, \alpha)$  be an LTS. Then the following are equivalent: (i) for any  $x, y \in X$ ,  $\text{str}(x) = \text{str}(y)$  if and only if  $\text{tr}(x) = \text{tr}(y)$ ; (ii) for any  $x \in X$ ,  $\alpha(x) \neq \emptyset$ .

### Specification

We use the following language for specifying processes: Starting with a fixed countably infinite set  $\{v_1, v_2, \dots\}$  of *variables*, the set of *terms* is given by the grammar

$$v \mid ae \mid e_1 + e_2 \mid \mu\nu e$$

where  $v$  is  $v_i$  for some  $i \in \mathbb{N}$ ,  $a \in A$ , and  $e, e_1, e_2$  are terms.

Intuitively, the process  $ae$  emits  $a$  and then turns into  $e$ , and  $e_1 + e_2$  is the process that nondeterministically branches into  $e_1$  and  $e_2$ . The process  $\mu\nu e$  is like  $e$ , but with instances of  $v$  that appear free in  $e$  acting like `goto` expressions that return the process to  $\mu\nu e$ .

► **Definition 2.3.** A (process) *term* is a term  $e$  in which every occurrence of a variable  $v$  appears both within the scope of a  $\mu\nu (-)$  ( $e$  is closed) and within the scope of an  $a(-)$  ( $e$  is guarded). The set of process terms is written  $\text{Term}$ . The set of process terms themselves form the LTS  $(\text{Term}, \gamma)$  defined in Figure 3.

In Figure 3, we use the notation  $e[g/v]$  to denote the expression obtained by replacing each *free* occurrence of  $v$  in  $e$  (one which does not appear within the scope of a  $\mu\nu (-)$  operator) with the expression  $g$ . Given  $e \in \text{Term}$ , the process *specified by  $e$*  is the LTS  $\langle e \rangle_\gamma$ .

$$\begin{array}{ll}
\text{(ID)} & e + e \equiv e \\
\text{(CM)} & e_2 + e_1 \equiv e_1 + e_2 \\
\text{(AS)} & e_1 + (e_2 + e_3) \equiv (e_1 + e_2) + e_3 \\
\text{(DS)} & a(e_1 + e_2) \equiv ae_1 + ae_2 \\
\text{(FP)} & \mu v e \equiv e[\mu v e/v] \\
\text{(CN)} & \frac{(\forall i) e_i \equiv f_i}{g[\vec{e}/\vec{v}] \equiv g[\vec{f}/\vec{v}]} \\
\text{(AE)} & \frac{}{\mu v e \equiv \mu v e[v/w]} \\
\text{(UA)} & \frac{g \equiv e[g/v]}{g \equiv \mu v e}
\end{array}$$

■ **Figure 4** The axioms and rules of the provable equivalence relation in addition to those of equational logic (not shown). Here,  $e, e_i, f, f_i, g \in \text{Term}$  for all  $i$ . In (CN),  $g$  has precisely the free variables  $v_1, \dots, v_n$ , and no variable that appears free in  $f_i$  is bound in  $g$  for any  $i$ . In (AE),  $v$  does not appear free in  $e$ .

► **Remark 2.4.** The set of process terms, as we have named them, is the fragment of Milner’s fixed-point calculus from [22] consisting of only the terms that specify productive LTSs.

Labelled transition systems specified by process terms are finite and productive, and conversely, every finite productive process is trace-equivalent to some process term.

► **Lemma 2.5** ([22, Proposition 5.1]). *For any  $e \in \text{Term}$ , the set of terms reachable from  $e$  in  $(\text{Term}, \gamma)$  is finite. Conversely, if  $x$  is a state in a finite productive LTS  $(X, \alpha)$ , then there is a process term  $e$  such that  $\text{tr}(e) = \text{tr}_\alpha(x)$ .*

### Axiomatization of trace equivalence

Given an interpretation of process terms as states in an LTS, and given the notion of trace equivalence, one might ask if there is an algebraic or proof-theoretic account of trace equivalence of process terms. Rabinovich showed in [25] that a complete inference system for trace equivalence can be obtained by adapting earlier work of Milner [22]. The axioms of the complete inference system include equations like  $e_1 + e_2 = e_2 + e_1$  and  $a(e_1 + e_2) = ae_1 + ae_2$ , which are intuitively true for trace equivalence.

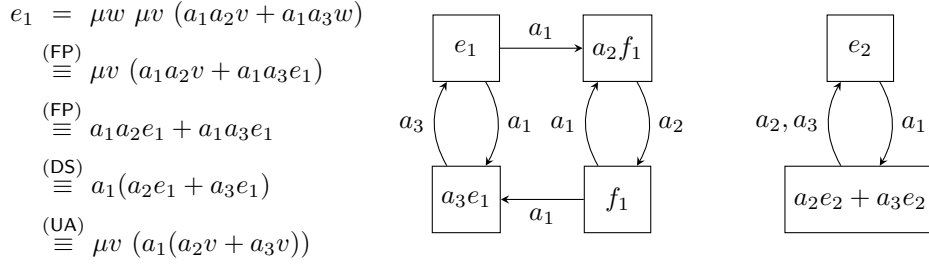
To be more precise, given any function with domain  $\text{Term}$ , say  $\sigma : \text{Term} \rightarrow Z$ , call an equivalence relation  $\sim$  *sound with respect to  $\sigma$*  if  $e \sim f$  implies  $\sigma(e) = \sigma(f)$ , and *complete with respect to  $\sigma$*  if  $\sigma(e) = \sigma(f)$  implies  $e \sim f$ . Then the smallest equivalence relation  $\equiv$  on  $\text{Term}$  containing all the pairs derivable from the axioms and inference rules appearing in Figure 4 is sound and complete with respect to  $\text{tr} = \text{tr}_\gamma : \text{Term} \rightarrow \mathcal{P}(A^*)$ .

► **Definition 2.6.** *Given  $e_1, e_2 \in \text{Term}$ , we say that  $e_1$  and  $e_2$  are provably equivalent if  $e_1 \equiv e_2$ , and call  $\equiv$  provable equivalence.*

► **Theorem 2.7** (Rabinovich [25]). *Let  $e_1, e_2 \in \text{Term}$ . Then  $e_1 \equiv e_2$  iff  $\text{tr}(e_1) = \text{tr}(e_2)$ .*

► **Example 2.8.** Consider the processes specified by  $e_1 = \mu v \mu w (a_1 a_2 v + a_1 a_3 w)$  and  $e_2 = \mu v (a_1 (a_2 v + a_3 v))$ . The traces emitted by both  $e_1$  and  $e_2$  are those that alternate between  $a_1$  and either  $a_2$  or  $a_3$ . We can show these expressions are trace equivalent via the formal deduction in Figure 5.

Rabinovich’s theorem tells us that, up to provable equivalence, our specification language consisting of process terms is really a specification language for languages of traces. In what follows, we are going to give an alternative semantics to process terms by using LTSs to



■ **Figure 5** Deducing  $e_1 \equiv e_2$ . Above,  $f_1 = \mu v (a_1 a_2 v + a_1 a_3 e_1)$ .

generate fractal subsets of metric spaces. The main result of our paper is that these two semantics coincide: Two process terms are trace equivalent if and only if they generate the same fractals. This is the content of Sections 3 and 4 below.

### 3 Fractals from Labelled Transition Systems

In the Sierpiński gasket  $\mathbf{S}$  from Figure 1, every point of  $\mathbf{S}$  corresponds to a stream of letters from the alphabet  $\{a, b, c\}$ , and every stream corresponds to a unique point. To obtain the point corresponding to a particular stream  $(a_1, a_2, a_3, \dots)$  with each  $a_i \in \{a, b, c\}$ , start with any  $p \in \mathbb{R}^2$  and compute the limit  $\lim_{n \in \mathbb{N}} \sigma_{a_1} \cdots \sigma_{a_n}(p)$ . The point in the fractal corresponding to  $(a_1, a_2, a_3, \dots)$  does not depend on  $p$  because  $\sigma_a, \sigma_b, \sigma_c$  in Figure 1 are contraction operators.

► **Definition 3.1.** Given a metric space  $(M, d)$ , a contraction operator on  $(M, d)$  is a function  $h : M \rightarrow M$  such that for some  $r \in [0, 1)$ ,  $d(h(x), h(y)) \leq r d(x, y)$  for any  $x, y \in M$ . The number  $r$  is called a contraction coefficient of  $h$ . The set of contraction operators on  $(M, d)$  is written  $\text{Con}(M, d)$ .

For example, with the Sierpiński gasket (Figure 1) associated to the contractions  $\sigma_a, \sigma_b$ , and  $\sigma_c$ ,  $r = 1/2$  is a contraction coefficient for all three maps. Now, given  $p, q \in \mathbb{R}^2$ ,

$$d(\sigma_{a_1} \cdots \sigma_{a_n}(p), \sigma_{a_1} \cdots \sigma_{a_n}(q)) \leq \frac{1}{2^n} d(p, q)$$

for all  $n$ , so it follows that  $\lim_{n \in \mathbb{N}} \sigma_{a_1} \cdots \sigma_{a_n}(p) = \lim_{n \in \mathbb{N}} \sigma_{a_1} \cdots \sigma_{a_n}(q)$ . For any finite set of contraction operators  $\{\sigma_{a_1}, \dots, \sigma_{a_n}\}$  indexed by  $A$  and acting on a complete metric space  $(M, d)$ , every stream from  $A$  corresponds to a unique point in  $M$ .

► **Definition 3.2.** A contraction operator interpretation is a function  $\sigma : A \rightarrow \text{Con}(M, d)$ . We usually write  $\sigma_a = \sigma(a)$ . Given  $\sigma : A \rightarrow \text{Con}(M, d)$  and a stream  $(a_1, \dots)$  from  $A$ , define

$$\sigma_\omega : A^\omega \rightarrow M \quad \sigma_\omega(a_1, \dots) = \lim_{n \in \mathbb{N}} \sigma_{a_1} \cdots \sigma_{a_n}(x) \tag{2}$$

where  $x \in M$  is arbitrary. The self-similar set corresponding to a contraction operator interpretation  $\sigma$  is the set  $\mathbf{S}_\sigma = \{\sigma_\omega(a_1, \dots) \mid (a_1, \dots) \text{ is a stream from } A\}$ .

► **Remark 3.3.** Note that in (2), the contraction operators corresponding to the initial trace  $(a_1, \dots, a_n)$  are applied in reverse order. That is,  $\sigma_{a_n}$  is applied before  $\sigma_{a_{n-1}}$ ,  $\sigma_{a_{n-2}}$  is applied before  $\sigma_{a_{n-1}}$ , and so on.



## Regular Subfractals

Generalizing the fractals of Mandelbrot [17], Hutchinson introduced self-similar sets in [13] and gave a comprehensive account of their theory. In op. cit., Hutchinson defines a self-similar set to be the invariant set of an *iterated function system*. In our terminology, an iterated function system is equivalent to a contraction operator interpretation of a finite set  $A$  of actions, and the invariant set is the total set of points obtained from streams from  $A$ . The fractals constructed from a LTS paired with a contraction operator interpretation generalize Hutchinson's self-similar sets to nonempty compact sets of points obtained from certain subsets of the streams, namely the subsets emitted by the LTS.

Write  $\mathbf{K}(M, d)$  for the set of nonempty compact subsets of  $(M, d)$ . Given a state  $x$  of a productive LTS  $(X, \alpha)$  and a contraction operator interpretation  $\sigma : A \rightarrow \text{Con}(M, d)$ , we define  $\llbracket - \rrbracket_{\alpha, \sigma} : X \rightarrow \mathbf{K}(M, d)$  by

$$\llbracket x \rrbracket_{\alpha, \sigma} = \{ \sigma_\omega(a_1, \dots) \mid (a_1, \dots) \text{ emitted by } x \} \quad (3)$$

and call this the set *generated by the state*  $x$ . As we will see,  $\llbracket x \rrbracket_{\alpha, \sigma}$  is always nonempty and compact.

► **Definition 3.4.** *Given a process term  $e \in \text{Term}$  and a contraction operator interpretation  $\sigma : A \rightarrow \text{Con}(M, d)$ , the regular subfractal semantics of  $e$  corresponding to  $\sigma$  is  $\llbracket e \rrbracket_\sigma = \llbracket e \rrbracket_{\gamma, \sigma}$ .*

For example, the set of points depicted in Figure 2 is the regular subfractal semantics of  $\mu v (av + b(bv + cv) + c(bv + cv))$  corresponding to the interpretation  $\sigma$  given in that figure. The regular subfractal semantics of  $e$  is a proper subset of the Sierpiński Gasket, and in particular does not contain the point corresponding to  $(c, a, b, c, b, c, \dots)$ .

## Systems and Solutions

Self-similar sets are often characterized as the unique nonempty compact sets that solve systems of equations of the form

$$K = \sigma_1(K) \cup \dots \cup \sigma_n(K)$$

with each  $\sigma_i$  a contraction operator on a complete metric space. For example, the Sierpiński gasket is the unique nonempty compact solution to  $K = \sigma_a(K) \cup \sigma_b(K) \cup \sigma_c(K)$ . In this section, we are going to provide a similar characterization for regular subfractals that will play an important role in the completeness proof in Section 4.

One way to think of an  $n$ -state LTS  $(X, \alpha)$  is as a system of formal equations

$$x_i = a_{k_1} x_{j_1} + \dots + a_{k_m} x_{j_m}$$

indexed by  $X = \{x_1, \dots, x_n\}$ , where  $x_i \xrightarrow{a_{k_l}}_\alpha x_{j_l}$  for  $k_1, \dots, k_m, j_1, \dots, j_m \leq n$ .

► **Definition 3.5.** *Given a contraction operator interpretation  $\sigma : A \rightarrow \text{Con}(M, d)$ , and an LTS  $(X, \alpha)$ , we call a function  $\varphi : X \rightarrow \mathbf{K}(M, d)$  a  $(\sigma)$ -solution to  $(X, \alpha)$  if for any  $x \in X$ ,*

$$\varphi(x) = \bigcup_{x \xrightarrow{a}_{\alpha} y} \sigma_a(\varphi(y))$$

► **Example 3.6.** Let  $\mathbf{S}$  be the Sierpiński gasket as a subset of  $\mathbb{R}^2$ . Let  $(X, \alpha)$  be the LTS in Figure 1. Then we have a single state,  $x$ , with  $x \xrightarrow{a, b, c}_\alpha x$ . The function  $\varphi : X \rightarrow \mathbf{K}(\mathbb{R}^2, d)$  given by  $\varphi(s) = \mathbf{S}$  is a solution to  $(X, \alpha)$ , because  $\mathbf{S} = \sigma_a(\mathbf{S}) \cup \sigma_b(\mathbf{S}) \cup \sigma_c(\mathbf{S})$ .

## 14:8 Fractals from Regular Behaviours

Finite productive LTSs have unique solutions.

► **Lemma 3.7.** *Let  $(M, d)$  be a complete metric space,  $\sigma : A \rightarrow \text{Con}(M, d)$ , and  $(X, \alpha)$  be a finite productive LTS. Then  $(X, \alpha)$  has a unique solution  $\varphi_\alpha$ .*

The proof of Lemma 3.7 makes use of the *Hausdorff metric* on  $\mathbf{K}(M, d)$ , defined

$$d(K_1, K_2) = \max \left\{ \sup_{u \in K_1} \inf_{v \in K_2} d(u, v), \sup_{v \in K_2} \inf_{u \in K_1} d(u, v) \right\} \quad (4)$$

This equips  $\mathbf{K}(M, d)$  with the structure of a metric space. If  $M$  is complete, so is  $\mathbf{K}(M, d)$ . Incidentally, we need to restrict to *nonempty* sets in (4). This is the primary motivation for the guardedness condition which we imposed on our terms. We also recall the *Banach fixed-point theorem*, which allows for the computation of fixed-points by iteration.

► **Theorem 3.8** (Banach [2]). *Let  $(M, d)$  be a complete nonempty metric space and  $f : M \rightarrow M$  a contraction map. Then  $\lim_{n \in \mathbb{N}} f^n(q)$  is the unique fixed-point of  $f$ .*

Fix a complete nonempty metric space  $(M, d)$ , a productive finite LTS  $(X, \alpha)$ , and a contraction operator interpretation  $\sigma : A \rightarrow \text{Con}(M, d)$ . To compute the solution to  $(X, \alpha)$ , we iteratively apply a matrix-like operator to the set  $\mathbf{K}(M, d)^X$  of vectors  $[K_{x_1}, \dots, K_{x_n}]$  with entries in  $\mathbf{K}(M, d)$  indexed by  $X$ . Formally, we define

$$[\alpha]_\sigma : \mathbf{K}(M, d)^X \rightarrow \mathbf{K}(M, d)^X \quad ([\alpha]_\sigma \vec{K})_x = \bigcup_{x \xrightarrow{\alpha} y} \sigma_a(K_y)$$

for each  $x \in X$ . Intuitively,  $[\alpha]_\sigma$  acts like an  $X \times X$ -matrix of unions of contractions.

**Proof of Lemma 3.7.** Every fixed-point of  $[\alpha]_\sigma$  corresponds to a solution of  $(X, \alpha)$ . Given a fixed-point  $\vec{F}$ , i.e.,  $[\alpha]_\sigma \vec{F} = \vec{F}$ , and defining  $\varphi : X \rightarrow \mathbf{K}(M, d)^X$  by  $\varphi(x) = F_x$ , we see that

$$\varphi(x) = F_x = ([\alpha]_\sigma \vec{F})_x = \bigcup_{x \xrightarrow{\alpha} y} \sigma_a(F_y) = \bigcup_{x \xrightarrow{\alpha} y} \sigma_a(\varphi(y))$$

Conversely, if  $\varphi : X \rightarrow \mathbf{K}(M, d)$  is a solution to  $(X, \alpha)$ , then defining  $F_x = \varphi(x)$  we have

$$F_x = \varphi(x) = \bigcup_{x \xrightarrow{\alpha} y} \sigma_a(\varphi(y)) = \bigcup_{x \xrightarrow{\alpha} y} \sigma_a(F_y) = ([\alpha]_\sigma \vec{F})_x$$

for each  $x \in X$ . Thus, it suffices to show that  $[\alpha]_\sigma$  has a unique fixed-point. By the Banach Fixed-Point Theorem 3.8, we just need to show that  $[\alpha]_\sigma$  is a contraction operator. That is,  $[\alpha]_\sigma \in \text{Con}(\mathbf{K}(M, d))$ , where  $d$  is the Hausdorff metric. This point is standard in the fractals literature; cf. [13]. ◀

### Fractal Semantics and Solutions

Recall that the fractal semantics of a process term  $e$  with respect to a contraction operator interpretation  $\sigma : A \rightarrow \text{Con}(M, d)$  is the set  $\llbracket e \rrbracket_\sigma$  of limits of streams applied to points in the complete metric space  $(M, d)$ .

► **Theorem 3.9.** *Let  $(X, \alpha)$  be a finite productive LTS and let  $x \in X$ . Given a complete metric space  $(M, d)$ , and  $\sigma : A \rightarrow \text{Con}(M, d)$ ,*

1.  $\llbracket x \rrbracket_{\alpha, \sigma} \in \mathbf{K}(M, d)$ , i.e.,  $\llbracket x \rrbracket_{\alpha, \sigma}$  is nonempty and compact.
2.  $\llbracket - \rrbracket_{\alpha, \sigma} : X \rightarrow \mathbf{K}(M, d)$  is the unique solution to  $(X, \alpha)$ .

In particular,  $(\text{Term}, \gamma)$  is locally finite, and so by Lemma 3.7 has a unique solution. Theorem 3.9 therefore implies that this solution is  $\llbracket - \rrbracket_\sigma$ .

Given a solution  $\varphi$  and a state  $x$ , call  $\varphi(x)$  the  $x$ -component of the solution  $\varphi$ . We obtain the following, which can be seen as an analogue of Kleene's theorem for regular expressions [15], as a direct consequence of Theorem 3.9.

► **Theorem 3.10.** *A subset of a self-similar set is a regular subfractal if and only if it is a component of a solution to a finite productive LTS.*

## 4 Fractal Equivalence is Traced

We have seen that finite productive LTSs (LTSs that only emit infinite streams) can be specified by process terms. We also introduced a family of fractal sets called regular subfractals, those subsets of self-similar sets obtained from the streams emitted by a finite productive LTS. An LTS itself is representative of a certain system of equations, and set-wise the system of equations is solved by the regular subfractals corresponding to it. Going from process terms to LTSs to regular subfractals, we see that a process term is representative of a sort of *uninterpreted fractal recipe*, which tells us how to obtain a regular subfractal from an interpretation of action symbols as contractions on a complete metric space.

► **Definition 4.1.** *Given  $e, f \in \text{Term}$ , we write  $e \approx f$  if for every complete metric space  $(M, d)$  and every contraction operator interpretation  $\sigma : A \rightarrow \text{Con}(M, d)$ ,  $\llbracket e \rrbracket_\sigma = \llbracket f \rrbracket_\sigma$ . We say that  $e$  and  $f$  are fractal equivalent or that they are equivalent fractal recipes when  $e \approx f$ .*

► **Theorem 4.2.** *Let  $e, f \in \text{Term}$ . Then  $e \approx f$  if and only if  $\text{str}(e) = \text{str}(f)$ .*

In essence, this is a soundness/completeness theorem for our version of Rabinovich's logic with respect to its fractal semantics that we presented. Our proof relies on the logical characterization of trace equivalence that we saw in Theorem 2.7.

► **Lemma 4.3 (Soundness).** *For any  $e, f \in \text{Term}$ , if  $e \equiv f$ , then  $e \approx f$ .*

► **Theorem 4.4 (Completeness).** *For any  $e, f \in \text{Term}$ , if  $e \approx f$ , then  $e \equiv f$ .*

**Proof.** Consider the space  $(A^\omega, d)$  of streams from  $A$  with the metric below:

$$d((a_1, \dots), (b_1, \dots)) = \inf \{ 2^{-n} \mid (\forall i \leq n) a_i = b_i \}$$

This space is the *Cantor set on  $A$  symbols*, a compact metric space. For any productive LTS  $(X, \alpha)$  and  $x \in X$ ,  $\text{str}(x)$  is a nonempty closed subset of  $(A^\omega, d)$ , for the following reason: Given a Cauchy sequence  $\{(a_1^{(i)}, \dots)\}_{i \in \mathbb{N}}$  in  $\text{str}(x)$ , let  $(a_1, \dots)$  be its limit in  $(A^\omega, d)$ . Then  $x$  emits every finite initial segment of  $(a_1, \dots)$  because for any  $N \in \mathbb{N}$  there is an  $m \in \mathbb{N}$  such that  $(a_1, \dots, a_m, a_{m+1}^{(m)}, \dots) \in \text{str}(x)$  for  $m > N$ . By compactness of  $(A^\omega, d)$ , we therefore have  $\text{str}(x) \in \mathbf{K}(A^\omega, d)$ , so  $\text{str} : X \rightarrow \mathbf{K}(A^\omega, d)$ .

For each  $a \in A$ , let  $\sigma_a : A^\omega \rightarrow A^\omega$  be the map  $\sigma_a(a_1, \dots) = (a, a_1, \dots)$ . Then  $\sigma : A \rightarrow \text{Con}(A^\omega, d)$ . By construction,  $\text{str}(x) = \bigcup_{x \rightarrow y} \sigma_a(\text{str}(y))$  for any  $x \in X$ . By the uniqueness of fixed points we saw in Lemma 3.7, we therefore have  $\text{str}(x) = \llbracket x \rrbracket_{\alpha, \sigma}$ .

To finish the proof, consider  $(\text{Term}, \gamma)$ . If  $e, f \in \text{Term}$  and  $e \approx f$ , then in particular,  $\text{str}(e) = \text{str}(f)$ , because  $\text{str} = \llbracket - \rrbracket_{\gamma, \sigma}$  with  $\sigma : A \rightarrow \text{Con}(A^\omega, d)$  as above. Since  $(\text{Term}, \gamma)$  is productive,  $\text{tr}(e) = \text{str}(e)$  and  $\text{tr}(f) = \text{str}(f)$ , so in particular,  $e$  and  $f$  are trace equivalent. By Rabinovich's Theorem, Theorem 2.7,  $e \equiv f$ , as desired. ◀

## 5 A Calculus of Subfractal Measures

Aside from showing the existence of self-similar sets and their correspondence with contraction operator interpretations (in Hutchinson's terminology, iterated function systems), Hutchinson also shows that every probability distribution on the contractions corresponds to a unique measure, called the *invariant measure*, that satisfies a certain recursive equation and whose support is the self-similar set. In this section, we replay the story up to this point, but with Hutchinson's invariant measure construction instead of the invariant (self-similar) set construction. We make use of a probabilistic version of LTSs called *labelled Markov chains*, as well as a probabilistic version of Milner's specification language introduced by Stark and Smolka [27] to specify fractal measures. Similar to how fractal equivalence coincides with trace equivalence, *fractal measure equivalence* is equivalent to a probabilistic version of trace equivalence due to Kerstan and König [14].

### Invariant measures

Recall that a *Borel probability measure* on a metric space  $(M, d)$  is a  $[0, \infty]$ -valued function  $\rho$  defined on the Borel subsets of  $M$  (the smallest  $\sigma$ -algebra containing the open balls of  $(M, d)$ ) that is countably additive and assigns  $\rho(\emptyset) = 0$  and  $\rho(M) = 1$ .

Hutchinson shows in [13] that, given  $\sigma: A \rightarrow \text{Con}(M, d)$ , each probability distribution  $\rho: A \rightarrow [0, 1]$  on  $A$  gives rise to a unique Borel probability measure  $\hat{\rho}$ , called the *invariant measure*, satisfying the equation below and supported by the self-similar set  $\mathbf{S}_\sigma$ :

$$\hat{\rho}(B) = \sum_{a \in A} \rho(a) \sigma_a^\# \hat{\rho}(B)$$

Here and elsewhere, the *pushforward measure*  $f^\# \hat{\rho}$  with respect to a continuous map  $f$  is defined by  $f^\# \hat{\rho}(B) = \hat{\rho}(f^{-1}(B))$  for any Borel subset  $B$  of  $(M, d)$ .

We can view the specification  $\rho$  of the invariant measure  $\hat{\rho}$  as a one-state Markov process with a loop labelled with each letter from  $A$ , similar to how self-similar sets are specified with a one-state productive LTS. We can adapt this construction to multiple states by moving from probability distributions on  $A$  to *labelled Markov chains*, where again, the labels are interpreted as contraction maps.

### Labelled Markov Chains

Let  $\mathcal{D}$  denote the finitely supported probability distribution functor on the category of sets.

► **Definition 5.1.** A labelled Markov chain (LMC) is a pair  $(X, \beta)$  consisting of a set  $X$  of states and a function  $\beta: X \rightarrow \mathcal{D}(A \times X)$ . A homomorphism of LMCs  $h: (X, \beta_X) \rightarrow (Y, \beta_Y)$  is a function  $h: X \rightarrow Y$  such that  $\mathcal{D}(h) \circ \beta_X = \beta_Y \circ h$ . We write  $x \xrightarrow{r|a}_\beta y$  if  $\beta(x)(a, y) = r$ , often dropping the symbol  $\beta$  if it is clear from context.

As we have already seen, given a contraction operator interpretation  $\sigma: A \rightarrow \text{Con}(M, d)$ , every state  $x$  of a productive LTS  $(X, \alpha)$  with labels in  $A$  corresponds to a regular subfractal  $\llbracket x \rrbracket_{\alpha, \sigma}$  of  $\mathbf{S}_\sigma$ . This regular subfractal is defined to be the continuous image of the set  $\text{str}(x)$  under the map  $\sigma_\omega: (A^\omega, d_\sigma) \rightarrow (M, d)$ , where  $d_\sigma$  is determined by the contraction coefficients of the  $\sigma_a$ 's as follows: Given a nonzero contraction coefficient  $c_a$  of  $\sigma_a$  for each  $a \in A$ , define  $d_\sigma((a_1, \dots), (b_1, \dots)) = \prod_{i=1}^n c_{a_i}$ , where  $n$  is the least index such that  $a_{n+1} \neq b_{n+1}$ . The family  $\llbracket x \rrbracket_{\alpha, \sigma}$  is characterized by its satisfaction of the equations representing the LTS  $(X, \alpha)$ .

Every LMC  $(X, \beta)$  has an *underlying* LTS  $(X, \bar{\beta})$ , where  $\bar{\beta}(x) = \{(a, y) \mid \beta(x)(a, y) > 0\}$ . For each  $x \in X$ , we are going to define a probability measure  $\hat{\beta}_\sigma(x)$  on  $\mathbf{S}_\sigma$  whose support is  $\llbracket x \rrbracket_{\bar{\beta}, \sigma}$ , and that satisfies a recursive system of equations represented by the LMC  $(X, \beta)$ . Roughly,  $\hat{\beta}_\sigma(x)$  is the pushforward of a certain Borel probability measure  $\hat{\beta}(x)$  on  $A^\omega$  that does not depend on the contraction operator interpretation  $\sigma$ .

We begin by topologizing  $A^\omega$ , using as a basis the sets of the form

$$B_{a_1 \dots a_n} = \{(a_1, \dots, a_n, b_1, \dots) \mid (b_1, \dots) \in A^\omega\}$$

Given a state  $x$  of a LMC  $(X, \beta)$  and a word  $w = a_1 \dots a_n$ , we follow Kerstan and König [14] and define the *trace measure* of the basic open set  $B_w$  by

$$\hat{\beta}(x)(B_w) = \sum \{r_1 \dots r_n \mid x \xrightarrow{r_1|a_1} x_1 \rightarrow \dots \xrightarrow{r_n|a_n} x_n\} \quad (5)$$

where  $\hat{\beta}(B_\epsilon) = \hat{\beta}(A^\omega) = 1$ . This defines a unique Borel probability measure on  $(A^\omega, d)$ .

► **Proposition 5.2.** *Let  $j: A^* \rightarrow [0, 1]$  satisfy  $j(w) = \sum_{a \in A} j(wa)$  for any  $w \in A^*$  and  $j(\epsilon) = 1$ , where  $\epsilon$  is the empty word. Then there is a unique Borel probability measure  $\rho$  on  $(A^\omega, d)$  such that for any  $w \in A^*$ ,  $\rho(B_w) = j(w)$ .*

**Proof.** This is an easy consequence of the Identity and Extension Theorems for  $\sigma$ -finite premeasures. See Propositions 2.3 to 2.5 of [14]. ◀

In particular, given any LMC  $(X, \beta)$ ,  $\hat{\beta}(x)(B_w) = \sum_{a \in A} \hat{\beta}(x)(B_wa)$ , so there is a unique Borel probability measure  $\hat{\beta}(x)$  on  $A^\omega$  such that (5) holds for any basic open set  $B_w$ .

► **Definition 5.3.** *Let  $(X, \beta)$  be a LMC, and  $\sigma: A \rightarrow \text{Con}(M, d)$  be a contraction operator interpretation in a complete metric space. For each  $x \in X$ , we define the regular subfractal measure corresponding to  $x$  to be  $\hat{\beta}_\sigma(x) = \sigma_a^\# \hat{\beta}(x)$ .*

Intuitively, the regular subfractal measure of a state in a LMC under a contraction operator interpretation computes the probability that, if run stochastically according to the probabilities labelling the edges, the sequence of points of  $M$  observed in the run eventually lands within a given Borel subset of  $(M, d)$ .

### Systems of Probabilistic Equations

Given a complete metric space  $(M, d)$ , let  $\mathbf{P}(M, d)$  be the set of Borel probability measures on  $(M, d)$ . In previous sections, we made use of the fact that, when  $\sigma: A \rightarrow \text{Con}(M, d)$ , we can see  $\mathbf{K}(M, d)$  as a *semilattice with operators*, i.e., union acts as a binary operation  $\cup: \mathbf{K}(M, d)^2 \rightarrow \mathbf{K}(M, d)$  and each  $\sigma_a: \mathbf{K}(M, d) \rightarrow \mathbf{K}(M, d)$  distributes over  $\cup$ . Analogously, equipped with  $\sigma: A \rightarrow \text{Con}(M, d)$ ,  $\mathbf{P}(M, d)$  is a *convex algebra with operators*. Formally, for any  $r \in [0, 1]$ , there is a binary operation  $\oplus_r: \mathbf{P}(M, d)^2 \rightarrow \mathbf{P}(M, d)$  defined  $(\rho_1 \oplus_r \rho_2)(B) = r\rho_1(B) + (1-r)\rho_2(B)$ , over which each  $\sigma_a^\#$  distributes, i.e.,

$$\sigma_a^\#(\rho_1 \oplus_r \rho_2) = \sigma_a^\# \rho_1 \oplus_r \sigma_a^\# \rho_2$$

We also make use of a summation notation defined by

$$r_1 \cdot \rho_1 \oplus \dots \oplus r_n \cdot \rho_n = \rho_n \oplus_{r_n} \left( \frac{r_1}{1-r_n} \cdot \rho_1 \oplus \dots \oplus \frac{r_{n-1}}{1-r_n} \cdot \rho_{n-1} \right)$$

for any  $r_1, \dots, r_n \in [0, 1)$ .

## 14:12 Fractals from Regular Behaviours

Given a contraction operator interpretation, an LMC  $(X, \beta)$  can be thought of as a system of equations with one side a polynomial term in a convex algebra with operators,

$$x_i = r_{i1} \cdot a_{i1}x_{k_1} \oplus r_{i2} \cdot a_{i2}x_{k_2} \oplus \cdots \oplus r_{im} \cdot a_{im}x_{k_m}$$

where  $X = \{x_1, \dots, x_n\}$  and  $x_i \xrightarrow{r_{ij}|a_{ij}} x_{k_m}$  for each  $i, j \leq m$ .

► **Definition 5.4.** Let  $(X, \beta)$  be a LMC, and let  $\sigma: A \rightarrow \text{Con}(M, d)$ . A solution to  $(X, \beta)$  is a function  $\varphi: X \rightarrow \mathbf{P}(M, d)$  such that for any  $x \in X$  and any Borel set  $B$ ,

$$\varphi(x)(B) = \sum_{x \xrightarrow{r|a} y} r \sigma_a^\#(\varphi(y))(B)$$

Every finite LMC admits a unique solution, and moreover, the unique solution is the regular subfractal measure from Definition 5.3.

► **Theorem 5.5.** Let  $(X, \beta)$  be a LMC,  $x \in X$ , and  $\sigma: A \rightarrow \text{Con}(M, d)$ . Then the map  $\hat{\beta}_\sigma: X \rightarrow \mathbf{P}(M, d)$  is the unique solution to  $(X, \beta)$ .

Since the support of  $\hat{\beta}(x)$  is precisely  $\text{str}(x)$ , the support of  $\hat{\beta}_\sigma(x)$  is precisely  $\sigma_\omega(\text{str}(x))$ , which we have already seen is the regular subfractal determined by the state  $x$  of the underlying LTS of  $(X, \beta)$ .

### Probabilistic Process Algebra

Finally, we introduce a syntax for specifying LMCs. Our specification language is essentially the *productive* fragment of Stark and Smolka's process calculus [27], meaning that the expressions do not involve deadlock and all variables are guarded.

► **Definition 5.6.** The set of probabilistic terms is given by the grammar

$$v \mid ae \mid e_1 \oplus_r e_2 \mid \mu v e$$

Here  $r \in [0, 1]$ , and otherwise we make the same stipulations as in Definition 2.3. The set of probabilistic process terms  $\mathbf{PTerm}$  consists of the closed and guarded probabilistic terms.

Instead of *languages* of streams, the analog of trace semantics appropriate for probabilistic process terms is a measure-theoretic semantics consisting of trace measures introduced earlier in this section (Equation (5)).

► **Definition 5.7.** We define the LMC  $(\mathbf{PTerm}, \delta)$  in Figure 6 and call it the syntactic LMC. The trace measure semantics  $\text{trm}(e)$  of a probabilistic process term  $e$  is defined to be  $\text{trm}(e) = \hat{\delta}(x)$ . Given  $\sigma: A \rightarrow \text{Con}(M, d)$ , the subfractal semantics of  $e \in \mathbf{PTerm}$  corresponding to  $\sigma$  is  $\hat{\delta}_\sigma(e)$ .

Intuitively, the trace measure semantics of a process term  $e$  assigns a Borel set of streams  $B$  the probability that  $e$  eventually emits a word in  $B$ . Trace measure semantics can be computed inductively as follows.

► **Lemma 5.8.** For any  $w \in A^*$ ,  $a \in A$ ,  $e, e_i \in \mathbf{PTerm}$ , and  $r \in [0, 1]$ ,  $\text{trm}(e)(A^\omega) = 1$  and

$$\text{trm}(ae)(B_w) = \begin{cases} \text{trm}(e)(B_u) & w = au \\ 0 & \text{otherwise} \end{cases}$$

$$\text{trm}(e_1 \oplus_r e_2)(B_w) = r \text{trm}(e_1)(B_w) + (1 - r) \text{trm}(e_2)(B_w)$$

$$\text{trm}(\mu v e)(B_w) = \text{trm}(e[\mu v e/v])(B_w)$$

$$\delta(ae)(b, f) = \begin{cases} 1 & f = e \text{ and } b = a \\ 0 & \text{otherwise} \end{cases}$$

$$\delta(e_1 \oplus_r e_2)(b, f) = r\delta(e_1)(b, f) + (1-r)\delta(e_2)(b, f)$$

$$\delta(\mu v e)(b, f) = \delta(e[\mu v e/v])(b, f)$$

■ **Figure 6** The LMC structure  $(\text{PTerm}, \delta)$ . Above,  $a, b \in A$ ,  $\sum r_i = 1$ , and  $e, e_i, f \in \text{PTerm}$ .

(ID)	$e \oplus_r e \equiv e$	(CN)	$\frac{(\forall i) e_i \equiv f_i}{g[\vec{e}/\vec{v}] \equiv g[\vec{f}/\vec{v}]}$
(CM)	$e_1 \oplus_r e_2 \equiv e_2 \oplus_{1-r} e_1$	(AE)	$\frac{\mu w e \equiv \mu v e[v/w]}{g \equiv e[g/v]}$
(AS)	$(e_1 \oplus_r e_2) \oplus_s e_3 \equiv e_1 \oplus_{rs} (e_2 \oplus_{\frac{s(1-r)}{1-rs}} e_3)$	(UA)	$\frac{g \equiv e[g/v]}{g \equiv \mu v e}$
(DS)	$a(e_1 \oplus_r e_2) \equiv ae_1 \oplus_r ae_2$		
(FP)	$\mu v e \equiv e[\mu v e/v]$		

■ **Figure 7** Axioms for probabilistic trace equivalence. Above,  $e, e_1, e_2 \in \text{PTerm}$ ,  $a \in A$ ,  $r, s \in [0, 1]$ , and  $rs \neq 1$ . Also, in (AE),  $v$  is not free in  $e$ .

Similar to the situation with trace semantics and regular subfractals, trace measure semantics and subfractal measure semantics identify the same probabilistic process terms.

► **Theorem 5.9.** *Let  $e, f \in \text{PTerm}$ . Then  $\text{trm}(e) = \text{trm}(f)$  if and only if for any contraction operator interpretation  $\sigma: A \rightarrow \text{Con}(M, d)$ ,  $\hat{\delta}_\sigma(e) = \hat{\delta}_\sigma(f)$ .*

### Axiomatization

Figure 7 outlines an inference system for determining when the subfractal measures corresponding to two expressions coincide.

► **Definition 5.10.** *Given  $e, f \in \text{PTerm}$ , write  $e \equiv f$  and say that  $e$  and  $f$  are provably equivalent if the equation  $e = f$  can be derived from inference rules in Figure 7.*

► **Theorem 5.11 (Soundness).** *For any  $e, f \in \text{PTerm}$ , if  $e \equiv f$ , then for any complete metric space  $(M, d)$  and any  $\sigma: A \rightarrow \text{Con}(M, d)$ ,  $\hat{\delta}_\sigma(e) = \hat{\delta}_\sigma(f)$ .*

Unlike the situation with trace equivalence, it is not known if these axioms are complete with respect to subfractal measure semantics. We leave this as a conjecture.

► **Conjecture 5.12 (Completeness).** *Figure 7 is a complete axiomatization of trace measure semantics. That is, for any  $e, f \in \text{PTerm}$ , if for any complete metric space  $(M, d)$  and any  $\sigma: A \rightarrow \text{Con}(M, d)$  we have  $\hat{\delta}_\sigma(e) = \hat{\delta}_\sigma(f)$ , then  $e \equiv f$ .*

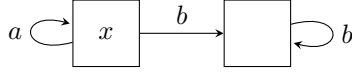
We expect that Conjecture 5.12 can be proven in a similar manner to Theorem 4.4.

## 6 A Question about Regular Subfractals

Certain regular subfractals that have been generated by LTSs with multiple states happen to coincide with self-similar sets using a different alphabet of action symbols and under a different contraction operator interpretation. For example, the twisted Sierpiński gasket in Figure 2 is the self-similar set generated by the iterated function system consisting of the compositions  $\sigma_a, \sigma_b\sigma_b, \sigma_b\sigma_c, \sigma_c\sigma_b$ , and  $\sigma_c\sigma_c$ .

► **Question 1.** Is every regular subfractal a self-similar set? In other words, are there regular subfractals that can only be generated by a multi-state LTS?

► **Example 6.1.** To illustrate the subtlety of this question, consider the following LTS.



The state  $x$  emits  $(a, a, \dots)$  (an infinite stream of  $a$ 's) and  $(a, \dots, a, b, b, \dots)$ , a stream with some finite number (possibly 0) of  $a$ 's followed by an infinite stream of  $b$ 's. Now let  $M = \mathbb{R}$  with Euclidean distance and consider the contraction operator interpretation  $\sigma_a(r) = \frac{1}{2}r$  and  $\sigma_b(r) = \frac{1}{2}r + \frac{1}{2}$ . Let  $K = \{0\} \cup \{\frac{1}{2^n} \mid n \geq 0\}$ . Then  $K$  is the component of the solution at  $x$ . This example is interesting because unlike the Twisted Sierpiński gasket in Figure 2, there is no obvious finite set of compositions  $\sigma_a$  and  $\sigma_b$  such that  $K$  is the self-similar set generated by that iterated function system.

There is an LTS  $(X, \alpha)$  with  $X$  a singleton set  $\{x\}$ , and a contraction operator interpretation  $\sigma_x$  whose solution is  $K$ . We take the set of action labels underlying  $X$  to be  $B = \{f, g, h\}$  and use the contraction operator interpretation  $\sigma_f(r) = 0$ ,  $\sigma_g(r) = 1$  and  $\sigma_h(r) = \frac{1}{2}r$ . It is easy to verify that  $K = \bigcup_{i \in \{f, g, h\}} \sigma_i(K)$ .

But we claim that  $K$  is not obtainable using a single-state LTS and *the same contractions*  $\sigma_a(r) = \frac{1}{2}r$  and  $\sigma_b(r) = \frac{1}{2}r + \frac{1}{2}$ , or using *any (finite) compositions* of  $\sigma_a$  and  $\sigma_b$ . Indeed, suppose there were such a finite collection  $\sigma_1, \dots, \sigma_n$  consisting of (finite) compositions of  $\sigma_a$  and  $\sigma_b$  such that  $K = \bigcup_{i=1}^n \sigma_i(K)$ . Since  $1 \in K$ , we must be using the stream  $(b, b, b, \dots)$  (since if there is an  $a$  at position  $n$ , the number obtained would be  $\leq 1 - \frac{1}{2^n} < 1$ ), so some  $\sigma_i$  must consist of a composition of  $\sigma_b$  some number  $m \geq 1$  of times with itself. Similarly, the only way to obtain 0 is with  $(a, a, a, \dots)$ , so there must be some  $\sigma_j$  which is a composition of  $\sigma_a$  some number of times  $p \geq 1$  with itself. But then  $\lim_{n \rightarrow \infty} \sigma_i \circ \sigma_j \circ \sigma_i^n(r) = 1 - (\frac{2^p - 1}{2^{m+p}}) > \frac{1}{2}$ , since  $m, p \geq 1$ . That point must be in the subset of  $\mathbb{R}$  generated by this LTS. However, it is not in  $K$ , since  $\frac{1}{2} < 1 - (\frac{2^p - 1}{2^{m+p}}) < 1$ .

More generally, we cannot obtain  $K$  using a single-state LTS even if we allowed finite sums of compositions of  $\sigma_a$  and  $\sigma_b$ .

Once again, it is possible to find a single state LTS whose corresponding subset of  $\mathbb{R}$  is  $K$ , but to do this we needed to *change the alphabet and also the contractions*. Perhaps un-coincidentally, the constant operators are exactly the limits of the two contractions from the original interpretation. Our question is whether this can always be done.

On the other hand, the thesis of Boore [7] may contain an answer to Question 1. Boore presents a (family) of 2-state GIFS whose *attractors*, total unions of their regular subfractals, are not self-similar. Attractors of GIFSs are not precisely the same as regular subfractals, so additional work is required to adapt Boore's work to answer Question 1.

## 7 Related Work

This paper is part of a larger effort of examining topics in continuous mathematics from the standpoint of coalgebra and theoretical computer science. The topic itself is quite old, and originates perhaps with Pavlovic and Escardó's paper "Calculus in Coinductive Form" [23]. Another early contribution is Pavlovic and Pratt [24]. These papers proposed viewing some structures in continuous mathematics – the real numbers, for example, and power series expansions – in terms of final coalgebras and streams. The next stage in this line of work was a set of papers specifically about fractal sets and final coalgebras. For example, Leinster [16]



offered a very general theory of self-similarity that used categorical modules in connection with the kind of gluing that is prominent in constructions of self-similar sets. In a different direction, papers like [4] showed that for some very simple fractals (such as the Sierpiński gasket treated here), the final coalgebras were Cauchy completions of the initial algebras.

**Generalizations of IFSs.** Many generalizations of Hutchinson’s self-similar sets have appeared in the literature. The generalization that most closely resembles our own is that of an attractor for a *directed-graph iterated function system* (GIFS) [18]. A LTS paired with a contraction operator interpretation is equivalent data to that of a GIFS, and equivalent statements to Lemma 3.7 can be found for example in [9, 18, 19]. As opposed to the regular subfractal corresponding to one state, as we have studied above, the geometric object studied in the GIFSs literature is typically the union of the regular subfractals corresponding to all the states (in our terminology), and properties such as Hausdorff dimension and connectivity are emphasized. We also distinguish our structures from GIFSs because we need to allow the interpretations of the labels to vary in our semantics.

Another generalization is Mihail and Miculescu’s notion of attractor for a *generalized iterated function system* [19]. A generalized IFS is essentially that of Hutchinson’s IFS with multi-arity contractions – equivalent to a single-state labelled transition system where labels have “higher arity”. A common generalization of GIFSs and generalized IFSs could be achieved by considering coalgebras of the form  $X \rightarrow \mathcal{P}(\coprod_{n \in \mathbb{N}} A_n \times X^n)$  and interpreting each  $a \in A_n$  as an  $n$ -ary contraction. We suspect that a similar story to the one we have outlined in this paper is possible for this common generalization.

**Process algebra.** The process terms we use to specify labelled transition systems and labelled Markov chains are fragments of known specification languages. Milner used process terms to specify LTSs in [22], and we have repurposed his small-step semantics here. Stark and Smolka use probabilistic process terms to specify labelled Markov chains (in our terminology) in [27], and we have used them for the same purpose. Both of these papers also include complete axiomatizations of bisimilarity, and we have also repurposed their axioms.

However, fractal semantics is strictly coarser than bisimilarity, and in particular, bisimilarity of process terms is trace equivalence. Rabinovich added a single axiom to Milner’s axiomatization to obtain a sound and complete axiomatization of trace equivalence of expressions [25], which allowed us to derive Theorem 4.4. In contrast, the axiomatization of trace equivalence for probabilistic processes is only well-understood for *finite* traces, see Silva and Sokolova’s [26], which our probabilistic process terms do not exhibit. We use the trace semantics of Kerstan and König [14] because it takes into account infinite traces. Infinite trace semantics has yet to see a complete axiomatization in the literature.

**Other types of syntax.** In this paper, we used the specification language of  $\mu$ -terms as our basic syntax. As it happens, there are two other flavors of syntax that we could have employed. These are *iteration theories* [5], and terms in the Formal Language of Recursion *FLR*, especially its *FLR<sub>0</sub>* fragment. The three flavors of syntax for fixed point terms are compared in a number of papers: In [12], it was shown that there is an equivalence of categories between *FLR<sub>0</sub>* structures and iteration theories, and Bloom and Ésik make a similar connection between iteration theories and the  $\mu$ -calculus in [6]. Again, these results describe general matters of equivalence, but it is not completely clear that for a specific space or class of spaces that they are equally powerful or equally convenient specification languages. We feel this matter deserves some investigation.

**Equivalence under hypotheses.** A specification language fairly close to iteration theories was used by Milius and Moss to reason about fractal constructions in [21] under the guise of *interpreted solutions* to recursive program schemes [20]. Moreover, [21] contains important examples of reasoning about the equality of fractal sets under assumptions about the contractions. Based on the general negative results on reasoning from hypotheses in the logic of recursion [12], we would not expect a completeness theorem for fractal equivalence under hypotheses. However, we do expect to find sound logical systems which account for interesting phenomena in the area.

## 8 Conclusion

This paper connects fractals to trace semantics, a topic originating in process algebra. This connection is our main contribution, because it opens up a line of communication between two very different areas of study. The study of fractals is a well-developed area, and like most of mathematics it is pursued without a special-purpose specification language. When we viewed process terms as recipes for fractals, we provided a specification language that was not present in the fractals literature. Of course, one also needs a contraction operator interpretation to actually define a fractal, but the separation of syntax (the process terms) and semantics (the fractals obtained using contraction operator interpretations of the syntax) is something that comes from the tradition of logic and theoretical computer science. Similarly, the use of a logical system and the emphasis on soundness and completeness is a new contribution here.

All of the above opens questions about fractals and their specifications. Our most concrete question was posed in Section 6. We would also like to know if we can obtain completeness theorems allowing for extra equations in the axiomatization. Lastly, and most speculatively, since LTSs (and other automata) appear so frequently in decision procedures from process algebra and verification, we would like to know if our semantics perspective on fractals can provide new complexity results in fractal geometry.

We hope we have initiated a line of research where questions and answers come from both the analytic side and from theoretical computer science.

---

## References

- 1 Jos C. M. Baeten. A brief history of process algebra. *Theor. Comput. Sci.*, 335(2-3):131–146, 2005. doi:10.1016/j.tcs.2004.07.036.
- 2 Stefan Banach. Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales. *Fundamenta Mathematicae*, 3:133–181, 1922.
- 3 Jan A. Bergstra, Alban Ponse, and Scott A. Smolka, editors. *Handbook of Process Algebra*. North-Holland / Elsevier, 2001. doi:10.1016/b978-0-444-82830-9.x5017-6.
- 4 Prasad Bhattacharya, Lawrence S. Moss, Jayampathy Ratnayake, and Robert Rose. Fractal sets as final coalgebras obtained by completing an initial algebra. In Franck van Breugel, Elham Kashefi, Catuscia Palamidessi, and Jan Rutten, editors, *Horizons of the Mind. A Tribute to Prakash Panangaden - Essays Dedicated to Prakash Panangaden on the Occasion of His 60th Birthday*, volume 8464 of *Lecture Notes in Computer Science*, pages 146–167. Springer, 2014.
- 5 Stephen L. Bloom and Zoltán Ésik. *Iteration Theories - The Equational Logic of Iterative Processes*. EATCS Monographs on Theoretical Computer Science. Springer, 1993. doi:10.1007/978-3-642-78034-9.
- 6 Stephen L. Bloom and Zoltán Ésik. Solving polynomial fixed point equations. In *Mathematical foundations of computer science 1994 (Košice, 1994)*, volume 841 of *Lecture Notes in Comput. Sci.*, pages 52–67. Springer, Berlin, 1994.

- 7 Graeme C. Boore. *Directed graph iterated function systems*. PhD thesis, University of St. Andrews, 2011. URL: <http://hdl.handle.net/10023/2109>.
- 8 G. A. Edgar and R. Daniel Mauldin. Multifractal decompositions of digraph recursive fractals. *Proceedings of the London Mathematical Society*, s3-65(3):604–628, 1992. doi: 10.1112/plms/s3-65.3.604.
- 9 Gerald A. Edgar. *Measure, Topology, and Fractal Geometry*. Springer New York, NY, 1 edition, 1990. doi:10.1007/978-1-4757-4134-6.
- 10 Kenneth J. Falconer. *The Geometry of Fractal Sets*. Cambridge Tracts in Mathematics. Cambridge University Press, 1986. URL: <https://www.cambridge.org/us/academic/subjects/mathematics/abstract-analysis/geometry-fractal-sets>.
- 11 C. A. R. Hoare. Communicating sequential processes. *Commun. ACM*, 21(8):666–677, 1978. doi:10.1145/359576.359585.
- 12 A. J. C. Hurkens, Monica McArthur, Yiannis N. Moschovakis, Lawrence S. Moss, and Glen T. Whitney. Erratum: “The logic of recursive equations”. *J. Symbolic Logic*, 64, 1999.
- 13 John E. Hutchinson. Fractals and self similarity. *Indiana University Mathematics Journal*, 30(5):713–747, 1981. URL: <http://www.jstor.org/stable/24893080>.
- 14 Henning Kerstan and Barbara König. Coalgebraic Trace Semantics for Continuous Probabilistic Transition Systems. *Logical Methods in Computer Science*, Volume 9, Issue 4, December 2013. doi:10.2168/LMCS-9(4:16)2013.
- 15 S. C. Kleene. Representation of events in nerve nets and finite automata. In Claude Shannon and John McCarthy, editors, *Automata Studies*, pages 3–41. Princeton University Press, Princeton, NJ, 1956.
- 16 Tom Leinster. A general theory of self-similarity. *Adv. Math.*, 226(4):2935–3017, 2011.
- 17 Benoit B. Mandelbrot. *Fractals: Form, Chance, and Dimension*. Mathematics Series. W. H. Freeman, 1977. URL: [https://books.google.com/books?id=avw\\_AQAAIAAJ](https://books.google.com/books?id=avw_AQAAIAAJ).
- 18 R Daniel Mauldin and S. C. Williams. Hausdorff dimension in graph directed constructions. *Transactions of the American Mathematical Society*, 309(2):811–829, 1988. doi:10.1090/S0002-9947-1988-0961615-4.
- 19 Alexandru Mihail and Radu Miculescu. Generalized ifss on noncompact spaces. *Fixed Point Theory and Applications*, 1(584215), 2010. doi:10.1155/2010/584215.
- 20 Stefan Milius and Lawrence S. Moss. The category-theoretic solution of recursive program schemes. *Theor. Comput. Sci.*, 366(1-2):3–59, 2006. doi:10.1016/j.tcs.2006.07.002.
- 21 Stefan Milius and Lawrence S. Moss. Equational properties of recursive program scheme solutions. *Cah. Topol. Géom. Différ. Catég.*, 50(1):23–66, 2009.
- 22 Robin Milner. A complete inference system for a class of regular behaviours. *J. Comput. Syst. Sci.*, 28(3):439–466, 1984. doi:10.1016/0022-0000(84)90023-0.
- 23 Dusko Pavlovic and Martín Hötzel Escardó. Calculus in coinductive form. In *Thirteenth Annual IEEE Symposium on Logic in Computer Science, Indianapolis, Indiana, USA, June 21-24, 1998*, pages 408–417. IEEE Computer Society, 1998.
- 24 Dusko Pavlovic and Vaughan R. Pratt. The continuum as a final coalgebra. *Theor. Comput. Sci.*, 280(1-2):105–122, 2002.
- 25 Alexander Moshe Rabinovich. A complete axiomatisation for trace congruence of finite state behaviors. In Stephen D. Brookes, Michael G. Main, Austin Melton, Michael W. Mislove, and David A. Schmidt, editors, *Mathematical Foundations of Programming Semantics, 9th International Conference, New Orleans, LA, USA, April 7-10, 1993, Proceedings*, volume 802 of *Lecture Notes in Computer Science*, pages 530–543. Springer, 1993. doi:10.1007/3-540-58027-1\_25.
- 26 Alexandra Silva and Ana Sokolova. Sound and complete axiomatization of trace semantics for probabilistic systems. In Michael W. Mislove and Joël Ouaknine, editors, *Twenty-seventh Conference on the Mathematical Foundations of Programming Semantics, MFPS 2011, Pittsburgh, PA, USA, May 25-28, 2011*, volume 276 of *Electronic Notes in Theoretical Computer Science*, pages 291–311. Elsevier, 2011. doi:10.1016/j.entcs.2011.09.027.

## 14:18 Fractals from Regular Behaviours

- 27 Eugene W. Stark and Scott A. Smolka. A complete axiom system for finite-state probabilistic processes. In Gordon D. Plotkin, Colin Stirling, and Mads Tofte, editors, *Proof, Language, and Interaction, Essays in Honour of Robin Milner*, pages 571–596. The MIT Press, 2000.

# Coinductive Control of Inductive Data Types

Paige Randall North

Department of Mathematics and Department of Information and Computing Sciences, Utrecht University, The Netherlands

Maximilien Péroux

Department of Mathematics, University of Pennsylvania, Philadelphia, PA, USA

---

## Abstract

We combine the theory of inductive data types with the theory of universal measurings. By doing so, we find that many categories of algebras of endofunctors are actually enriched in the corresponding category of coalgebras of the same endofunctor. The enrichment captures all possible partial algebra homomorphisms, defined by measuring coalgebras. Thus this enriched category carries more information than the usual category of algebras which captures only total algebra homomorphisms. We specify new algebras besides the initial one using a generalization of the notion of initial algebra.

**2012 ACM Subject Classification** Theory of computation → Categorical semantics; Theory of computation → Algebraic semantics; Theory of computation → Type structures

**Keywords and phrases** Inductive types, enriched category theory, algebraic data types, algebra, coalgebra

**Digital Object Identifier** 10.4230/LIPIcs.CALCO.2023.15

**Related Version** *Full Version*: <https://arxiv.org/abs/2303.16793>

**Funding** *Paige Randall North*: This material is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-21-1-0334.

## 1 Introduction

In both the tradition of functional programming and categorical logic, one takes the perspective that most data types should be obtained as initial algebras of certain endofunctors (to use categorical language). For instance, the natural numbers are obtained as the initial algebra of the endofunctor  $X \mapsto X + 1$ , assuming that the category in question (often the category of sets) has a terminal object  $1$  and a coproduct  $+$ . Much theory has been developed around this approach, which culminated in the notion of W-types [5].

In another tradition, for  $k$  a field, it has been long understood (going back at least to Wraith, according to [3], and Sweedler [10]) that the category of  $k$ -algebras is naturally enriched over the category of  $k$ -coalgebras, a fact which has admitted generalization to several other settings (e.g. [3, 11, 8, 6]). In this paper, we extend this theory to the setting of an endofunctor on a category – in particular those endofunctors that are considered in the theory of W-types.

This work is thus the beginning of a development of an analogue of the theory of W-types – not based on the notion of initial objects in a *category* of algebras, but rather on generalized notions of initial objects in a *coalgebra enriched category* of algebras. Our main result (Theorem 31) states that the categories of algebras of endofunctors considered in the theory of W-types are often enriched in their corresponding categories of coalgebras. The hom-coalgebras of our enriched category carry more information than the hom-sets in the unenriched category that is usually considered in the theory of W-types. Because of our passage to the enriched setting, we have more precise control than in the unenriched setting,



© Paige Randall North and Maximilien Péroux;

licensed under Creative Commons License CC-BY 4.0

10th Conference on Algebra and Coalgebra in Computer Science (CALCO 2023).

Editors: Paolo Baldan and Valeria de Paiva; Article No. 15; pp. 15:1–15:17

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

## 15:2 Coinductive Control of Inductive Data Types

and we are able to specify more data types than just those which are captured by the theory of W-types. We do this by generalizing the notion of initial algebra, taking inspiration from the notion of weighted limits. This general theory is presented in Section 3.

But first, in Section 2, we begin our paper with an enlightening example which serves as an illustration of the relevance of our enriched theory and as a motivation for the more general setting. Therein, we provide explicit calculations for the case of algebras over the endofunctor  $X \mapsto X + 1$  on  $\mathbf{Set}$ . In that example, we illustrate that it is appropriate to interpret the elements of the hom-coalgebras as *partial homomorphisms*.

Indeed, in the classical Sweedler theory, the enrichment in coalgebras can also be understood as encoding a notion of partial homomorphism. Though we do not study  $k$ -algebras in this paper, we conclude this introduction with details of that classical theory. A *measuring* from a  $k$ -algebra  $A$  to a  $k$ -algebra  $B$ , in the sense of Sweedler [10], is a  $k$ -coalgebra  $C$  together with a linear homomorphism  $\phi: C \otimes_k A \rightarrow B$  that is compatible with the multiplication and identities of  $A$  and  $B$ . A measuring from  $A$  to  $B$  is equivalently a  $k$ -coalgebra  $C$  together with a  $k$ -linear map  $\phi: C \rightarrow A \rightarrow B$  such that

$$\phi_c(aa') = \sum_{i=1}^n \phi_{c_i^{(1)}}(a)\phi_{c_i^{(2)}}(a'), \quad \text{and} \quad \phi_c(1_A) = \varepsilon(c)1_B,$$

for all  $c \in C$  and  $a, a' \in A$  where  $\Delta(c) = \sum_{i=1}^n c_i^{(1)} \otimes c_i^{(2)}$  is the comultiplication  $\Delta: C \rightarrow C \otimes_k C$  and  $\varepsilon: C \rightarrow k$  is the counit of  $C$ . Therefore the  $k$ -linear maps  $\phi_c: A \rightarrow B$  can be regarded as partial algebra homomorphisms, and the elements  $c \in C$  can be interpreted as measuring how far each partial homomorphism  $\phi_c$  is from being a total homomorphism. For instance when  $\Delta(c) = c \otimes c$ , we have that  $\phi_c: A \rightarrow B$  is a total algebra homomorphism. Now we proceed to tell an analogous story about endofunctors.

### 2 Illustrative example: $\text{id} + 1$

In this section, we illustrate our results in the context of one example: the endofunctor that sends  $X \mapsto X + 1$  (the coproduct of  $X$  and a terminal set 1) in  $\mathbf{Set}$ , the category of sets. The initial algebra of this endofunctor is  $\mathbb{N}$ , the natural numbers, and thus this endofunctor is one of the most basic and important examples in the theory of W-types.

This section is one very long worked example of our general, categorical theory which follows in Section 3.

We first review the classical story in Section 2.1, and afterwards our goal is to explain how the category of algebras is naturally enriched in the category of coalgebras of this functor and how we can use this extra structure to generalize the notion of *initial algebra* to capture more algebras than just  $\mathbb{N}$ . So, in Section 2.2 we explore by hand a notion of partial homomorphism between algebras that will be captured more formally later in the enrichment. Next, in Section 2.3, we explore the structure that this enrichment gives us. In Section 2.4, we introduce a computational tool and compute explicitly some of the hom-objects of our enrichment. Finally, in Section 2.5, we use this extra structure to generalize the notion of initial object, and we describe some of the algebras that can be specified in this way.

Note that many of the proofs in this paper were relegated to the appendices, which do not appear with this, published, version. Thus, we repeatedly reference proofs in the full version, [7].

## 2.1 Preliminaries

Here, we review the established theory regarding algebras and coalgebras of  $\text{id} + 1$  that we will use. See, for instance, [9, Ch. 3] for details.

We let  $\mathbf{Alg}$  denote the category of algebras of  $\text{id} + 1$ , and we let  $\mathbf{CoAlg}$  denote the category of coalgebras of  $\text{id} + 1$ . Recall that an algebra is a pair  $(A, \alpha)$  of a set  $A$  together with a function  $\alpha : A + 1 \rightarrow A$  (equivalently, a *successor* endofunction  $\alpha|_A : A \rightarrow A$  and a *zero*  $\alpha|_1 : 1 \rightarrow A$ ), and a coalgebra is a pair  $(C, \chi)$  of a set  $C$  together with a function  $\chi : C \rightarrow C + 1$ , i.e., a partial endofunction. The initial object of  $\mathbf{Alg}$  is  $(\mathbb{N}, \alpha_{\mathbb{N}})$ , where  $\mathbb{N}$  is the usual natural numbers,  $\alpha_{\mathbb{N}}|_{\mathbb{N}}$  is the usual successor function  $x \mapsto x + 1$  and  $\alpha_{\mathbb{N}}|_1$  picks out  $0 \in \mathbb{N}$ . The terminal object of  $\mathbf{CoAlg}$  is  $(\mathbb{N}^{\infty}, \chi_{\mathbb{N}^{\infty}})$  where  $\mathbb{N}^{\infty}$  is the extended natural numbers  $\mathbb{N} + \{\infty\}$ , and the map  $\chi_{\mathbb{N}^{\infty}} : \mathbb{N}^{\infty} \rightarrow \mathbb{N}^{\infty} + 1$  takes  $0 \in \mathbb{N}^{\infty}$  to the element  $\mathfrak{t} \in 1$  and all other  $x \in \mathbb{N}^{\infty}$  to  $x - 1 \in \mathbb{N}^{\infty}$ .

Note that because  $\mathbb{N}$  is initial in  $\mathbf{Alg}$ , any algebra  $(A, \alpha_A)$  gets a function  $!_A : \mathbb{N} \rightarrow A$ , and thus it will be useful write  $n_A$  for  $!_A(n)$ . That is,  $0_A$  is the zero of  $A$ ,  $1_A$  is the successor of  $0_A$ , etc. For  $a \in A$ , we will often also write  $a + 1$  as shorthand for  $\alpha_A(a)$  (especially when the algebra structure morphism, here  $\alpha_A$ , does not have an explicit name).

Dually, because  $\mathbb{N}^{\infty}$  is terminal in  $\mathbf{CoAlg}$ , there is a function  $\llbracket - \rrbracket : C \rightarrow \mathbb{N}^{\infty}$  for any coalgebra  $(C, \chi_C)$ , and we will say that the *index* of a  $c \in C$  is  $\llbracket c \rrbracket$ . Then the elements of  $C$  that have index 0 are those  $c$  such that  $\chi_C(c) = \mathfrak{t}$ , those that have index 1 are all those other  $c$  such that  $\chi_C^2(c) = \mathfrak{t}$ , etc. For  $c \in C$  where  $\llbracket c \rrbracket \neq 0$ , we will also often write  $c - 1$  to denote  $\chi_C(c)$  (especially when the coalgebra structure morphism does not have an explicit name).

Besides  $\mathbb{N}$ , the *initial algebra*, we will often consider *preinitial algebras*, that is, algebras  $A$  for which  $!_A : \mathbb{N} \rightarrow A$  is epic. The nontrivial preinitial algebras are of the form  $\mathfrak{n} := (\{0, 1, \dots, n\}, \alpha_{\mathfrak{n}})$  for any  $n \in \mathbb{N}$ . Here,  $\alpha_{\mathfrak{n}}$  is the algebra structure that  $\{0, 1, \dots, n\}$  inherits as the quotient of  $\mathbb{N}$  in  $\mathbf{Set}$  that identifies all  $m \geq n$  (see [7, Example 39] and the preceding [7, Lemma 38]).

Dually, besides  $\mathbb{N}^{\infty}$ , we will often consider *subterminal coalgebras*, that is, coalgebras  $C$  for which  $\llbracket - \rrbracket : C \rightarrow \mathbb{N}^{\infty}$  is monic. The nontrivial ones are  $\mathfrak{n}^{\circ}$  with underlying subset  $\{0, 1, \dots, n\}$  of  $\mathbb{N}^{\infty}$ ,  $\mathbb{N}^{-}$  with underlying subset  $\mathbb{N}$ , and  $\mathbb{1}$  with underlying subset  $\{\infty\}$ . These all inherit coalgebra structures from  $\mathbb{N}^{\infty}$  (see [7, Example 43] and the preceding [7, Lemma 42]).

## 2.2 Partial homomorphisms

Consider algebras  $A$  and  $B$ . We are, first of all, most interested in algebra homomorphisms  $f : A \rightarrow B$  (which we might call *total* algebra homomorphisms to distinguish them from the notion of *partial* algebra homomorphisms which we are about to introduce). This means that we have (H1)  $f(0_A) = 0_B$  and (H2)  $f(a + 1) = f(a) + 1$  for all  $a \in A$ . If  $A$  is  $\mathbb{N}$ , we know that there is a total algebra homomorphism  $\mathbb{N} \rightarrow B$ , and we can use (H1) and (H2) to inductively construct this homomorphism.

But depending on the nature of  $A$  and  $B$ , it might happen that we can only guarantee (H1) and (H2) hold for some, but not all,  $a \in A$ , and thus an attempt to construct a total algebra homomorphism  $A \rightarrow B$  inductively might fail at some point. Perhaps  $A$  is a preinitial algebra  $\mathfrak{n}$  and  $B$  is  $\mathbb{N}$ . We can try to construct a total homomorphism, so we set  $f(0_{\mathfrak{n}}) := 0_{\mathbb{N}}$  following (H1),  $f(1_{\mathfrak{n}}) := 1_{\mathbb{N}}$  following (H2), etc. This works until we get to  $n_{\mathfrak{n}}$ . Since  $n_{\mathfrak{n}}$  is the successor of both  $(n - 1)_{\mathfrak{n}}$  and  $n_{\mathfrak{n}}$ , (H2) tells us to send  $n_{\mathfrak{n}}$  both to  $n_{\mathbb{N}}$  and  $(n + 1)_{\mathbb{N}}$ . We might say that induction worked only up to the  $n$ th step, or that we can define a *n-partial homomorphism*.

## 15:4 Coinductive Control of Inductive Data Types

We formalize this idea in the following way, in which we inductively construct partial homomorphisms in an attempt to approximate a total homomorphism. In our first attempt at formalizing this idea, Construction 1 below, we make the simplifying assumption that  $A$  is preinitial – simplifying because then there is at most one homomorphism  $A \rightarrow B$ . We will almost immediately drop this assumption in the more general Definition 2.

► **Construction 1** (Partial induction). *We seek to inductively approximate a homomorphism  $A \rightarrow B$  when  $A$  is preinitial. We define a sequence of functions  $f_c : A \rightarrow B$  as follows.*

**Initial step (P1).** *Define  $f_0 : A \rightarrow B$  by  $f_0(a) := 0_B$  for all  $a \in A$ .*

**Inductive step.** *Define  $f_{c+1} : A \rightarrow B$  by:*

$$\text{(P2)} \quad f_{c+1}(0_A) := 0_B;$$

$$\text{(P3)} \quad f_{c+1}(a + 1) := f_c(a) + 1.$$

*We stop when  $f_{c+1}$  is not well-defined.*

*If we have defined  $f_c$  for all  $c \in \mathbb{N}$ , then we will say that we have defined an  $\infty$ -partial homomorphism. Otherwise, if we have only defined  $f_c$  for all  $c \in \{0, \dots, n\}$ , we will say that we have defined an  $n$ -partial homomorphism.*

Since  $A$  is preinitial, it is of the form  $\mathfrak{n}$  or  $\mathbb{N}$ , and so every element is of the form  $x_A$  for  $x \in \mathbb{N}$ . Thus,  $f_c(x_A)$  is  $x_B$  for  $x \leq c$  and otherwise  $c_B$ . In particular, if  $A = \mathfrak{n}$ , then  $f_n = f_m$  for all  $m \geq n$ . Now we can see that there is an  $\infty$ -partial homomorphism  $A \rightarrow B$  if and only if there is a total homomorphism  $f : A \rightarrow B$ . Indeed, if we have an  $\infty$ -partial homomorphism  $A \rightarrow B$  consisting of an  $f_c : A \rightarrow B$  for all  $c \in \mathbb{N}$ , then we can define a “diagonal” total homomorphism  $f : A \rightarrow B$  by  $f(x_A) := f_x(x_A)$ . Conversely, if we have a total homomorphism  $f : A \rightarrow B$ , there is no obstruction to the inductive steps in defining a  $\infty$ -partial homomorphism. Thus, we can conflate the notions of a  $\infty$ -partial homomorphism and a total homomorphism  $A \rightarrow B$  when  $A$  is preinitial.

Notice that in the term “ $n$ -partial homomorphism” in the above Construction 1,  $n$  takes values in  $\mathbb{N}^\infty$ , the terminal coalgebra of our endofunctor. In fact, the above construction follows the similar pattern of the measurements of algebras over a field that we mentioned in the introduction. So now we make the following definition in which we encode the coalgebra directly. This allows us to generalize Construction 1, dropping the hypothesis that  $A$  is preinitial.

► **Definition 2** (Measuring, cf. Definition 18 and Proposition 23). *Consider algebras  $A$  and  $B$  and a coalgebra  $C$ . A measuring from  $A$  to  $B$  by  $C$  is a function  $f : C \rightarrow A \rightarrow B$  such that:*

$$\text{(M1)} \quad f_c(0_A) = 0_B \text{ for all } c \in C;$$

$$\text{(M2)} \quad f_c(a + 1) = 0_B \text{ for all } \llbracket c \rrbracket = 0 \text{ and for all } a \in A;$$

$$\text{(M3)} \quad f_c(a + 1) = f_{c-1}(a) + 1 \text{ for } \llbracket c \rrbracket \geq 1 \text{ and for all } a \in A.$$

*We write  $\mu_C(A, B)$  for the set of measurements from  $A$  to  $B$ . This defines a functor  $\mu : \text{CoAlg}^{\text{op}} \times \text{Alg}^{\text{op}} \times \text{Alg} \rightarrow \text{Set}$ .*

*For a measuring  $f$  and an element  $c \in C$ , we call  $f_c$  a  $C$ -partial homomorphism.*

► **Example 3.** Suppose that  $A$  is preinitial, so that in particular every element of  $A$  is of the form  $0_A$  or  $a + 1$ .

Then there is a measuring from  $A$  to  $B$  by  $\mathfrak{n}^\circ$  if the induction of Construction 1 creates an  $n$ -partial homomorphism, and in this case the functions of the form  $f_c$  constructed in Construction 1 are the same as those specified in Definition 2.

There is a measuring by  $\mathbb{N}^-$  if the induction never fails, and again the functions  $f_c$  from Construction 1 and Definition 2 coincide. Now, note that exhibiting a measuring by  $\mathbb{N}^\infty$  amounts to exhibiting a measuring by  $\mathbb{N}^-$  together with a total algebra homomorphism  $f_\infty$ . For such an  $A$ , then, exhibiting a measuring by  $\mathbb{N}^-$  is equivalent to exhibiting one by  $\mathbb{N}^\infty$ .



The reader might wonder why Definition 2 does not require (M2')  $f_c(x) = 0_B$  for any  $\llbracket c \rrbracket = 0$  and any  $x$ , but rather only requires  $f_c(x) = 0_B$  when  $\llbracket c \rrbracket = 0$  and  $x$  is either the zero or a successor. In the previous example, when  $A$  is preinitial, every  $x \in A$  is either the zero or a successor, so there is no difference between these two requirements. In the following example, we consider an algebra  $A$  where this is not the case, and illustrate why we only stipulate (M2) and not (M2').

► **Example 4.** Consider the algebra  $A$  with underlying set  $\mathbb{N} + \mathbb{N}$ , where we will notate the elements of the first copy of  $\mathbb{N}$  as  $n_A$ , and the elements of the second copy as  $n'$ . The zero of  $A$  is then  $0_A$  and the successors are given by  $n_A + 1 := (n + 1)_A$  and  $n' + 1 := (n + 1)'$ . Total homomorphisms  $A \rightarrow \mathbb{N}$  are determined by the image of  $0'$  in  $\mathbb{N}$ .

If we require (M2') instead of (M2) in Definition 2, then in a measuring by  $\mathbb{N}^\infty$ ,  $f_0(0') = 0_{\mathbb{N}}$  by (M2'), and in general  $f_n(n') = n_{\mathbb{N}}$  by (M3).

However, following Definition 2 as written, in a measuring by  $\mathbb{N}^\infty$ ,  $f_0(0')$  may be anything, say  $z \in \mathbb{N}$  and then general  $f_n(n') = (z + n)_{\mathbb{N}}$ .

Thus, Definition 2 does generalize the idea of inductively approximating a total homomorphism  $A \rightarrow \mathbb{N}$  from Construction 1.

Now notice another difference between Construction 1 and Definition 2. In Construction 1 we continue the induction as far as we can, but there is nothing of this flavor in Definition 2. For instance, if there is a total algebra homomorphism  $A \rightarrow B$ , then in the process described by Construction 1, we will inductively construct  $f_c$  for all  $c \in \mathbb{N}$ . However, following Definition 2, we could say that  $A \rightarrow B$  is measured by  $\emptyset$  (which only amounts to exhibiting  $f_0$ ), without making any claim about it being measured by other coalgebras – it does not ask us to find any kind of maximum coalgebra  $C$  that measures  $A \rightarrow B$ . To remedy this, we make the following definition.

► **Definition 5** (Universal measuring, cf. Definition 20). *Let  $A$  and  $B$  be algebras.*

*We define the category of measurings from  $A$  to  $B$  to be the category whose objects are pairs  $(C; f)$  of a coalgebra  $C$  and a measuring  $f : C \rightarrow A \rightarrow B$ , and whose morphisms  $(C; f) \rightarrow (D; g)$  are coalgebra homomorphisms  $d : C \rightarrow D$  such that  $f = gd$ .*

*The universal measuring from  $A$  to  $B$ , denoted  $(\text{Alg}(A, B); u)$ , is the terminal object in the category of measurings from  $A$  to  $B$ . That is, if  $(C; f)$  is a measuring from  $A$  to  $B$ , then there is a unique morphism  $! : C \rightarrow \text{Alg}(A, B)$  that makes the following diagram commute.*

$$\begin{array}{ccc}
 C & \xrightarrow{f} & A \rightarrow B \\
 \downarrow ! & \nearrow u & \\
 \text{Alg}(A, B) & & 
 \end{array}$$

► **Example 6.** Again, suppose that  $A$  is preinitial. In this case, the universal measuring is a subterminal coalgebra [7, Lemma 37]. We have shown that if the induction of Construction 1 creates an  $n$ -partial homomorphism, then the maximum subterminal coalgebra that measures  $A \rightarrow B$  is  $\mathfrak{n}^\circ$ , so this is the universal measuring. And if the induction of Construction 1 creates an total homomorphism, then the maximum subterminal coalgebra that measures  $A$  to  $B$  is  $\mathbb{N}^\infty$  itself, so this is the universal measuring. We will also show this fact more directly (i.e., without reference to [7, Lemma 37]) in Section 2.4 below.

Since composing an arbitrary coalgebra homomorphism  $C \rightarrow \text{Alg}(A, B)$  with  $u$  produces a measuring  $C \rightarrow A \rightarrow B$ , we obtain a bijection, natural in  $C, A, B$ ,

$$\mu_C(A, B) \cong \text{CoAlg}(C, \text{Alg}(A, B))$$

## 15:6 Coinductive Control of Inductive Data Types

showing that  $\mu_{-}(A, B)$  is represented by  $\underline{\text{Alg}}(A, B)$ . In Theorem 25, we will see that  $\underline{\text{Alg}}(A, B)$  always exists (for this and other endofunctors of interest). The coalgebra  $\underline{\text{Alg}}(A, B)$  will constitute the hom-coalgebra from  $A$  to  $B$  of our enriched category of algebras (Theorem 31).

Now, note that a measuring by the coalgebra  $\mathbb{1}$  is a total algebra homomorphism. Thus,

$$\text{Alg}(A, B) \cong \mu_{\mathbb{1}}(A, B) \cong \text{CoAlg}(\mathbb{1}, \underline{\text{Alg}}(A, B))$$

and so we find the hom-sets of the category of algebras can be easily extracted from  $\underline{\text{Alg}}(A, B)$  – a statement that aligns with our intuition that  $\text{Alg}(A, B)$  is the set of total algebra homomorphisms and  $\underline{\text{Alg}}(A, B)$  is the coalgebra of partial algebra homomorphisms.

### 2.3 Composing partial homomorphisms

We will only prove that the universal measuring coalgebras form the hom-objects of our enriched category in Theorem 31, but we can already work out how this enrichment behaves. Thus, in this section, we describe the composition and identities of this enriched category.

Given algebras  $A, B$ , and  $T$ , we can always compose total homomorphisms  $f : A \rightarrow B$  and  $g : B \rightarrow T$  to form a total homomorphism  $g \circ f : A \rightarrow T$ . We wish to do the same for our partial homomorphisms. Consider coalgebras  $C$  and  $D$ , a  $C$ -partial homomorphism  $f_c : A \rightarrow B$ , and a  $D$ -partial homomorphism  $g_d : B \rightarrow T$ . We can compose  $g_d$  and  $f_c$  as functions to obtain  $g_d \circ f_c : A \rightarrow T$ , and we claim that this is a  $(D \times C)$ -partial homomorphism. Indeed  $D \times C$  has a coalgebra structure where  $\llbracket (d, c) \rrbracket = \min(\llbracket d \rrbracket, \llbracket c \rrbracket)$  and  $(d, c) - 1 = (d - 1, c - 1)$  if  $\llbracket (d, c) \rrbracket > 0$  for  $(d, c) \in D \times C$ . This induces a symmetric monoidal structure on  $\text{CoAlg}$  for which  $\mathbb{1}$  is the unit (Proposition 30), and one can verify that  $g_d \circ f_c$  is a  $(D \times C)$ -partial homomorphism.

Now we have constructed a function

$$\begin{aligned} \mu_D(B, T) \times \mu_C(A, B) &\longrightarrow \mu_{D \times C}(A, T) \\ (g, f) &\longmapsto g \circ f \end{aligned}$$

where  $g \circ f : (D \times C) \rightarrow A \rightarrow T$  is defined by  $(g \circ f)_{(d, c)} = g_d \circ f_c$ . Thus, by the universal property of  $\underline{\text{Alg}}$ , we obtain a function

$$\text{CoAlg}(D, \underline{\text{Alg}}(B, T)) \times \text{CoAlg}(C, \underline{\text{Alg}}(A, B)) \longrightarrow \text{CoAlg}(D \times C, \underline{\text{Alg}}(A, T)).$$

Applying this function to  $(\text{id}_{\underline{\text{Alg}}(B, T)}, \text{id}_{\underline{\text{Alg}}(A, B)})$ , we obtain a composition morphism  $\circ : \underline{\text{Alg}}(B, T) \times \underline{\text{Alg}}(A, B) \rightarrow \underline{\text{Alg}}(A, T)$  such that for  $(d, c) \in \underline{\text{Alg}}(B, T) \times \underline{\text{Alg}}(A, B)$ , we have  $u_d \circ u_c = u_{d \circ c}$  (where  $u$  is as in Definition 20).

Similarly, for any algebra  $A$  we might ask if there is an identity  $\text{id}_A : \mathbb{1} \rightarrow \underline{\text{Alg}}(A, A)$ . We showed above that  $\text{Alg}(A, A) \cong \text{CoAlg}(\mathbb{1}, \underline{\text{Alg}}(A, A))$ . Thus, we take the image of  $\text{id}_A \in \text{Alg}(A, A)$  under this bijection.

We leave it as an exercise for the interested reader to show by hand that this constitutes an enrichment of  $\text{Alg}$  in  $(\text{CoAlg}, \otimes, \mathbb{1})$ , i.e., that all the axioms for an enriched category are satisfied by this choice of composition and identities. We will instead leave this result (Theorem 31) to the general setting.

### 2.4 The convolution algebra

We now give an alternative representation of  $\mu_C(A, B)$  that can be directly defined and computed. In this section, we will be able to use it to compute  $\underline{\text{Alg}}(A, B)$  without appealing to [7, Lemma 37].

In Definition 2, we defined  $\mu_C(A, B)$  to be a certain subset of  $\text{Set}(C, \text{Set}(A, B)) \cong \text{Set}(A, \text{Set}(C, B))$ . We now identify that subset as the subset of (total) algebra homomorphisms  $A \rightarrow \text{Set}(C, B)$  with a particular *convolution* algebra structure on  $\text{Set}(C, B)$ .

► **Definition 7** (Convolution algebra, cf. Definition 27). *Given a coalgebra  $(C, \chi_C)$  and an algebra  $(B, \alpha_B)$ , define the convolution algebra  $[C, B]$  to be the algebra whose underlying set is  $\text{Set}(C, B)$ , whose zero is the constant function  $C \rightarrow B$  at  $0_B$ , and where  $f + 1$  is defined by*

$$(f + 1)(c) = \begin{cases} 0_B & \text{if } \llbracket c \rrbracket = 0; \\ f(c - 1) + 1 & \text{if } \llbracket c \rrbracket > 0. \end{cases}$$

This defines a functor  $[-, -] : \text{CoAlg}^{\text{op}} \times \text{Alg} \rightarrow \text{Alg}$ .

Given a coalgebra  $(C, \chi_C)$  and an algebra  $(B, \alpha_B)$ , a function  $m : C \rightarrow A \rightarrow B$  is a measuring if and only if the associated  $\tilde{m} : A \rightarrow C \rightarrow B$  (under the bijection  $\tilde{\cdot} : \text{Set}(C, \text{Set}(A, B)) \rightarrow \text{Set}(A, \text{Set}(C, B))$ ) underlies a homomorphism  $A \rightarrow [C, B]$  of algebras. Indeed, (M1) of Definition 2 for  $m$  is equivalent to (H1)  $\tilde{m}(0_A) = 0_{[C, B]}$  and criteria (M2) and (M3) for  $f$  are equivalent to (H2)  $\tilde{m}(a + 1) = \tilde{m}(a) + 1$ .

Therefore, we find the following string of bijections, natural in  $C, A, B$ ,

$$\mu_C(A, B) \cong \text{CoAlg}(C, \underline{\text{Alg}}(A, B)) \cong \text{Alg}(A, [C, B]) \quad (1)$$

so that we see that  $\mu_C(-, B)$  is represented by  $[C, B]$ . We can even find a representation for  $\mu_C(-, B)$ , but we will leave this for the more general setting (Theorem 22). The interested reader is encouraged to calculate that other representation in this example.

In practice, when we want to compute  $\underline{\text{Alg}}(A, B)$ , we will compute  $[C, B]$  and then apply the universal property above. We do that now, computing some of the results of Example 6 without appealing to [7, Lemma 37].

► **Example 8.** We compute  $\underline{\text{Alg}}(\mathfrak{n}, B)$  using the right-hand bijection in Equation (1).

We first observe the following for any coalgebra  $Z$ .

$$\text{Alg}(\mathfrak{n}, Z) \cong \begin{cases} * & \text{if } n_Z = (n + 1)_Z \\ \emptyset & \text{otherwise} \end{cases}$$

Since we are considering  $Z := [C, B]$ , we need to understand when  $n_{[C, B]} = (n + 1)_{[C, B]}$ . By definition,  $0_{[C, A]}$  is the constant function at  $0_A$ . Then  $1_{[C, A]}$  is the function that takes every  $c \in C$  of index 0 to  $0_B$ , and every other  $c \in C$  to  $1_B$ . Inductively, we can show that  $n_{[C, B]}(c) = \min(\llbracket c \rrbracket, n)_B$ . Thus,  $n_{[C, B]} = (n + 1)_{[C, B]}$  means that  $\min(\llbracket c \rrbracket, n)_B = \min(\llbracket c \rrbracket, n + 1)_B$  for all  $c \in C$ , and this holds if and only if  $\llbracket c \rrbracket \leq n$  for all  $c \in C$  or  $n_B = (n + 1)_B$ . Now we have the following.

$$\text{CoAlg}(C, \underline{\text{Alg}}(\mathfrak{n}, B)) \cong \text{Alg}(\mathfrak{n}, [C, B]) \cong \begin{cases} * & \text{if } \llbracket c \rrbracket \leq n \text{ for all } c \in C \\ * & \text{if } n_B = (n + 1)_B \\ \emptyset & \text{otherwise} \end{cases} \quad (2)$$

In the case that  $n_B = (n + 1)_B$ , we find that  $\underline{\text{Alg}}(\mathfrak{n}, B)$  has the universal property of the terminal object,  $\mathbb{N}^\infty$ .

Now suppose that  $n_B \neq (n + 1)_B$ . Since  $\text{CoAlg}(C, \mathfrak{n}^\circ) = *$  if and only if  $\llbracket c \rrbracket \leq n$  for all  $c \in C$ ,  $\underline{\text{Alg}}(\mathfrak{n}, B)$  has the universal property of  $\mathfrak{n}^\circ$ .

## 15:8 Coinductive Control of Inductive Data Types

Now we have calculated the following

$$\underline{\text{Alg}}(\mathfrak{n}, B) = \begin{cases} \mathbb{N}^\infty & \text{if } n_B = (n+1)_B \\ \mathfrak{n}^\circ & \text{otherwise} \end{cases}$$

This aligns with our expectations, since there is a total homomorphism  $\mathfrak{n} \rightarrow B$  if  $n_B = (n+1)_B$  but there is only an  $n$ -partial homomorphism  $\mathfrak{n} \rightarrow B$  otherwise.

Finally, note that taking  $B := \mathbb{N}$ , we have calculated  $\underline{\text{Alg}}(\mathfrak{n}, \mathbb{N})$ , the *dual* (Definition 29) of  $\mathfrak{n}$ , to be  $\mathfrak{n}^\circ$ .

### 2.5 Generalizing initial objects

Now we turn to the question of specifying algebras other than  $\mathbb{N}$  via a generalization of the notion of initial algebra.

The fact that  $\mathbb{N}$  is the initial object in  $\text{Alg}$  means that the algebra structure on an algebra  $A$  specifies exactly one total algebra homomorphism  $\mathbb{N} \rightarrow A$ , and this can be constructed inductively. Now we have introduced the notion of partial homomorphism which can be constructed by partial induction (Construction 1). Thus, we might ask if we can formalize a notion of being initial with respect to partial homomorphisms and partial induction.

Our calculations in this section so far have perhaps given us the intuition that the algebra  $\mathfrak{n}$  represents  $n$ -partial homomorphisms in the way that  $\mathbb{N}$  represents total homomorphisms. Indeed, from Equation (2), there is a unique measuring  $f : \mathfrak{n}^\circ \rightarrow \mathfrak{n} \rightarrow B$  for any algebra  $B$ . Now we try to capture and elucidate this fact by rephrasing it to say that  $\mathfrak{n}$  is a certain kind of initial object with respect to such partial homomorphisms.

There are multiple equivalent definitions of initial object, and we choose the one that is amenable to generalization. We choose to define an initial object in a category  $\mathcal{C}$  as an object  $I \in \mathcal{C}$  such that there is a unique function  $1 \rightarrow \mathcal{C}(I, X)$  for all  $X \in \mathcal{C}$ . Now we have brought to the surface a parameter, here 1, that we can vary, inspired by the theory of weighted limits.

► **Definition 9** (*C*-initial algebra, cf. Definition 35). *For a coalgebra  $C$ , we define a  $C$ -initial algebra to be an algebra  $A$  such that there is a unique coalgebra morphism  $C \rightarrow \underline{\text{Alg}}(A, X)$  for all algebras  $X$ .*

► **Remark 10.** One may wonder what would happen if for a *set*  $S$ , we defined an  $S$ -initial algebra to be an algebra  $A$  such that there is a unique *function*  $S \rightarrow \text{Alg}(A, X)$  for all  $X \in \text{Alg}$ . But every algebra is an  $\emptyset$ -initial algebra, and an  $S$ -initial algebra is an initial algebra for any  $S \neq \emptyset$  (because functions  $S \rightarrow T$  are unique only when  $S = \emptyset$  or  $T \cong 1$ ). Thus, we need to consider  $\underline{\text{Alg}}(A, X)$  and not just  $\text{Alg}(A, X)$  to obtain interesting  $C$ -initial algebras.

► **Example 11.** We have shown in Example 8 that  $\mathfrak{n}$  is an  $\mathfrak{n}^\circ$ -initial algebra.

Since  $\text{Alg}(A, X) \cong \text{CoAlg}(\mathbb{1}, \underline{\text{Alg}}(A, X))$ , the initial algebra  $\mathbb{N}$  is the (only)  $\mathbb{1}$ -initial algebra. In fact, since  $\underline{\text{Alg}}(\mathbb{N}, X) = \mathbb{N}^\infty$  for all  $X$  by [7, Lemma 37] or by a similar computation to Example 8, we find that  $\mathbb{N}$  is a  $C$ -initial algebra for any subterminal coalgebra (i.e.,  $\emptyset, \mathfrak{n}^\circ, \mathbb{N}^-, \mathbb{N}^\infty$ ).

Now we see that for instance, both  $\mathfrak{n}$  and  $\mathbb{N}$  are  $\mathfrak{n}^\circ$ -initial algebras. Thus,  $\mathfrak{n}^\circ$ -initial algebras are not determined up to isomorphism as initial algebras are. This captures the fact that for an algebra  $B$ , we can construct  $n$ -partial homomorphisms from both  $\mathfrak{n}$  and  $\mathbb{N}$  to  $B$ .

► **Definition 12** (Terminal  $C$ -initial algebra, cf. Definition 35). *Consider the category whose objects are  $C$ -initial algebras, and whose morphisms  $A \rightarrow B$  are total algebra homomorphisms  $A \rightarrow B$ . Then we call the terminal object of this category the terminal  $C$ -initial algebra.*

► **Example 13.** Since the only  $\mathbb{1}$ -initial algebra is  $\mathbb{N}$ , it is also the terminal  $C$ -initial algebra.

We want to show that  $\mathfrak{n}$  is the terminal  $\mathfrak{n}^\circ$ -initial algebra. However, we need another computational tool. This is in fact an alternate generalization of the notion of initial algebra.

Above, we might have observed that an initial object can be characterized as the limit of the identity functor and then, following the theory of weighted limits, considered objects  $\lim^C \text{id}_{\text{Alg}}$  with the following universal property.

$$\text{Alg}(A, \lim^C \text{id}_{\text{Alg}}) \cong \lim_{X \in \text{Alg}} \text{CoAlg}(C, \text{Alg}(A, X))$$

We can immediately calculate (Proposition 36) that  $\lim^C \text{id}_{\text{Alg}}$  is  $C^* := [C, \mathbb{N}]$ , the dual of  $C$  (Definition 29). By the bijection above, there is a unique total algebra homomorphism from each  $C$ -initial object to  $\lim^C \text{id}_{\text{Alg}}$ . This will help us understand the possible structure that a  $C$ -initial object can have. But first, we must understand the structure of  $C^*$ .

► **Example 14.** Let  $C := \mathfrak{n}^\circ$ . Then elements of  $[\mathfrak{n}^\circ, \mathbb{N}]$  are sequences of  $n+1$  natural numbers.

The successor of a sequence  $(a_i)_{i=0}^n$  is  $(b_i)_{i=0}^n$  where  $b_0 = 0$  and  $b_{i+1} = a_i + 1$ . Notice that the successor of  $(b_i)_{i=0}^n$  is  $(c_i)_{i=0}^n$  where  $c_0 = 0$ ,  $c_1 = 1$  and otherwise  $c_{i+2} = a_i + 2$ . Thus, we can inductively show that the  $(n+1)$ -st successor of any element of  $[\mathfrak{n}^\circ, \mathbb{N}]$  is the sequence  $(i)_{i=0}^n$ , and the successor of this sequence is itself.

We claim that the unique morphism  $!_{[\mathfrak{n}^\circ, \mathbb{N}]} : \mathbb{N} \rightarrow [\mathfrak{n}^\circ, \mathbb{N}]$  factors through  $\mathfrak{n}$ . We have  $m_{[\mathfrak{n}^\circ, \mathbb{N}]} = (\min(i, m))_{i=0}^n$ . Thus, the restriction of the map  $!_{[\mathfrak{n}^\circ, \mathbb{N}]}$  to  $\{0, \dots, n\} \subset \mathbb{N}$  is injective, and  $n_{[\mathfrak{n}^\circ, \mathbb{N}]} = m_{[\mathfrak{n}^\circ, \mathbb{N}]}$  for all  $m \geq n$ .

► **Example 15.** Now we can show that  $\mathfrak{n}$  is the terminal  $\mathfrak{n}^\circ$ -initial algebra. In this calculation, we use of the law of excluded middle for the only time in this paper.

Consider an  $\mathfrak{n}^\circ$ -initial algebra  $A$ .

First, we show that every  $a \in A$  is either the basepoint or a successor. So suppose that there is an element  $a \in A$  that is not a basepoint or successor, and consider an algebra  $B$  with more than one element. Then for any  $b \in B$  and any measure  $f : \mathfrak{n}^\circ \rightarrow A \rightarrow B$ , we can form a measure  $\tilde{f} : \mathfrak{n}^\circ \rightarrow A \rightarrow B$  such that  $\tilde{f}_n(a) = b$  and  $\tilde{f}$  agrees with  $f$  everywhere else, since Definition 2 imposes no requirements on  $\tilde{f}_n(a)$ . Thus, there are multiple measures  $\mathfrak{n}^\circ \rightarrow A \rightarrow B$ , equivalently total algebra homomorphisms  $\mathfrak{n}^\circ \rightarrow \text{Alg}(A, B)$ , so we find a contradiction.

Now, we consider the unique map  $A \rightarrow [\mathfrak{n}^\circ, \mathbb{N}]$  and claim that this factors through the injection  $\mathfrak{n} \rightarrow [\mathfrak{n}^\circ, \mathbb{N}]$ , so that there is a unique  $A \rightarrow \mathfrak{n}$ . Since every element of  $A$  is either a basepoint or a successor, every element of  $A$  is either of the form  $n_A$  or has infinitely many predecessors. The elements of the form  $n_A$  are mapped those to of the form  $n_{[\mathfrak{n}^\circ, \mathbb{N}]}$ , and the elements who have infinitely many predecessors can only be mapped to the “top element”  $n_{[\mathfrak{n}^\circ, \mathbb{N}]} = (i)_{i=0}^n$ , since this is the only element which has an  $m$ -th predecessor for any  $m \in \mathbb{N}$ . Thus, the unique  $A \rightarrow [\mathfrak{n}^\circ, \mathbb{N}]$  indeed factors through  $\mathfrak{n}$ .

Thus we have shown how to specify algebras of the form  $\mathfrak{n}$  in a way analogous to the specification of  $\mathbb{N}$  as an initial algebra. After determining an algebra structure on a set  $A$ , we obtain a unique  $n$ -partial algebra homomorphism  $\mathfrak{n} \rightarrow A$ .

### 3 General theory

In this section, we now generalize the results of the previous section. So fix a symmetric monoidal category  $(\mathcal{C}, \otimes, \mathbb{1})$  and a lax symmetric monoidal endofunctor  $(F, \nabla, \eta)$  (defined below in Definition 16) on  $\mathcal{C}$ .

#### 3.1 Measuring coalgebras

In this section, we define the general notion of *measuring* for  $F$ . Note that in Section 2.2 above, it was convenient to define a measuring to be a certain kind of function  $C \rightarrow A \rightarrow B$ , but here we first define the notion of measuring without requiring the monoidal structure to be closed. That is, we define a measuring to be a certain kind of function  $C \otimes A \rightarrow B$ .

► **Definition 16.** *That  $(F, \nabla, \eta)$  is a lax symmetric monoidal endofunctor means that  $F$  is an endofunctor on  $\mathcal{C}$  with*

(L1) *a natural transformation  $\nabla_{X,Y}: F(X) \otimes F(Y) \rightarrow F(X \otimes Y)$ , for all  $X, Y \in \mathcal{C}$ ; and*

(L2) *a morphism  $\eta: \mathbb{1} \rightarrow F(\mathbb{1})$  in  $\mathcal{C}$ ;*

*such that  $(F, \nabla, \eta)$  is associative, unital and commutative, as described in [7, Appendix A.2].*

► **Example 17.** In Section 2, we considered the (cartesian closed) symmetric monoidal category  $(\mathbf{Set}, \times, 1)$ . For the endofunctor  $\text{id}+1$ , we define  $\nabla_{X,Y}: (X+1) \times (Y+1) \rightarrow (X \times Y)+1$  to take  $(x, y) \mapsto (x, y)$ ,  $(\mathfrak{t}, y) \mapsto \mathfrak{t}$ ,  $(x, \mathfrak{t}) \mapsto \mathfrak{t}$ ,  $(\mathfrak{t}, \mathfrak{t}) \mapsto \mathfrak{t}$  for  $x \in X, y \in Y, \mathfrak{t} \in 1$ . We define  $\eta: 1 \rightarrow 1+1$  to be the inclusion into the first summand.

► **Definition 18 (Measuring, cf. Definition 2).** *Consider algebras  $(A, \alpha)$  and  $(B, \beta)$ , and a coalgebra  $(C, \chi)$ . We call a map  $\phi: C \otimes A \rightarrow B$  a measuring from  $A$  to  $B$  if it makes the following diagram commute.*

$$\begin{array}{ccccc}
 & & F(C) \otimes F(A) & \xrightarrow{\nabla_{C,A}} & F(C \otimes A) & \xrightarrow{F(\phi)} & F(B) \\
 & \nearrow \chi \otimes \text{id} & & & & & \downarrow \beta \\
 C \otimes F(A) & & & & C \otimes A & \xrightarrow{\phi} & B \\
 & \searrow \text{id} \otimes \alpha & & & & & 
 \end{array}$$

We denote by  $\mu_C(A, B)$  the set of all measurings  $C \otimes A \rightarrow B$ .

If  $\phi: C \otimes A \rightarrow B$  is a measuring,  $a: (A', \alpha') \rightarrow (A, \alpha)$  and  $b: (B, \beta) \rightarrow (B', \beta')$  are algebra homomorphisms, and  $c: (C', \chi') \rightarrow (C, \chi)$  is a coalgebra homomorphism, then one can check that the composite

$$C' \otimes A' \xrightarrow{c \otimes a} C \otimes A \xrightarrow{\phi} B \xrightarrow{b} B'$$

is a measuring. Therefore, the assignment  $C, A, B \mapsto \mu_C(A, B)$  underlies a functor

$$\mu: \mathbf{CoAlg}^{\text{op}} \times \mathbf{Alg}^{\text{op}} \times \mathbf{Alg} \rightarrow \mathbf{Set}.$$

We shall see that this functor is representable in each of its variables under reasonable hypotheses.

► **Example 19.** The monoidal unit  $\mathbb{1}$  of  $\mathcal{C}$  is a coalgebra via the lax symmetric monoidal structure  $\eta: \mathbb{1} \rightarrow F(\mathbb{1})$ . Thus morphisms  $A \rightarrow B$  in  $\mathcal{C}$  are in bijection with morphisms  $\mathbb{1} \otimes A \rightarrow B$  in  $\mathcal{C}$ , and one can check that a morphism  $A \rightarrow B$  in  $\mathcal{C}$  is an algebra homomorphism if and only if  $\mathbb{1} \otimes A \rightarrow B$  is a measure. Thus,  $\mu_{\mathbb{1}}(A, B) \cong \mathbf{Alg}(A, B)$ .

► **Definition 20** (Universal measuring, cf. Definition 5). *Let  $A$  and  $B$  be algebras.*

*We define the category of measurings from  $A$  to  $B$  to be the category whose objects are pairs  $(C; f)$  of a coalgebra  $C$  and a measuring  $f : C \otimes A \rightarrow B$ , and whose morphisms  $(C; f) \rightarrow (D; g)$  are coalgebra homomorphisms  $d : C \rightarrow D$  such that  $f = g(d \otimes A)$ .*

*The universal measuring from  $A$  to  $B$ , denoted  $(\underline{\text{Alg}}(A, B), \text{ev})$ , is the terminal object (if it exists) in the category of measurings from  $A$  to  $B$ . That is, if  $(C; f)$  is a measuring from  $A$  to  $B$ , then there is a unique morphism  $! : C \rightarrow \underline{\text{Alg}}(A, B)$  that makes the following diagram commute.*

$$\begin{array}{ccc} C \otimes A & \xrightarrow{f} & B \\ \downarrow ! \otimes A & \searrow \text{ev} & \\ \underline{\text{Alg}}(A, B) \otimes A & & \end{array}$$

If a universal measuring  $(\underline{\text{Alg}}(A, B), \text{ev})$  exists, then we obtain a representation  $\underline{\text{Alg}}(A, B)$  for  $\mu_{-}(A, B) : \text{CoAlg}^{\text{op}} \rightarrow \text{Set}$ . That is, we have the following bijection, natural in  $C, A, B$ .

$$\mu_C(A, B) \cong \text{CoAlg}(C, \underline{\text{Alg}}(A, B)).$$

In the following sections, we will show that if  $\mathcal{C}$  is closed and locally presentable and  $F$  is accessible, then the universal measuring always exists.

### 3.2 Local presentability, accessibility, and the measuring tensor

We will now usually require that  $\mathcal{C}$  be *locally presentable* and  $F$  is *accessible* [2, Def. 1.17 & 2.17]. Then  $\text{Alg}$  and  $\text{CoAlg}$  are also locally presentable, the forgetful functor  $\text{Alg} \rightarrow \mathcal{C}$  has a left adjoint  $\text{Fr}$ , and the forgetful functor  $\text{CoAlg} \rightarrow \mathcal{C}$  has a right adjoint  $\text{Cof}$  [2, Cor. 2.75 & Ex. 2.j]. We will also use that these categories, as locally presentable categories, are complete and cocomplete.

► **Example 21.**  $\text{Set}$  is locally presentable and  $\text{id} + 1$  is accessible.

If  $\mathcal{C}$  is locally presentable and  $F$  is accessible, then for a coalgebra  $(C, \chi)$ , and algebras  $(A, \alpha)$  and  $(B, \beta)$ , a map  $\phi : C \otimes A \rightarrow B$  uniquely determines an algebra homomorphism  $\phi' : \text{Fr}(C \otimes A) \rightarrow (B, \beta)$ . Notice then that a map  $\phi : C \otimes A \rightarrow B$  is a measuring if and only if both composites from  $\text{Fr}(C \otimes FA)$  to  $(B, \beta)$  coincide in the following diagram.

$$\text{Fr}(C \otimes FA) \xrightarrow[f]{\text{Fr}(\text{id}_C \otimes \alpha)} \text{Fr}(C \otimes A) \xrightarrow{\phi'} (B, \beta)$$

In the above,  $f$  is obtained as adjunct under the free-forgetful adjunction of the composition

$$C \otimes FA \xrightarrow{\chi \otimes \text{id}} FC \otimes FA \xrightarrow{\nabla_{C,A}} F(C \otimes A) \xrightarrow{F(i)} F(\text{Fr}(C \otimes A)) \xrightarrow{\alpha_{\text{Fr}}} \text{Fr}(C \otimes A),$$

in which  $i$  is the unit of the free-forgetful adjunction and  $\alpha_{\text{Fr}}$  is the algebra structure on the free algebra  $\text{Fr}(C \otimes A)$ . We have now shown the following.

► **Theorem 22.** *Suppose that  $\mathcal{C}$  is locally presentable and  $F$  is accessible. Consider a coalgebra  $C$  and an algebra  $A$ . Then the coequalizer of the following diagram in  $\text{Alg}$  exists, and we denote it by  $C \triangleright A$  and call it the measuring tensor of  $C$  and  $A$ .*

$$\text{Fr}(C \otimes FA) \rightrightarrows \text{Fr}(C \otimes A) \dashrightarrow C \triangleright A.$$

## 15:12 Coinductive Control of Inductive Data Types

Given any algebra  $B$ , a measuring  $\phi : C \otimes A \rightarrow B$  uniquely corresponds to an algebra homomorphism  $C \triangleright A \rightarrow B$ . In other words, we obtain a natural identification

$$\mu_C(A, B) \cong \text{Alg}(C \triangleright A, B).$$

That is, the functor  $\mu_C(A, -) : \text{Alg} \rightarrow \text{Set}$  is represented by  $C \triangleright A$ .

In the following sections, we will also construct representing objects for  $\mu_C(-, B)$  and  $\mu_-(A, B)$ .

### 3.3 Measurements as partial homomorphisms

Now we will often assume that the symmetric monoidal structure of  $\mathcal{C}$  is closed. Whenever we do, we will denote the internal hom by  $\underline{\mathcal{C}}(-, -)$ . In this section, we provide a dual description of measurings when  $\mathcal{C}$  is closed, generalizing Definition 2.

Note that since  $F$  is lax monoidal, it is also *lax closed*: that is, there is a map

$$\widetilde{\nabla}_{X, Y} : F(\underline{\mathcal{C}}(X, Y)) \rightarrow \underline{\mathcal{C}}(FX, FY)$$

natural in  $X, Y \in \mathcal{C}$ . Indeed, this is the adjunct under the adjunction  $- \otimes FX \dashv \underline{\mathcal{C}}(FX, -)$  of the composition

$$F(\underline{\mathcal{C}}(X, Y)) \otimes F(X) \xrightarrow{\nabla_{\underline{\mathcal{C}}(X, Y), X}} F(\underline{\mathcal{C}}(X, Y) \otimes X) \xrightarrow{F(\text{ev}_X)} F(Y),$$

in which  $\text{ev}_X$  is the counit of the adjunction  $- \otimes X \dashv \underline{\mathcal{C}}(X, -)$ .

Given a closed monoidal structure, we can connect the notion of measuring with our notion of partial homomorphism from Section 2.

► **Proposition 23** (cf. Definition 2). *Suppose that  $\mathcal{C}$  is closed. Given algebras  $(A, \alpha)$  and  $(B, \beta)$  and a coalgebra  $(C, \chi)$ , a map  $\phi : C \otimes A \rightarrow B$  is a measuring if and only if its adjunct  $\tilde{\phi} : C \rightarrow \underline{\mathcal{C}}(A, B)$  fits in the following commutative diagram*

$$\begin{array}{ccccc} & & F(C) & \xrightarrow{F(\tilde{\phi})} & F(\underline{\mathcal{C}}(A, B)) & \xrightarrow{\widetilde{\nabla}_{A, B}} & \underline{\mathcal{C}}(FA, FB) \\ & \nearrow \chi & & & & & \downarrow \beta_* \\ C & & & & \underline{\mathcal{C}}(A, B) & \xrightarrow{\alpha^*} & \underline{\mathcal{C}}(FA, B) \end{array}$$

where  $\alpha^*$  denotes precomposition by  $\alpha$  and  $\beta_*$  denotes postcomposition by  $\beta$ . We shall also refer to the pair  $(C; \tilde{\phi})$  as a measuring.

► **Example 24.** Note that the cartesian monoidal structure on  $\text{Set}$  is closed, and that the above recovers Definition 2.

This approach allows us to reformulate the notion of measuring as certain coalgebra homomorphisms which we now describe. If  $\mathcal{C}$  is locally presentable and  $F$  is accessible, then given a coalgebra  $(C, \chi)$  and algebras  $(A, \alpha)$  and  $(B, \beta)$ , a map  $\phi : C \rightarrow \underline{\mathcal{C}}(A, B)$  in  $\mathcal{C}$  uniquely determines a coalgebra homomorphism  $\phi' : (C, \chi_C) \rightarrow \text{Cof}(\underline{\mathcal{C}}(A, B))$ . A map  $\phi : C \rightarrow \underline{\mathcal{C}}(A, B)$  is a measuring if and only if both composites from  $(C, \chi_C)$  to  $\text{Cof}(\underline{\mathcal{C}}(FA, B))$  in the following diagram coincide.

$$(C, \chi_C) \xrightarrow{\phi'} \text{Cof}(\underline{\mathcal{C}}(A, B)) \xrightarrow[\text{f}]{\text{Cof}(\underline{\mathcal{C}}(\alpha, B))} \text{Cof}(\underline{\mathcal{C}}(FA, B))$$



In the above,  $f$  is the adjunct under the cofree-forgetful adjunction of the following composite.

$$\mathrm{Cof}(\underline{\mathcal{C}}(A, B)) \xrightarrow{\chi_{\mathrm{Cof}}} F(\mathrm{Cof}(\underline{\mathcal{C}}(A, B))) \xrightarrow{F(\varepsilon)} F(\underline{\mathcal{C}}(A, B)) \xrightarrow{\widetilde{\nabla}_{A,B}} \underline{\mathcal{C}}(F(A), F(B)) \xrightarrow{\beta_*} \underline{\mathcal{C}}(F(A), B).$$

Here  $\chi_{\mathrm{Cof}}$  is the coalgebraic structure on the cofree coalgebra, and  $\varepsilon$  is the counit of the cofree-forgetful adjunction.

Now we can use this to guarantee the existence of a universal measuring.

► **Theorem 25** (Proof in [7, Appendix A.3]). *Suppose that  $\mathcal{C}$  is locally presentable and closed and that  $F$  is accessible. Given algebras  $A$  and  $B$ , then the universal measuring coalgebra  $\underline{\mathrm{Alg}}(A, B)$  exists and is obtained as the following equalizer diagram in  $\mathrm{CoAlg}$*

$$\underline{\mathrm{Alg}}(A, B) \overset{\mathrm{eq}}{\dashrightarrow} \mathrm{Cof}(\underline{\mathcal{C}}(A, B)) \rightrightarrows \mathrm{Cof}(\underline{\mathcal{C}}(F(A), B)),$$

with  $\widetilde{\mathrm{ev}}: \underline{\mathrm{Alg}}(A, B) \rightarrow \underline{\mathcal{C}}(A, B)$  obtained as the composition of the equalizer map  $\mathrm{eq}$  together with the counit  $\mathrm{Cof}(\underline{\mathcal{C}}(A, B)) \rightarrow \underline{\mathcal{C}}(A, B)$  of the cofree-forgetful adjunction.

► **Corollary 26.** *Suppose that  $\mathcal{C}$  is locally presentable and closed and that  $F$  is accessible. Given algebras  $A$  and  $B$ , the functor  $\mu_-(A, B) : \mathrm{CoAlg}^{\mathrm{op}} \rightarrow \mathrm{Set}$  is represented by  $\underline{\mathrm{Alg}}(A, B)$ .*

### 3.4 Measuring via the convolution algebra

We will now describe the last representable object for the measuring functor.

► **Definition 27** (Convolution algebra, cf. Definition 7). *Suppose that  $\mathcal{C}$  is closed. Given a coalgebra  $(C, \chi)$  and an algebra  $(A, \alpha)$  in  $\mathcal{C}$ , we define an algebra structure on  $\underline{\mathcal{C}}(C, A)$ , called the convolution algebra, which is denoted  $[(C, \chi), (A, \alpha)]$  or simply  $[C, A]$ , as follows. The algebra structure  $F[C, A] \rightarrow [C, A]$  is the composition*

$$F(\underline{\mathcal{C}}(C, A)) \xrightarrow{\widetilde{\nabla}_{C,A}} \underline{\mathcal{C}}(FC, FA) \xrightarrow{\alpha_* \chi^*} \underline{\mathcal{C}}(C, A),$$

where  $\alpha_* \chi^*$  denotes postcomposition by  $\alpha$  and precomposition by  $\chi$ . The convolution algebra construction lifts the internal hom to a functor

$$[-, -] : \mathrm{CoAlg}^{\mathrm{op}} \times \mathrm{Alg} \longrightarrow \mathrm{Alg}.$$

The convolution algebra provides a representing object for  $\mu_C(-, B) : \mathrm{Alg}^{\mathrm{op}} \rightarrow \mathrm{Set}$ . Indeed, we have the following bijection natural in  $C, A, B$ .

$$\mu_C(A, B) \cong \mathrm{Alg}(A, [C, B]).$$

In other words, a measuring  $\phi : C \otimes A \rightarrow B$  corresponds to an algebra homomorphism  $\phi' : A \rightarrow [C, B]$  under the bijection  $\mathcal{C}(C \otimes A, B) \cong \mathcal{C}(A, \underline{\mathcal{C}}(C, B))$ . Indeed, notice that  $\phi'$  is a homomorphism if and only if the following diagram, adjunct to the one appearing in Definition 18, commutes.

$$\begin{array}{ccc} & & F([C, B]) \xrightarrow{\widetilde{\nabla}_{C,B}} [F(C), F(B)] \\ & \nearrow^{F(\phi')} & \\ F(A) & & \\ & \searrow_{\alpha} & A \xrightarrow{\phi'} [C, B] \end{array} \quad \begin{array}{c} \downarrow \beta_* \chi^* \\ \\ \end{array}$$

## 15:14 Coinductive Control of Inductive Data Types

► **Remark 28.** The convolution algebra also provides an alternative characterization of the algebra  $C \triangleright A$  and coalgebra  $\underline{\text{Alg}}(A, B)$ . As limits in  $\text{Alg}$  and colimits in  $\text{CoAlg}$  are determined in  $\mathcal{C}$  [1] and the internal hom  $\underline{\mathcal{C}}(-, -) : \mathcal{C}^{\text{op}} \times \mathcal{C} \rightarrow \mathcal{C}$  preserves limits, the functor  $[-, -] : \text{CoAlg}^{\text{op}} \times \text{Alg} \rightarrow \text{Alg}$  also preserves limits. Moreover, fixing a coalgebra  $C$ , the induced functor  $[C, -] : \text{Alg} \rightarrow \text{Alg}$  is accessible since filtered colimits in  $\text{Alg}$  are computed in  $\mathcal{C}$  (see [1, 5.6]). Therefore, by the adjoint functor theorem [2, 1.66], the functor  $[C, -]$  is a right adjoint. Its left adjoint is precisely  $C \triangleright - : \text{Alg} \rightarrow \text{Alg}$ . Indeed, for any algebras  $A$  and  $B$ , we obtain the following bijection, natural in  $C, A, B$ .

$$\text{Alg}(C \triangleright A, B) \cong \text{Alg}(A, [C, B]).$$

Notice we can also determine the universal measuring by using the adjoint functors. Fixing now an algebra  $B$ , the opposite functor  $[-, B]^{\text{op}} : \text{CoAlg} \rightarrow \text{Alg}^{\text{op}}$  preserves colimits, where the domain is locally presentable and the codomain is essentially locally small. By the adjoint functor theorem [2, 1.66] and [4, 5.5.2.10], this functor is a left adjoint. Its right adjoint is precisely the functor  $\underline{\text{Alg}}(-, B) : \text{Alg}^{\text{op}} \rightarrow \text{CoAlg}$ . Indeed, for any algebra  $A$  and  $B$  and any coalgebra  $C$ , we have the following bijection, natural in  $C, A, B$ .

$$\text{CoAlg}(C, \underline{\text{Alg}}(A, B)) \cong \text{Alg}(A, [C, B]).$$

Combining the identifications, we see that the measuring functor is representable in each factor:

$$\mu_C(A, B) \cong \text{CoAlg}(C, \underline{\text{Alg}}(A, B)) \cong \text{Alg}(A, [C, B]) \cong \text{Alg}(C \triangleright A, B).$$

In other words, for any algebra  $A$  and  $B$  and any coalgebra  $C$ , the following data are equivalent.

$C \otimes A \rightarrow B$ measuring	$C \rightarrow \underline{\mathcal{C}}(A, B)$ measuring	$C \rightarrow \underline{\text{Alg}}(A, B)$ coalgebra homomorphism	$A \rightarrow [C, B]$ algebra homomorphism	$C \triangleright A \rightarrow B$ algebra homomorphism
--	--	---	---	---

► **Definition 29.** Assuming that  $\mathcal{C}$  is locally presentable and  $F$  is accessible,  $\text{Alg}$  has an initial object which we denote by  $N$ .

Let  $(-)^* : \text{CoAlg}^{\text{op}} \rightarrow \text{Alg}$  denote the functor  $[-, N]$ , and call  $C^*$  the dual algebra of  $C$  for any coalgebra  $C$ .

Let  $(-)^{\circ} : \text{Alg}^{\text{op}} \rightarrow \text{CoAlg}$  denote the functor  $\underline{\text{Alg}}(-, N)$ , and call  $A^{\circ}$  the dual coalgebra of  $A$  for any algebra  $A$ .

These functors form a dual adjunction since we have the following bijection, natural in  $C, A$ :

$$\text{Alg}(A, C^*) \cong \text{CoAlg}(C, A^{\circ}).$$

### 3.5 Measuring as an enrichment

We now come to the main punchline of the general theory presented in this paper: that  $\underline{\text{Alg}}(-, -)$  gives the category of algebras an enrichment in coalgebras. First, we describe how to compose measurings.

► **Proposition 30.** The category  $\text{CoAlg}$  has a symmetric monoidal structure for which the forgetful functor  $\text{CoAlg} \rightarrow \mathcal{C}$  is strong symmetric monoidal.

**Proof.** Suppose  $(C, \chi_C)$  and  $(D, \chi_D)$  are coalgebras. Then  $C \otimes D$  has the following coalgebra structure.

$$C \otimes D \xrightarrow{\chi_C \otimes \chi_D} F(C) \otimes F(D) \xrightarrow{\nabla_{C,D}} F(C \otimes D)$$

The morphism  $\eta : \mathbb{1} \rightarrow F(\mathbb{1})$  provides the coalgebraic structure on  $\mathbb{1}$ . One can verify that  $(\mathbf{CoAlg}, \otimes, (\mathbb{1}, \eta))$  is a symmetric monoidal category (see details in [7, Appendix A.4]). ◀

Now we can prove our main theorem.

► **Theorem 31** (Proof in [7, Appendix A.5]). *Suppose that  $\mathcal{C}$  is locally presentable and closed and that  $F$  is accessible. Then the category  $\mathbf{Alg}$  is enriched, tensored, and powered over the symmetric monoidal category  $\mathbf{CoAlg}$  respectively via*

$$\mathbf{Alg}^{\mathrm{op}} \times \mathbf{Alg} \xrightarrow{\mathbf{Alg}(-, -)} \mathbf{CoAlg}, \quad \mathbf{CoAlg} \times \mathbf{Alg} \xrightarrow{- \triangleright -} \mathbf{Alg}, \quad \mathbf{CoAlg}^{\mathrm{op}} \times \mathbf{Alg} \xrightarrow{[-, -]} \mathbf{Alg}.$$

► **Example 32** (Details in [7, Appendix A.7]). Suppose that  $\mathcal{C}$  is locally presentable and closed. The following endofunctors on  $\mathcal{C}$  are accessible and lax symmetric monoidal.

- (id) The identity endofunctor  $\mathrm{id}_{\mathcal{C}}$ .
- (A) The constant endofunctor that sends each object to a fixed commutative monoid  $A$  in  $\mathcal{C}$ .
- (GF) The composition  $GF$  of accessible, lax symmetric monoidal endofunctors  $F$  and  $G$ .
- ( $F \otimes G$ ) The pointwise tensor product  $F \otimes G$  of accessible, lax symmetric monoidal endofunctors  $F$  and  $G$ , assuming  $\mathcal{C}$  is closed.
- ( $F + G$ ) The pointwise coproduct  $F + G$  of an accessible, lax symmetric monoidal endofunctor  $F$  and an accessible endofunctor  $G$  equipped with natural transformations  $GX \otimes GY \rightarrow G(X \otimes Y)$ ,  $\lambda : FX \otimes GY \rightarrow G(X \otimes Y)$ ,  $\rho : GX \otimes FY \rightarrow G(X \otimes Y)$  satisfying the axioms described in [7, Appendix A.7], assuming  $\mathcal{C}$  is closed.
- ( $\mathrm{id}^A$ ) The exponential  $\mathrm{id}^A$  for any object  $A$  of  $\mathcal{C}$ , assuming the monoidal product on  $\mathcal{C}$  is cartesian closed.
- ( $W$ -types) A polynomial endofunctor associated to a morphism  $f : X \rightarrow Y$  in  $\mathbf{Set}$ , given a commutative monoid structure on  $Y$  and an oplax symmetric monoidal structure on the preimage functor  $f^{-1} : \mathcal{C} \rightarrow \mathbf{Set}$ .
- (d.e.s.) A discrete equational system, assuming that the monoidal structure on  $\mathcal{C}$  is cocartesian and that  $\mathcal{C}$  has binary products that preserve filtered colimits.

On some occasions, the category of coalgebras of  $F$  can be interesting while its category of algebras is less so. For instance, given an alphabet  $\Sigma$ , coalgebras over the endofunctor  $F(X) = 2 \times X^\Sigma$  in  $\mathbf{Set}$  are automata but the initial algebra remains  $\emptyset$ . To remedy this, we can extend our main result into the following theorem.

► **Theorem 33** (Proof in [7, Appendix A.6]). *Suppose that  $\mathcal{C}$  is locally presentable and closed and that  $F$  is also accessible. Let  $G : \mathcal{C} \rightarrow \mathcal{C}$  be a  $\mathcal{C}$ -enriched functor that is accessible. Then  $\mathbf{Alg}_{GF}$  is enriched, tensored and powered over  $\mathbf{CoAlg}_F$ .*

► **Example 34.** If  $F(X) = 2 \times X^\Sigma$ , we could consider  $G = \mathrm{id} + 1$ , and thus  $\mathbf{Alg}_{GF}$  has  $\mathbb{N}$  as an initial object and remains enriched in automata.

The enrichment of algebras in coalgebras specify a pairing of coalgebras

$$\underline{\mathbf{Alg}}(B, T) \otimes \underline{\mathbf{Alg}}(A, B) \longrightarrow \underline{\mathbf{Alg}}(A, T),$$

## 15:16 Coinductive Control of Inductive Data Types

regarded as an enriched composition, for any algebras  $A$ ,  $B$  and  $T$ . In more details, the above coalgebra homomorphism is induced by the measuring of

$$(\underline{\text{Alg}}(B, T) \otimes \underline{\text{Alg}}(A, B)) \otimes A \xrightarrow{\text{id} \otimes \text{ev}_{A, B}} \underline{\text{Alg}}(B, T) \otimes B \xrightarrow{\text{ev}_{B, T}} T.$$

In other words, the enrichment is recording precisely that we can compose a measuring  $C \otimes A \rightarrow B$  with  $D \otimes B \rightarrow T$  to obtain a measuring  $(D \otimes C) \otimes A \rightarrow T$ . In particular, our above discussion shows that  $\underline{\text{Alg}}(A, A)$  is always a monoid object in the symmetric monoidal category  $(\text{CoAlg}, \otimes, \mathbb{1})$ .

### 3.6 General $C$ -initial objects

Now we generalize Section 2.5. We can use the extra structure in the enriched category of algebras to specify more algebras than we could in the unenriched category of algebras.

► **Definition 35** ( $C$ -initial algebra, cf. Definition 9 and Definition 12). *Suppose that  $\mathcal{C}$  is locally presentable and closed and that  $F$  is accessible.*

*Given a coalgebra  $C$ , we say an algebra  $A$  is a  $C$ -initial algebra if there exists a unique map  $C \rightarrow \underline{\text{Alg}}(C, X)$ , for all algebras  $X$ .*

*The terminal  $C$ -initial algebra is the terminal object, if it exists, in the subcategory of  $\text{Alg}$  spanned by the  $C$ -initial algebras.*

We end with a result that helped us calculate some terminal  $C$ -initial algebras in Section 2.5.

► **Proposition 36** (Proof in [7, Appendix A.8]). *Suppose that  $\mathcal{C}$  is locally presentable and closed and that  $F$  is accessible. There is a unique map from any  $C$ -initial algebra to  $C^*$ .*

## 4 Conclusions & Vista

In this paper, we have shown that given a closed symmetric monoidal category  $\mathcal{C}$  and an accessible lax symmetric monoidal endofunctor  $F$  on  $\mathcal{C}$ , the category of algebras of  $F$  is enriched, tensored, and cotensored in the category of coalgebras of  $F$ . The algebras of such a functor are of central importance in theoretical computer science, and we hope that identifying such extra structure can shed light on these studies. Indeed, we have demonstrated one use case: we can now specify  $C$ -initial algebras in an analogous way to initial algebras. We identified a large class of examples of endofunctors that are encompassed by our theory. Thus, we have established the beginning of an *enriched* analogue of the theory of  $W$ -types. We have also worked out concretely the results for the endofunctor  $\text{id} + 1$  on  $\text{Set}$ , which suggested a meaningful interpretation of the enrichment as partial algebra homomorphisms.

In future work, we will present similar meaningful interpretations for other endofunctors of our theory. Our future plans involve incorporating features, such as  $C$ -initial algebras, of this new enriched theory into concrete programming languages like Haskell or Agda.

We also seek to extend the results of Example 15 into more general settings and provide conditions for the existence of the terminal  $C$ -initial algebras. We will also develop more robust theory from Theorem 33. Our partial algebra homomorphisms remain total functions: it would be interesting to develop a theory that encodes maps that are partial both as a function and as algebra homomorphisms. Lastly in Example 32, when we consider the constant functor at an object  $A$ , we must choose a commutative monoid structure on  $A$ . What if we had two different monoidal structures on  $A$ ? There are other such choices that are needed in Example 32, for instance in our motivating example of  $W$ -types. We seek to understand how these choices interact with one another.

---

**References**

---

- 1 Jiří Adámek and Hans-E. Porst. On varieties and covarieties in a category. *Mathematical Structures in Computer Science*, 13(2):201–232, 2003. doi:10.1017/S0960129502003882.
- 2 Jiří Adámek and Jiří Rosický. *Locally presentable and accessible categories*, volume 189 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, Cambridge, 1994. doi:10.1017/CB09780511600579.
- 3 Martin Hyland, Ignacio López Franco, and Christina Vasilakopoulou. Hopf measuring comonoids and enrichment. *Proc. Lond. Math. Soc. (3)*, 115(5):1118–1148, 2017. doi:10.1112/plms.12064.
- 4 Jacob Lurie. *Higher topos theory*, volume 170 of *Annals of Mathematics Studies*. Princeton University Press, Princeton, NJ, 2009. doi:10.1515/9781400830558.
- 5 Per Martin-Löf. *Intuitionistic type theory*. Naples: Bibliopolis, 1984.
- 6 Dylan McDermott, Exequiel Rivas, and Tarmo Uustalu. Sweedler theory of monads. In Patricia Bouyer and Lutz Schröder, editors, *Foundations of Software Science and Computation Structures*, pages 428–448, Cham, 2022. Springer International Publishing. doi:10.1007/978-3-030-99253-8\_22.
- 7 Paige Randall North and Maximilien Péroux. Coinductive control of inductive data types, 2023. arXiv:2303.16793.
- 8 Maximilien Péroux. The coalgebraic enrichment of algebras in higher categories. *Journal of Pure and Applied Algebra*, 226(3):106849, 2022. doi:10.1016/j.jpaa.2021.106849.
- 9 Jan Rutten. *The Method of Coalgebra: exercises in coinduction*. C, 2019. URL: <https://ir.cwi.nl/pub/28550/>.
- 10 Moss E. Sweedler. *Hopf algebras*. Mathematics Lecture Note Series. W. A. Benjamin, Inc., New York, 1969.
- 11 Christina Vasilakopoulou. Enriched duality in double categories:  $\mathcal{V}$ -categories and  $\mathcal{V}$ -cocategories. *J. Pure Appl. Algebra*, 223(7):2889–2947, 2019. doi:10.1016/j.jpaa.2018.10.003.



# Weakly Markov Categories and Weakly Affine Monads

Tobias Fritz ✉


Department of Mathematics, Universität Innsbruck, Austria

Fabio Gadducci ✉ 

Department of Computer Science, University of Pisa, Italy

Paolo Perrone ✉ 

Department of Computer Science, University of Oxford, UK

Davide Trotta ✉ 

Department of Computer Science, University of Pisa, Italy

---

## Abstract

Introduced in the 1990s in the context of the algebraic approach to graph rewriting, *gs-monoidal categories* are symmetric monoidal categories where each object is equipped with the structure of a commutative comonoid. They arise for example as Kleisli categories of commutative monads on cartesian categories, and as such they provide a general framework for effectful computation. Recently proposed in the context of categorical probability, *Markov categories* are *gs-monoidal categories* where the monoidal unit is also terminal, and they arise for example as Kleisli categories of commutative *affine monads*, where *affine* means that the monad preserves the monoidal unit.

The aim of this paper is to study a new condition on the *gs-monoidal structure*, resulting in the concept of *weakly Markov categories*, which is intermediate between *gs-monoidal categories* and *Markov ones*. In a *weakly Markov category*, the morphisms to the monoidal unit are not necessarily unique, but form a group. As we show, these categories exhibit a rich theory of conditional independence for morphisms, generalising the known theory for *Markov categories*. We also introduce the corresponding notion for commutative monads, which we call *weakly affine*, and for which we give two equivalent characterisations.

The paper argues that these monads are relevant to the study of categorical probability. A case at hand is the monad of finite non-zero measures, which is *weakly affine* but not *affine*. Such structures allow to investigate probability without normalisation within an elegant categorical framework.

**2012 ACM Subject Classification** Theory of computation → Models of computation

**Keywords and phrases** String diagrams, *gs-monoidal* and *Markov categories*, categorical probability, *affine monads*

**Digital Object Identifier** 10.4230/LIPIcs.CALCO.2023.16

**Funding** Tobias Fritz: FWF P 35992-N.

Fabio Gadducci: MIUR PRIN 2017FTXR “IT-MaTTeRS”.

Paolo Perrone: ERC Grant “BLaSt – A Better Language for Statistics”.

Davide Trotta: MIUR PRIN 2017FTXR “IT-MaTTeRS”.

## 1 Introduction

The idea of *gs-monoidal categories*, which are symmetric monoidal categories equipped with copy and discard morphisms making every object a comonoid, was first introduced in the context of algebraic approaches to term graph rewriting [4], and then developed in a series of papers [5, 6, 7]. Two decades later, similar structures have been rediscovered independently in the context of categorical probability theory, in particular in [2] and [10], under the names of *copy-discard (CD) categories* and *Markov categories*. While “*CD-categories*” and “*gs-monoidal categories*” are synonyms, *Markov categories* have the additional condition that



© Tobias Fritz, Fabio Gadducci, Paolo Perrone, and Davide Trotta;  
licensed under Creative Commons License CC-BY 4.0

10th Conference on Algebra and Coalgebra in Computer Science (CALCO 2023).

Editors: Paolo Baldan and Valeria de Paiva; Article No. 16; pp. 16:1–16:17

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

the monoidal unit is the terminal object (i.e. every morphism commutes with the discard maps), a condition corresponding to normalisation of probability. See [16, Remark 2.2] for a more detailed history of these ideas.

A canonical way of obtaining a gs-monoidal category is as the Kleisli category of a commutative monad on a cartesian monoidal category. As argued in [24], commutative monads can be seen as generalising theories of distributions of some kind, and the fact that their Kleisli categories are gs-monoidal can be seen as the correspondence between distributions and (possibly unnormalised) probability theory. In particular, when the monad is affine (i.e. it preserves the monoidal unit [23, 19]), the Kleisli category is Markov – this can be seen as the correspondence between normalised distributions and probability theory.

In this work we introduce and study an intermediate notion between gs-monoidal and Markov categories, which we call *weakly Markov categories*. These are defined as gs-monoidal categories where for every object its morphisms to the monoidal unit form a group (Definition 3.2). Weakly Markov categories can be interpreted intuitively as gs-monoidal categories where each morphism is discardable up to an invertible normalisation (see Proposition 3.4 for the precise mathematical statement). The choice of the name is due to the fact that every Markov category is (trivially) weakly Markov.

In parallel to weakly Markov categories, we also introduce *weakly affine monads*, which are commutative monads on cartesian monoidal categories preserving the (internal) group structure of the terminal object (Definition 3.5). As a particular concrete example of relevance to probability and measure theory, we consider the monad of finite non-zero measures on **Set** (Example 3.7), and we use it as a running example in the rest of the work. As we show (see Proposition 3.6), a commutative monad on a cartesian monoidal category is weakly affine if and only if its Kleisli category is weakly Markov, analogously to what happens with affine monads and Markov categories.

Markov categories come equipped with a notion of *conditional independence*, which has been one of the main motivations for their use in categorical probability and statistics [2, 10, 15]. It is noteworthy that a notion of conditional independence can also be given for any gs-monoidal category. As we show, for weakly Markov categories it has convenient properties that can be considered “up-to-normalisation” versions of their corresponding Markov-categorical counterpart. These concepts allow us to provide an equivalent condition for weak affinity of a monad, namely a pullback condition on the associativity diagram of the structural morphisms  $c_{X,Y} : TX \times TY \rightarrow T(X \times Y)$  (Theorem 4.7), widely generalising the elementary statement that a monoid is a group if and only if its associativity diagram is a pullback (Proposition 2.1). As such, we believe that weakly affine monads are relevant to the study of categorical probability, as they allow to investigate probability without normalisation within an elegant categorical framework.

## Categorical probability

In the past few years, we have seen a rapid increase in the interest for categorical methods in probability and information theory, and we briefly sketch the basic ideas in order to provide context for this paper.

The first works on categories of stochastic maps, almost as old as category theory and information theory themselves, have been proposed by Lawvere [25] and independently by Chentsov [1]. Subsequently, similar intuitions have been expressed in terms of monads by Giry [18], as well as by Świrszcz in the context of convex analysis [29], and by Jones and Plotkin in computer science [22]. These monads are collectively and informally known as *probability monads*.



Today, Markov categories [2, 10], which generalise categories of stochastic maps and Kleisli categories of probability monads, have been used to express probabilistic concepts *synthetically*, i.e. in terms of basic axioms that the categories satisfy, and of which the usual measure-theoretic proofs are a concrete instance.

The advantages of a categorical approach to probability theory are multiple. First of all, it facilitates an almost-verbatim translation of probabilistic ideas into programming languages, in particular probabilistic programming languages, even in the case of highly complex models. Also, categorical probability comes with a graphical formalism similar to the one of Bayesian networks (see [15] for more details), allowing to represent the structure of stochastic interactions in terms of a graph, for easier interpretation by a human. It is a high-order language, in the sense that it expresses visually some ideas of measure-theoretic significance without requiring measure theory itself, similar to how high-level programming languages spare the programmer from working directly with machine code. Finally, the categorical formalism *complements* the traditional measure-theoretic one in the sense that several concepts which are hard to express or prove with one method are easier to approach using the other method, once the main structures are in place. In this sense, categorical probability is a novel, additional box of tools which provides shortcuts to proofs that would otherwise be lengthy and counterintuitive.

Among concepts that have been expressed and proven this way, we have de Finetti's theorem [14], the Kolmogorov extension theorem and the Kolmogorov and Hewitt-Savage zero-one laws [17], a categorical d-separation criterion [15], theorems on multinomial and hypergeometric distributions [21], theorems on sufficient statistics [10] and on comparison of statistical experiments [13], data processing inequalities in information theory [27], the ergodic decomposition theorem in dynamical systems [26], and results on fresh name generation in theoretical computer science [12].

## Outline

In Section 2 we review the main structures used in this work, in particular group and monoid objects, gs-monoidal and Markov categories, and their interaction with commutative monads.

In Section 3 we define the main original concepts, namely weakly Markov categories and weakly affine monads. We study their relationship and we prove that a commutative monad on a cartesian monoidal category is weakly affine if and only if its Kleisli category is weakly Markov (Proposition 3.6). We then turn to concrete examples using finite measures and group actions (Section 3.3).

In Section 4 we extend the concept of conditional independence from Markov categories to general gs-monoidal categories. We specialise to the weakly Markov case and show that the situation is then similar to what happens in Markov categories, but in a certain precise sense only up to normalisation. We use this formalism to equivalently reformulate weak affinity in terms of a pullback condition (Theorem 4.7). Together with the newly introduced concepts, this result can be considered the main outcome of our work.

Finally, in the concluding Section 5, we pose further questions, such as when we can iterate the construction of weakly Markov categories by means of weakly affine monads, and the relation to strongly affine monads in the sense of Jacobs [20].

## 2 Background

In this section, we develop some relevant background material for later reference. To begin, the following categorical characterisation of groups will be useful to keep in mind.

## 16:4 Weakly Markov Categories and Weakly Affine Monads

► **Proposition 2.1.** *A monoid  $(M, m, e)$  in  $\mathbf{Set}$  is a group if and only if the associativity square*

$$\begin{array}{ccc} M \times M \times M & \xrightarrow{m \times \text{id}} & M \times M \\ \downarrow \text{id} \times m & & \downarrow m \\ M \times M & \xrightarrow{m} & M \end{array} \quad (1)$$

is a pullback.

**Proof.** The square (1) is a pullback of sets if and only if given  $a, g, h, c \in M$  such that  $ag = hc$ , there exists a unique  $b \in M$  such that  $g = bc$  and  $h = ab$ . First, suppose that  $G$  is a group. Then the only possible choice of  $b$  is

$$b = a^{-1}h = gc^{-1}$$

which is unique by uniqueness of inverses.

Conversely, suppose that (1) is a pullback. We can set  $g, h = e$  and  $c = a$  so that  $ae = ea = a$ . Instantiating the pullback property on these elements gives  $b$  such that  $ab = e$  and  $ba = e$ , that is,  $b = a^{-1}$ . ◀

Proposition 2.1 holds generally for a monoid object in a cartesian monoidal category, where the element-wise proof still applies thanks to the following standard observation.

► **Remark 2.2.** Given an object  $M$  in a cartesian monoidal category  $\mathcal{D}$ , there is a bijection between internal monoid structures on  $M$  and monoid structures on every hom-set  $\mathcal{D}(X, M)$  such that pre-composition with any  $f : X \rightarrow Y$  defines a monoid homomorphism

$$\mathcal{D}(Y, M) \longrightarrow \mathcal{D}(X, M).$$

The proof is straightforward by the Yoneda lemma. It follows that Proposition 2.1 holds for internal monoids in cartesian monoidal categories in general.

For the consideration of categorical probability, we now recall the simplest version of a commutative monad of measures. It works with measures taking values in any semiring instead of  $[0, \infty)$  (see e.g. [8, Section 5.1]), but we restrict to the case of  $[0, \infty)$  for simplicity.

► **Definition 2.3.** Let  $X$  be a set. Denote by  $MX$  the set of *finitely supported measures on  $X$* , i.e. the functions  $m : X \rightarrow [0, \infty)$  that are zero for all but a finite number of  $x \in X$ . Given a function  $f : X \rightarrow Y$ , denote by  $Mf : MX \rightarrow MY$  the function sending  $m \in MX$  to the assignment

$$(Mf)(m) : y \longmapsto \sum_{x \in f^{-1}(y)} m(x).$$

This makes  $M$  into a functor, and even a monad with the unit and multiplication maps

$$\begin{array}{ccc} X & \xrightarrow{\delta} & MX \\ x & \longmapsto & \delta_x, \end{array} \quad \begin{array}{ccc} MMX & \xrightarrow{E} & MX \\ \xi & \longmapsto & E\xi, \end{array}$$

where

$$\delta_x(x') = \begin{cases} 1 & x = x', \\ 0 & x \neq x', \end{cases} \quad (E\xi)(x) = \sum_{m \in MX} \xi(m) m(x).$$

Call  $M$  the *measure monad* on  $\mathbf{Set}$ .

Denote also by  $DX \subseteq MX$  the subset of *probability measures*, i.e. those finitely supported  $p : X \rightarrow [0, \infty)$  such that

$$\sum_{x \in X} p(x) = 1.$$

$D$  forms a sub-monad of  $M$  called the *distribution monad*.

It is known that  $M$  is a commutative monad [8]. The corresponding lax monoidal structure

$$MX \times MY \xrightarrow{c} M(X \times Y)$$

is exactly the formation of product measures given by  $c(m, m')(x, y) = m(x)m'(y)$ . Also  $D$  is a commutative monad with the induced lax monoidal structure, since the product of probability measures is again a probability measure.

### 2.1 GS-monoidal and Markov categories

We recall here the basic definitions adopting the graphical formalism of string diagrams, referring to [28] for some background on various notions of monoidal categories and their associated diagrammatic calculus.

► **Definition 2.4.** A **gs-monoidal category** is a symmetric monoidal category  $(\mathcal{C}, \otimes, I)$  with a commutative comonoid structure on each object  $X$  consisting of a comultiplication and a counit

$$\text{copy}_X = \begin{array}{c} \text{---} \\ \cup \\ \bullet \\ | \\ X \end{array} \quad \text{del}_X = \begin{array}{c} \bullet \\ | \\ X \end{array}$$

which satisfy the commutative comonoid equations

These comonoid structures must be multiplicative with respect to the monoidal structure, meaning that it satisfies the equations

## 16:6 Weakly Markov Categories and Weakly Affine Monads

► **Definition 2.5.** A morphism  $f : X \rightarrow Y$  in a gs-monoidal category is called **copyable** or **functional** if

It is called **discardable** or **full** if

► **Example 2.6.** The category **Rel** of sets and relations with the monoidal operation  $\otimes : \mathbf{Rel} \times \mathbf{Rel} \rightarrow \mathbf{Rel}$  given by the direct product of sets is a gs-monoidal category [7]. In this gs-monoidal category, the copyable arrows are precisely the partial functions, and the discardable arrows are the total relations.

► **Remark 2.7.** It is well-known that if every morphism is copyable and discardable, or equivalently if the copy and discard maps are natural, then the monoidal product is the categorical product, and thus the category is cartesian monoidal [9]. In other words, the following conditions are equivalent for a gs-monoidal category  $\mathcal{C}$

- $\mathcal{C}$  is cartesian monoidal;
- every morphism is copyable and discardable;
- the copy and discard maps are natural.

In recent work [11] it has been shown that gs-monoidal categories naturally arise in several ways, such as Kleisli categories of commutative monads or span categories. In the following proposition, we recall the result regarding Kleisli categories.

► **Proposition 2.8.** *Let  $T$  be a commutative monad on a cartesian monoidal category  $\mathcal{D}$ . Then its Kleisli category  $\mathbf{Kl}_T$  is canonically a gs-monoidal category with the copy and discard structure induced by that of  $\mathcal{D}$ .*

► **Example 2.9.** The Kleisli categories of the monads  $M$  and  $D$  of Definition 2.3 are gs-monoidal. We can write their Kleisli categories concretely as follows

- a morphism  $k : X \rightarrow Y$  of  $\mathbf{Kl}_M$  is a *matrix* with rows indexed by  $Y$  and columns indexed by  $X$ , and non-negative entries  $k(y|x)$  such that for each  $x \in X$ , the number  $k(y|x)$  is non-zero only for finitely many  $y$ ;
- a morphism  $k : X \rightarrow Y$  of  $\mathbf{Kl}_D$  is a morphism of  $\mathbf{Kl}_M$  such that moreover, for all  $x \in X$ , the sum of each column satisfies

$$\sum_{y \in Y} k(y|x) = 1$$

If  $X$  and  $Y$  are finite, such a matrix is called a *stochastic matrix*.

In both categories, identities are identity matrices, composition is matrix composition, monoidal structure is the cartesian product on objects and the Kronecker product on matrices, and the copy and discard maps are the images of the standard copy and discard maps on **Set** under the Kleisli inclusion functor.

*Markov categories* [10] represent one of the more interesting specialisations of the notion of gs-monoidal category. Based on the interpretation of their arrows as generalised Markov kernels, they are considered the foundation of a categorical approach to probability theory.

► **Definition 2.10.** A gs-monoidal category is said to be a **Markov category** if any (hence all) of the following equivalent conditions are satisfied

- the monoidal unit is terminal;
- the discard maps are natural;
- every morphism is discardable.

We recall from [23, 19] the notion of *affine monad*.

► **Definition 2.11.** A monad  $T$  on a cartesian monoidal category is called **affine** if  $T1 \cong 1$ .

It was observed in [10, Corollary 3.2] that if the monad preserves the terminal object, then every arrow of the Kleisli category is discardable, and this makes the Kleisli category into a Markov category. Since the converse is easy to see, we have the following addendum to Proposition 2.8.

► **Proposition 2.12.** *Let  $T$  be a commutative monad on a cartesian monoidal category  $\mathcal{D}$ . Then  $\text{Kl}_T$  is Markov if and only if  $T$  is affine.*

► **Example 2.13.** The distribution monad  $D$  of Definition 2.3 is affine, and so its Kleisli category (Example 2.9) is a Markov category. It is one of the simplest examples of categories of relevance for categorical probability.

The measure monad  $M$  is not affine, as it is easy to see that  $M1 \cong [0, \infty)$ , and so its Kleisli category is not Markov.

### 3 Weakly Markov categories and weakly affine monads

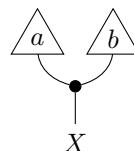
In this section, we introduce an intermediate level between gs-monoidal and Markov called *weakly Markov*, and its corresponding notion for monads, which we call *weakly affine*.

#### 3.1 The monoid of effects

In a gs-monoidal category  $\mathcal{C}$  we call a *state* a morphism from the monoidal unit  $p : I \rightarrow X$ , and *effect* a morphism to the monoidal unit  $a : X \rightarrow I$ . As is standard convention, we represent such morphisms as triangles as follows



Effects, i.e. elements of the set  $\mathcal{C}(X, I)$ , form canonically a commutative monoid as follows: the monoidal unit is the discard map  $X \rightarrow I$ , and given  $a, b : X \rightarrow I$ , their product  $ab$  is given by copying<sup>1</sup>



<sup>1</sup> See also e.g. the  $\odot$  product in [3, Proposition 3.10].

## 16:8 Weakly Markov Categories and Weakly Affine Monads

If a morphism  $f : X \rightarrow Y$  is copyable and discardable, the pre-composition with  $f$  induces a morphism of monoids  $\mathcal{C}(Y, I) \rightarrow \mathcal{C}(X, I)$ .

► **Remark 3.1.** The monoidal unit  $I$  of a monoidal category is canonically a monoid object via the coherence isomorphisms  $I \otimes I \cong I$  and  $I \cong I$ . However, in a general (i.e. not necessarily cartesian) gs-monoidal category  $\mathcal{C}$ , the monoid structure on  $\mathcal{C}(X, I)$  is not, as in Remark 2.2, coming from considering the presheaf represented by  $I$ . Indeed, in order for Remark 2.2 to hold, we would need that every pre-composition is a morphism of monoids. As remarked above, this fails in general unless all morphisms are copyable and discardable (i.e. if  $\mathcal{C}$  is not cartesian monoidal).

Let us now consider the case where the gs-monoidal structure comes from a commutative monad on a cartesian monoidal category  $\mathcal{D}$ . In this case, the monoid structure on Kleisli morphisms  $X \rightarrow 1$  does come from the canonical internal monoid structure on  $T1$  (and from the one on  $1$ ) in  $\mathcal{D}$ . Indeed,  $T1$  is a monoid object with the following unit and multiplication [24, Section 10]

$$1 \xrightarrow{\eta} T1, \quad T1 \times T1 \xrightarrow{c_{1,1}} T(1 \times 1) \xrightarrow{\cong} T1.$$

For example, for the monad of measures  $M$ , we obtain  $M1 = [0, \infty)$  with its usual multiplication. The resulting monoid structure on Kleisli morphisms  $X \rightarrow 1$  is now given as follows. The unit is given by

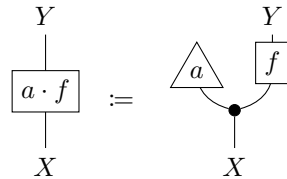
$$X \xrightarrow{\text{del}_X} 1 \xrightarrow{\eta} T1,$$

and the multiplication of Kleisli morphisms  $f, g : X \rightarrow 1$  represented by  $f^\#, g^\# : X \rightarrow T1$  is the Kleisli morphism represented by

$$X \xrightarrow{\text{copy}_X} X \times X \xrightarrow{f^\# \times g^\#} T1 \times T1 \xrightarrow{c_{1,1}} T(1 \times 1) \xrightarrow{\cong} T1.$$

For the monad of measures  $M$ , Kleisli morphisms  $X \rightarrow 1$  are represented by functions  $X \rightarrow [0, \infty)$ , and this description shows that their product is the point-wise product.

For a general  $\mathcal{C}$ , the commutative monoid  $\mathcal{C}(X, I)$  acts on the set  $\mathcal{C}(X, Y)$ : given  $a : X \rightarrow I$  and  $f : X \rightarrow Y$ , the resulting  $a \cdot f$  is given as follows



It is straightforward to see that this indeed amounts to an action of the monoid  $\mathcal{C}(X, I)$  on the set  $\mathcal{C}(X, Y)$ . For the monad of measures  $M$ , this action is given by point-wise rescaling.

Moreover, for a general  $\mathcal{C}$  the operation

$$\begin{aligned} \mathcal{C}(X, Y) \times \mathcal{C}(X, Z) &\longrightarrow \mathcal{C}(X, Y \otimes Z) \\ (f, g) &\longmapsto f \cdot g := (f \otimes g) \circ \text{copy}_X \end{aligned}$$

commutes with this action in each variable (separately).

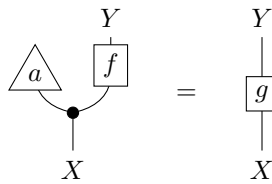
### 3.2 Main definitions

► **Definition 3.2.** A gs-monoidal category  $\mathcal{C}$  is called **weakly Markov** if for every object  $X$ , the monoid  $\mathcal{C}(X, I)$  is a group.

Clearly, every Markov category is weakly Markov: for every object  $X$ , the monoid  $\mathcal{C}(X, I)$  is the trivial group.

► **Definition 3.3.** Given two parallel morphisms  $f, g : X \rightarrow Y$  in a weakly Markov category  $\mathcal{C}$ , we say that  $f$  and  $g$  are called **equivalent**, denoted  $f \sim g$ , if they lie in the same orbit for the action of  $\mathcal{C}(X, I)$ , i.e. if there is  $a \in \mathcal{C}(X, I)$  such that  $a \cdot f = g$ .

Note that if  $a \cdot f = g$  for some  $a$ , then  $a$  is unique. This can be seen by discarding  $Y$  in the following diagram



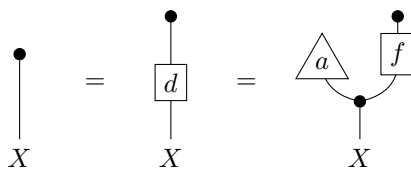
which shows that taking  $a := (\text{del}_Y f)^{-1} \cdot (\text{del}_Y g)$  is the only possibility. In other words, the action of  $\mathcal{C}(X, I)$  on  $\mathcal{C}(X, Y)$  is free, i.e. it has trivial stabilisers.

For the next statement, let us first call the *mass* of a morphism  $f : X \rightarrow Y$  in a gs-monoidal category  $\mathcal{C}$  the morphism  $m_f := \text{del}_Y \circ f : X \rightarrow I$ . Note that  $f$  is discardable if and only if  $m_f = \text{del}_X$ , i.e. if its mass is the unit of the monoid  $\mathcal{C}(X, I)$ .

► **Proposition 3.4.** Every morphism  $f : X \rightarrow Y$  in a weakly Markov category is equivalent to a unique discardable morphism.

We call the discardable morphism the *normalisation* of  $f$  and denote it by  $n_f : X \rightarrow Y$ .

**Proof.** Consider the mass  $m_f$ , and denote its group inverse by  $m_f^{-1}$ . The morphism  $n_f := m_f^{-1} \cdot f$  is discardable and equivalent to  $f$ . Suppose now that  $d : X \rightarrow Y$  is discardable and equivalent to  $f$ , i.e. there exists  $a : X \rightarrow I$  such that  $d = a \cdot f$ . Since  $d$  is discardable



which means that  $a = m_f^{-1}$ , i.e.  $d = n_f$ . ◀

In other words, every morphism  $f$  can be written as its mass times its normalisation. Let us now look at the Kleisli case.

► **Definition 3.5.** A commutative monad  $T$  on a cartesian monoidal category is called **weakly affine** if  $T1$  with its canonical internal commutative monoid structure is a group.

This choice of terminology is motivated by the following proposition, which can be seen as a “weakly” version of Proposition 2.12.

► **Proposition 3.6.** Let  $\mathcal{D}$  be a cartesian monoidal category and  $T$  a commutative monad on  $\mathcal{D}$ . Then the Kleisli category of  $T$  is weakly Markov if and only if  $T$  is weakly affine.

## 16:10 Weakly Markov Categories and Weakly Affine Monads

**Proof.** First, suppose that  $T1$  is an internal group, and denote by  $\iota : T1 \rightarrow T1$  its inversion map. The inverse of a Kleisli morphism  $a : X \rightarrow 1$  in  $\text{Kl}_T(X, 1)$  represented by  $a^\sharp : X \rightarrow T1$  is represented by  $\iota \circ a^\sharp$ : indeed, the following diagram in  $\mathcal{D}$  commutes

$$\begin{array}{ccccccc}
 X & \xrightarrow{\text{copy}_X} & X \times X & & & & \\
 \downarrow a^\sharp & & \downarrow a^\sharp \times a^\sharp & \searrow a^\sharp \times (\iota \circ a^\sharp) & & & \\
 T1 & \xrightarrow{\text{copy}_{T1}} & T1 \times T1 & \xrightarrow{\text{id} \times \iota} & T1 \times T1 & \xrightarrow{c_{1,1}} & T(1 \times 1) \\
 \downarrow \text{del}_{T1} & & & & & & \downarrow \cong \\
 1 & \xrightarrow{\eta} & & & & & T1
 \end{array}$$

where the bottom rectangle commutes since  $\iota$  is the inversion map for  $T1$ . The analogous diagram with  $\iota \times \text{id}$  in place of  $\text{id} \times \iota$  similarly commutes.

Conversely, suppose that for every  $X$ , the monoid structure on  $\text{Kl}_T(X, 1)$  has inverses. Then in particular we can take  $X = T1$ , and the inverse of the Kleisli morphism  $\text{id} : T1 \rightarrow T1$  is an inversion map for  $T1$ .  $\blacktriangleleft$

This result can also be thought of in terms of the Yoneda embedding, via Remark 2.2: since the Yoneda embedding preserves and reflects pullbacks (and all limits), the associativity square for  $T1$  is a pullback in  $\mathcal{D}$  if and only if the associativity squares of all the monoids  $\mathcal{D}(X, T1)$  are pullbacks. Note that Remark 2.2 applies since we are assuming that  $\mathcal{D}$  is *cartesian* monoidal. In the proof of Proposition 3.6, this is reflected by the fact in the main diagram, the morphism  $a^\sharp$  commutes with the copy maps.

### 3.3 Examples of weakly affine monads

Every affine monad is a weakly affine monad. Below you find a few less trivial examples.

► **Example 3.7.** Let  $M^* : \mathbf{Set} \rightarrow \mathbf{Set}$  be the monad assigning to every set the set of finitely supported discrete *non-zero* measures on  $M^*$ , or equivalently let  $M^*(X)$  for any set  $X$  be the set of non-zero finitely supported functions  $X \rightarrow [0, \infty)$ . It is a sub-monad  $M^* \subseteq M$ , meaning that the monad structure is defined in terms of the same formulas as for the monad of measures  $M$  (Definition 2.3). Similarly, the lax structure components

$$c_{X,Y} : M^*X \times M^*Y \longrightarrow M^*(X \times Y)$$

are also given by the formation of product measures, or equivalently point-wise products of functions  $X \rightarrow [0, \infty)$ .

Since  $M^*1 \cong (0, \infty) \not\cong 1$ , this monad is not affine. However the monoid structure of  $(0, \infty)$  induced by  $M^*$  is the usual multiplication of positive real numbers, which form a group. Therefore  $M^*$  is weakly affine, and its Kleisli category is weakly Markov.

On the other hand, if the zero measure is included, we have  $M1 \cong [0, \infty)$  which is not a group under multiplication, so  $M$  is not weakly affine.

► **Example 3.8.** Let  $A$  be a commutative monoid. Then the functor  $T_A := A \times -$  on  $\mathbf{Set}$  has a canonical structure of commutative monad, where the lax structure components  $c_{X,Y}$  are given by multiplying elements in  $A$  while carrying the elements of  $X$  and  $Y$  along.

Since  $T_A 1 \cong A$ , the monad  $T_A$  is weakly affine if and only if  $A$  is a group, and affine if and only if  $A \cong 1$ .

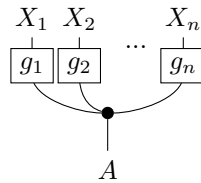


► **Example 3.9.** As for negative examples, consider the free abelian group monad  $F$  on **Set**. Its functor takes a set  $X$  and forms the set  $FX$  of finite multisets (with repetition, where order does not matter) of elements of  $X$  and their formal inverses. We have that  $F1 \cong \mathbb{Z}$ , which is an abelian group under addition. However, the monoid structure on  $F1$  induced by the monoidal structure of the monad corresponds to the *multiplication* on  $\mathbb{Z}$ , which does not have inverses. Therefore  $F$  is not weakly affine.

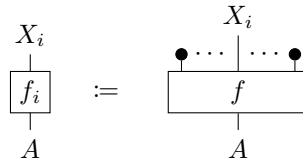
#### 4 Conditional independence in weakly Markov categories

Markov categories have a rich theory of conditional independence in the sense of probability theory [15]. It is noteworthy that some of those ideas can be translated and generalised to the setting of weakly Markov categories.

► **Definition 4.1.** A morphism  $f : A \rightarrow X_1 \otimes \cdots \otimes X_n$  in a gs-monoidal category  $\mathcal{C}$  is said to exhibit **conditional independence of the  $X_i$  given  $A$**  if and only if it can be expressed as a product of the following form



Note that this formulation is a bit different from the earlier definitions given in [2, Definition 6.6] and [10, Definition 12.12], which were formulated for morphisms in Markov categories and state that  $f$  exhibits conditional independence if the above holds with the  $g_i$  being the *marginals* of  $f$ , which are

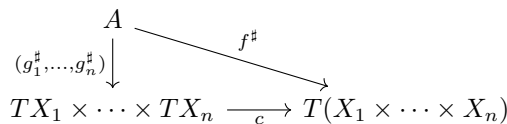


Indeed, in a Markov category, conditional independence in our sense holds if and only if it holds with  $g_i = f_i$  [10, Lemma 12.11]. We also say that  $f$  is the *product of its marginals*.

► **Example 4.2.** In the Kleisli category of the distribution monad  $D$ , which is Markov, a morphism  $f : A \rightarrow X \otimes Y$  exhibits conditional independence if and only if its value at every  $a \in A$  is the product of its marginals [10, Section 12].

Here is what conditional independence looks like in the Kleisli case.

► **Proposition 4.3.** Let  $\mathcal{D}$  be a cartesian monoidal category and  $T$  a commutative monad on  $\mathcal{D}$ . Then a Kleisli morphism represented by  $f^\sharp : A \rightarrow T(X_1 \times \cdots \times X_n)$  exhibits conditional independence of the  $X_i$  given  $A$  if and only if it factors as



for some Kleisli maps  $g_i^\sharp : A \rightarrow TX_i$ , where the map  $c$  above is the one obtained by iterating the lax monoidal structure (which is unique by associativity).

16:12 Weakly Markov Categories and Weakly Affine Monads

**Proof.** In terms of the base category  $\mathcal{D}$ , a Kleisli morphism in the form of Definition 4.1 reads as follows

$$A \xrightarrow{\text{copy}} A \times \cdots \times A \xrightarrow{g_1^\# \times \cdots \times g_n^\#} TX_1 \times \cdots \times TX_n \xrightarrow{c} T(X_1 \times \cdots \times X_n).$$

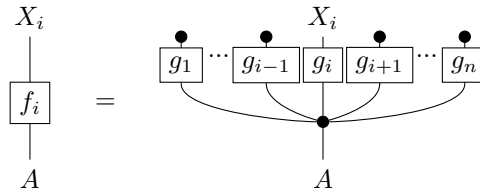
Therefore  $f^\# : A \rightarrow T(X_1 \times \cdots \times X_n)$  exhibits the conditional independence if and only if it is of the form above. ◀

► **Example 4.4.** In the Kleisli category of the measure monad  $M$ , and for any objects, the morphism  $A \rightarrow X_1 \otimes \cdots \otimes X_n$  given by the zero measure on every  $a \in A$  exhibits conditional independence of its outputs given its input. For example, for  $A = 1$ , the zero measure on  $X \times Y$  is the product of the zero measure on  $X$  and the zero (or any other) measure on  $Y$ . Notice that both marginals of the zero measure are zero measures – therefore, the factors appearing in the product are not necessarily related to the marginals.

In a weakly Markov category, the situation is similar to the Markov case discussed above, but up to equivalence: an arrow exhibits conditional independence if and only if it is *equivalent* to the product of its marginals.

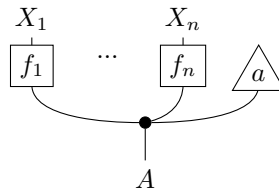
► **Proposition 4.5.** *Let  $f : A \rightarrow X_1 \otimes \cdots \otimes X_n$  be a morphism in a weakly Markov category  $\mathcal{C}$ . Then  $f$  exhibits conditional independence of the  $X_i$  given  $A$  if and only if it is equivalent to the product of all its marginals.*

**Proof.** Denote the marginals of  $f$  by  $f_1, \dots, f_n$ . Suppose that  $f$  is a product as in Definition 4.1. By marginalising, for each  $i = 1, \dots, n$  we get



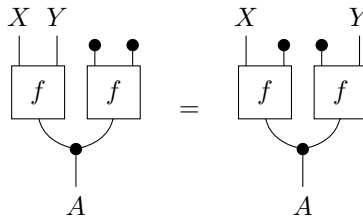
Therefore for each  $i$  we have that  $f_i \sim g_i$ .

Conversely, suppose that  $f$  is equivalent to the product of its marginals, i.e. that there exists  $a : X \rightarrow I$  such that  $f$  is equal to the following



One can then choose  $g_i = f_i$  for all  $i < n$ , and  $g_n = a \cdot f_n$ , so that  $f$  is in the form of Definition 4.1. ◀

► **Remark 4.6.** For  $n = 2$ , a morphism  $f : A \rightarrow X \otimes Y$  in a weakly Markov category  $\mathcal{C}$  exhibits conditional independence of  $X$  and  $Y$  given  $A$  if and only if the equation below holds



Indeed this arises as a consequence of Proposition 4.5 by noting that both sides of the equation describe the same element of  $\mathcal{C}(A, I)$  upon marginalising.

## 4.1 Main result

The concept of conditional independence for weakly Markov categories allows us to give an equivalent characterisation of weakly affine monads. The condition is a pullback condition on the associativity diagram, and it recovers Proposition 2.1 when applied to the monads of the form  $A \times -$  for  $A$  a commutative monoid.

► **Theorem 4.7.** *Let  $\mathcal{D}$  be a cartesian monoidal category and  $T$  a commutative monad on  $\mathcal{D}$ . Then the following conditions are equivalent*

1.  $T$  is weakly affine;
2. the Kleisli category  $\text{Kl}_T$  is weakly Markov;
3. for all objects  $X, Y$ , and  $Z$ , the following associativity diagram is a pullback

$$\begin{array}{ccc}
 T(X) \times T(Y) \times T(Z) & \xrightarrow{\text{id} \times c_{Y,Z}} & T(X) \times T(Y \times Z) \\
 c_{X,Y} \times \text{id} \downarrow & & \downarrow c_{X,Y \times Z} \\
 T(X \times Y) \times T(Z) & \xrightarrow{c_{X \times Y,Z}} & T(X \times Y \times Z)
 \end{array} \tag{2}$$

In order to prove the theorem above, we will exploit the following property of weakly Markov categories.

► **Lemma 4.8** (localised independence property). *Let  $\mathcal{C}$  be a weakly Markov category. Whenever a morphism  $f : A \rightarrow X \otimes Y \otimes Z$  exhibits conditional independence of  $X \otimes Y$  (jointly) and  $Z$  given  $A$ , as well as conditional independence of  $X$  and  $Y \otimes Z$  given  $A$ , then it exhibits conditional independence of  $X, Y$ , and  $Z$  given  $A$ .*

**Proof of Lemma 4.8.** Let us then assume that  $f : A \rightarrow X \otimes Y \otimes Z$  exhibits conditional independence of  $X \otimes Y$  (jointly) and  $Z$  given  $A$ , as well as conditional independence of  $X$  and  $Y \otimes Z$  given  $A$ . By marginalising out  $X$ , we have that  $f_{YZ}$  exhibits conditional independence of  $Y$  and  $Z$  given  $A$ . Since by hypothesis  $f$  exhibits conditional independence of  $X$  and  $Y \otimes Z$  given  $A$ , by Proposition 4.5 it follows that  $f$  is equivalent to the product of  $f_X$  and  $f_{YZ}$ . But, again by Proposition 4.5,  $f_{YZ}$  is equivalent to the product of  $f_Y$  and  $f_Z$ , so it follows that  $f$  is equivalent to the product of all its marginals. Using Proposition 4.5 in the other direction, this means that  $f$  exhibits conditional independence of  $X, Y$ , and  $Z$  given  $A$ . ◀

We are now ready to prove the theorem.

**Proof of Theorem 4.7.** We already know that  $1 \Leftrightarrow 2$ : see Proposition 3.6. We then focus on the correspondence between the first and third item.

$1 \Rightarrow 3$ : By the universal property of products, a cone over the cospan in (2) consists of maps  $g_1^\sharp : A \rightarrow T(X)$ ,  $g_{23}^\sharp : A \rightarrow T(Y \times Z)$ ,  $g_{12}^\sharp : A \rightarrow T(X \times Y)$  and  $g_3^\sharp : A \rightarrow T(Z)$  such that the following diagram commutes

$$\begin{array}{ccc}
 A & \xrightarrow{(g_1^\sharp, g_{23}^\sharp)} & T(X) \times T(Y \times Z) \\
 (g_{12}^\sharp, g_3^\sharp) \searrow & & \downarrow c_{X,Y \times Z} \\
 & T(X) \times T(Y) \times T(Z) & \xrightarrow{\text{id} \times c_{Y,Z}} & T(X) \times T(Y \times Z) \\
 & \downarrow c_{X,Y} \times \text{id} & & \downarrow c_{X,Y \times Z} \\
 & T(X \times Y) \times T(Z) & \xrightarrow{c_{X \times Y,Z}} & T(X \times Y \times Z)
 \end{array}$$

## 16:14 Weakly Markov Categories and Weakly Affine Monads

By Proposition 4.3, this amounts to a Kleisli morphism  $f^\# : A \rightarrow T(X \times Y \times Z)$  exhibiting conditional independence of  $X$  and  $Y \otimes Z$  given  $A$ , as well as of  $X \otimes Y$ , and  $Z$  given  $A$ . By the localised independence property (Lemma 4.8), we then have that  $f$  exhibits conditional independence of all  $X$ ,  $Y$  and  $Z$  given  $A$ , and so, again by Proposition 4.3,  $f^\#$  factors through the product  $TX \times TY \times TZ$ . More specifically, by marginalising over  $Z$ , we have that  $g_{12}^\#$  factors through  $TX \times TY$ , i.e. the following diagram on the left commutes for some  $h_1^\# : A \rightarrow TX$  and  $h_2^\# : A \rightarrow TY$ , and similarly, by marginalising over  $X$ , the diagram on the right commutes for some  $\ell_2^\# : A \rightarrow TY$  and  $\ell_3^\# : A \rightarrow TZ$

$$\begin{array}{ccc} A & \xrightarrow{g_{12}^\#} & \\ (h_1^\#, h_2^\#) \downarrow & & \\ TX \times TY & \xrightarrow{c_{X,Y}} & T(X \times Y) \end{array} \quad \begin{array}{ccc} A & \xrightarrow{g_{23}^\#} & \\ (\ell_2^\#, \ell_3^\#) \downarrow & & \\ TY \times TZ & \xrightarrow{c_{Y,Z}} & T(Y \times Z) \end{array}$$

In other words, we have that the upper and the left curved triangles in the following diagram commute

$$\begin{array}{ccc} A & \xrightarrow{(g_1^\#, g_{23}^\#)} & \\ & \searrow^{(g_1^\#, \ell_2^\#, \ell_3^\#)} & \\ & & T(X) \times T(Y) \times T(Z) \xrightarrow{\text{id} \times c_{Y,Z}} T(X) \times T(Y \times Z) \\ & \searrow_{(h_1^\#, h_2^\#, g_3^\#)} & \downarrow c_{X,Y} \times \text{id} \qquad \downarrow c_{X,Y \times Z} \\ & & T(X \times Y) \times T(Z) \xrightarrow{c_{X \times Y, Z}} T(X \times Y \times Z) \end{array}$$

By marginalising over  $Y$  and  $Z$ , and by weak affinity of  $T$ , there exists a unique  $a^\# : A \rightarrow T1$  such that  $h_1 = a \cdot g_1$ . Therefore

$$g_{12} = h_1 \cdot h_2 = (a \cdot g_1) \cdot h_2 = g_1 \cdot (a \cdot h_2),$$

and so in the diagram above we can equivalently replace  $h_1$  and  $h_2$  with  $g_1$  and  $a \cdot h_2$ . Similarly, by marginalising over  $X$  and  $Y$ , there exists a unique  $c^\# : A \rightarrow T1$  such that  $\ell_3 = c \cdot g_3$ , so that

$$g_{23} = \ell_2 \cdot \ell_3 = \ell_2 \cdot (c \cdot g_3) = (c \cdot \ell_2) \cdot g_3$$

and in the diagram above we can replace  $\ell_2$  and  $\ell_3$  with  $c \cdot \ell_2$  and  $g_3$ , as follows

$$\begin{array}{ccc} A & \xrightarrow{(g_1^\#, g_{23}^\#)} & \\ & \searrow^{(g_1^\#, (c \cdot \ell_2)^\#, g_3^\#)} & \\ & \searrow_{(g_1^\#, (a \cdot h_2)^\#, g_3^\#)} & \\ & & T(X) \times T(Y) \times T(Z) \xrightarrow{\text{id} \times c_{Y,Z}} T(X) \times T(Y \times Z) \\ & & \downarrow c_{X,Y} \times \text{id} \qquad \downarrow c_{X,Y \times Z} \\ & & T(X \times Y) \times T(Z) \xrightarrow{c_{X \times Y, Z}} T(X \times Y \times Z) \end{array}$$

Now, marginalising over  $X$  and  $Z$ , we see that necessarily  $a \cdot h_2 = c \cdot \ell_2$ . Therefore there is a unique map  $A \rightarrow TX \times TY \times TZ$  making the whole diagram commute, which means that (2) is a pullback.

3  $\Rightarrow$  1: If  $T$  is weakly affine, then taking  $X = Y = Z = 1$  in (2) shows that this monoid must be an abelian group: we obtain a unique arrow  $\iota: T1 \rightarrow T1$  making the following diagram commute

$$\begin{array}{ccccc}
 T1 & & & & \\
 \downarrow (id, \iota, id) & \searrow (id, \eta_1 \text{del}_{T1}) & & & \\
 T1 \times T1 \times T1 & \xrightarrow{id \times c_{1,1}} & T1 \times T(1 \times 1) & \xrightarrow{\cong} & T1 \times T1 \\
 \downarrow c_{1,1} \times id & & \downarrow c_{1,1} \times 1 & & \downarrow c_{1,1} \\
 T(1 \times 1) \times T1 & \xrightarrow{c_{1 \times 1, 1}} & T(1 \times 1 \times 1) & \xrightarrow{\cong} & T(1 \times 1) \\
 \downarrow \cong & & \downarrow \cong & & \downarrow \cong \\
 T1 \times T1 & \xrightarrow{c_{1,1}} & T(1 \times 1) & \xrightarrow{\cong} & T1 \\
 \uparrow (\eta_1 \text{del}_{T1}, id) & & & & 
 \end{array}$$

and the commutativity shows that  $\iota$  satisfies the equations making it the inversion map for a group structure.  $\blacktriangleleft$

► **Example 4.9.** In the Kleisli category of the measure monad  $Kl_M$  (which is not weakly affine) consider the following diagram

$$\begin{array}{ccc}
 MX \times MY \times MZ & \xrightarrow{id \times c_{Y,Z}} & MX \times M(Y \times Z) \\
 \downarrow c_{X,Y} \times id & & \downarrow c_{X,Y \times Z} \\
 M(X \times Y) \times MZ & \xrightarrow{c_{X \times Y, Z}} & M(X \times Y \times Z)
 \end{array}$$

In the top-right corner  $MX \times M(Y \times Z)$ , take the pair  $(0, p)$  where  $p$  is any non-zero measure on  $Y \times Z$ , and similarly, in the bottom-left corner take the pair  $(q, 0)$  where  $q$  is any non-zero measure on  $X \times Y$ . Following the diagram, both pairs are mapped to the zero measure in the bottom-right corner. If the diagram was a pullback, we would be able to express the top-right and bottom-left corners as coming from the same triple in  $MX \times MY \times MZ$ , that is, there would exist a measure  $m$  on  $Y$  such that  $m \cdot 0 = p$  and  $0 \cdot m = q$ . Since  $p$  and  $q$  are non-zero, this is not possible.

► **Remark 4.10.** It is worth noting that the pullback condition on the associativity square is not equivalent to the localised independence property of Lemma 4.8: recall that a zero measure always exhibits conditional independence of all its outputs (Example 4.4). Therefore, for zero measures, the localised independence property is always trivially valid, and hence the Kleisli category of the measures monad  $M$  satisfies it in general. However, the example above shows explicitly that the pullback property fails.

For now it is an open question whether the localised independence property for a Kleisli category is reflected by an equivalent condition on the monad.

## 5 Conclusions and future work

Our paper introduces weakly Markov categories and weakly affine monads and explore their relationship. More explicitly, our main result (Theorem 4.7) establishes a tight correspondence between the algebraic properties of  $T1$  and the universal properties of certain commutative squares given by the structural arrows of  $T$  for a commutative monad  $T$  on a cartesian category. We believe that this theorem suggests at least two directions

- generalising the statement to weakly affine monads on weakly Markov categories;
- generalising other Markov-categorical notions, such as the positivity axiom, to weakly Markov or even gs-monoidal categories.

We will provide further details on these potential directions in what follows.

**Regarding possible generalisations.** In Theorem 4.7, we provide a characterisation of weakly affine monads on cartesian monoidal categories. Taking inspiration from the case of affine monads on Markov categories [10, Corollary 3.2], it seems natural to consider whether our main result can be extended to commutative monads on *weakly Markov categories*.

However, this problem is non-trivial and requires clever adjustments to the main definitions. The crucial point is that, in general, the structure of the internal group of  $T1$  and the structure of the group  $\mathcal{D}(X, T1)$  are not necessarily related in the current definitions. One approach could be to introduce a form of *compatibility* for  $T1$  and  $\mathcal{D}(X, T1)$  by defining a weakly affine monad on a weakly Markov category as a commutative monad such that  $T1$  is an internal group and  $\mathcal{D}(X, T1)$  is a group with the composition and units induced by those of  $T1$ . With this change, for example, Proposition 3.6 would work for any weakly Markov category, but Theorem 4.7 would likely fail as its proof involves the universal property of products.

**On the positivity axiom.** A strong monad  $T$  on a cartesian monoidal category is *strongly affine* [20] if for every pair of objects  $X$  and  $Y$  the following diagram is a pullback

$$\begin{array}{ccc} X \times TY & \xrightarrow{s} & T(X \times Y) \\ \downarrow \pi_1 & & \downarrow T\pi_1 \\ X & \xrightarrow{\eta} & TX \end{array}$$

where  $s$  denotes the strength and  $\eta$  denotes the unit of the monad. Every strongly affine monad is affine. The corresponding condition on the Markov category  $\text{Kl}_T$  has recently been characterised as an information flow axiom called *positivity* [12, Section 2].

For a generic commutative monad, the diagram above may even fail to commute (take the measure monad  $M$  and start with  $(x, 0)$  in the top left corner). One can however consider the following diagram, which reduces to the one above (up to isomorphism) in the affine case

$$\begin{array}{ccc} X \times TY & \xrightarrow{s} & T(X \times Y) \\ \downarrow \text{id} \times T(\text{del}_Y) & & \downarrow T(\text{id} \times \text{del}_Y) \\ X \times T1 & \xrightarrow{s} & T(X \times 1) \cong TX \end{array}$$

and which always commutes by naturality of the strength. One can then call the monad  $T$  *positive* if this second diagram is a pullback. Upon defining *positive gs-monoidal categories* analogously to positive Markov categories, one may conjecture that  $T$  is positive if and only if  $\text{Kl}_T$  is positive. This would generalise the existing result for Markov categories.

---

## References

- 1 Nikolai N. Chentsov. The categories of mathematical statistics. *Doklady Akademii Nauk SSSR*, 164(3):511–514, 1965.
- 2 Kenta Cho and Bart Jacobs. Disintegration and Bayesian inversion via string diagrams. *Mathematical Structures in Computer Science*, 29(7):938–971, 2019.
- 3 Bob Coecke, Bill Edwards, and Robert W. Spekkens. Phase groups and the origin of non-locality for qubits. In Bob Coecke, Prakash Panangaden, and Peter Selinger, editors, *QPL@MFPS 2009*, volume 270 of *ENTCS*, pages 15–36. Elsevier, 2009.
- 4 Andrea Corradini and Fabio Gadducci. A 2-categorical presentation of term graph rewriting. In Eugenio Moggi and Giuseppe Rosolini, editors, *CTCS 1997*, volume 1290 of *LNCS*, pages 87–105. Springer, 1997.
- 5 Andrea Corradini and Fabio Gadducci. An algebraic presentation of term graphs, via gs-monoidal categories. *Applied Categorical Structures*, 7(4):299–331, 1999.

- 6 Andrea Corradini and Fabio Gadducci. Rewriting on cyclic structures: equivalence between the operational and the categorical description. *RAIRO - Theoretical Informatics and Applications - Informatique Théorique et Applications*, 33(4-5):467–493, 1999.
- 7 Andrea Corradini and Fabio Gadducci. A functorial semantics for multi-algebras and partial algebras, with applications to syntax. *Theoretical Computer Science*, 286(2):293–322, 2002.
- 8 Dion Coumans and Bart Jacobs. Scalars, monads, and categories. In Chris Heunen, Mehrnoosh Sadrzadeh, and Edward Grefenstette, editors, *Quantum Physics and Linguistics - A Compositional, Diagrammatic Discourse*, pages 184–216. Oxford University Press, 2013.
- 9 Thomas Fox. Coalgebras and cartesian categories. *Communications in Algebra*, 4(7):665–667, 1976.
- 10 Tobias Fritz. A synthetic approach to Markov kernels, conditional independence and theorems on sufficient statistics. *Advances in Mathematics*, 370:107239, 2020.
- 11 Tobias Fritz, Fabio Gadducci, Davide Trotta, and Andrea Corradini. Lax completeness for gs-monoidal categories. *CoRR*, abs/2205.06892, 2022.
- 12 Tobias Fritz, Tomáš Gonda, Nicholas G. Houghton-Larsen, Paolo Perrone, and Dario Stein. Dilations and information flow axioms in categorical probability. *CoRR*, abs/2211.02507, 2022.
- 13 Tobias Fritz, Tomáš Gonda, Paolo Perrone, and Eigil F. Rischel. Representable Markov categories and comparison of statistical experiments in categorical probability. *CoRR*, abs/2010.07416, 2020.
- 14 Tobias Fritz, Tomáš Gonda, and Paolo Perrone. de Finetti’s theorem in categorical probability. *Journal of Stochastic Analysis*, 2(4):6:1–6:26, 2021.
- 15 Tobias Fritz and Andreas Klingler. The  $d$ -separation criterion in categorical probability. *Journal of Machine Learning Research*, 24(46):1–49, 2023.
- 16 Tobias Fritz and Wendong Liang. Free gs-monoidal categories and free Markov categories. *Applied Categorical Structures*, 31(2):21:1–21:31, 2023.
- 17 Tobias Fritz and Eigil F. Rischel. Infinite products and zero-one laws in categorical probability. *Compositionality*, 2(3), 2020.
- 18 Michèle Giry. A categorical approach to probability theory. In Bernhard Banaschewski, editor, *Categorical Aspects of Topology and Analysis*, volume 915 of *LNM*, pages 68–85. Springer, 1982.
- 19 Bart Jacobs. Semantics of weakening and contraction. *Annals of Pure and Applied Logic*, 69(1):73–106, 1994.
- 20 Bart Jacobs. Affine monads and side-effect-freeness. In Ichiro Hasuo, editor, *CMCS 2016*, volume 9608 of *LNCS*, pages 53–72. Springer, 2016.
- 21 Bart Jacobs. Multinomial and hypergeometric distributions in Markov categories. In Ana Sokolova, editor, *MFPS XXXVII*, volume 351 of *ENTCS*, pages 98–115, 2021.
- 22 Claire Jones and Gordon D. Plotkin. A probabilistic powerdomain of evaluations. In *LICS 1989*, pages 186–195. IEEE Computer Society, 1989.
- 23 Anders Kock. Bilinearity and cartesian closed monads. *Mathematica Scandinavica*, 29(2):161–174, 1971.
- 24 Anders Kock. Commutative monads as a theory of distributions. *Theory and Applications of Categories*, 26:97–131, 2012.
- 25 Francis W. Lawvere. The category of probabilistic mappings. Unpublished notes, 1962.
- 26 Sean Moss and Paolo Perrone. A category-theoretic proof of the ergodic decomposition theorem. *Ergodic Theory and Dynamical Systems*, 2023. To appear, available at arXiv:2207.07353.
- 27 Paolo Perrone. Markov categories and entropy. *CoRR*, abs/2212.11719, 2022.
- 28 Peter Selinger. A survey of graphical languages for monoidal categories. In Bob Coecke, editor, *New Structures for Physics*, volume 813 of *LNP*, pages 289–355. Springer, 2011.
- 29 Tadeusz Świrszcz. Monadic functors and convexity. *Bulletin de l’Académie Polonaise des Sciences, Série des Sciences Mathématiques, Astronomiques et Physiques*, XXII(1):39–42, 1974.





# Many-Valued Coalgebraic Logic: From Boolean Algebras to Primal Varieties

Alexander Kurz ✉

Fowler School of Engineering, Chapman University, Orange, CA, USA

Wolfgang Poiger ✉

Department of Mathematics, University of Luxembourg, Esch-sur-Alzette, Luxembourg

---

## Abstract

We study many-valued coalgebraic logics with primal algebras of truth-degrees. We describe a way to lift algebraic semantics of classical coalgebraic logics, given by an endofunctor on the variety of Boolean algebras, to this many-valued setting, and we show that many important properties of the original logic are inherited by its lifting. Then, we deal with the problem of obtaining a concrete axiomatic presentation of the variety of algebras for this lifted logic, given that we know one for the original one. We solve this problem for a class of presentations which behaves well with respect to a lattice structure on the algebra of truth-degrees.

**2012 ACM Subject Classification** Theory of computation → Modal and temporal logics; Theory of computation → Categorical semantics; Theory of computation → Algebraic semantics

**Keywords and phrases** coalgebraic modal logic, many-valued logic, primal algebras, algebraic semantics, presenting functors

**Digital Object Identifier** 10.4230/LIPIcs.CALCO.2023.17

**Funding** *Wolfgang Poiger*: Supported by the Luxembourg National Research Fund under the project PRIDE17/12246620/GPS.

## 1 Introduction

Both many-valued modal logics (see, e.g., [10, 8, 5, 12, 36]) and two-valued coalgebraic logics (see, e.g., [27, 29, 17, 18]) have received increased attention in recent years. Nonetheless, the literature on the combination of these two topics seems, as of yet, sparse (examples include [2, 1, 23]). In this paper, we use methods from universal algebra and category-theory to study algebraic semantics of many-valued coalgebraic logics.

In the classical (two-valued) case, algebraic semantics for coalgebraic logics have been described in [17] as follows. Given an endofunctor  $T$  on the category  $\mathbf{Set}$ , an *abstract coalgebraic logic* for  $T$  consists of an endofunctor  $L$  on the variety  $\mathbf{BA}$  of Boolean algebras together with a natural transformation  $\delta$  determining the semantics (see Definition 1). One can then relate  $T$ -coalgebras and  $L$ -algebras via  $\delta$  and a dual adjunction between  $\mathbf{Set}$  and  $\mathbf{BA}$ . In particular, we call such a coalgebraic logic *concrete* if the functor  $L$  comes equipped

$$T \left( \begin{array}{c} \curvearrowright \\ \mathbf{Set} \\ \curvearrowleft \end{array} \right) \begin{array}{c} \xrightarrow{P} \\ \xleftarrow{S} \end{array} \left( \begin{array}{c} \mathbf{BA} \\ \curvearrowright \\ \curvearrowleft \end{array} \right) L \quad \delta: LP \Rightarrow PT$$

■ **Figure 1** Classical abstract coalgebraic logic for  $T$ .

with a *presentation by operations and equations* in the sense of [4, 20, 22]. Essentially, this corresponds to an axiomatization of the variety  $\mathbf{Alg}(L)$  of  $L$ -algebras. For example, considering classical modal logic, where  $T = \mathcal{P}$  is the covariant powerset functor (that is,  $T$ -coalgebras are Kripke frames), the functor  $L$  has a presentation by one unary operation  $\Box$  with two equations  $\Box(x \wedge y) = \Box x \wedge \Box y$  and  $\Box 1 = 1$  (that is,  $L$ -algebras are modal algebras).



© Alexander Kurz and Wolfgang Poiger;

licensed under Creative Commons License CC-BY 4.0

10th Conference on Algebra and Coalgebra in Computer Science (CALCO 2023).

Editors: Paolo Baldan and Valeria de Paiva; Article No. 17; pp. 17:1–17:17

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

It is well-known that the variety BA is generated by the two-element Boolean algebra  $\mathbf{2}$ , that is  $\text{BA} = \mathbb{HSP}(\mathbf{2})$ . In this paper, we consider the many-valued case where BA is replaced by a variety  $\mathcal{A} = \mathbb{HSP}(\mathbf{D})$ , generated by another finite algebra  $\mathbf{D}$ . More specifically, we study the case where the algebra  $\mathbf{D}$  is *primal*.

An algebra  $\mathbf{D}$  with carrier set  $D$  is primal [11, 31, 7] if every map  $f: D^k \rightarrow D$  is definable by a term  $t_f(x_1, \dots, x_k)$  of  $\mathbf{D}$ . It is well-known that the Boolean algebra  $\mathbf{2}$  is primal, and primal algebras (e.g., the Post-chains, see Example 6) may be seen as many-valued generalizations of this algebra. Indeed, Hu [13] showed that if  $\mathbf{D}$  is primal, then the variety  $\mathcal{A}$  it generates is categorically equivalent to the variety of Boolean algebras BA (and vice versa). Utilizing such a categorical equivalence, we lift an abstract coalgebraic logic  $(L, \delta)$  over BA to an abstract coalgebraic logic  $(L', \delta')$  over  $\mathcal{A}$  (see Figure 4). The logic thus obtained inherits many useful properties of the original one, such as (one-step) completeness and expressivity.

In particular, if  $L$  has a presentation by operations and equations, the same is true for  $L'$ , so at first glance it may seem straightforward to lift concrete coalgebraic logics in a similar manner. However, as we illustrate in this paper, this task turns out to be far from trivial. While the lifting guarantees the *existence* of a presentation of  $L'$ , it offers no indication of what this presentation looks like or how it can be explicitly obtained from a presentation of  $L$ . To answer these questions, we delve deeper into the algebraic structure of  $\mathbf{D}$ . For certain classes of functors  $L$ , we show that there is a systematic way to obtain a presentation of  $L'$  directly from a presentation of  $L$ . In particular, this method applies to classical modal logic as described above.

This work should be seen in the larger context of many-valued coalgebraic logics which have been of interest to the community for a range of potential applications, from AI and cyber-physical systems to the reasoning about software quality. Another (not necessarily coalgebraic) application of many-valued reasoning are semiring-based algorithms for solving soft constraints (see, e.g., [34] for a recent example). From the point of view of some of these applications of many-valued logics, a restriction of our approach is that the dualising algebra of truth-degrees is finite and, correspondingly, the topological duality is zero-dimensional. It remains to be seen in future work whether the techniques we develop to extend Boolean modal logics to many-valued modal logics can be generalized to a continuum of truth-degrees. Our next step, still keeping to the finite case, will be to generalize from primal to semi-primal algebras of truth values (see Question 3).

The paper is structured as follows. In Section 2, we give an overview of coalgebraic logic (Subsection 2.1) and of primal algebras (Subsection 2.2). In Section 3, we show how to lift abstract coalgebraic logics over BA to ones over  $\mathcal{A}$  (see Definition 11), and we show that important properties are preserved under this lifting (see Theorem 12). In Section 4, we present some methods which, under various circumstances, allow us to obtain a presentation of the lifted logic from a presentation of the original one (see Theorems 15 and 18). We also show how these methods can be applied to classical modal logic (see Example 17) and to neighborhood semantics (see Example 19). Lastly, in Section 5, we give a short summary and collect some open questions for further research.

## 2 Preliminaries

In this section, we recall the most important notions used in this paper. In Subsection 2.1, we give a short summary of coalgebraic logics and their algebraic semantics [17]. We distinguish between abstract coalgebraic logics, in which the algebraic semantics correspond to an endofunctor  $L$  on a variety without further specification, and concrete coalgebraic logics,

in which this functor  $L$  is given together with an explicit presentation by operations and equations [4]. We also recall two important properties of coalgebraic logics, namely one-step completeness [29, 17] and expressivity [30, 16, 35, 15].

In Subsection 2.2, we recall the definition of primality [11] and provide some examples of primal algebras which have previously been considered in logic. Note that the unary terms  $T_1$  and  $T_0$  defined in Example 7 reoccur in later sections of this paper. Regarding the variety generated by a primal algebra, we recall Hu's Theorem [13, 14].

## 2.1 Abstract and Concrete Coalgebraic Logics

Coalgebraic (modal) logic, introduced by Moss [27], offers a uniform framework for the logical study of transition systems modeled by coalgebras. In this paper, we follow the approach to coalgebraic logic developed in [17] (for an overview of the various approaches to coalgebraic logic we refer the reader to [18]). It builds on the following dual adjunction between the category  $\mathbf{Set}$  and the variety  $\mathbf{BA}$  of Boolean algebras, defined by two contravariant functors  $P: \mathbf{Set} \rightarrow \mathbf{BA}$  and  $S: \mathbf{BA} \rightarrow \mathbf{Set}$ . Intuitively, the functor  $P$  is the contravariant powerset functor and  $S$  is the functor sending a Boolean algebra to its set of ultrafilters. Formally, we will describe them in a way which is more convenient to generalize to other algebras later on.

The functor  $P: \mathbf{Set} \rightarrow \mathbf{BA}$  assigns the Boolean algebra  $P(X) = \mathbf{2}^X$  to the set  $X$ , where  $\mathbf{2} = (\{0, 1\}, \wedge, \vee, \neg, 0, 1)$  is the two-element Boolean algebra. A map  $f: X \rightarrow X'$  gets sent to  $Pf: \mathbf{2}^{X'} \rightarrow \mathbf{2}^X$  defined by composition  $\beta \mapsto \beta \circ f$ .

The functor  $S$  assigns the set of homomorphisms  $S(\mathbf{B}) = \mathbf{BA}(\mathbf{B}, \mathbf{2})$  to a Boolean algebra  $\mathbf{B} \in \mathbf{BA}$  (note that  $\mathbf{BA}(\mathbf{B}, \mathbf{2})$  can be identified with the set of ultrafilters of  $\mathbf{B}$ ) and sends a homomorphism  $h: \mathbf{B} \rightarrow \mathbf{B}'$  to the map  $Sh: \mathbf{BA}(\mathbf{B}', \mathbf{2}) \rightarrow \mathbf{BA}(\mathbf{B}, \mathbf{2})$ , again defined by composition  $u \mapsto u \circ h$ .

It is well-known that  $P$  and  $S$  form a dual adjunction between the categories  $\mathbf{Set}$  and  $\mathbf{BA}$ . The corresponding natural transformations  $\eta: 1_{\mathbf{BA}} \Rightarrow PS$  and  $\varepsilon: 1_{\mathbf{Set}} \Rightarrow SP$  are given by evaluations, that is, for all  $\mathbf{B} \in \mathbf{BA}$  and  $X \in \mathbf{Set}$  we have

$$\begin{array}{ccc} \eta_{\mathbf{B}}: \mathbf{B} \rightarrow \mathbf{2}^{\mathbf{BA}(\mathbf{B}, \mathbf{2})} & & \varepsilon_X: X \rightarrow \mathbf{BA}(\mathbf{2}^X, \mathbf{2}) \\ b \mapsto \text{ev}_b & & x \mapsto \text{ev}_x \end{array}$$

where  $\text{ev}_b(h) = h(b)$  and  $\text{ev}_x(f) = f(x)$ .

Classical coalgebraic logics are built “on top” of this dual adjunction, relating coalgebras over the base category  $\mathbf{Set}$  to algebras over the base category  $\mathbf{BA}$ . Since we are not only interested in the classical case (that is, we aim to replace  $\mathbf{BA}$  by other varieties later on), we use the following general definition.

► **Definition 1** (Abstract coalgebraic logic). *Let  $\mathcal{V}$  be a variety,  $\Pi: \mathbf{Set} \rightarrow \mathcal{V}$  and  $\Sigma: \mathcal{V} \rightarrow \mathbf{Set}$  be two contravariant functors forming a dual adjunction and let  $T$  be an endofunctor on  $\mathbf{Set}$ . An abstract coalgebraic logic for  $T$  is a pair  $(L, \delta)$ , consisting of an endofunctor  $L$  on  $\mathcal{V}$  and a natural transformation  $\delta: L\Pi \Rightarrow \Pi T$ .*

$$T \left( \begin{array}{c} \curvearrowright \\ \mathbf{Set} \\ \curvearrowleft \end{array} \right) \begin{array}{c} \xrightarrow{\Pi} \\ \xleftarrow{\Sigma} \end{array} \left( \begin{array}{c} \mathcal{V} \\ \curvearrowright \\ \curvearrowleft \end{array} \right) L \quad \delta: L\Pi \Rightarrow \Pi T$$

■ **Figure 2** Abstract coalgebraic logic for  $T$  over a variety  $\mathcal{V}$ .

## 17:4 Many-Valued Coalgebraic Logic: From Boolean Algebras to Primal Varieties

Given a  $T$ -coalgebra  $\gamma: X \rightarrow T(X)$ , applying  $\Pi$  yields  $\Pi\gamma: \Pi T(X) \rightarrow \Pi(X)$ . Composing with  $\delta_X$ , we obtain an  $L$ -algebra  $\Pi\gamma \circ \delta_X: L\Pi(X) \rightarrow \Pi(X)$ . To illustrate these notions, we recall how classical modal logic arises as a special case of a coalgebraic logic (for more details see [17]).

► **Example 2** (Classical modal logic). For a general introduction to classical modal logic, we refer the reader to [3]. The category of Kripke frames with bounded morphisms is isomorphic to the category  $\text{Coalg}(\mathcal{P})$  of coalgebras for the covariant powerset functor  $\mathcal{P}: \text{Set} \rightarrow \text{Set}$ .

The variety of modal algebras, on the other hand, can be identified with the category  $\text{Alg}(L)$  of algebras for an endofunctor  $L: \mathbf{BA} \rightarrow \mathbf{BA}$  defined as follows. If  $\mathbf{B}$  is a Boolean algebra,  $L(\mathbf{B})$  is the free Boolean algebra generated by the set of formal expressions  $\{\Box b \mid b \in B\}$ , quotiented by the equations  $\Box 1 = 1$  and  $\Box(b_1 \wedge b_2) = \Box b_1 \wedge \Box b_2$ .

The corresponding natural transformation  $\delta: L\mathcal{P} \Rightarrow \mathcal{P}\mathcal{P}$  is defined as follows. For a set  $X$ , the component  $\delta_X: L\mathcal{P}(X) \rightarrow \mathcal{P}\mathcal{P}(X)$  is the unique homomorphism which maps a generator  $\Box Y$  (where  $Y \subseteq X$ ) to  $\{Z \subseteq X \mid Z \subseteq Y\}$ . For a Kripke frame  $\gamma: W \rightarrow \mathcal{P}(W)$ , the algebra  $\mathcal{P}\gamma \circ \delta_W$  is known as the *complex algebra* of the frame.

In this example, the category  $\text{Alg}(L)$  is a variety, since the functor  $L$  has a *presentation* by one unary operation  $\Box$  and two equations  $\Box 1 = 1$  and  $\Box(x \wedge y) = \Box x \wedge \Box y$ . For further information about *presentations of functors by operations and equations* we refer the reader to [4, 20, 22].

► **Definition 3** (Concrete coalgebraic logic). A concrete coalgebraic logic is an abstract coalgebraic logic  $(L, \delta)$  together with a presentation of the functor  $L$  by operations and equations.

It is shown in [22, Theorem 4.7] that an endofunctor  $L$  on a variety has a presentation by operations and equations if and only if it preserves sifted colimits.

Two important properties of coalgebraic logics are (one-step) completeness [29, 17] and expressivity [30, 16, 35, 15].

► **Definition 4** (One-step completeness, expressivity). A coalgebraic logic  $(L, \delta)$  is called

- one-step complete if  $\delta$  is a component-wise monomorphism, and
- expressive if the adjoint-transpose  $\delta^\dagger$  of  $\delta$  is a component-wise monomorphism.

Classical modal logic (see Example 2) is one-step complete but not expressive. However, if we replace  $\mathcal{P}$  by the finite powerset functor  $\mathcal{P}_{\text{fin}}$  (i.e., if we only consider *image-finite* Kripke frames), the logic becomes expressive.

## 2.2 Primal Algebras

It is well-known that every function  $f: \{0, 1\}^k \rightarrow \{0, 1\}$  (where  $k \geq 1$ ) is term-definable in the two-element Boolean algebra  $\mathbf{2}$ . In 1953, Foster [11] initiated the general study of algebras with this property, introducing the following notion.

► **Definition 5** (Primal algebra). A finite algebra  $\mathbf{D}$  with carrier set  $D$  is called primal if every function  $f: D^k \rightarrow D$  (where  $k \geq 1$ ) is term-definable in  $\mathbf{D}$ .

Next we give some examples of primal algebras which have a connection to logic, starting with a well-known example of an early many-valued logic.

► **Example 6** (Post chain). The  $(n + 1)$ -element *Post chain* is the algebra

$$\mathbf{P}_n = \left( \left\{ 0, \frac{1}{n}, \dots, \frac{n-1}{n}, 1 \right\}, \wedge, \vee, ', 0, 1 \right),$$

where  $\wedge$  and  $\vee$  are the usual lattice operations and the unary operation  $'$  is defined by  $0' = 1$  and  $\left(\frac{i}{n}\right)' = \left(\frac{i-1}{n}\right)$  for  $0 < i \leq n$ . For every  $n \geq 1$ , the algebra  $\mathbf{P}_n$  is primal [11, Theorem 35].

In our next example, we show that every finite bounded lattice can be turned into a primal algebra in a canonical way. Modal expansions of similar structures have been studied in [25].

► **Example 7.** Let  $(L, \wedge, \vee, 0, 1)$  be a finite bounded lattice. Consider the algebra

$$\mathbf{L} = (L, \wedge, \vee, \{T_\ell\}_{\ell \in L}, \{\hat{\ell}\}_{\ell \in L}),$$

with unary operations

$$T_\ell(x) = \begin{cases} 1 & \text{if } x = \ell \\ 0 & \text{if } x \neq \ell \end{cases}$$

as well as constants  $\hat{\ell}$  for every  $\ell \in L$  (in particular, for the bounds 0 and 1). The algebra  $\mathbf{L}$  is primal. For instance, every unary function  $f: L \rightarrow L$  is definable by the “generalized disjunctive normal form”

$$t_f(x) = \bigvee_{\ell \in L} (T_\ell(x) \wedge \widehat{f(\ell)}).$$

We can proceed similarly with functions  $f: L^k \rightarrow L$  of higher arity, using the terms  $T_{(\ell_1, \dots, \ell_k)}(x_1, \dots, x_k) = T_{\ell_1}(x_1) \wedge \dots \wedge T_{\ell_k}(x_k)$  for every  $(\ell_1, \dots, \ell_k) \in L^k$ .

Other examples of primal algebras in logic which we don’t describe in detail here include the four-valued bilattice studied in [32] and the “Boolean-like” algebras studied in [33].

Not surprisingly, primal algebras have a lot in common with the two-element Boolean algebra. From a category-theoretical perspective, this resemblance is subsumed by *Hu’s Theorem*, which we will state now.

► **Theorem 8** (Hu’s Theorem [13, 14]). *A variety  $\mathcal{A}$  is categorically equivalent to BA if and only if there is a primal algebra  $\mathbf{D} \in \mathcal{A}$  such that  $\mathcal{A} = \mathbb{HSP}(\mathbf{D})$ .*

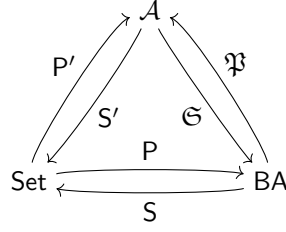
In the following sections we will relate “classical” coalgebraic logics  $(\mathbf{L}, \delta)$ , where  $\mathbf{L}$  is an endofunctor on BA, to “primal” coalgebraic logics  $(\mathbf{L}', \delta')$  where  $\mathbf{L}'$  is an endofunctor on the variety  $\mathcal{A}$  generated by a primal algebra. Even though Theorem 8 implies that BA and  $\mathcal{A}$  are categorically equivalent, we will see that this is a non-trivial task, since presentations of functors are usually not preserved under categorical equivalences.

### 3 Lifting Abstract Coalgebraic Logics

For the remainder of this paper, we adopt the following framework.

► **Assumption 9.** *Let  $\mathbf{D}$  be a primal algebra, based on a bounded lattice  $\mathbf{D}^b = (D, \wedge, \vee, 0, 1)$ .*

We use  $\mathcal{A} = \mathbb{HSP}(\mathbf{D})$  to denote the variety generated by  $\mathbf{D}$ . Note that the assumption that  $\mathbf{D}$  comes equipped with a lattice structure can essentially be made without loss of generality, since every possible lattice-order on  $D$  is term-definable in a primal algebra  $\mathbf{D}$ .



■ **Figure 3** Functors between  $\text{Set}$ ,  $\text{BA}$  and  $\mathcal{A}$ .

To set the scene, we now describe various functors relating our base categories  $\text{Set}$ ,  $\text{BA}$  and  $\mathcal{A}$ . The entire constellation is summarized in Figure 3.

Due to Theorem 8, we know that  $\mathcal{A}$  is categorically equivalent to  $\text{BA}$ . Since  $\mathbf{D}$  is based on a bounded lattice, we have an explicit algebraic description of two functors  $\mathfrak{G}: \mathcal{A} \rightarrow \text{BA}$  and  $\mathfrak{B}: \text{BA} \rightarrow \mathcal{A}$  establishing such an equivalence [21].

The *Boolean skeleton functor*  $\mathfrak{G}: \mathcal{A} \rightarrow \text{BA}$  sends an algebra  $\mathbf{A} \in \mathcal{A}$  to the Boolean algebra

$$\mathfrak{G}(\mathbf{A}) = (\mathfrak{G}(A), \wedge, \vee, T_0, 0, 1)$$

on the carrier set

$$\mathfrak{G}(A) = \{a \in A \mid T_1(a) = a\}.$$

Here,  $\wedge$  and  $\vee$  are the lattice operations of  $\mathbf{A}$ , and  $T_0$  and  $T_1$  are terms defining the unary operations from Example 7 (such terms exist since  $\mathbf{D}$  is primal), interpreted in  $\mathbf{A}$ . It is shown in [26, Lemma 3.11] that  $\mathfrak{G}(\mathbf{A})$  forms a Boolean algebra. To a homomorphism  $g: \mathbf{A} \rightarrow \mathbf{A}'$  the functor  $\mathfrak{G}$  assigns its restriction  $\mathfrak{G}g = g|_{\mathfrak{G}(\mathbf{A})}$ .

The *Boolean power functor*  $\mathfrak{B}: \text{BA} \rightarrow \mathcal{A}$  sends a Boolean algebra  $\mathbf{B}$  to the Boolean power  $\mathbf{D}[\mathbf{B}]$  defined as follows [11, 6]. The carrier set of  $\mathbf{D}[\mathbf{B}]$  is the set of functions  $\xi: D \rightarrow B$  which satisfy  $\xi(d_1) \wedge \xi(d_2) = 0$  for all  $d_1 \neq d_2$  and  $\bigvee \{\xi(d) \mid d \in D\} = 1$  (for the definition of the algebra operations we refer the reader to [6]). To a Boolean homomorphism  $h: \mathbf{B} \rightarrow \mathbf{B}'$  the functor  $\mathfrak{B}$  assigns the homomorphism defined by composition  $\mathfrak{B}h(\xi) = h \circ \xi$ .

A proof of the fact that  $\mathfrak{G}$  and  $\mathfrak{B}$  form a categorical equivalence between  $\text{BA}$  and  $\mathcal{A}$  may be found in [21, Corollary 4.12].

The contravariant functors  $P: \text{Set} \rightarrow \text{BA}$  and  $S: \text{BA} \rightarrow \text{Set}$  were already described in Subsection 2.1, and the contravariant functors  $P': \text{Set} \rightarrow \mathcal{A}$  and  $S': \mathcal{A} \rightarrow \text{BA}$  are defined similarly.

That is, the functor  $P'$  assigns the algebra  $P'(X) = \mathbf{D}^X$  to a set  $X$  and sends a map  $f: X \rightarrow X'$  to the homomorphism  $P'f: \mathbf{D}^{X'} \rightarrow \mathbf{D}^X$  defined by composition  $\alpha \mapsto \alpha \circ f$ .

The functor  $S'$  assigns the set of homomorphisms  $S'(\mathbf{A}) = \mathcal{A}(\mathbf{A}, \mathbf{D})$  to an algebra  $\mathbf{A} \in \mathcal{A}$  and sends a homomorphism  $h: \mathbf{A} \rightarrow \mathbf{A}'$  to the map  $S'h: \mathcal{A}(\mathbf{A}', \mathbf{D}) \rightarrow \mathcal{A}(\mathbf{A}, \mathbf{D})$  defined by composition  $u \mapsto u \circ h$ . Like in the case where  $\mathbf{D} = \mathbf{2}$ , the functors  $P'$  and  $S'$  establish a dual adjunction between  $\text{Set}$  and  $\mathcal{A}$ . The corresponding natural transformations  $\eta': 1_{\mathcal{A}} \Rightarrow P'S'$  and  $\varepsilon': 1_{\text{Set}} \Rightarrow S'P'$  are again given by evaluations (see Subsection 2.1).

We collect some useful properties of the functors appearing in Figure 3 and the natural transformations corresponding to the two dual adjunctions in the following.

► **Proposition 10.** *The functors  $P, S, P', S', \mathfrak{B}, \mathfrak{G}$  and the natural transformations  $\varepsilon, \eta, \varepsilon', \eta'$  satisfy the following properties.*

(a)  $\Phi_{\mathbf{A}}: \mathcal{A}(\mathbf{A}, \mathbf{D}) \rightarrow \text{BA}(\mathfrak{G}(\mathbf{A}), \mathbf{2})$  given by restriction  $u \mapsto u|_{\mathfrak{G}(\mathbf{A})}$  defines a natural isomorphism  $S' \cong S\mathfrak{G}$ . There also exists a natural isomorphism  $S \cong S'\mathfrak{B}$ .

- (b)  $\Psi_X: \mathbf{2}^X \rightarrow \mathfrak{S}(\mathbf{D}^X)$ , which identifies  $\mathbf{2}^X$  with a subset of  $\mathbf{D}^X$  in the obvious way defines a natural isomorphism  $\mathfrak{P} \cong \mathfrak{S}\mathfrak{P}'$ . There also exists a natural isomorphism  $\mathfrak{P}' \cong \mathfrak{P}\mathfrak{P}$ .
- (c)  $\varepsilon = \mathfrak{S}\Psi \circ \Phi\mathfrak{P}' \circ \varepsilon'$  and  $\mathfrak{S}\eta' = \Psi\mathfrak{S}' \circ \mathfrak{P}\Phi \circ \eta\mathfrak{S}$ .

**Proof.** In both (a) and (b), the second statement is an immediate consequence of the first one because  $\mathfrak{P}$  and  $\mathfrak{S}$  form a categorical equivalence. A proof of the first part of (a) can be found in [21, Proposition 4.3].

For the first part of (b), note that  $\Psi_X$  is well-defined since  $\beta \in \mathbf{2}^X$  satisfies  $T_1(\beta(x)) = \beta(x)$  in every component  $x \in X$ . Since the Boolean operations are defined component-wise, it is a homomorphism, and it is clearly injective. It is also surjective, since whenever an element  $\alpha \in \mathbf{D}^X$  has a component with  $\alpha(x) \notin \{0, 1\}$ , we have  $T_1(\alpha(x)) \neq \alpha(x)$ . Naturality is straightforward by definition.

For (c), we need to show that the following diagrams commute for all  $X \in \text{Set}$  and  $\mathbf{A} \in \mathcal{A}$ .

$$\begin{array}{ccc}
 X & \xrightarrow{\varepsilon_X} & \text{BA}(\mathbf{2}^X, \mathbf{2}) \\
 \varepsilon'_X \downarrow & & \uparrow \mathfrak{S}\Psi_X \\
 \mathcal{A}(\mathbf{D}^X, \mathbf{D}) & \xrightarrow{\Phi_{\mathbf{D}^X}} & \text{BA}(\mathfrak{S}(\mathbf{D}^X), \mathbf{2})
 \end{array}
 \qquad
 \begin{array}{ccc}
 \mathfrak{S}(\mathbf{A}) & \xrightarrow{\mathfrak{S}\eta'_\mathbf{A}} & \mathfrak{S}(\mathbf{D}^{\mathcal{A}(\mathbf{A}, \mathbf{D})}) \\
 \eta_{\mathfrak{S}(\mathbf{A})} \downarrow & & \uparrow \Psi_{\mathcal{A}(\mathbf{A}, \mathbf{D})} \\
 \mathbf{2}^{\text{BA}(\mathfrak{S}(\mathbf{A}), \mathbf{2})} & \xrightarrow{\mathfrak{P}\Phi_\mathbf{A}} & \mathbf{2}^{\mathcal{A}(\mathbf{A}, \mathbf{D})}
 \end{array}$$

For the diagram on the left, given  $x \in X$ , we compute

$$\mathfrak{S}\Psi_X \circ \Phi_{\mathbf{D}^X} \circ \varepsilon'_X(x) = \mathfrak{S}\Psi_X \circ \Phi_{\mathbf{D}^X}(\text{ev}_x) = \mathfrak{S}\Psi_X(\text{ev}_x|_{\mathfrak{S}(\mathbf{D}^X)}) = \text{ev}_x|_{\mathfrak{S}(\mathbf{D}^X)} \circ \Psi_X,$$

which, on  $\beta \in \mathbf{2}^X$ , is given by  $\text{ev}_x|_{\mathfrak{S}(\mathbf{D}^X)} \circ \Psi_X(\beta) = \text{ev}_x|_{\mathfrak{S}(\mathbf{D}^X)}(\beta) = \beta(x)$ . Thus, it coincides with  $\varepsilon_X(x)(\beta) = \text{ev}_x(\beta) = \beta(x)$ .

For the diagram on the right, given  $b \in \mathfrak{S}(\mathbf{A})$ , similarly we compute

$$\Psi_{\mathcal{A}(\mathbf{A}, \mathbf{D})} \circ \mathfrak{P}\Phi_\mathbf{A} \circ \eta_{\mathfrak{S}(\mathbf{A})}(b) = \Psi_{\mathcal{A}(\mathbf{A}, \mathbf{D})} \circ \mathfrak{P}\Phi_\mathbf{A}(\text{ev}_b) = \Psi_{\mathcal{A}(\mathbf{A}, \mathbf{D})}(\text{ev}_b \circ \Phi_\mathbf{A}),$$

which is given on  $u \in \mathcal{A}(\mathbf{A}, \mathbf{D})$  by  $\Psi_{\mathcal{A}(\mathbf{A}, \mathbf{D})}(\text{ev}_b \circ \Phi_\mathbf{A})(u) = \Psi_{\mathcal{A}(\mathbf{A}, \mathbf{D})}(\text{ev}_b(u|_{\mathfrak{S}(\mathbf{A})})) = u(b)$ . This coincides with  $\mathfrak{S}\eta'_\mathbf{A}(b)(u) = \eta'_\mathbf{A}|_{\mathfrak{S}(\mathbf{A})}(b)(u) = u(b)$ , finishing the proof.  $\blacktriangleleft$

Suppose we are given an endofunctor  $\mathsf{T}$  on  $\text{Set}$  and an abstract coalgebraic logic  $(\mathsf{L}, \delta)$  for  $\mathsf{T}$  which is classical in the sense that  $\mathsf{L}$  is an endofunctor on  $\text{BA}$ . We now lift this to an abstract coalgebraic logic  $(\mathsf{L}', \delta')$  where  $\mathsf{L}'$  is an endofunctor on  $\mathcal{A}$ . The entire situation is summarized in Figure 4.

► **Definition 11** (Lifting of a coalgebraic logic). *Let  $(\mathsf{L}, \delta)$  be an abstract coalgebraic logic for  $\mathsf{T}: \text{Set} \rightarrow \text{Set}$  with  $\mathsf{L}: \text{BA} \rightarrow \text{BA}$ . Then*

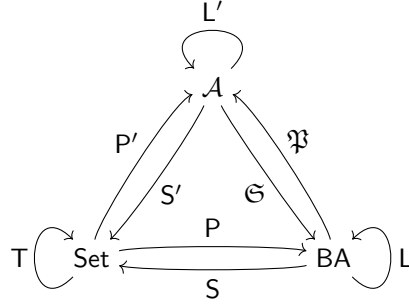
$$\mathsf{L}' = \mathfrak{P}\mathfrak{L}\mathfrak{S} \text{ and } \delta' = \mathfrak{P}\delta$$

*defines an abstract coalgebraic logic  $(\mathsf{L}', \delta')$  for  $\mathsf{T}$ , which we call the lifting of  $(\mathsf{L}, \delta)$  to  $\mathcal{A}$ .*

This is well-defined since, by Proposition 10(b), the natural transformation  $\mathfrak{P}\delta: \mathfrak{P}\mathfrak{L}\mathfrak{P} \rightarrow \mathfrak{P}\mathfrak{P}\mathfrak{T}$  can be identified with one from  $\mathfrak{P}\mathfrak{L}\mathfrak{P} \cong \mathfrak{P}\mathfrak{L}\mathfrak{S}\mathfrak{P}' = \mathsf{L}'\mathfrak{P}'$  to  $\mathfrak{P}\mathfrak{P}\mathfrak{T} \cong \mathfrak{P}'\mathsf{T}$ .

► **Theorem 12.** *Let  $(\mathsf{L}', \delta')$  be the lifting of a coalgebraic logic  $(\mathsf{L}, \delta)$  to  $\mathcal{A}$ .*

- (a) *If  $\mathsf{L}$  has a presentation by operations and equations, then  $\mathsf{L}'$  has one as well.*  
(b) *If  $(\mathsf{L}, \delta)$  is one-step complete, then so is  $(\mathsf{L}', \delta')$ .*  
(c) *If  $(\mathsf{L}, \delta)$  is expressive, then so is  $(\mathsf{L}', \delta')$ .*



■ **Figure 4** Classical coalgebraic logic and its lifting.

**Proof.**

- (a) Recall that an endofunctor on a variety has a presentation if and only if it preserves sifted colimits [22, Theorem 4.7]. Of course, if  $L$  preserves sifted colimits then, by definition, so does  $L'$ .
- (b) If  $\delta$  is a component-wise monomorphism, then so is  $\delta'$ , since  $\mathfrak{P}$  preserves monomorphisms.
- (c) We show that  $(\delta')^\dagger = \delta^\dagger \mathfrak{G}$  holds up to natural isomorphism, from which the statement follows since it implies that if  $\delta^\dagger$  is a component-wise monomorphism, then so is  $(\delta')^\dagger$ . So we want to show that the following diagram commutes.

$$\begin{array}{ccccccc}
 \text{TS}' & \xrightarrow{\varepsilon' \text{TS}'} & \text{S}'\text{P}'\text{TS}' & \xrightarrow{\text{S}'\delta'\text{S}'} & \text{S}'\text{L}'\text{P}'\text{S}' & \xrightarrow{\text{S}'\text{L}'\eta'} & \text{S}'\text{L}' \\
 \downarrow & & \downarrow & & \downarrow & & \downarrow \\
 \text{TS}\mathfrak{G} & \xrightarrow{\varepsilon \text{TS}\mathfrak{G}} & \text{SPT}\mathfrak{S}\mathfrak{G} & \xrightarrow{\text{S}\delta\text{S}\mathfrak{G}} & \text{SLP}\mathfrak{S}\mathfrak{G} & \xrightarrow{\text{SL}\eta\mathfrak{G}} & \text{SL}\mathfrak{G}
 \end{array}$$

Here, by definition, the top edge of the diagram is the adjoint-transpose  $(\delta')^\dagger$  and the bottom edge is  $\delta^\dagger \mathfrak{G}$ . All vertical arrows are natural isomorphisms obtained via  $\Phi$  and  $\Psi$  from Proposition 10. The diagram  $D_2$  commutes by definition of  $\delta'$ , using that  $\text{S}'\delta' = \text{S}'\mathfrak{P}\delta$  and  $\text{S}'\mathfrak{P} \cong \text{S}$  by Proposition 10(a). To finish the proof we show that  $D_1$  and  $D_3$  commute as well.

To see that  $D_1$  commutes, we apply the first equation of Proposition 10(c) to compute

$$\text{SPT}\Phi \circ \text{S}\Psi\text{TS}' \circ \Phi\text{P}'\text{TS}' \circ \varepsilon' \text{TS}' = \text{SPT}\Phi \circ (\text{S}\Psi \circ \Phi\text{P}' \circ \varepsilon')\text{TS}' = \text{SPT}\Phi \circ \varepsilon \text{TS}',$$

which coincides with  $\varepsilon \text{TS}\mathfrak{G} \circ \text{T}\Phi$ .

Similarly, to see that  $D_3$  commutes we apply the second equation of Proposition 10(c) to compute

$$\text{SL}\eta\mathfrak{G} \circ \text{SLP}\Phi \circ \text{SL}\Psi\text{S}' \circ \Phi\text{L}'\text{P}'\text{S}' = \text{SL}(\Psi\text{S}' \circ \text{P}\Phi \circ \eta\mathfrak{G}) \circ \Phi\text{L}'\text{P}'\text{S}' = \text{SL}\mathfrak{G}\eta' \circ \Phi\text{L}'\text{P}'\text{S}',$$

which coincides with  $\Phi\text{L}' \circ \text{S}'\text{L}'\eta'$ .  $\blacktriangleleft$

If  $(L, \delta)$  is a concrete coalgebraic logic for  $\mathbb{T}$  with  $L: \text{BA} \rightarrow \text{BA}$ , then the initial  $L$ -algebra exists and corresponds to the *Lindenbaum-Tarski* algebra of the variety  $\text{Alg}(L)$ . If  $(L, \delta)$  is a coalgebraic logic for  $\mathbb{T}$  and  $\gamma: X \rightarrow \mathbb{T}(X)$  is a coalgebra, then the unique map from the Lindenbaum-Tarski algebra into the  $L$ -algebra  $\text{P}\gamma \circ \delta_X$  determines semantics of formulas. In this context, it is known that one-step completeness of  $(L, \delta)$  implies completeness for



the resulting logic [20, Theorem 6.15]. Since the proof only uses properties of  $\mathbf{BA}$  which are invariant under categorical equivalence, it can easily be adapted to coalgebraic logics over  $\mathcal{A}$ . Thus, parts (a) and (b) of Theorem 12 imply the following.

► **Corollary 13.** *Let  $(\mathbf{L}', \delta')$  be the lifting of the coalgebraic logic  $(\mathbf{L}, \delta)$ , where  $\mathbf{L}$  has a presentation. If  $(\mathbf{L}, \delta)$  is complete, then so is  $(\mathbf{L}', \delta')$ .*

So we showed that the lifting  $(\mathbf{L}', \delta')$  of a coalgebraic logic  $(\mathbf{L}, \delta)$  inherits desirable properties from the original logic, which is satisfactory from a theoretical point of view. From a more “practical” point of view, one important question still needs to be answered, namely that of a concrete presentation of  $\mathbf{L}'$  and its relationship to a presentation of  $\mathbf{L}$ . Indeed, Theorem 12(a) only states that the *existence* of a presentation is preserved, without any explicit way of obtaining it from the original one. In the following section, we give some partial solutions to this problem.

## 4 Lifting Presentations of Functors

We aim to relate presentations of  $\mathbf{L}: \mathbf{BA} \rightarrow \mathbf{BA}$  to presentations of the corresponding lifted functor  $\mathbf{L}' = \mathfrak{P}\mathbf{L}\mathfrak{G}: \mathcal{A} \rightarrow \mathcal{A}$ . Not surprisingly, to do this we need to delve deeper into the algebraic structure of  $\mathbf{D}$ .

Since  $\mathbf{D}$  is based on a bounded lattice and primal (Assumption 9), for every  $d \in D$ , the unary function  $\tau_d: D \rightarrow D$  defined by

$$\tau_d(x) = \begin{cases} 1 & \text{if } d \leq x \\ 0 & \text{if } d \not\leq x \end{cases}$$

is well-defined and term-definable in  $\mathbf{D}$ . Note that  $\tau_0$ , being of constant value 1, carries no relevant information. Thus, we only consider  $\tau_d$  for  $d \in D^+ := D \setminus \{0\}$  in the following. Also note that  $\tau_1$  coincides with  $T_1$  from Example 7. Given an element  $e \in D$ , the map  $\tau_{(\cdot)}(e): D^+ \rightarrow 2$  defined by  $d \mapsto \tau_d(e)$  fully determines the element  $e$  via

$$e = \bigvee \{d \mid \tau_d(e) = 1\}.$$

In the following, we characterize all maps of this form by their lattice-theoretic properties.

► **Lemma 14.** *Let  $\mathcal{T}: D^+ \rightarrow 2$  be a map which, for all  $d_1, d_2 \in D^+$ , satisfies*

$$\mathcal{T}(d_1 \vee d_2) = \mathcal{T}(d_1) \wedge \mathcal{T}(d_2). \quad (1)$$

*Then  $\mathcal{T} = \tau_{(\cdot)}(e)$  for  $e = \bigvee \{d \mid \mathcal{T}(d) = 1\}$ .*

**Proof.** The case  $e = 0$  can only occur if  $\mathcal{T}(d) = 0$  for all  $d \in D^+$ , which implies  $\mathcal{T}(d) = 0 = \tau_d(0)$  for all  $d \in D$ . Now assume that  $e \neq 0$ . First we show that  $\mathcal{T}(e) = 1$ . Since  $e$  is a finite join we apply (1) to find

$$\mathcal{T}(e) = \mathcal{T}(\bigvee \{d \mid \mathcal{T}(d) = 1\}) = \bigwedge \{\mathcal{T}(d) \mid \mathcal{T}(d) = 1\} = 1.$$

Furthermore, since (1) implies that  $\mathcal{T}$  is order-reversing, we have  $\mathcal{T}(c) = 1$  for all  $c \leq e$  as well. Now let  $c \not\leq e$ . Then we have  $\mathcal{T}(c) = 0$ , since otherwise  $\mathcal{T}(c) = 1$  leads to the contradiction

$$e = \bigvee \{d \mid \mathcal{T}(d) = 1\} \geq e \vee c > e.$$

Altogether, we have shown that  $\mathcal{T}(d) = 1$  if and only if  $e \geq d$ , so  $\mathcal{T}(d) = \tau_d(e)$ . ◀

## 17:10 Many-Valued Coalgebraic Logic: From Boolean Algebras to Primal Varieties

Suppose that  $L: \mathbf{BA} \rightarrow \mathbf{BA}$  has a presentation by one unary operation  $\square$  and equations which are satisfied by the terms  $\tau_d$ , in the sense that all the equations obtained by replacing  $\square$  by any  $\tau_d$  hold in  $\mathbf{D}$ . Prominent examples of such equations are  $\square(x \wedge y) = \square x \wedge \square y$  and  $\square 1 = 1$  from Example 2.

Under these circumstances, we can find a presentation of the corresponding lifted functor  $L': \mathcal{A} \rightarrow \mathcal{A}$  as follows. The idea is to “approach” a presentation of  $L'$  by introducing a modal operator for every  $d \in D^+$ , intended to correspond to  $\tau_d \square$  for the “lifted”  $\square'$ . However, only if these modal operators are “consistent” in the sense of Lemma 14, we can replace them by a single operator again.

For simplicity, we only consider the case of one unary operation in the following, but there is a straightforward generalization of Theorem 15 to presentations of  $L$  by one operation which is not necessarily unary (the operations  $\square_d$  and  $\square'$  will simply have the same arity).

► **Theorem 15.** *Let  $L: \mathbf{BA} \rightarrow \mathbf{BA}$  have a presentation by one unary operation  $\square$  and equations which are satisfied (in  $\mathbf{D}$ ) by all  $\tau_d$ ,  $d \in D^+$ . Let  $L' = \mathfrak{P}L\mathfrak{S}$ .*

(a) *The functor  $L'$  can be presented by unary operations  $\square_d$  for every  $d \in D^+$  and the following equations.*

- *The equations for  $\square$ , where  $\square$  is replaced by  $\square_1$ .*
- $\square_1 \tau_d(x) = \square_d x$  for all  $d \in D^+$ .
- $T_1(\square_d x) = \square_d x$  for all  $d \in D^+$ .

(b) *If, in the variety  $\mathbf{Alg}(L')$  axiomatized by the presentation of (a), the equation*

$$\square_{d_1 \vee d_2} x = \square_{d_1} x \wedge \square_{d_2} x \tag{2}$$

*holds, then  $L'$  can also be presented by one unary operation  $\square'$  and the following equations.*

- *The equations for  $\square$ , where  $\square$  is replaced by  $\square'$ .*
- $\square' \tau_d(x) = \tau_d(\square' x)$  for all  $d \in D^+$ .

**Proof.**

(a) Let  $L^+: \mathcal{A} \rightarrow \mathcal{A}$  be the functor presented by the operations  $\square_d$  and equations as in the statement. We want to show that  $L'$  is naturally isomorphic to  $L^+$ . Since both these functors are finitary (because they preserve sifted colimits, in particular they preserve filtered colimits), it suffices to show that their restrictions to finite algebras are naturally isomorphic. The restrictions of  $P$  and  $S$  to the categories  $\mathbf{Set}^{fin}$  of finite sets and  $\mathbf{BA}^{fin}$  of finite Boolean algebras form a dual equivalence. Similarly, the restrictions of  $P'$  and  $S'$  form a dual equivalence between  $\mathbf{Set}^{fin}$  and  $\mathcal{A}^{fin}$ . Therefore, it suffices to show that

$$S'L^+P' \cong SLP,$$

since, due to Proposition 10, for the right-hand side we have further natural isomorphisms  $SLP \cong S'\mathfrak{P}L\mathfrak{S}P' = S'L^+P'$ . Spelling this out, we want to find a bijection between the sets of homomorphisms  $\mathcal{A}(L^+(\mathbf{D}^X), \mathbf{D})$  and  $\mathbf{BA}(L(\mathbf{2}^X), \mathbf{2})$  which is natural in  $X \in \mathbf{Set}$ . By definition of  $L^+$ , the set  $\mathcal{A}(L^+(\mathbf{D}^X), \mathbf{D})$  can be naturally identified with the collection of all maps (whose domain is simply a set of formal expressions)

$$f: \{\square_d a \mid d \in D^+, a \in D^X\} \rightarrow D, \text{ where } f \text{ respects the equations of } L^+.$$

Similarly, the set  $\mathbf{BA}(L(\mathbf{2}^X), \mathbf{2})$  can be naturally identified with the collection of all maps

$$g: \{\square b \mid b \in \mathbf{2}^X\} \rightarrow \mathbf{2}, \text{ where } g \text{ respects the equations of } L.$$

Given  $f$  as above, we assign to it  $g_f$  defined by

$$g_f(\square b) = f(\square_1 b).$$

This is well-defined, since  $T_1(f(\square_1 b)) = f(\square_1 b)$  implies  $f(\square_1 b) \in 2$ , and  $g_f$  respects the equations of  $\mathbf{L}$ , because  $f$  does for  $\square$  replaced by  $\square_1$ .

Conversely, given  $g$  as above, we assign to it  $f_g$  defined by

$$f_g(\square_d a) = g(\square \tau_d(a)).$$

Since the equations of  $\mathbf{L}$  are satisfied by  $\tau_d$  and respected by  $g$ , they are also respected by  $f_g$ . The remaining equations of  $\mathbf{L}^+$  are respected by  $f_g$ , since, for all  $d \in D^+$  we can directly verify

$$f_g(\square_1 \tau_d(a)) = g(\square T_1(\tau_d(a))) = g(\square \tau_d(a)) = f_g(\square_d a),$$

where we used  $T_1(\tau_d(a)) = \tau_d(a)$  since  $\tau_d(a) \in 2^X$  and

$$T_1(f_g(\square_d a)) = T_1(g(\square \tau_d a)) = g(\square \tau_d a) = f_g(\square_d a),$$

where we used  $T_1(g(\square \tau_d a)) = g(\square \tau_d a)$  since  $g(\square \tau_d a) \in 2$ .

Now we show that these two assignments are mutually inverse. For this we compute

$$f_{g_f}(\square_d a) = g_f(\square \tau_d a) = f(\square_1 \tau_d a) = f(\square_d a),$$

where in the last equation we used that  $f$  respects the corresponding equation of  $\mathbf{L}^+$  and

$$g_{f_g}(\square b) = f_g(\square_1 b) = g(\square T_1(b)) = g(\square b),$$

where in the last equation we used  $b \in 2^X$  again.

For naturality, we need to show that, given a map  $m: X_1 \rightarrow X_2$ , the following diagram commutes.

$$\begin{array}{ccc} \mathcal{A}(\mathbf{L}^+(\mathbf{D}^{X_1}), \mathbf{D}) & \xrightarrow{g_{(\cdot)}} & \mathbf{BA}(\mathbf{L}(2^{X_1}), \mathbf{2}) \\ \downarrow S'L^+P'm & & \downarrow \text{SLP}m \\ \mathcal{A}(\mathbf{L}^+(\mathbf{D}^{X_2}), \mathbf{D}) & \xrightarrow{g_{(\cdot)}} & \mathbf{BA}(\mathbf{L}(2^{X_2}), \mathbf{2}) \end{array}$$

Let  $f: \{\square_d a \mid d \in D^+, a \in D^{X_1}\} \rightarrow D$  be given as before. On the one hand, for  $\alpha \in D^{X_2}$  and  $\beta \in 2^{X_2}$  we have  $S'L^+P'm(f)(\square_d \alpha) = f(\square_d(\alpha \circ m))$  and therefore  $g_{S'L^+P'm(f)}(\square \beta) = f(\square_1(\beta \circ m))$ . On the other hand,  $\text{SLP}m(g_f)(\square \beta) = g_f(\square(\beta \circ m)) = f(\square_1(\beta \circ m))$ . Thus, the diagram commutes.

- (b) Let  $\mathbf{L}^*: \mathcal{A} \rightarrow \mathcal{A}$  be defined by one unary operation  $\square'$  and equations as in the statement and let  $\mathbf{L}^+$  be defined as in the proof of (a). For the same reason as before, it suffices to show

$$S'L^*P' \cong S'L^+P'.$$

Again,  $S'L^+P'(X) = \mathcal{A}(\mathbf{L}^+(\mathbf{D}^X), \mathbf{D})$  is essentially the collection of maps

$$f: \{\square_d a \mid d \in D^+, a \in D^X\} \rightarrow D, \text{ where } f \text{ respects the equations of } \mathbf{L}^+,$$

## 17:12 Many-Valued Coalgebraic Logic: From Boolean Algebras to Primal Varieties

and  $S'L^*P'(X)$  is essentially the collection of maps

$$h: \{\Box a \mid a \in D^X\} \rightarrow D, \text{ where } h \text{ respects the equations of } L^*.$$

Given  $h$  as above, we assign to it

$$f_h(\Box_d a) = h(\Box' \tau_d a).$$

Checking that this is well-defined is routine by now, the only non-trivial part being

$$T_1(f_h(\Box_d a)) = T_1(h(\Box' \tau_d(a))) = h(\Box' T_1(\tau_d(a))) = f_h(\Box_d a),$$

which uses the fact that  $h$  respects the corresponding equation  $\Box' T_1(x) = T_1(\Box' x)$  of  $L^*$ . Conversely, given  $f$  as above, we assign to it

$$h_f(\Box' a) = \bigvee \{c \mid f(\Box_c a) = 1\}.$$

First, given  $d \in D^+$ , using that  $\tau_c \circ \tau_d = \tau_d$  holds for all  $c \in D^+$ , we note

$$h_f(\Box' \tau_d(a)) = \bigvee \{c \mid f(\Box_c \tau_d(a)) = 1\} = \bigvee \{c \mid f(\Box_1 \tau_c(\tau_d(a))) = 1\} = \bigvee \{c \mid f(\Box_d a) = 1\}.$$

Since, on the right-hand side, the formula  $f(\Box_d a) = 1$  is independent of  $c$ , this join is either equal to  $\bigvee \emptyset = 0$  if  $f(\Box_d a) = 0$  or  $\bigvee D^+ = 1$  if  $f(\Box_d a) = 1$ . On the other hand, by assumption we can apply Lemma 14, which yields

$$\tau_d(h_f(\Box' a)) = \tau_d(\bigvee \{c \mid f(\Box_c a) = 1\}) = f(\Box_d a)$$

as well. The two assignments thus defined are mutually inverse since

$$f_{h_f}(\Box_d a) = h_f(\Box' \tau_d(a)) = \bigvee \{c \mid f(\Box_c \tau_d(a)) = 1\} = f(\Box_d a)$$

holds again by Lemma 14 and

$$h_{f_h}(\Box' a) = \bigvee \{c \mid h(\Box' \tau_c(a)) = 1\} = \bigvee \{c \mid \tau_c(h(\Box' a)) = 1\} = h(\Box' a).$$

Analogous to (a), it is straightforward to show that the isomorphism thus defined is natural.  $\blacktriangleleft$

In particular, part (b) of this theorem applies if the “original” operation  $\Box$  preserves meets, as shown in the following.

► **Corollary 16.** *Let  $L$  be as in Theorem 15, such that  $\Box(x \wedge y) = \Box x \wedge \Box y$  holds in the variety  $\text{Alg}(L)$ . Then  $L' = \mathfrak{P}L\mathfrak{S}$  can be presented by one unary operation  $\Box'$  and the following equations.*

- *The equations for  $\Box$ , where  $\Box$  is replaced by  $\Box'$ .*
- *$\Box' \tau_d(x) = \tau_d(\Box' x)$  for all  $d \in D^+$ .*

**Proof.** We verify equation (2) from Theorem 15(b) by

$$\Box_{d_1 \vee d_2} x = \Box_1 \tau_{d_1 \vee d_2}(x) = \Box_1(\tau_{d_1}(x) \wedge \tau_{d_2}(x)) = \Box_1 \tau_{d_1}(x) \wedge \Box_1 \tau_{d_2}(x) = \Box_{d_1} x \wedge \Box_{d_2} x,$$

and the statement immediately follows from there.  $\blacktriangleleft$

If  $(\mathbf{L}, \delta)$  is a concrete coalgebraic logic for  $\mathbb{T}$ , where  $\mathbf{L}: \mathbf{BA} \rightarrow \mathbf{BA}$  is endowed with a presentation such that the conditions of Theorem 15 are satisfied, it is now easy to describe the lifting  $(\mathbf{L}', \delta')$  as a concrete coalgebraic logic as well. The only missing piece is an explicit description of the natural transformation  $\delta': \mathbf{L}'\mathbf{P}' \Rightarrow \mathbf{P}'\mathbb{T}$ . Similar to the proof of Theorem 15, for a set  $X$ , the component  $\delta'_X: \mathbf{L}'(\mathbf{D}^X) \rightarrow \mathbf{D}^{\mathbb{T}(X)}$  is defined on  $Y \in \mathbb{T}(X)$  by

$$\delta'_X(\Box_d a)(Y) = \delta_X(\Box \tau_d(a))(Y).$$

Given that the additional condition of part (b) of Theorem 15 is also satisfied, it can be described as

$$\delta'_X(\Box a)(Y) = \bigvee \{d \mid \delta(\Box \tau_d(a)) = 1\}.$$

In the following, we show that the machinery developed works well with respect to the way classical modal logic is described as a concrete coalgebraic logic in Example 2.

► **Example 17** (Lifting classical modal logic). Let  $(\mathbf{L}, \delta)$  be the coalgebraic logic for  $\mathcal{P}$  which corresponds to classical modal logic as in Example 2, in particular  $\mathbf{L}: \mathbf{BA} \rightarrow \mathbf{BA}$  is presented by a unary operation  $\Box$  and the equations  $\Box(x \wedge y) = \Box x \wedge \Box y$  and  $\Box 1 = 1$ .

Let  $(\mathbf{L}', \delta')$  be the lifting of  $(\mathbf{L}, \delta)$  to  $\mathcal{A}$ . By Corollary 16, we know that  $\mathbf{L}'$  has a presentation by a unary operation  $\Box'$  and equations

$$\Box'(x \wedge y) = \Box' x \wedge \Box' y, \quad \Box' 1 = 1 \quad \text{and} \quad \tau_d(\Box' x) = \Box' \tau_d(x) \quad \text{for all } d \in D^+.$$

The natural transformation  $\delta'$  has components  $\delta'_X: \mathbf{L}'(\mathbf{D}^X) \rightarrow \mathbf{D}^{\mathcal{P}(X)}$ , defined by

$$\delta'_X(\Box' a)(Y) = \bigvee \{d \mid \delta_X(\Box \tau_d(a))(Y) = 1\}.$$

Now, since  $\delta_X(\Box \tau_d(a))(Y) = 1 \Leftrightarrow \forall y \in Y : \tau_d(a(y)) = 1 \Leftrightarrow \forall y \in Y : a(y) \geq d$  we can rewrite this as

$$\bigvee \{d \mid \delta_X(\Box \tau_d(a))(Y) = 1\} = \bigvee \{d \mid \bigwedge_{y \in Y} a(y) \geq d\} = \bigvee \{d \mid \tau_d(\bigwedge_{y \in Y} a(y)) = 1\} = \bigwedge_{y \in Y} a(y).$$

Thus, this corresponds to the usual semantics of a many-valued box over Kripke frames defined via meet (see, e.g., [5, 12]). Since we know that  $(\mathbf{L}, \delta)$  is one-step complete (and thus complete), by Theorem 12(b) (and Corollary 13) the logic  $(\mathbf{L}', \delta')$  is one-step complete (and thus complete) as well (similar results are shown in [25, 12]). Furthermore, from Theorem 12(c) we conclude that, replacing  $\mathcal{P}$  by the finite-powerset functor  $\mathcal{P}_{fin}$ , the logic  $(\mathbf{L}', \delta')$  is expressive for image-finite frames (this can also be proved directly along the lines of [24]).

The applicability of Theorem 15 does depend on the specific choice of a presentation of  $\mathbf{L}$ . For instance, the functor  $\mathbf{L}$  in the example above can also be presented by one unary operator  $\Diamond$  with equations  $\Diamond(x \vee y) = \Diamond x \vee \Diamond y$  and  $\Diamond 0 = 0$ . If  $\mathbf{D}$  is not linear, it is easy to check that  $\tau_d(x \vee y) = \tau_d(x) \vee \tau_d(y)$  does not hold in general (simply choose incomparable elements  $x$  and  $y$  and set  $d = x \vee y$ ). Therefore, this presentation can not be lifted by this method. However, the following order-dual version of Theorem 15 can be applied in this case.

For every  $d \in D^- := D \setminus \{1\}$ , the unary operation  $\kappa_d: D \rightarrow D$  defined by

$$\kappa_d(x) = \begin{cases} 1 & \text{if } d \geq x \\ 0 & \text{if } d \not\geq x \end{cases}$$

is term-definable in  $\mathbf{D}$ . Not surprisingly, the following can be shown completely analogous to what we did before.

## 17:14 Many-Valued Coalgebraic Logic: From Boolean Algebras to Primal Varieties

► **Theorem 18.** *Let  $L: \mathbf{BA} \rightarrow \mathbf{BA}$  have a presentation by one unary operation  $\diamond$  and equations which are satisfied by all  $\kappa_d, d \in D^-$ . Let  $L' = \mathfrak{P}L\mathfrak{G}$ .*

(a) *The functor  $L'$  can be presented by unary operations  $\diamond_d$  for every  $d \in D^-$  and the following equations.*

- *The equations for  $\diamond$ , where  $\diamond$  is replaced by  $\diamond_0$ .*
- $\diamond_0 T_0(\kappa_d(x)) = \diamond_d x$  for all  $d \in D^-$ .
- $T_1(\diamond_d x) = \diamond_d x$  for all  $d \in D^-$ .

(b) *If, in the variety  $\mathbf{Alg}(L')$  axiomatized by the presentation of (a), the equation*

$$\diamond_{d_1 \wedge d_2} x = \diamond_{d_1} x \vee \diamond_{d_2} x \tag{3}$$

*holds, then  $L'$  can also be presented by one unary operation  $\diamond'$  and the following equations.*

- *The equations for  $\diamond$ , where  $\diamond$  is replaced by  $\diamond'$ .*
- $\diamond' \kappa_d(x) = \kappa_d(\diamond' x)$  for all  $d \in D^-$ .

Analogous to Corollary 16, equation (3) of Theorem 18 can be deduced if  $\diamond(x \vee y) = \diamond x \vee \diamond y$  holds in  $\mathbf{Alg}(L)$ . Thus, another way to concretely present the lifting  $(L', \delta')$  of classical modal logic (Example 17) is by one unary operation  $\diamond'$  satisfying

$$\diamond'(x \vee y) = \diamond' x \vee \diamond' y, \quad \diamond' 1 = 1 \quad \text{and} \quad \kappa_d(\diamond' x) = \diamond' \kappa_d(x) \quad \text{for all } d \in D^-.$$

The semantics of  $\diamond'$  are (as usual for many-valued diamonds over Kripke frames) defined by joins, that is, for  $a \in \mathbf{D}^X$  and  $Y \in \mathcal{P}(X)$  we have  $\delta'_X(\diamond' a)(Y) = \bigvee_{y \in Y} a(y)$ .

We finish this section with an example to illustrate a situation where part (a) of Theorem 15 can be applied, but part (b) can not.

► **Example 19 (Neighborhood frames).** To deal with non-normal modal logics, one typically considers neighborhood semantics (for an introduction see, e.g., [28]). *Neighborhood frames* are coalgebras for the neighborhood functor  $\mathcal{N}: \mathbf{Set} \rightarrow \mathbf{Set}$ , given by  $\mathcal{N} = \wp \circ \wp$ , where  $\wp$  is the contravariant powerset functor.

Let  $(L, \delta)$  be the following concrete coalgebraic logic over  $\mathcal{N}$ . The functor  $L: \mathbf{BA} \rightarrow \mathbf{BA}$  has a presentation by one unary operation  $\square$  and no (i.e., the empty set of) equations. The natural transformation  $\delta$  has components  $\delta_X: L(\mathbf{2}^X) \rightarrow \mathbf{2}^{\mathcal{N}(X)}$  defined by

$$\delta_X(\square b)(N) = N(b),$$

in other words,  $\delta_X(\square b)(N) = 1$  if and only if the subset  $b \in \mathbf{2}^X$  is an element of the collection of neighborhoods  $N$ .

Since the presentation of  $L$  doesn't include any equations, it trivially satisfies the conditions of Theorem 15. Therefore, the lifting  $(L', \delta')$  of the above logic to  $\mathcal{A}$  can be described as follows. The functor  $L': \mathcal{A} \rightarrow \mathcal{A}$  has a presentation by unary operations  $\square_d$  for all  $d \in D^+$  with equations

$$\square_1 \tau_d(x) = \square_d x \quad \text{and} \quad T_1(\square_d x) = \square_d x \quad \text{for all } d \in D^+.$$

The semantics  $\delta'$  can be described by

$$\delta'_X(\square_d a)(N) = \delta_X(\square \tau_d(a))(N) = N(\tau_d(a)),$$

which means that  $\delta'_X(\square_d a) = 1$  if and only if the subset  $\{x \in X \mid a(x) \geq d\}$  is an element of the collection of neighborhoods  $N$ . Since  $(L, \delta)$  is one-step complete, we again have that  $(L', \delta')$  is complete.

Therefore, it can easily be shown by counter-example that  $\Box_{d_1 \vee d_2} x = \Box_{d_1} x \wedge \Box_{d_2} x$  does not hold in  $\text{Alg}(\mathbf{L}')$ , which means that the above presentation can not be simplified to the one using a single unary operation via Theorem 15(b). At this point, the question whether or not the presentation can be simplified differently remains open.

However, if we replace the functor  $\mathcal{N}$  by the one which only allows collections of neighborhoods which are closed under finite intersections and supersets, we know that there is a corresponding concrete coalgebraic logic  $(\mathbf{L}, \delta)$  such that the presentation of  $\mathbf{L}$  contains the equation  $\Box(x \wedge y) = \Box x \wedge \Box y$ . Thus, Corollary 16 applies in this case.

This concludes the main sections of this paper. In the last section we briefly summarize our results and discuss some potential directions for future research along similar lines.

## 5 Conclusion and Open Questions

We showed how to lift classical coalgebraic logics  $(\mathbf{L}, \delta)$  over  $\mathbf{BA}$  to many-valued coalgebraic logics  $(\mathbf{L}', \delta')$  over  $\mathcal{A}$ , the variety generated by a primal algebra  $\mathbf{D}$ . On the level of abstract coalgebraic logics, it can be shown by purely category-theoretical means that the logic thus lifted inherits important properties like one-step completeness and expressivity from the original logic. On the level of concrete coalgebraic logics, we showed how one may lift a given presentation of  $\mathbf{L}$  by operations and equations to a presentation of  $\mathbf{L}'$ , making use of algebraic properties and a lattice structure of  $\mathbf{D}$ . As of yet, there is no fully general method to do this. However, prominent examples like the modal logics for Kripke frames and neighborhood frames are covered by our results. In the following, we propose some open questions for future research.

As Example 19 illustrates, applying Theorem 15 does not always yield a presentation by a single unary operation. However, such a presentation could still exist in such situations.

► **Question 1.** *Suppose that  $\mathbf{L}: \mathbf{BA} \rightarrow \mathbf{BA}$  has a presentation by a single unary operation and equations. Does there always exist a presentation of  $\mathbf{L}'$  by a single unary operation as well?*

If it is true, a follow-up question would be how these two presentations relate to each other in general. If it is false, a follow-up problem would be to classify the presentations of  $\mathbf{L}$  for which it is true.

The following question arises if we start with a presentation of  $\mathbf{L}$  with more than one, possibly infinitely-many, operations (for example, the multi-modal logic for the distribution functor described in [9]).

► **Question 2.** *Given that the functor  $\mathbf{L}: \mathbf{BA} \rightarrow \mathbf{BA}$  has a presentation by more than one operations operations and equations, can we still obtain a presentation of  $\mathbf{L}'$  with methods similar to the ones developed in this paper?*

Further generalizations of results of this paper may be obtained by weakening Assumption 9 about  $\mathbf{D}$  being primal. We summarize this in the following general question.

► **Question 3.** *Let  $\mathcal{V}$  be some variety generated by some algebra. Is there a canonical way to lift abstract coalgebraic logics  $(\mathbf{L}, \delta)$  over  $\mathbf{BA}$  to abstract coalgebraic logics  $(\mathbf{L}', \delta')$  over  $\mathcal{V}$  and to relate presentations of  $\mathbf{L}$  and  $\mathbf{L}'$ ?*

We plan to generalize the results of this paper to the case of  $\mathbf{D}$  being *semi-primal* in future work (the first step towards that direction has been taken in [21], where we study the category-theoretical relationship between  $\mathcal{V}$  and  $\mathbf{BA}$  in this case).

Lastly, one could keep Assumption 9, but change the approach to coalgebraic logic (for an overview of the various approaches see [18]).

► **Question 4.** *Develop and study the theory of coalgebraic logic with a primal algebra  $\mathbf{D}$  of truth-degrees using other approaches to coalgebraic logic.*

Many-valued nabla modalities and many-valued predicate liftings have, for example, been investigated in [2] and [1, 23]. As follow-up research, one could study the relationship between the various approaches to coalgebraic logic in the many-valued setting (similar to [19]).

---

#### References

- 1 M. Bílková and M. Dostál. Expressivity of many-valued modal logics, coalgebraically. In J. Väänänen, Å. Hirvonen, and R. de Queiroz, editors, *Logic, Language, Information, and Computation*, pages 109–124. Springer Berlin Heidelberg, 2016.
- 2 M. Bílková, A. Kurz, D. Petrişan, and J. Velebil. Relation lifting, with an application to the many-valued cover modality. *Logical Methods in Computer Science*, 9(4):739–790, 2013.
- 3 P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2001.
- 4 M. M. Bonsangue and A. Kurz. Presenting functors by operations and equations. In L. Aceto and A. Ingólfssdóttir, editors, *Foundations of Software Science and Computation Structures*, pages 172–186. Springer Berlin Heidelberg, 2006.
- 5 F. Bou, F. Esteva, L. Godo, and R. O. Rodríguez. On the minimum many-valued modal logic over a finite residuated lattice. *Journal of Logic and Computation*, 21(5):739–790, 2011.
- 6 S. Burris. Boolean powers. *Algebra Universalis*, 5:341–360, 1975.
- 7 S. Burris and H.P. Sankappanavar. *A Course in Universal Algebra*. Graduate Texts in Mathematics. Springer, 1981.
- 8 X. Caicedo and R. Rodríguez. Standard Gödel modal logics. *Studia Logica*, 94:189–214, 2010.
- 9 C. Cirstea and D. Pattinson. Modular construction of modal logics. In P. Gardner and N. Yoshida, editors, *CONCUR 2004 - Concurrency Theory*, pages 258–275. Springer Berlin Heidelberg, 2004.
- 10 M. C. Fitting. Many-valued modal logics. *Fundamenta Informaticae*, 15(3-4):35–254, 1991.
- 11 A. L. Foster. Generalized "Boolean" theory of universal algebras. Part i. *Mathematische Zeitschrift*, 58:306–336, 1953.
- 12 G. Hansoul and B. Teheux. Extending Łukasiewicz logics with a modality: Algebraic approach to relational semantics. *Studia Logica*, 101:505–545, 2013.
- 13 T.-K. Hu. Stone duality for primal algebra theory. *Mathematische Zeitschrift*, 110:180–198, 1969.
- 14 T.-K. Hu. On the topological duality for primal algebra theory. *Algebra Universalis*, 1:152–154, 1971.
- 15 B. Jacobs and A. Sokolova. Exemplaric expressivity of modal logics. *Journal of Logic and Computation*, 20(5):1041–1068, 2009.
- 16 B. Klin. Coalgebraic modal logic beyond sets. *Electronic Notes in Theoretical Computer Science*, 173:177–201, 2007. Proceedings of the 23rd Conference on the Mathematical Foundations of Programming Semantics (MFPS XXIII).
- 17 C. Kupke, A. Kurz, and D. Pattinson. Algebraic semantics for coalgebraic logics. *Electronic Notes in Theoretical Computer Science*, 106:219–241, 2004.
- 18 C. Kupke and D. Pattinson. Coalgebraic semantics of modal logics: An overview. *Theoretical Computer Science*, 412(38):5070–5094, 2011.
- 19 A. Kurz and R. Leal. Modalities in the Stone age: A comparison of coalgebraic logics. *Theoretical Computer Science*, 430:88–116, 2012. Mathematical Foundations of Programming Semantics (MFPS XXV).
- 20 A. Kurz and D. Petrişan. Presenting functors on many-sorted varieties and applications. *Information and Computation*, 208(12):1421–1446, 2010.
- 21 A. Kurz, W. Poiger, and B. Teheux. New perspectives on semi-primal varieties. Preprint available at <https://arxiv.org/abs/2301.13406>, 2023.



- 22 A. Kurz and J. Rosický. Strongly complete logics for coalgebras. *Logical Methods in Computer Science*, 8(3):1–32, 2012.
- 23 C.-Y. Lin and C.-J. Liau. Many-valued coalgebraic modal logic: One-step completeness and finite model property. Preprint available at <https://arxiv.org/abs/2012.05604>, 2022.
- 24 M. Marti and G. Metcalfe. Expressivity in chain-based modal logics. *Archive for Mathematical Logic*, 57:361–380, 2018.
- 25 Y. Maruyama. Algebraic study of lattice-valued logic and lattice-valued modal logic. In R. Ramanujam and S. Sarukkai, editors, *Logic and Its Applications. ICLA*, pages 170–184. Springer Berlin Heidelberg, 2009.
- 26 Y. Maruyama. Natural duality, modality, and coalgebra. *Journal of Pure and Applied Algebra*, 216(3):565–580, 2012.
- 27 L. S. Moss. Coalgebraic logic. *Annals of Pure and Applied Logic*, 96(1):277–317, 1999.
- 28 E. Pacuit. *Neighborhood Semantics for Modal Logic*. Short Textbooks in Logic. Springer, 2017.
- 29 D. Pattinson. Coalgebraic modal logic: Soundness, completeness and decidability of local consequence. *Theoretical Computer Science*, 309(1):177–193, 2003.
- 30 D. Pattinson. Expressive logics for coalgebras via terminal sequence induction. *Notre Dame Journal of Formal Logic*, 45(1):19–33, 2004.
- 31 R. W. Quackenbush. Primality: The influence of Boolean algebras in universal algebra. In *Georg Grätzer. Universal Algebra. Second Edition*, pages 401–416. Springer, New York, 1979.
- 32 U. Rivieccio, A. Jung, and R. Jansana. Four-valued modal logic: Kripke semantics and duality. *Journal of Logic and Computation*, 27(1):155–199, 2017.
- 33 A. Salibra, A. Bucciarelli, A. Ledda, and F. Paoli. Classical logic with  $n$  truth values as a symmetric many-valued logic. *Foundations of Science*, 28:115–142, 2023.
- 34 A. Schiendorfer, A. Knapp, G. Anders, and W. Reif. MiniBrass: Soft constraints for MiniZinc. *Constraints*, 23(4):403–450, 2018.
- 35 L. Schröder. Expressivity of coalgebraic modal logic: The limits and beyond. *Theoretical Computer Science*, 390(2):230–247, 2008. Foundations of Software Science and Computational Structures.
- 36 A. Vidal, F. Esteva, and L. Godo. On modal extensions of product fuzzy logic. *Journal of Logic and Computation*, 27(1):299–336, 2017.



# Composition and Recursion for Causal Structures

Henning Basold   

LIACS, Leiden University, The Netherlands

Tanjona Ralaivaosaona 

LIACS, Leiden University, The Netherlands

---

## Abstract

Causality appears in various contexts as a property where present behaviour can only depend on past events, but not on future events. In this paper, we compare three different notions of causality that capture the idea of causality in the form of restrictions on morphisms between coinductively defined structures, such as final coalgebras and chains, in fairly general categories. We then focus on one presentation and show that it gives rise to a traced symmetric monoidal category of causal morphisms. This shows that causal morphisms are closed under sequential and parallel composition and, crucially, under recursion.

**2012 ACM Subject Classification** Theory of computation → Categorical semantics

**Keywords and phrases** Causal morphisms, Final Coalgebras, Final Chains, Metric Maps, Guarded Recursion, Traced Symmetric Monoidal Category

**Digital Object Identifier** 10.4230/LIPIcs.CALCO.2023.18

## 1 Introduction

Causality appears in various fields of science as the property that the output of a system at given time only depends on past and present inputs. This is particularly well-understood for computations on streams and various approaches to define causal maps on streams have been proposed [7]. More generally, distributive laws have been identified to give rise, and in the category of sets also coincide with, causal maps [14]. Such distributive laws provide a very neat formalism for constructing simultaneously several causal maps but are notoriously difficult to use in compositional specifications [5]. Our aim here is to provide a compositional framework for causal maps, in which such maps can be constructed by sequential composition, parallel composition and recursion. This framework is built around the idea of graphical calculi that arise from traced monoidal categories that allow us to construct and reason about morphisms with string diagrams.

The first question that arises is what causal maps are in general. A robust definition can be given by considering maps on final coalgebras. Suppose that  $F$  is a functor on some category  $\mathbf{C}$  and that it has a final coalgebra with carrier  $\nu F$ , which arises as the limit of a sequence of approximations that we denote by  $\Phi F$ . The final coalgebra  $\nu F$  comes with projections  $p_i: \nu F \rightarrow (\Phi F)_i$  that allow us to inspect an element in  $\nu F$  up to stage  $i$  of the approximation. Intuitively, a map  $f: \nu F \rightarrow \nu F$  is causal if the  $i$ th approximation of its output only depends on the  $i$ th approximation of the input. This notion has been formalised by Rot and Pous [14] and we recap the formal definition in Section 3. For the purpose of this introduction, it suffices to say that one can show that causal maps can equivalently be represented by chain maps  $\Phi F \rightarrow \Phi F$ , which are families of maps for every approximation stage that are consistent across approximation stages. Formally, one considers  $\Phi F$  as a diagram in  $\mathbf{C}$  and a chain map is then a natural transformation.

Thus, there are two equivalent ways of approaching causality. Why would we choose one over the other? Causal maps on final coalgebras have the advantage that they are easy to understand and calculate. However, to attain our goal of compositional reasoning for causal maps, it is better to let go of these for a moment and work with chain maps instead. This



© Henning Basold and Tanjona Ralaivaosaona;  
licensed under Creative Commons License CC-BY 4.0

10th Conference on Algebra and Coalgebra in Computer Science (CALCO 2023).

Editors: Paolo Baldan and Valeria de Paiva; Article No. 18; pp. 18:1–18:17

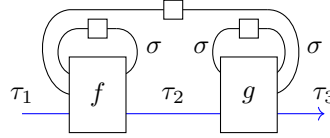
Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

## 18:2 Composition and Recursion for Causal Structures

gives us access to powerful tools for recursion that is akin to that of domain theory [4, 3]. Using these tools and some ideas from monoidal categories, we will be able to draw diagrams such as those in Figure 1.



■ **Figure 1** Circuit with feedback loops and parameters.

The interpretation of Figure 1 is that  $f$  and  $g$  are two causal maps that connected in various ways, including recursive feedback loops. Each of the maps has a small feedback loop and then they are tied together in one big loop. On the loops are small boxes that can be seen as registers that store information in between computation steps. It should be noted that this is an analogy that works well for streams but may fail for other cases. However, we like to place these boxes in the loop because we will show that the feedback is only defined if an initial condition is provided, which can be interpreted as initial values in the registers. Next, there are blue edges with labels  $\tau_k$ . These edges are parameters of the maps that we cannot do recursion with but have more flexible types. This can be useful if we consider causal maps that have additional inputs and outputs that may not even stem from final coalgebras.

The approach to compositional reasoning for causal maps we propose based on the above ideas is that one starts with a set of known causal maps, obtained either directly as chain maps or the construction we provide in the paper. Then one can build arbitrarily complex compositions and loops around these maps using the formalism of traced monoidal and tensored categories. Once construction and reasoning are done, causal maps can be easily obtained from the chain maps by taking limits. All of this works fairly generally, as long as the assumptions in Section 2.2 are fulfilled and that suitable initial conditions for recursion are provided.

### Contributions and Outline

We contribute in Section 4 a framework for working compositionally with chain maps. This framework consists of a construction of string diagrams that differentiate between interfaces for recursion and for parameters. These come about as certain symmetric monoidal, enriched, and tensored categories. For such categories, we show that a trace operator can be obtained relative to the recursion interface of morphisms. To enable the use of this framework, we prove in Section 3 the correspondence between chain maps and causal maps, from which we obtain a very flexible method of composition and recursion for causal maps. We also show in Section 3.1 a third way to define causal maps in terms of a metric that is induced on  $\nu F$  by the diagram  $\Phi F$ . This metric view allows us to understand causality better in certain examples, like streams and partial computations. In Section 5, we discuss applications to probabilistic computations and we pay particular attention to linear maps, which turn out to be automatically causal. Our framework provides then an alternative view on the various calculi for linear circuits. We end with some concluding remarks in Section 6.

Before we begin with the actual work, we recall in the following Section 2 some background on (enriched) monoidal categories and guarded recursion, and we prove some small results to get the theory of the ground.

## 2 Preliminaries

We follow the convention to use boldface letters  $\mathbf{C}$  for categories, capital letters such as  $X$  for objects, lower case letters for morphisms, capital letters such as  $F$  for functors, small Greek letters like  $\mu$  for natural transformations, and  $\alpha, \beta$  for ordinals. We denote by  $\omega$  the ordinal of the natural numbers. Finally,  $\sigma, \tau, \gamma$  will be for  $\alpha^{\text{op}}$ -indexed diagrams in some category.

Recall [11] that a *symmetric monoidal category (SMC)* is a category  $\mathbf{C}$  with tensor product functor  $\otimes: \mathbf{C} \times \mathbf{C} \rightarrow \mathbf{C}$  and a tensor unit  $I \in \mathbf{C}$  with the associativity, unit and symmetry isomorphisms. An SMC is *closed* if for every object  $X \in \mathbf{C}$ , the functor  $\text{Id} \otimes X: \mathbf{C} \rightarrow \mathbf{C}$  has a right-adjoint. In particular, a *Cartesian closed category (CCC)* is a closed SMC with products acting as tensor and exponentials as their right adjoint:  $- \times X \dashv -^X$ . Let  $\mathbf{V}$  be a SMC. A  $\mathbf{V}$ -category  $\mathbf{C}$  is a  $\mathbf{V}$ -enriched category, which means that its morphisms  $\mathbf{C}(X, Y)$  are objects in  $\mathbf{V}$ , and composition and identity are morphisms  $c_{X,Y,Z}: \mathbf{C}(Y, Z) \otimes \mathbf{C}(X, Y) \rightarrow \mathbf{C}(X, Z)$  and  $u_X: I \rightarrow \mathbf{C}(X, X)$  in  $\mathbf{V}$  subject to the corresponding associativity and unit axioms [10, 6]. For morphisms  $f: X \rightarrow Y$  in a Cartesian closed category  $\mathbf{C}$ , we denote by  $\ulcorner f \urcorner: \mathbf{1} \rightarrow Y^X$  the “code” of  $f$  given by the Cartesian closure. The CCC  $\mathbf{C}$  is a  $\mathbf{C}$ -category (self-enriched) by taking  $\ulcorner \text{id} \urcorner: \mathbf{1} \rightarrow X^X$  as unit and the composition  $\text{comp}_{X,Y,Z}: Z^Y \times Y^X \rightarrow Z^X$  is given by the exponential adjunction. A functor  $F: \mathbf{C} \rightarrow \mathbf{C}$  is called *strong* if there is a natural family of morphisms  $F_{X,Y}: Y^X \rightarrow FY^{FX}$ , such that  $F_{X,Y} \circ \ulcorner f \urcorner = \ulcorner Ff \urcorner$  for all  $f: X \rightarrow Y$ . This makes  $F$  a  $\mathbf{C}$ -functor for the self-enrichment of  $\mathbf{C}$ .

Let  $\mathbf{C}$  be a category and  $F: \mathbf{C} \rightarrow \mathbf{C}$  a functor. An *F-coalgebra* (or just coalgebra) is a morphism  $c: X \rightarrow FX$  in  $\mathbf{C}$ . If we need to be explicit about the carrier  $X$ , we also write  $(X, c)$ . A coalgebra homomorphism from  $(X, c)$  to  $(Y, d)$  is a morphism  $f: X \rightarrow Y$  in  $\mathbf{C}$ , satisfying  $Ff \circ c = d \circ f$ . A coalgebra  $(Y, d)$  is *final* if it is final in the category of  $F$ -coalgebras  $\text{CoAlg}(F)$ , i.e., if for every coalgebra  $(X, c)$  there exists a unique coalgebra homomorphism from  $(X, c)$  to  $(Y, d)$ .

Given a category  $\mathbf{C}$ , the *category of descending  $\alpha$ -chains in  $\mathbf{C}$* , here denoted by  $\overleftarrow{\mathbf{C}}$ , is the functor category  $[\alpha^{\text{op}}, \mathbf{C}]$ . Objects of  $\overleftarrow{\mathbf{C}}$  are functors  $\sigma: \alpha^{\text{op}} \rightarrow \mathbf{C}$ , which assign each  $i < \alpha$  an object  $\sigma_i$  of  $\mathbf{C}$  and each pair  $i \leq j$  a morphism  $\sigma(i \leq j): \sigma_j \rightarrow \sigma_i$  in  $\mathbf{C}$ . A morphism  $f: \sigma \rightarrow \tau$  in  $\overleftarrow{\mathbf{C}}$  is a natural transformation, which means that it is an  $\alpha$ -indexed family of morphisms such that  $f_i \circ \sigma(i \leq j) = \tau(i \leq j) \circ f_j$  holds. Such  $f$  will often be called a *chain map* for simplicity. We also record here that the chain category construction gives rise to a 2-functor  $\overleftarrow{(-)}: \mathbf{Cat} \rightarrow \mathbf{Cat}$  on the category of categories. In particular, a functor  $F: \mathbf{C} \rightarrow \mathbf{D}$  gives rise to a functor  $\overleftarrow{F}: \overleftarrow{\mathbf{C}} \rightarrow \overleftarrow{\mathbf{D}}$  by post-composition with diagrams (point-wise application) and similarly for natural transformations. Finally, let us denote by  $K: \mathbf{C} \rightarrow \overleftarrow{\mathbf{C}}$  the constant functor which assigns an object  $X$  of  $\mathbf{C}$  to the constant chain given by  $KX_i = X$  and  $KX(i \leq j) = \text{id}_X$ . If  $\mathbf{C}$  has  $\alpha^{\text{op}}$ -limits, then we assume them to be given as an adjunction  $\langle K \dashv L, \eta, \epsilon \rangle: \mathbf{C} \rightarrow \overleftarrow{\mathbf{C}}$ , where  $L: \overleftarrow{\mathbf{C}} \rightarrow \mathbf{C}$  assigns to a chain its limit.

### 2.1 Domain Theory of Chains

It is well known [1, 8] that if  $F: \mathbf{C} \rightarrow \mathbf{C}$  has a final coalgebra, then there is a limit ordinal  $\alpha$  for which  $F$  is  $\alpha^{\text{op}}$ -continuous (preserves limits of  $\alpha^{\text{op}}$ -diagrams) and the final coalgebra is given by the limit of the so-called final chain. The main tool of this paper is this final chain and we shall therefore recap recursion theory for such chains, see [13, 4, 3].

The category  $\overleftarrow{\mathbf{C}}$  of  $\alpha^{\text{op}}$ -chains has properties that are akin to that of domains used in recursion theory, with the main difference that fixed point theorems require guardedness via the so-called later modality. We assume in what follows that  $\mathbf{C}$  is Cartesian closed, which implies that  $\overleftarrow{\mathbf{C}}$  is also a CCC, and that  $\mathbf{C}$  has sufficiently many limits, cf. Section 2.2.

The *later modality* is a functor  $\blacktriangleright: \overleftarrow{\mathbf{C}} \rightarrow \overleftarrow{\mathbf{C}}$  defined on objects by  $(\blacktriangleright \sigma)_i = \lim_{j < i} \sigma_j$  and it comes with a natural transformation  $\text{next}: \text{Id} \rightarrow \blacktriangleright$ . Since products preserve limits, there are natural isomorphisms  $\delta_{\sigma, \tau}^{\blacktriangleright}: \blacktriangleright \sigma \times \blacktriangleright \tau \rightarrow \blacktriangleright(\sigma \times \tau)$  and  $\varepsilon^{\blacktriangleright}: \mathbf{1} \rightarrow \blacktriangleright \mathbf{1}$ . If  $\omega$  is used as indexing ordinal, one can easily show that  $(\blacktriangleright \sigma)_0 \cong \mathbf{1}$  and  $(\blacktriangleright \sigma)_{n+1} \cong \sigma_n$  via a chain map.

We are interested in the category  $\overleftarrow{\mathbf{C}}$  here because it allows us to do so-called guarded recursion, which comes in the form of fixed point solution theorems for morphism and for functors analogue to those occurring in domain theory. However, what differentiates guarded recursion from domain theory is that we only find fixed points of contractive morphisms. A *solution* or fixed point of a morphism  $h: \tau \times \gamma \rightarrow \gamma$  in  $\overleftarrow{\mathbf{C}}$  is a morphism  $s: \tau \rightarrow \gamma$  with  $s = h \circ \langle \text{id}_\tau, s \rangle$ . We call a morphism  $h: \tau \times \gamma \rightarrow \gamma$  *contractive* if there is  $g: \tau \times \blacktriangleright \gamma \rightarrow \gamma$  with  $h = g \circ (\text{id}_\tau \times \text{next}_\gamma)$ . The main point is now that any contractive morphism  $h$  has a solution in  $\overleftarrow{\mathbf{C}}$ .

The isomorphisms  $\delta^{\blacktriangleright}$  and  $\varepsilon^{\blacktriangleright}$  make  $\blacktriangleright$  a (strong) monoidal functor and thus allow us to change the enriching base and obtain a  $\overleftarrow{\mathbf{C}}$ -category  $\overleftarrow{\mathbf{C}}_{\blacktriangleright}$  with the same objects as  $\overleftarrow{\mathbf{C}}$  but  $\overleftarrow{\mathbf{C}}_{\blacktriangleright}(\sigma, \tau) = \blacktriangleright(\tau^\sigma)$  as morphism object. The monoidal natural transformation  $\text{next}$  induces a  $\overleftarrow{\mathbf{C}}$ -functor  $\text{Next}: \overleftarrow{\mathbf{C}} \rightarrow \overleftarrow{\mathbf{C}}_{\blacktriangleright}$  by putting  $N_{\sigma, \tau} = \text{next}_{\tau^\sigma}: \tau^\sigma \rightarrow \blacktriangleright(\tau^\sigma)$ . A  $\overleftarrow{\mathbf{C}}$ -functor  $F: \overleftarrow{\mathbf{C}} \rightarrow \overleftarrow{\mathbf{C}}$  is called *locally contractive* if there is a  $\overleftarrow{\mathbf{C}}$ -functor  $G: \overleftarrow{\mathbf{C}}_{\blacktriangleright} \rightarrow \overleftarrow{\mathbf{C}}$  with  $G \circ \text{Next} = F$ . Explicitly, there is a family of morphisms  $G_{\sigma, \tau}: \blacktriangleright(\tau^\sigma) \rightarrow F\sigma^{F\tau}$  with  $F_{\sigma, \tau} = G_{\sigma, \tau} \circ \text{next}_{\sigma^\tau}$ ,  $G_{\sigma, \sigma} \circ \blacktriangleright \lceil \text{id} \rceil \circ \varepsilon^{\blacktriangleright} = \lceil \text{id} \rceil$  and  $\text{comp} \circ (C_{\sigma, \tau} \times C_{\gamma, \sigma}) = C_{\gamma, \tau} \circ \blacktriangleright \text{comp} \circ \delta^{\blacktriangleright}$ .

Throughout this paper, we will use that  $\blacktriangleright$  is locally contractive, and that if  $F$  and  $G$  are  $\overleftarrow{\mathbf{C}}$ -functors and at least one of them is locally contractive, then  $F \circ G$  is locally contractive. Moreover, we will need the following result.

**► Lemma 1.** *Given a functor  $F: \mathbf{C} \rightarrow \mathbf{C}$ , the functor  $\overleftarrow{F}: \overleftarrow{\mathbf{C}} \rightarrow \overleftarrow{\mathbf{C}}$  is a  $\overleftarrow{\mathbf{C}}$ -functor if and only if  $F$  is a  $\mathbf{C}$ -functor.*

What makes locally contractive functor interesting, is that they admit unique fixed points: Given a locally contractive functor  $F: \overleftarrow{\mathbf{C}} \rightarrow \overleftarrow{\mathbf{C}}$ , there is a unique chain  $\nu F$  with isomorphisms  $\text{obs}: \nu F \rightarrow F(\nu F)$  and  $\text{fold} = \text{obs}^{-1}: F(\nu F) \rightarrow \nu F$ . In this paper, we pick coinduction as our main principle and consider  $(\nu F, \text{obs})$  as final object in  $\text{CoAlg}(F)$ .

**► Lemma 2.** *There is a functor  $\Phi: \text{Endo}(\mathbf{C}) \rightarrow \overleftarrow{\mathbf{C}}$  given on objects by  $\Phi F = \nu(\blacktriangleright \overleftarrow{F})$ , which exists because  $\blacktriangleright \circ \overleftarrow{F}$  is locally contractive. We call  $\Phi F$  the final chain of  $F$ .*

**Proof.** Given a natural transformation  $\alpha: F \rightarrow G$ , we define  $\Phi \alpha$  coinductively as in the following diagram.

$$\begin{array}{ccc}
 \Phi F & \xrightarrow{\Phi \alpha} & \Phi G \\
 \downarrow \text{obs} & & \downarrow \text{obs} \\
 \blacktriangleright \overleftarrow{F}(\Phi F) & & \blacktriangleright \overleftarrow{G}(\Phi G) \\
 \downarrow \blacktriangleright \alpha_{\Phi F} & & \\
 \blacktriangleright \overleftarrow{G}(\Phi F) & \xrightarrow{\blacktriangleright \overleftarrow{G}(\Phi \alpha)} & \blacktriangleright \overleftarrow{G}(\Phi G)
 \end{array}$$

Preservation of identities and composition follow by standard arguments from finality. ◀

If  $F$  preserves  $\alpha^{\text{op}}$ -limits, that is, if  $L\overleftarrow{F} \cong FL$ , then the limit adjunction  $K \dashv L$  lifts to an adjunction  $\overleftarrow{K} \dashv \overleftarrow{L}$  with  $\overleftarrow{K}: \text{CoAlg}(F) \rightarrow \text{CoAlg}(\blacktriangleright \overleftarrow{F})$ , see [3]. In particular,  $\overleftarrow{L}(\Phi F, \text{obs})$  is a final  $F$ -coalgebra with carrier  $L(\Phi F)$ .

## 2.2 Assumptions

Given the above, we assume the following for the remainder of the paper:  $\mathbf{C}$  is a Cartesian closed category;  $\alpha$  is a limit ordinal;  $\mathbf{C}$  has  $\alpha^{\text{op}}$ -limits and  $\partial(\alpha \downarrow i)^{\text{op}}$ -limits, where  $\partial(\alpha \downarrow i)$  is the category that contains all  $j < i$ ;  $F$  is a strong functor on  $\mathbf{C}$  that preserves  $\alpha^{\text{op}}$ -limits.

## 3 Causality

In this section, we extend the definition of  $\omega$ -causal operators [14, Def. 8.1] to arbitrary categories but we do not define causal algebra. Although, our definition can be easily extended to causal algebras. For this purpose, we assume that  $F$  preserves  $\alpha^{\text{op}}$ -limits and thus  $L\Phi F$  can be taken as the carrier  $\nu F$  of a final  $F$ -coalgebra. We denote by  $(\nu F, (p_i)_{i < \alpha})$ , the universal cone defining a limit for  $\Phi F$  and we define causal morphisms on  $\nu F$  as follows.

► **Definition 3.** A morphism  $f: \nu F \rightarrow \nu F$  is causal if for every object  $X$  of  $\mathbf{C}$ , morphisms  $e_1, e_2: X \rightarrow \nu F$  and  $i < \alpha$ : if  $p_i \circ e_1 = p_i \circ e_2$ , then  $p_i \circ f \circ e_1 = p_i \circ f \circ e_2$ . Diagrammatically:

$$\begin{array}{ccc}
 & \nu F & \\
 e_1 \nearrow & & \searrow p_i \\
 X & & (\Phi F)_i \\
 e_2 \searrow & & \nearrow p_i \\
 & \nu F &
 \end{array}
 \implies
 \begin{array}{ccc}
 & \nu F \xrightarrow{f} \nu F & \\
 e_1 \nearrow & & \searrow p_i \\
 X & & (\Phi F)_i \\
 e_2 \searrow & & \nearrow p_i \\
 & \nu F \xrightarrow{f} \nu F &
 \end{array}$$

We denote the set of causal morphisms on  $\nu F$  by  $\text{Caus}(\nu F, \nu F) \subseteq \mathbf{C}(\nu F, \nu F)$ .

In the following theorem we compare two characterisations of causal morphisms on  $\nu F$ .

► **Theorem 4.** There is a map  $\lambda: \overleftarrow{\mathbf{C}}(\Phi F, \Phi F) \rightarrow \text{Caus}(\nu F, \nu F)$  with  $\lambda(g) = Lg$ . If there is a section  $s: \Phi F \rightarrow KL\Phi F$  of  $\epsilon_{\Phi F}$  in  $\overleftarrow{\mathbf{C}}$ , i.e.  $\epsilon_{\Phi F} \circ s = \text{id}_{\Phi F}$ , then  $\lambda$  is an isomorphism.

**Proof.** We define  $\lambda: \overleftarrow{\mathbf{C}}(\Phi F, \Phi F) \rightarrow \text{Caus}(\nu F, \nu F)$  such that for each  $g: \Phi F \rightarrow \Phi F$ ,  $\lambda(g) = Lg$ . To show that  $\lambda(g)$  is causal we need to prove, by Definition 3, that if diagram (1) below commutes, then the outer diagram must also commute, for any  $\rho \in \overleftarrow{\mathbf{C}}$  and morphisms  $e_1, e_2: \rho \rightarrow \nu F$ . In the diagram, we use  $L\Phi F$  for  $\nu F$ .

$$\begin{array}{ccccc}
 & L\Phi F & \xrightarrow{Lg} & L\Phi F & \\
 e_2 \nearrow & & \searrow p_i & & \searrow p_i \\
 \rho & & & (\Phi F)_i & \xrightarrow{g_i} & (\Phi F)_i \\
 e_1 \searrow & & \nearrow p_i & & \nearrow p_i \\
 & L\Phi F & \xrightarrow{Lg} & L\Phi F &
 \end{array}
 \quad (1)$$

To prove that the outer diagram commutes, it is enough to prove that diagram (2) commutes. Because of naturality of the counit  $\epsilon$  of the adjunction  $\langle K \dashv L, \eta, \epsilon \rangle$ , the diagram below commutes.

$$\begin{array}{ccc}
 KL\Phi F & \xrightarrow{KLg} & KL\Phi F \\
 \epsilon_{\Phi F} \downarrow & & \downarrow \epsilon_{\Phi F} \\
 \Phi F & \xrightarrow{g} & \Phi F
 \end{array}$$

## 18:6 Composition and Recursion for Causal Structures

Hence diagram (2) commutes, as being the  $i^{\text{th}}$  component of the above commuting diagram. Therefore,  $\lambda(g)$  is causal.

Given the section  $s: \Phi F \rightarrow KL\Phi F$ , we define an inverse  $\chi: \text{Caus}(\nu F, \nu F) \rightarrow \overleftarrow{\mathbf{C}}(\Phi F, \Phi F)$  of  $\lambda$  on causal maps  $f: \nu F \rightarrow \nu F$  by letting  $\chi(f) = \Phi F \xrightarrow{s} KL\Phi F \xrightarrow{Kf} KL\Phi F \xrightarrow{\epsilon_{\Phi F}} \Phi F$ .  $\chi(g)$  is a chain map in  $\overleftarrow{\mathbf{C}}$  because it is a composition of chain maps in  $\overleftarrow{\mathbf{C}}$ . We have,  $(\chi \circ \lambda)(g) = g$ , since the following diagram commutes by naturality of  $\epsilon$  and  $s$  being a section.

$$\begin{array}{ccccc} \Phi F & \xrightarrow{s} & KL\Phi F & \xrightarrow{KLg} & KL\Phi F \\ & \searrow \text{id}_{\Phi F} & \downarrow \epsilon_{\Phi F} & & \downarrow \epsilon_{\Phi F} \\ & & \Phi F & \xrightarrow{g} & \Phi F \end{array}$$

We also have  $(\lambda \circ \chi)(f) = f$ : The following diagram commutes because of causality of  $f$ , naturality of  $\eta$ , and the triangular axiom of adjunction.

$$\begin{array}{ccccc} LKL\Phi F & \xrightarrow{LKf} & LKL\Phi F & & \\ \eta_{L\Phi F} \uparrow & & \eta_{L\Phi F} \uparrow & \searrow L\epsilon_{\Phi F} & \\ L\Phi F & \xrightarrow{f} & L\Phi F & \xrightarrow{\text{id}_{L\Phi F}} & L\Phi F \\ Ls \downarrow & & & \nearrow L\epsilon_{\Phi F} & \\ LKL\Phi F & \xrightarrow{LKf} & LKL\Phi F & & \end{array}$$

Thus  $\lambda$  is an isomorphism with inverse  $\chi$ . ◀

Importantly, this characterisation allows us to exploit all the domain-theoretic tools that are available in  $\overleftarrow{\mathbf{C}}$  to compose and reason about causal morphisms.

Let us pause for a moment to take a look at some examples in the category **Set**. First of all, we note that we generally get the required section in Theorem 4 because the limit projections split if the involved chains are non-empty. Thus, chain and causal maps are equivalent in **Set**. Let us explore more concretely the familiar examples of streams and partial computations.

► **Example 5.** Let  $S: \mathbf{Set} \rightarrow \mathbf{Set}$  be the functor defined by  $S(X) = R \times X$ , for some set  $R$ . The set  $R^\omega$  consists of streams over  $R$ , defined by  $R^\omega = [\mathbb{N}, R]$ . If we use  $\omega$  as ordinal for indexing, then the final chain  $\Phi S$  is isomorphic to the following chain.

$$\mathbf{1} \xleftarrow{!} R \xleftarrow{\pi_1} R^2 \xleftarrow{\pi_2} R^3 \xleftarrow{\dots} \dots$$

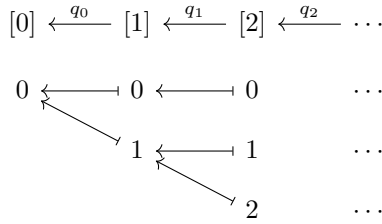
That is,  $(\Phi S)_0 \cong \mathbf{1}$  and for every  $i \in \mathbb{N}$ ,  $(\Phi S)_i \cong R^i$  via a chain map. Indeed,  $L\Phi S \cong R^\omega$  with the projections  $(p_i)_{i \in \mathbb{N}}$ , such that  $p_i: R^\omega \rightarrow R^i$  giving for every  $s \in R^\omega$  its first  $i$  elements. It is well known [7] that a function  $f: R^\omega \rightarrow R^\omega$  is causal if and only if for all  $k \in \mathbb{N}$ ,  $s, t \in R^\omega$ , if  $s(i) = t(i)$  for all  $i \leq k$ , then  $f(s)(k) = f(t)(k)$ . Which implicitly includes every  $i \leq k$ , that is  $f(s)(i) = f(t)(i)$ , and that is exactly Definition 3. From Theorem 4, we now obtain that we can equivalently see  $f$  as a chain map  $\chi(f): \Phi S \rightarrow \Phi S$ , where for  $u \in R^n$  we have  $\chi(f)_{n+1}(u) = f(u : s)$  for any stream  $s \in R^\omega$ . Note that this requires that  $R$  is inhabited.

► **Example 6.** For the functor  $N: \mathbf{Set} \rightarrow \mathbf{Set}$  given by  $N(X) = X + \mathbf{1}$ , where  $\mathbf{1} = \{*\}$ , one has  $\nu N \cong \mathbb{N} \cup \{\infty\}$ . and we use  $\omega$  as indexing ordinal. The final chain  $\Phi N$  is isomorphic to the following chain, in which  $[n] = \{k \in \mathbb{N} \mid 0 \leq k < n\}$ .

$$[0] \xleftarrow{!} [1] \xleftarrow{q_1} [2] \xleftarrow{q_2} [3] \xleftarrow{\dots} \dots$$



The projections  $q_i$  are the identity on numbers below  $i$  and truncate all higher numbers. Pictorially this looks as follows.



One can show [14, Ex. 8.4] that a map  $f: \nu N \rightarrow \nu N$  is causal if for all  $n, m$  and  $i \leq \min(n, m)$ , then  $f(n) = f(m)$  or  $i \leq \min(f(n), f(m))$ .

One may wonder where the last condition in Example 6 comes from. Let us, therefore, digress for a moment and explore yet another characterisation of causal morphisms.

### 3.1 Causality and Metric Maps

For the purpose of comparing causal maps with metric maps, we assume additionally that  $\mathbf{C}$  is locally small and that it has a *generator*  $G$ , which is an object such that the hom-functor  $\mathbf{C}(G, -): \mathbf{C} \rightarrow \mathbf{Set}$  is faithful. We will denote this functor by  $E = \mathbf{C}(G, -)$  and its action on a morphism  $f: X \rightarrow Y$  by  $f_*: EX \rightarrow EY$ . One can think of  $x \in EX$  as element of  $X$  and  $f_*(x) \in EY$  as its image under  $f$ . Moreover, we need that the functor  $F$  is  $\omega^{\text{op}}$ -continuous. These assumptions allow us to define a metric on final coalgebras and then prove that metric maps correspond to causal maps.

Let  $d: E(\nu F) \times E(\nu F) \rightarrow [0, 1]$  be the metric defined for  $e_1, e_2 \in E(\nu F)$  as follows.

$$d(e_1, e_2) = \sup\{2^{-i} \mid p_i \circ e_1 \neq p_i \circ e_2, i \in \mathbb{N}\} = \inf\{2^{-i} \mid p_i \circ e_1 = p_i \circ e_2, i \in \mathbb{N}\}$$

One can easily observe from Definition 3 that two outputs of causal morphisms  $f_*$  should not be more distant than their corresponding inputs. That is, causal functions are metric maps, in the following sense.

► **Definition 7.** *Let  $(X, d_X), (Y, d_Y)$  be two metric spaces. A function  $f: X \rightarrow Y$  is a metric map when for any elements  $x, y \in X$ , the following condition is fulfilled.*

$$d_Y(f(x), f(y)) \leq d_X(x, y)$$

*Metric spaces and metric maps form a category  $\mathbf{Met}$ .*

Now we can show the correspondence between causal morphisms and metric maps.

► **Theorem 8.** *The following are equivalent:*

1.  $f \in \mathbf{Caus}(\nu F, \nu F)$
2.  $f \in \mathbf{Met}((\nu F, d), (\nu F, d))$

**Proof.** (1  $\rightarrow$  2) By the universal property of sup, we need to prove  $2^{-l} \leq d(x, y)$  for all  $l$  with  $p_l \circ f_*(x) \neq p_l \circ f_*(y)$ . Given such an  $l$ , we get by causality of  $f$  that  $p_l \circ x \neq p_l \circ y$  and hence  $2^{-l} \leq d(x, y)$ . As this holds for all  $l$ , we get  $d(f_*(x), f_*(y)) \leq d(x, y)$ .

(2  $\rightarrow$  1) Conversely, let us assume that  $f$  is a metric map. That is  $d(f_*(x), f_*(y)) \leq d(x, y)$ , which implies that  $l \geq k$ . Hence, we have for all  $i < k$  the following.

$$p_i \circ x = p_i \circ y \implies f \circ p_i \circ x = f \circ p_i \circ y$$

Since  $f$  is a metric map, we also have  $p_i \circ f_*(x) = p_i \circ f_*(y)$ . Thus  $f$  is causal. ◀

Birkedal et al. [4] show that there is an adjunction between certain metric spaces and  $\overleftarrow{\mathbf{Set}}$ , and that there is a one-to-one correspondence between contractive maps in the metric sense and contractive maps in  $\overleftarrow{\mathbf{Set}}$ , see Section 2.1. One can think of Theorem 8 as a partial generalisation of this result, although we are mostly interested in it here to understand causality better in some examples.

► **Example 9.** Recall that we cited in Example 6 a rather odd looking characterisation of causal maps on partial computations. We can derive this characterisation from Theorem 8 as follows. Since if  $n = m$  we must have  $f(n) = f(m)$ , suppose without loss of generality  $n \neq m$ . For  $i \leq \min(n, m)$ , we get  $d(n, m) = 2^{-(\min(n, m)+1)}$ . If  $f$  is causal, we either have  $f(n) = f(m)$  or  $d(f(n), f(m)) = 2^{-(\min(f(n), f(m))+1)} \leq d(n, m)$ . By inspecting the two sides, we get that  $i \leq \min(n, m) \leq \min(f(n), f(m))$ , which is what we wanted to prove.

The results in Theorem 4 and Theorem 8 can be summed up as in the following diagram.

$$\begin{array}{ccc}
 & \text{Caus}(\nu F, \nu F) & \\
 \cong \nearrow & & \nwarrow \cong \\
 \overleftarrow{\mathbf{C}}(\Phi F, \Phi F) & \xrightarrow{\cong} & \text{Met}((\nu F, d), (\nu F, d))
 \end{array}$$

#### 4 Composition and Recursion

In this section, we construct for a fixed chain  $\sigma$  a symmetric monoidal category  $\mathbf{P}_\sigma$  together with a trace-like operator. This category allows us to construct diagrams of arbitrary causal morphisms with feedback loops. The SMC  $\mathbf{P}_\sigma$  will have as morphisms something one may think of building blocks with two kinds of interfaces: one for things of type  $\sigma$  over which we do recursion via traces and one type for parameter of arbitrary type. The diagram in Figure 1 displays the kind of circuit that we intend to build. Here, we build a circuit out of two causal morphisms  $f$  and  $g$ , where  $\tau_k$  are types of the parameters (blue wires) and the three loops going through small boxes indicate recursive feedback that goes through a *register* that can store elements of type  $\sigma$  (black wires). Such diagrams can be built, in the usual way, by parallel and sequential composition of morphisms and by looping interfaces of type  $\sigma$  back to inputs. What is not allowed are loops of types other than  $\sigma$ .

Let us first explain the nature of  $\mathbf{P}_\sigma$  and then we prove that it is a traced SMC. Recall that we can associate to any SMC, in this case,  $\overleftarrow{\mathbf{C}}$ , a canonical PROP [12]  $\mathbf{H}_\sigma$  with objects being natural numbers and morphisms given by  $\mathbf{H}_\sigma(n, m) = \overleftarrow{\mathbf{C}}(\sigma^n, \sigma^m)$ . In fact, any PROP is of this form [2]. In  $\mathbf{H}_\sigma$ , we could build diagrams with only black wires and our result Corollary 18 below will have as special case that this category is a traced SMC. However, we wish to have the extra flexibility of additional parameters, which we can achieve by creating a symmetric monoidal  $\overleftarrow{\mathbf{C}}$ -category that is tensored over  $\overleftarrow{\mathbf{C}}$ .

► **Theorem 10.** *Let  $(\mathbf{V}, \otimes, I)$  be a closed SMC and  $v \in \mathbf{V}$  some object. Denote by  $\mathbf{H}_v$  the  $\mathbf{V}$ -enriched PROP with natural numbers as objects and morphisms  $v^{\otimes n} \rightarrow v^{\otimes m}$  where  $v^{\otimes n}$  is the  $n$ -fold tensor product of  $v$ . There is a  $\mathbf{V}$ -enriched SMC  $\mathbf{P}_v$  with a fully faithful monoidal  $\mathbf{V}$ -functor  $(-): \mathbf{H}_v \rightarrow \mathbf{P}_v$  that is tensored over  $\mathbf{V}$ , which means that there is a monoidal functor  $\odot: \overline{\mathbf{V}} \times \mathbf{P}_v \rightarrow \mathbf{P}_v$  with natural isomorphisms  $\mathbf{P}_v(u \odot X, Y) \cong \mathbf{V}(u, \mathbf{P}_v(X, Y))$  for  $u \in \mathbf{V}$  and  $X, Y \in \mathbf{P}_v$ .*

**Proof.** We define  $\mathbf{P}_v$  to have as objects pairs  $(u, n)$  with  $u \in \mathbf{V}$  and  $n \in \mathbb{N}$ , and as morphisms we take

$$\mathbf{P}_v((u, n), (w, m)) = \mathbf{V}(u \otimes v^{\otimes n}, w \otimes v^{\otimes m}).$$

Since  $\mathbf{V}$  is closed, this makes  $\mathbf{P}_v$  immediately a  $\mathbf{V}$ -category. It is also symmetric monoidal with the product  $(u, n) \otimes_{\mathbf{P}_v} (w, m) = (u \otimes w, n + m)$  and unit  $I_{\mathbf{P}_v} = (I, 0)$ . The functor  $\mathbf{H}_v \rightarrow \mathbf{P}_v$  is given by  $\underline{n} = (I, n)$  and  $\underline{f} = I \otimes f$ . It is obviously monoidal and faithful, and that it is full follows from  $I$  being the monoidal unit. Finally, the tensor is defined by  $u \odot (w, n) = (u \otimes w, n)$  and we get immediately

$$\begin{aligned} \mathbf{P}_v(u \odot (x, n), (y, m)) &= \mathbf{P}_v((u \otimes x, n), (y, m)) \\ &= \mathbf{V}(u \otimes x \otimes v^{\otimes n}, y \otimes v^{\otimes m}) \\ &\cong \mathbf{V}(u \otimes, \mathbf{V}(x \otimes v^{\otimes n}, y \otimes v^{\otimes m})) \\ &= \mathbf{V}(u \otimes, \mathbf{P}_v((x, n), (y, m))) \end{aligned}$$

by  $\mathbf{V}$  being closed. Thus  $\mathbf{P}_v$  is also tensored over  $\mathbf{V}$ .  $\blacktriangleleft$

We now apply Theorem 10 to our situation of  $\alpha^{\text{op}}$ -chains to obtain for  $\sigma \in \overleftarrow{\mathbf{C}}$  a  $\overleftarrow{\mathbf{C}}$ -category  $\mathbf{P}_\sigma$  with pairs  $(\tau, n)$  of  $\tau \in \overleftarrow{\mathbf{C}}$  and  $n \in \mathbb{N}$  and

$$\mathbf{P}_\sigma((\tau, n), (\gamma, m)) = \overleftarrow{\mathbf{C}}(\tau \times \sigma^n, \gamma \times \sigma^m)$$

as hom-objects. We denote the monoidal product of  $\mathbf{P}_\sigma$  simply by  $\otimes$  and its unit by  $I$ . Since morphisms in  $\mathbf{P}_\sigma$  are particular morphisms in  $\overleftarrow{\mathbf{C}}$ , we make no distinction between, e.g.,  $\text{id}_{(\tau, n)}$  and  $\text{id}_{\tau \times \sigma^n}$  to lighten notation a bit.

Our goal now is to enable recursion in  $\mathbf{P}_\sigma$  via a trace operator [9]. Except that our trace will be *relative to*  $\mathbf{H}_\sigma$  in the sense that there is a family of maps

$$\text{Tr}_{X, Y}^k : \mathbf{P}_\sigma(X \otimes k, Y \otimes k) \rightarrow \mathbf{P}_\sigma(X, Y)$$

indexed by  $X, Y \in \mathbf{P}_\sigma$  and  $k \in \mathbf{H}_\sigma$  that fulfils the usual trace axioms. Since the functor  $\mathbf{H}_\sigma \rightarrow \mathbf{P}_\sigma$  is fully faithful, this will expose  $\mathbf{H}_\sigma$  as a proper traced SMC.

Whenever morphisms are defined by recursive equations, one has to provide boundary conditions to obtain a well-defined solution to the equations, even if they are implicit. In analogy with registers to create well-defined feedback loops as in Figure 1, an initial value that we place in the registers will take the role of boundary conditions in our case.

**► Definition 11.** We call a morphism  $i : \blacktriangleright \sigma \rightarrow \sigma$  in  $\overleftarrow{\mathbf{C}}$  an initial value. It gives rise to a morphism on powers of  $\sigma$  by  $\hat{i}^k = \blacktriangleright (\sigma^k) \xrightarrow{\delta^{\blacktriangleright}} (\blacktriangleright \sigma)^k \xrightarrow{i^k} \sigma^k$ . A morphism  $g : n \rightarrow m$  in  $\mathbf{H}_\sigma$  is compatible with  $i$  if  $\hat{i}^m \circ \blacktriangleright g = g \circ \hat{i}^n$ .

If  $\sigma \in [\omega^{\text{op}}, \mathbf{C}]$ , then an initial value  $i : \blacktriangleright \sigma \rightarrow \sigma$  consists of morphisms  $i_0 : \mathbf{1} \rightarrow \sigma_0$  and  $i_{n+1} : \sigma_n \rightarrow \sigma_{n+1}$  that are compatible with the chain  $\sigma$ . In the case of streams, see Example 5,  $i : \blacktriangleright (\Phi S) \rightarrow \Phi S$  picks out an element  $i_1 : \mathbf{1} \rightarrow R$  that all  $i_k : R^k \rightarrow R^{k+1}$  have to return as the first element. Compatibility of  $g$  with  $i$  means then that  $g_1 \circ i_1 = i_1$ , which is for example the case when  $i_1$  returns 0 and  $g$  is linear, see Section 5.1.

A good source of initial values for the final chain is pointed functors.

**► Proposition 12.** If  $F : \mathbf{C} \rightarrow \mathbf{C}$  is a pointed functor, i.e., comes with a natural transformation  $\eta : \text{Id} \rightarrow F$ , then there is an initial value  $\blacktriangleright \Phi F \rightarrow \Phi F$ .

**Proof.** The initial value is defined as the composite  $\blacktriangleright \Phi F \xrightarrow{\blacktriangleright \overleftarrow{\eta}_{\Phi F}} \blacktriangleright \overleftarrow{F} \Phi F \xrightarrow{\text{fold}} \Phi F$ .  $\blacktriangleleft$

In what follows, we assume an initial value to be given and construct the trace relative to it. Since  $\overleftarrow{\mathbf{C}}$  is Cartesian closed, we find that the morphism involved in our relative trace has a special shape.

We give the definition of morphisms with  $k$ -feedback loops as follows.

## 18:10 Composition and Recursion for Causal Structures

► **Definition 13.** A  $k$ -feedback morphism  $f \in \mathbf{P}_\sigma((\tau, n) \otimes_{\mathbf{P}_\sigma} \underline{k}, (\gamma, m) \otimes_{\mathbf{P}_\sigma} \underline{k})$  is of the form

$$f = \langle f_{\text{out}}, f_{\text{fb}} \rangle$$

such that  $f_{\text{out}} \in \mathbf{P}_\sigma((\tau, n) \otimes_{\mathbf{P}_\sigma} \underline{k}, (\gamma, m))$  refers to the output of  $f$  and  $f_{\text{fb}} \in \mathbf{P}_\sigma((\tau, n) \otimes_{\mathbf{P}_\sigma} \underline{k}, \underline{k})$  refers to the  $k$ -feedback loops of  $f$ , given by  $f_{\text{fb}} = \hat{\mathbf{i}}^k \circ \text{next}_{\sigma^k} \circ f_{\text{fb}}$ , where  $\hat{\mathbf{i}}^k \in \mathbf{P}_\sigma(\underline{k}, \underline{k})$  such that  $(\hat{\mathbf{i}}^k)_i: (\sigma_i)^k \rightarrow (\sigma_{i+1})^k$ .

The first step to defining a trace operator is to figure out the behaviour of the register involved in a feedback loop. To this end, let  $h: (\tau, n) \otimes \underline{k} \rightarrow \underline{k}$  be a morphism in  $\mathbf{P}_\sigma$  and consider the morphism  $\hat{\mathbf{i}}^k \circ \text{next}_{\sigma^k} \circ h: \tau \times \sigma^n \times \sigma^k \rightarrow \sigma^k$ , which is contractive with  $\hat{\mathbf{i}}^k \circ \blacktriangleright h \circ \delta \blacktriangleright \circ (\text{next}_{\tau \times \sigma^n} \times \text{id})$  because  $\text{next}$  is a monoidal natural transformation, as the following diagram shows, where  $X = \tau \times \sigma^n$ .

$$\begin{array}{ccccc}
 X \times \blacktriangleright(\sigma^k) & \xleftarrow{\text{id} \times \text{next}} & X \times \sigma^k & \xrightarrow{h} & \sigma^k \\
 \text{next} \times \text{id} \downarrow & & \text{next} \times \text{next} \swarrow & & \downarrow \text{next} \\
 \blacktriangleright X \times \blacktriangleright(\sigma^k) & \xrightarrow{\delta \blacktriangleright} & \blacktriangleright(X \times \sigma^k) & \xrightarrow{\blacktriangleright h} & \blacktriangleright(\sigma^k) \xrightarrow{\hat{\mathbf{i}}^k} \sigma^k
 \end{array}$$

We denote by  $s(h): (\tau, n) \rightarrow \underline{k}$  a solution for  $\hat{\mathbf{i}}^k \circ \text{next}_{\sigma^k} \circ h$ , that is, the unique morphism fulfilling the following equation.

$$s(h) = \hat{\mathbf{i}}^k \circ \text{next}_{\sigma^k} \circ h \circ \langle \text{id}_{(\tau, n)}, s(h) \rangle \quad (1)$$

We collect some properties of  $s(h)$  that we need to prove the trace axioms.

► **Lemma 14.** For any  $h: (\tau, n) \otimes \underline{k} \rightarrow \underline{k}$  and  $g: (\tau', n') \rightarrow (\tau, n)$  morphisms in  $\mathbf{P}_\sigma$ , if  $s(h)$  is a solution for  $\hat{\mathbf{i}}^k \circ \text{next}_{\sigma^k} \circ h$ , then  $s(h) \circ g$  is a solution for  $\hat{\mathbf{i}}^k \circ \text{next}_{\sigma^k} \circ h \circ (g \times \text{id}_{\underline{k}})$ .

**Proof.**  $s(h) \circ g$  is a solution for  $\hat{\mathbf{i}}^k \circ \text{next}_{\sigma^k} \circ h \circ (g \times \text{id}_{\underline{k}})$ , because

$$\begin{aligned}
 s(h) \circ g &= \hat{\mathbf{i}}^k \circ \text{next}_{\sigma^k} \circ h \circ \langle \text{id}_{(\tau, n)}, s(h) \rangle \circ g && \text{by def. of } s(h) \\
 &= \hat{\mathbf{i}}^k \circ \text{next}_{\sigma^k} \circ h \circ \langle g \circ \text{id}_{(\tau', n')}, s(h) \circ g \rangle \\
 &= \hat{\mathbf{i}}^k \circ \text{next}_{\sigma^k} \circ h \circ (g \times \text{id}_{\underline{k}}) \circ \langle \text{id}_{(\tau', n')}, s(h) \circ g \rangle && \blacktriangleleft
 \end{aligned}$$

The following lemma will allow us to prove the sliding axiom for tracing, but only for chain maps that are compatible with the initial value.

► **Lemma 15.** Suppose  $h': (\tau, n) \otimes \underline{k} \rightarrow \underline{k}'$  and  $g: \underline{k}' \rightarrow \underline{k}$  that is compatible with  $\mathbf{i}$ . If  $s(h' \circ (\text{id}_{(\tau, n)} \otimes g))$  is a solution for  $\hat{\mathbf{i}}^{k'} \circ \text{next}_{\sigma^{k'}} \circ h' \circ (\text{id}_{(\tau, n)} \otimes g)$ , then  $g \circ s(h' \circ (\text{id}_{(\tau, n)} \otimes g))$  is a solution for  $\hat{\mathbf{i}}^k \circ \text{next}_{\sigma^k} \circ g \circ h'$ .

**Proof.** Let  $s^{k'} = s(h' \circ (\text{id}_{(\tau, n)} \otimes g))$ , then  $g \circ s^{k'}$  is a solution for  $\hat{\mathbf{i}}^k \circ \text{next}_{\sigma^k} \circ g \circ h'$ , because

$$\begin{aligned}
 g \circ s^{k'} &= g \circ \hat{\mathbf{i}}^{k'} \circ \text{next}_{\sigma^{k'}} \circ h' \circ \langle \text{id}_{(\tau, n)}, g \circ s^{k'} \rangle, \\
 &= \hat{\mathbf{i}}^k \circ \blacktriangleright g \circ \text{next}_{\sigma^{k'}} \circ h' \circ \langle \text{id}_{(\tau, n)}, g \circ s^{k'} \rangle && g \text{ compatible with } \mathbf{i} \\
 &= \hat{\mathbf{i}}^k \circ \text{next}_{\sigma^k} \circ g \circ h' \circ \langle \text{id}_{(\tau, n)}, g \circ s^{k'} \rangle && \blacktriangleleft
 \end{aligned}$$

We propose a definition of a trace in  $\mathbf{P}_\sigma$  in the following theorem, followed by a proof that it satisfies the axioms of a trace [9].

► **Theorem 16.** For any  $X, Y, \underline{k} \in \mathbf{P}_\sigma$ , we define  $\mathrm{Tr}_{X,Y}^k: \mathbf{P}_\sigma(X \otimes \underline{k}, Y \otimes \underline{k}) \rightarrow \mathbf{P}_\sigma(X, Y)$  by

$$\mathrm{Tr}_{X,Y}^k(f) = f_{\mathrm{out}} \circ \langle \mathrm{id}_X, s(f_{\mathrm{fb}}) \rangle \quad (2)$$

a family of morphisms that satisfy the axioms of a trace, with the exception that dinaturality is relative to  $i$ -compatible morphisms.

**Proof.**

1. *Naturality on  $(\tau, n)$ :*  $\mathrm{Tr}_{-, (\gamma, m)}^k: \mathbf{P}_\sigma(- \otimes \underline{k}, (\gamma, m) \otimes \underline{k}) \rightarrow \mathbf{P}_\sigma(-, (\gamma, m))$  is a natural transformation.

Let  $f: (\tau, n) \otimes \underline{k} \rightarrow (\gamma, m) \otimes \underline{k}$  be  $k$ -feedback and  $g: (\tau', n') \rightarrow (\tau, n)$ , both morphisms in  $\mathbf{P}_\sigma$ . We need to show that

$$\mathrm{Tr}_{(\tau', n'), (\gamma, m)}^k(f \circ (g \otimes \mathrm{id}_{\underline{k}})) = \mathrm{Tr}_{(\tau, n), (\gamma, m)}^k(f) \circ g. \quad (3)$$

Since  $f$  is  $k$ -feedback, we have

$$(f \circ (g \otimes \mathrm{id}_{\underline{k}}))_{\mathrm{out}} = f_{\mathrm{out}} \circ (g \otimes \mathrm{id}_{\underline{k}}) \text{ and } (f \circ (g \otimes \mathrm{id}_{\underline{k}}))_{\mathrm{fb}} = f_{\mathrm{fb}} \circ (g \otimes \mathrm{id}_{\underline{k}}). \quad (4)$$

Hence, by Equation (2),

$$\mathrm{Tr}_{(\tau', n'), (\gamma, m)}^k(f \circ (g \otimes \mathrm{id}_{\underline{k}})) = f_{\mathrm{out}} \circ (g \otimes \mathrm{id}_{\underline{k}}) \circ \langle \mathrm{id}_{(\tau', n')}, s(f_{\mathrm{fb}} \circ (g \otimes \mathrm{id}_{\underline{k}})) \rangle \quad (5)$$

where  $s(f_{\mathrm{fb}} \circ (g \otimes \mathrm{id}_{\underline{k}}))$  is a solution for  $\hat{i}^k \circ \mathrm{next}_{\sigma^k} \circ f_{\mathrm{fb}} \circ (g \otimes \mathrm{id}_{\underline{k}})$ , and

$$s(f_{\mathrm{fb}} \circ (g \otimes \mathrm{id}_{\underline{k}})) = \hat{i}^k \circ \mathrm{next}_{\sigma^k} \circ f_{\mathrm{fb}} \circ (g \otimes \mathrm{id}_{\underline{k}}) \circ \langle \mathrm{id}_{(\tau', n')}, s(f_{\mathrm{fb}} \circ (g \otimes \mathrm{id}_{\underline{k}})) \rangle.$$

We also have,  $\mathrm{Tr}_{(\tau, n), (\gamma, m)}^k(f) \circ g = f_{\mathrm{out}} \circ \langle \mathrm{id}_{(\tau, n)}, s(f_{\mathrm{fb}}) \rangle \circ g$ , such that,  $s(f_{\mathrm{fb}})$  being the fixed point of  $\hat{i}^k \circ \mathrm{next}_{\sigma^k} \circ f_{\mathrm{fb}}$  and  $s(f_{\mathrm{fb}}) = \hat{i}^k \circ \mathrm{next}_{\sigma^k} \circ f_1 \circ \langle \mathrm{id}_{(\tau, n)}, s \rangle$ .

By Lemma 14, we get

$$\begin{aligned} \mathrm{Tr}_{(\tau', n'), (\gamma, m)}^k(f \circ (g \otimes \mathrm{id}_{\underline{k}})) &= f_{\mathrm{out}} \circ (g \otimes \mathrm{id}_{\underline{k}}) \circ \langle \mathrm{id}_{(\tau', n')}, s(f_{\mathrm{fb}}) \circ g \rangle, \\ &= f_{\mathrm{out}} \circ \langle \mathrm{id}_{(\tau, n)}, s(f_{\mathrm{fb}}) \rangle \circ g, \\ &= \mathrm{Tr}_{(\tau, n), (\gamma, m)}^k(f) \circ g. \end{aligned}$$

Hence, Equation (3).

2. *Naturality on  $(\gamma, m)$ :*  $\mathrm{Tr}_{(\tau, n), -}^k: \mathbf{P}_\sigma((\tau, n) \otimes \underline{k}, - \otimes \underline{k}) \rightarrow \mathbf{P}_\sigma((\tau, n), -)$  is a natural transformation.

Let  $f: (\tau, n) \otimes \underline{k} \rightarrow (\gamma, m) \otimes \underline{k}$  and  $g: (\gamma, m) \rightarrow (\gamma', m')$ , we need to show that

$$\mathrm{Tr}_{(\tau, n), (\gamma', m')}^k((g \otimes \mathrm{id}_{\underline{k}}) \circ f) = g \circ \mathrm{Tr}_{(\tau, n), (\gamma, m)}^k(f). \quad (6)$$

For the  $k$ -feedback morphism  $(g \otimes \mathrm{id}_{\underline{k}}) \circ f$ ,

$$((g \otimes \mathrm{id}_{\underline{k}}) \circ f)_{\mathrm{out}} = g \circ f_{\mathrm{out}}, \text{ and } ((g \otimes \mathrm{id}_{\underline{k}}) \circ f)_{\mathrm{fb}} = f_{\mathrm{fb}}.$$

By definition,  $\mathrm{Tr}_{(\tau, n), (\gamma', m')}^k((g \otimes \mathrm{id}_{\underline{k}}) \circ f) = g \circ f_{\mathrm{out}} \circ \langle \mathrm{id}_{(\tau, n)}, s(f_{\mathrm{fb}}) \rangle$ , and

$g \circ \mathrm{Tr}_{(\tau, n), (\gamma, m)}^k(f) = g \circ f_{\mathrm{out}} \circ \langle \mathrm{id}_{(\tau, n)}, s(f_{\mathrm{fb}}) \rangle$ . Hence, Equation (6).

3. *Dinaturality on  $\underline{k}$ :*  $\mathrm{Tr}_{(\tau, n), (\gamma, m)}^-: \mathbf{P}_\sigma((\tau, n) \otimes -, (\gamma, m) \otimes -) \rightarrow \mathbf{P}_\sigma((\tau, n), (\gamma, m))$  is a dinatural transformation, on the full subcategory  $\mathbf{H}_\sigma$  with objects of the form  $\underline{n} = (K\mathbf{1}, n)$  for all  $n \in \mathbb{N}$ , and if  $i_{\sigma^k}$  at every  $k \in \mathbb{N}$  satisfies for each  $g: \underline{k} \rightarrow \underline{k}'$ ,  $g \circ \hat{i}^k = \hat{i}^{k'} \circ g$ .

## 18:12 Composition and Recursion for Causal Structures

Let  $f: (\tau, n) \otimes \underline{k} \rightarrow (\gamma, m) \otimes \underline{k}'$  and  $g: \underline{k}' \rightarrow \underline{k}$ , we need to show that

$$\mathrm{Tr}_{(\tau, n), (\gamma, m)}^k((\mathrm{id}_{(\gamma, m)} \otimes g) \circ f) = \mathrm{Tr}_{(\tau, n), (\gamma, m)}^{k'}(f \circ (\mathrm{id}_{(\tau, n)} \otimes g)). \quad (7)$$

Note that  $(\mathrm{id}_{(\gamma, m)} \otimes g) \circ f$  is  $k$ -feedback with  $((\mathrm{id}_{(\gamma, m)} \otimes g) \circ f)_{\mathrm{out}} = f_{\mathrm{out}}$ , and  $((\mathrm{id}_{(\gamma, m)} \otimes g) \circ f)_{\mathrm{fb}} = (g \circ f)_{\mathrm{fb}}$ ; and  $f \circ (\mathrm{id}_{(\tau, n)} \otimes g)$  is  $k'$ -feedback, with  $(f \circ (\mathrm{id}_{(\tau, n)} \otimes g))_{\mathrm{out}} = f_{\mathrm{out}} \circ (\mathrm{id}_{(\tau, n)} \otimes g)$ , and  $(f \circ (\mathrm{id}_{(\tau, n)} \otimes g))_{\mathrm{fb}} = f_{\mathrm{fb}} \circ (\mathrm{id}_{(\tau, n)} \otimes g)$ ; such that  $f_{\mathrm{out}}: \tau \times \sigma^n \times \sigma^k \rightarrow \gamma \times \sigma^m$  and  $f_{\mathrm{fb}}: \tau \times \sigma^n \times \sigma^k \rightarrow \sigma^{k'}$ .

Then, by Theorem 16, we have

$$\mathrm{Tr}_{(\tau, n), (\gamma, m)}^k((\mathrm{id}_{(\gamma, m)} \otimes g) \circ f) = f_{\mathrm{out}} \circ \langle \mathrm{id}_{(\tau, n)}, s(g \circ f_{k'}) \rangle; \quad (8)$$

and

$$\begin{aligned} \mathrm{Tr}_{(\tau, n), (\gamma, m)}^{k'}(f \circ (\mathrm{id}_{(\tau, n)} \otimes g)) &= f_{\mathrm{out}} \circ (\mathrm{id}_{(\tau, n)} \otimes g) \circ \langle \mathrm{id}_{(\tau, n)}, s(f_{k'} \circ (\mathrm{id}_{(\tau, n)} \otimes g)) \rangle, \\ &= f_{\mathrm{out}} \circ \langle \mathrm{id}_{(\tau, n)}, g \circ s(f_{k'} \circ (\mathrm{id}_{(\tau, n)} \otimes g)) \rangle. \end{aligned}$$

Let  $s^{k'} = s(f_{k'} \circ (\mathrm{id}_{(\tau, n)} \otimes g))$ , a solution for  $\mathbf{i}_{\sigma^{k'}} \circ \mathrm{next}_{\sigma^{k'}} \circ f_{k'} \circ (\mathrm{id}_{(\tau, n)} \otimes g)$ , then by Lemma 15,  $g \circ s^{k'}$  is a solution for  $\hat{\mathbf{i}}^k \circ \mathrm{next}_{\sigma^k} \circ g \circ f_{k'}$ . Hence, we can substitute  $s(g \circ f_{k'})$  in Equation (8), by  $g \circ s^{k'}$ , and we get

$$\begin{aligned} \mathrm{Tr}_{(\tau, n), (\gamma, m)}^k((\mathrm{id}_{(\gamma, m)} \otimes g) \circ f) &= f_{\mathrm{out}} \circ \langle \mathrm{id}_{(\tau, n)}, s(g \circ f_{k'}) \rangle, \\ &= f_{\mathrm{out}} \circ \langle \mathrm{id}_{(\tau, n)}, g \circ s^{k'} \rangle, \\ &= \mathrm{Tr}_{(\tau, n), (\gamma, m)}^{k'}(f \circ (\mathrm{id}_{(\tau, n)} \otimes g)). \end{aligned}$$

► Remark 17. In the case where we do not have  $g \circ \mathbf{i}_{\sigma^k} = \mathbf{i}_{\sigma^{k'}} \circ \blacktriangleright g$ , dinaturality is not satisfied.

We have now seen that trace in Theorem 16 is a family of natural morphisms, we are left to check if they fulfill the three axioms of trace in [9], for symmetric monoidal categories.

4. *Vanishing 1:* Let  $f: (\tau, n) \otimes \underline{0} \rightarrow (\gamma, m) \otimes \underline{0}$  and  $\iota_r: - \otimes \underline{1} \rightarrow -$ , where  $\iota_r$  is the right unitor. Then we need to show, that

$$\mathrm{Tr}_{(\tau, n), (\gamma, m)}^0(f) = \iota_r(\gamma, m) \circ f \circ \iota_r(\tau, n)^{-1}. \quad (9)$$

Note that  $\mathrm{Tr}_{(\tau, n), (\gamma, m)}^0: \mathbf{P}_\sigma((\tau, n), (\gamma, m)) \rightarrow \mathbf{P}_\sigma((\tau, n), (\gamma, m))$

In this case,  $f$  is 0-feedback, therefore  $f_{\mathrm{out}} = f$ . Hence

$$\begin{aligned} \mathrm{Tr}_{(\tau, n), (\gamma, m)}^0(f) &= f \\ &= \iota_r(\gamma, m) \circ f \circ \iota_r(\tau, n)^{-1}. \end{aligned}$$

5. *Vanishing 2:* Let  $f: (\tau, n) \otimes \underline{1} \otimes \underline{1} \rightarrow (\gamma, m) \otimes \underline{1} \otimes \underline{1}$  We need to show that

$$\mathrm{Tr}_{(\tau, n), (\gamma, m)}^2(f) = \mathrm{Tr}_{(\tau, n), (\gamma, m)}^1(\mathrm{Tr}_{(\tau, n+1), (\gamma, m+1)}^1(f)) \quad (10)$$

We have,  $f$  is a 2-feedback with  $f = \langle f_{\mathrm{out}}, f_2 \rangle = \langle f_{\mathrm{out}}, f_{21}, f_1 \rangle = \langle f_{\mathrm{out}, 2\mathrm{out}}, f_1 \rangle$ . Then,

$$\mathrm{Tr}_{(\tau, n+1), (\gamma, n+1)}^1(f) = f_1 \circ \langle \mathrm{id}_{(\tau, n+1)}, s_1 \rangle \quad (11)$$

such that  $s_1$  is a solution for  $\hat{\mathbf{i}}^1 \circ \mathrm{next}_\sigma \circ f_1$ . Then

$$\mathrm{Tr}_{(\tau, n), (\gamma, m)}^1(\mathrm{Tr}_{(\tau, n+1), (\gamma, m+1)}^1(f)) = (f_1 \circ \langle \mathrm{id}_{(\tau, n+1)}, s_1 \rangle)_1 \circ \langle \mathrm{id}_{(\tau, n)}, s_2 \rangle, \quad (12)$$

such that  $s_2$  is a solution for  $\hat{i}^1 \circ \text{next}_\sigma \circ \langle f_{\text{out},2\text{out}} \circ \langle \text{id}_{(\tau,n+1)}, s_1 \rangle \rangle_2$  and  $s_2 = \hat{i}^1 \circ \text{next}_\sigma \circ f_{21} \circ \langle \text{id}_{(\tau,n+1)}, s_1 \rangle \circ \langle \text{id}_{(\tau,n)}, s_2 \rangle$ , where

$$\begin{aligned} \langle f_{\text{out},2\text{out}} \circ \langle \text{id}_{(\tau,n+1)}, s_1 \rangle \rangle_1 &= f_{\text{out}} \circ \langle \text{id}_{(\tau,n+1)}, s_1 \rangle && \text{and} \\ \langle f_{\text{out},2\text{out}} \circ \langle \text{id}_{(\tau,n+1)}, s_1 \rangle \rangle_2 &= f_{21} \circ \langle \text{id}_{(\tau,n+1)}, s_1 \rangle && . \end{aligned}$$

Hence  $\text{Tr}_{(\tau,n),(\gamma,m)}^1(\text{Tr}_{(\tau,n+1),(\gamma,m+1)}^1(f)) = f_{\text{out}} \circ \langle \text{id}_{(\tau,n+1)}, s_1 \rangle \circ \langle \text{id}_{(\tau,n)}, s_2 \rangle$ . On the other hand, We have  $f = \langle f_{\text{out}}, \langle f_{21}, f_1 \rangle \rangle$ , and  $\text{Tr}_{(\tau,n),(\gamma,m)}^2(f) = f_{\text{out}} \circ \langle \text{id}_{(\tau,n)}, s \rangle$ , where  $s$  is a solution for  $\hat{i}^2 \circ \text{next}_{\sigma^2} \circ \langle f_{21}, f_1 \rangle$ . We can show that  $t = \langle s_2, s_1 \circ \langle \text{id}_{(\tau,n)}, s_2 \rangle \rangle$  is a solution for  $\hat{i}^2 \circ \text{next}_{\sigma^2} \circ \langle f_{21}, f_1 \rangle$ . We have  $\langle \text{id}_{(\tau,n+1)}, s_1 \rangle \circ \langle \text{id}_{(\tau,n)}, s_2 \rangle = \langle \text{id}_{(\tau,n)}, t \rangle$ , where  $t = \langle s_2, s_1 \circ \langle \text{id}_{(\tau,n)}, s_2 \rangle \rangle = \hat{i}^2 \circ \text{next}_{\sigma^2} \circ \langle f_{21}, f_1 \rangle \circ \langle \langle \text{id}_{(\tau,n)}, t \rangle \rangle$ . Therefore,  $t$  is a solution for  $\hat{i}^2 \circ \text{next}_{\sigma^2} \circ \langle f_{21}, f_1 \rangle$ . Thus, we have the following.

$$\begin{aligned} \text{Tr}_{(\tau,n),(\gamma,m)}^2(f) &= f_{\text{out}} \circ \langle \text{id}_{(\tau,n)}, t \rangle \\ &= \text{Tr}_{(\tau,n),(\gamma,m)}^1(\text{Tr}_{(\tau,n+1),(\gamma,m+1)}^1(f)) \end{aligned}$$

6. *Superposing*: Let  $f: (\tau, n) \otimes \underline{1} \rightarrow (\gamma, m) \otimes \underline{1}$  and  $g: (\tau', n') \rightarrow (\gamma', m')$ , we need to show that

$$g \circ \text{Tr}_{(\tau,n),(\gamma,m)}^1(f) = \text{Tr}_{(\tau',n') \otimes (\tau,n), (\gamma',m') \otimes (\gamma,m)}^1(g \otimes f). \quad (13)$$

We have  $\text{Tr}_{(\tau',n') \otimes (\tau,n), (\gamma',m') \otimes (\gamma,m)}^1(g \otimes f) = (g \otimes f_1) \circ \langle \text{id}_{(\tau',n') \otimes (\tau,n)}, s \rangle$ , where  $s$  is a solution for  $\hat{i}^1 \circ \text{next}_\sigma \circ (g \otimes f)_2 = \hat{i}^1 \circ \text{next}_\sigma \circ f_1 \circ \pi_{(\tau,n+1)}$ . If  $s(f_1)$  is a solution for  $\hat{i}^1 \circ \text{next}_\sigma \circ f_1$ , then  $s(f_1) \circ \pi_{(\tau,n)}$  is a solution for  $\hat{i}^1 \circ \text{next}_\sigma \circ f_1 \circ \pi_{(\tau,n+1)}$ , because of the following

$$\begin{aligned} s(f_1) \circ \pi_{(\tau,n)} &= \hat{i}^1 \circ \text{next}_\sigma \circ f_1 \circ \langle \text{id}_{(\tau,n)}, s(f_1) \rangle \circ \pi_{(\tau,n)}, \\ &= \hat{i}^1 \circ \text{next}_\sigma \circ f_1 \circ \langle \pi_{(\tau,n)} \circ \text{id}_{(\tau',n') \otimes (\tau,n)}, s(f_1) \circ \pi_{(\tau,n)} \rangle. \end{aligned}$$

By definition,  $\text{Tr}_{(\tau,n),(\gamma,m)}^1(f) = f_1 \circ \langle \text{id}_{(\tau,n)}, s(f_1) \rangle$ . Hence,

$$\begin{aligned} \text{Tr}_{(\tau',n') \otimes (\tau,n), (\gamma',m') \otimes (\gamma,m)}^1(g \otimes f) &= (g \otimes f_1) \circ \langle \text{id}_{(\tau',n') \otimes (\tau,n)}, s(f_1) \circ \pi_{(\tau,n)} \rangle \\ &= g \otimes \text{Tr}_{(\tau,n),(\gamma,m)}^1(f) \end{aligned}$$

Therefore, we have Equation (13).

7. *Yanking*: We need to show, for the component at  $(\underline{1}, \underline{1})$  of the braiding, i.e.  $\xi_{\underline{1}, \underline{1}}$ , that

$$\text{Tr}_{(\underline{1}, \underline{1})}^1(\xi_{\underline{1}, \underline{1}}) = \text{id}_{\underline{1}}. \quad (14)$$

Note that  $\xi_{\underline{1}, \underline{1}} = \langle \pi_1, \pi_2 \rangle$ ,  $\text{Tr}_{(\underline{1}, \underline{1})}^1(\xi_{\underline{1}, \underline{1}}) = \pi_1 \circ \langle \text{id}_{\underline{1}}, s(\pi_2) \rangle$ , where  $s(\pi_2)$  is a solution for  $\pi_2$ .  $\text{id}_{\underline{1}}$  is a solution for  $\pi_2$ . Hence, Equation (14).

The dinaturality of  $\text{Tr}_{(\tau,n),(\gamma,m)}^-$  is only on  $\mathbf{P}_\sigma$ , and only fulfilled if for any  $g \in \overleftarrow{\mathbf{C}}(k, k)$ ,  $\hat{i}^k \circ \blacktriangleright g = g \circ \hat{i}^{k'}$ .  $\blacktriangleleft$

The following is a consequence of Theorem 16.

- **Corollary 18.**  $\text{Tr}_{n,m}^k$  is a trace operator on  $\mathbf{H}_\sigma$  if all  $g: k \rightarrow k$  are  $i$ -compatible.

**Proof.** This follows from Theorem 16 because the functor  $\mathbf{H}_\sigma \rightarrow \mathbf{P}_\sigma$  is fully faithful.  $\blacktriangleleft$

Going back to causality, by definition  $\text{Tr}_{(\tau,n),(\gamma,m)}^k(f)$  is a morphism in  $\overleftarrow{\mathbf{C}}$ . Therefore  $L(\text{Tr}_{(\tau,n),(\gamma,m)}^k(f))$  is causal by Theorem 4. As Theorem 4 establishes a bijective correspondence, we find that  $\text{Caus}(\nu F, \nu F)$  is closed under sequential composition, parallel composition and under recursion via trace. In the following section, we show some applications of this.

## 5 Applications

Before we come to concrete applications, we mention here that *distributive laws*, that is, natural transformations  $\delta: GF \rightarrow FG$ , induce morphisms  $\hat{\delta}: \overleftarrow{G}\Phi F \rightarrow \Phi F$  [3]. In particular, distributive laws  $\delta: \Sigma_n F \rightarrow F\Sigma_n$  for the functor  $\Sigma_n: \mathbf{C} \rightarrow \mathbf{C}$  given by  $\Sigma_n(X) = X^n$  allow us to define  $n$ -ary causal morphisms. If, moreover,  $F$  is pointed with  $\eta: \mathbf{Id} \rightarrow F$  and  $\delta \circ \Sigma_n \eta = \eta \Sigma_n$ , the induced map  $\hat{\delta}: (\Phi F)^n \rightarrow \Phi F$  is compatible with the initial value induced by  $\eta$ , see Proposition 12.

### 5.1 Linear Stream Functions

In this section, we look into functions over the set  $R^\omega$  of all streams over a commutative ring  $(R, +, \cdot, 0, 1)$ . The set  $R^\omega$  is a commutative ring, with the pointwise addition  $+$ , the convolution product  $\times$ , together with their respective unit stream, see [15]. Moreover, for any  $n \in \mathbb{N}$ ,  $(R^\omega)^n$  is an  $R^\omega$ -module and module homomorphisms are  $R^\omega$ -linear systems in the following sense.

► **Definition 19.** A system  $\langle f_1, \dots, f_m \rangle: (R^\omega)^n \rightarrow (R^\omega)^m$  is  $R^\omega$ -linear if for every  $i \in \{1, \dots, m\}$ ,  $f_i: (R^\omega)^n \rightarrow R^\omega$  is  $R^\omega$ -linear, i.e., for all streams  $u, v \in R^\omega$  and  $(s_1, \dots, s_n), (t_1, \dots, t_n) \in (R^\omega)^n$

$$f((u \times (s_1, \dots, s_n)) + (v \times (t_1, \dots, t_n))) = (u \times f(s_1, \dots, s_n)) + (v \times f(t_1, \dots, t_n))$$

where  $f(s_1, \dots, s_n) = (z_1 \times s_1) + \dots + (z_n \times s_n)$  for some fixed rational streams<sup>1</sup>  $z_1, \dots, z_n \in R^\omega$ .

We consider the above linear systems because they are characterization of finite stream circuits, possibly with feedback loops under the condition that each loop passes through at least one register, see [15].

► **Theorem 20.** Every linear stream operator  $f: (R^\omega)^n \rightarrow R^\omega$  is causal.

**Proof.** For every  $(s_1, \dots, s_n), (t_1, \dots, t_n), (z_1, \dots, z_n) \in (R^\omega)^n$  and  $k \in \mathbb{N}$ , we assume for all  $i \leq k$  and  $1 \leq j \leq n$  that  $s_j(i) = t_j(i)$ . We have  $f(s_1, \dots, s_n)(k) = \sum_{j=1}^n \sum_{i=0}^k z_j(i) \cdot s_j(k-i)$  and  $f(t_1, \dots, t_n)(k) = \sum_{j=1}^n \sum_{i=0}^k z_j(i) \cdot t_j(k-i)$ .

For all  $i \leq k$ ,  $k-i \leq k$ . Hence,  $s_j(k-i) = t_j(k-i)$  for all  $1 \leq j \leq n$ . Thus, for all  $k \in \mathbb{N}$ ,  $f(s_1, \dots, s_n)(k) = f(t_1, \dots, t_n)(k)$ . ◀

We have seen that  $R^\omega \cong L\Phi S$  where  $\Phi S$  is isomorphic to an  $\omega^{\text{op}}$ -chain as described in Example 5. We aim to define stream circuits with feedback loops with initial condition [15] as the trace of functions on the final chain  $\Phi S$ .

Consider the pointed functor  $(S, \eta^S)$ , where  $S = R \times \mathbf{Id}$ , the functor from Example 5 and  $\eta^S: \mathbf{Id} \rightarrow S$  is a natural transformation defined for a fixed  $r \in R$  such that  $\mu_X(u) = (r, u)$ , for every  $u \in X$ . Then we get a chain map  $i: \blacktriangleright \Phi S \rightarrow \Phi S$  defined by  $i_0: \mathbf{1} \rightarrow R$  and  $i_n: R^n \rightarrow R^{n+1}$  with  $i_n(u) = (r, u)$  for every  $n \in \mathbb{N}$  and  $u \in R^n$ . Moreover,  $(\pi_n \circ i_n)(u) = (r, \pi_{n-1}(u))$  as given in the following.

$$\begin{array}{ccccccc} \mathbf{1} & \xleftarrow{!} & \mathbf{1} & \xleftarrow{!} & R & \xleftarrow{\pi_1} & R^2 & \xleftarrow{\quad} & \dots \\ \downarrow ! & & \downarrow i_0 & & \downarrow i_1 & & \downarrow i_2 & & \\ \mathbf{1} & \xleftarrow{!} & R & \xleftarrow{\pi_1} & R^2 & \xleftarrow{\pi_2} & R^3 & \xleftarrow{\quad} & \dots \end{array}$$

<sup>1</sup> A rational stream is a product of polynomial streams and inverse of a polynomial stream, see [15, Def. 3.32].



The morphism  $\text{next}: \Phi S \rightarrow \blacktriangleright \Phi S$  is defined for every  $n \in \mathbb{N}$  by  $\text{next}_n: R^{n+1} \rightarrow R^n$  such that  $\text{next}_n = \pi_n$ . Hence, for every  $u \in R^{n+1}$ ,  $(i_n \circ \text{next}_n)(u) = (r, \pi_{n-1}(u))$ . Note that, for  $r = 0$  the latter can be implemented by a register with initial value 0 [15] and the trace of a function  $f: (\Phi S)^{n+1} \rightarrow (\Phi)^{m+1}$ , given by  $f = \langle f_{\text{out}}, f_{\text{fb}} \rangle$  such that  $f_{\text{out}}: (\Phi S)^{n+1} \rightarrow (\Phi S)^m$  and  $f_{\text{fb}}: (\Phi S)^{n+1} \rightarrow \Phi S$ , is defined by

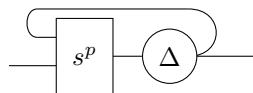
$$\text{Tr}_{n,m}^k(f) = f_{\text{out}} \circ \langle \text{id}_n, s(f_{\text{fb}}) \rangle$$

where  $s(f_{\text{fb}})$  is a fixed point for  $i \circ \text{next} \circ f_{\text{fb}}$ .

Since the trace of a chain map is a chain map, it is as well causal by Theorem 4.

## 5.2 Probabilistic Computations

Let us denote by  $\mathcal{D}: \mathbf{Set} \rightarrow \mathbf{Set}$  the (functor of the) finite probability distribution monad. The elements of  $\mathcal{D}(X)$  are maps  $d: X \rightarrow [0,1]$  that have only finitely many elements in the support  $\text{supp}(d) = \{x \in X \mid d(x) \neq 0\}$  and such that  $\sum_{x \in \text{supp}(d)} d(x) = 1$ . On maps  $f: X \rightarrow Y$ ,  $\mathcal{D}$  is defined by  $\mathcal{D}(f)(d)(y) = \sum_{f(x)=y} d(x)$ . We can now consider probabilistic stream systems, also known as labelled Markov chains, which are coalgebras for the composed functor  $\mathcal{D}_R = \mathcal{D}(R \times \text{Id})$ .



■ **Figure 2** Diagram for computing discounted sum  $ds_p$ .

Let us construct a discounted sum operation  $ds_p: \Phi \mathcal{D}_R \rightarrow \Phi \mathcal{D}_R$  for  $p \in [0,1]$  as the diagram displayed in Figure 2. First of all, the convex sum induces a distributive law  $c^p: \Sigma_2 \mathcal{D}_R \rightarrow \mathcal{D}_R$  given by  $c_X^p(d_1, d_2)(r, x, y) = pd_1(r, x) + (1-p)d_2(r, y)$ . This gives us a causal map  $\hat{c}^p: (\Phi \mathcal{D}_R)^2 \rightarrow \Phi \mathcal{D}_R$ . Finally, we obtain  $ds_p$  as  $\text{Tr}(\Delta \circ \hat{c}^p)$ , where  $\Delta$  is the diagonal map  $\Phi \mathcal{D}_R \rightarrow (\Phi \mathcal{D}_R)^2$ .

Note that  $\hat{c}^p$  is *not* compatible with the initial value induced by the unit  $\eta^{\mathcal{D}}$  of the distribution monad, which is defined by  $\eta_X^{\mathcal{D}}(x) = 1$ . In particular, we obtain  $(s^p \circ \Sigma_2 \eta^{\mathcal{D}})(x, y) = p\eta^{\mathcal{D}}(x) + (1-p)\eta^{\mathcal{D}}(y)$  and this is not a Dirac distribution given by  $\eta^{\mathcal{D}}$ , unless  $x = y$ .

## 5.3 Remark

A potential example that one could additionally consider is the category of presheaves  $\text{PSh}(P) = [P^{\text{op}}, \mathbf{Set}]$  on a preordered set  $P$ . The category  $\text{PSh}(P)$  is Cartesian closed and for a limit preserving functor  $F$ , the carrier of a final coalgebra for  $F$  is a presheaf, which is a functor  $\nu F: P^{\text{op}} \rightarrow \mathbf{Set}$ . Hence a causal morphism  $f: \nu F \rightarrow \nu F$  is a natural transformation and the corresponding chain map is a morphism between a final chain, which is a diagram in  $\hat{\text{PSh}}(P) = [\alpha^{\text{op}}, \text{PSh}(P)] = [\alpha^{\text{op}}, [P^{\text{op}}, \mathbf{Set}]]$ , for a limit ordinal  $\alpha$ . Moreover,  $\text{PSh}(P)$  has a generator. Therefore, one could investigate the meaning of causality using theorem 4 and theorem 8.

## 6 Summary, Related Work and Future Work

We have defined causal morphisms on the carrier of a final coalgebra  $\nu F$  for a limit preserving endofunctor  $F$  on arbitrary cartesian closed categories  $\mathbf{C}$ . We have seen, based on the construction of a final coalgebra via final chains, that there is a one-to-one correspondence between causal maps in  $\text{Caus}(\nu F, \nu F)$  and chain maps in  $\overline{\mathbf{C}}(\Phi F, \Phi F)$ , where  $\nu F$  is isomorphic to the limit of  $\Phi F$ . For a locally small category with a generator, we equipped  $\nu F$  with a metric and found out that causal morphisms are metric maps and vice versa. Additionally, we have constructed on a category of descending chains a (parameterised) traced symmetric monoidal category, on which causal morphisms (simply chain maps between final chains) are closed under sequential and parallel composition and under recursion via the trace operator.

[16] and [14] both give a definition of causal functions via finite approximations, but both work on **Set** and give the equivalence between causal functions on final coalgebras and morphisms on their finite approximations. We can easily extend our definition to causal algebras, as in [14], which gives us the inspiration to more general notion of causality. [16] introduced recursion in their work, which could be achieved in a traced symmetric monoidal category. They also defined linear causal maps, but for our case, it is enough to talk about linearity since we show that linear maps are causal.

For future work, we consider working on other cartesian closed categories such as  $G - \mathbf{Set}$  of sets with group actions from  $G$ , particularly nominal set; and also on the CCC of quasi-Borel spaces on which one can formalize some probability theory. One could use monoidal closed categories instead of cartesian closed and see how everything works out. We would also like to extend the notion of causality to more general continuity properties.

---

### References

- 1 J Adámek, S Milius, and LS Moss. Initial algebras, terminal coalgebras, and the theory of fixed points of functors, 2019.
- 2 John C. Baez, Brandon Coya, and Franciscus Rebro. Props in Network Theory, June 2018. [arXiv:1707.08321](https://arxiv.org/abs/1707.08321).
- 3 Henning Basold. Coinduction in Flow: The Later Modality in Fibrations. In Markus Roggenbach and Ana Sokolova, editors, *8th Conference on Algebra and Coalgebra in Computer Science (CALCO 2019)*, volume 139 of *LIPICs*, pages 8:1–8:22, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPICs.CALCO.2019.8.
- 4 Lars Birkedal, Rasmus Ejlers Møgelberg, Jan Schwinghammer, and Kristian Støvring. First steps in synthetic guarded domain theory: Step-indexing in the topos of trees. *LMCS*, 8(4), 2012. doi:10.2168/LMCS-8(4:1)2012.
- 5 Jasmin Christian Blanchette, Aymeric Bouzy, Andreas Lochbihler, Andrei Popescu, and Dmitriy Traytel. Friends with Benefits. In Hongseok Yang, editor, *Programming Languages and Systems*, Lecture Notes in Computer Science, pages 111–140, Berlin, Heidelberg, 2017. Springer. doi:10.1007/978-3-662-54434-1\_5.
- 6 Francis Borceux. *Handbook of Categorical Algebra: Volume 2: Categories and Structures*, volume 2 of *Encyclopedia of Mathematics and Its Applications*. Cambridge University Press, Cambridge, 1994. doi:10.1017/CB09780511525865.
- 7 Helle Hvid Hansen, Clemens Kupke, and Jan Rutten. Stream differential equations: specification formats and solution methods. *arXiv preprint arXiv:1609.08367*, 2016.
- 8 Bart Jacobs. *Introduction to coalgebra*, volume 59. Cambridge University Press, 2017.
- 9 André Joyal, Ross Street, and Dominic Verity. Traced monoidal categories. In *Mathematical proceedings of the cambridge philosophical society*, volume 119, pages 447–468. Cambridge University Press, 1996.

- 10 Max Kelly. *Basic Concepts of Enriched Category Theory*. Number 64 in Lecture Notes in Mathematics. Cambridge University Press, reprints in theory and applications of categories, no. 10 (2005) edition, 1982.
- 11 Saunders Mac Lane. *Categories for the Working Mathematician*. Number 5 in Graduate Texts in Mathematics. Springer, 2 edition, 1998.
- 12 Saunders MacLane. Categorical algebra. *Bulletin of the American Mathematical Society*, 71(1):40–106, 1965.
- 13 Luís Monteiro. Observation systems. *Electronic Notes in Theoretical Computer Science*, 33:261–275, 2000.
- 14 Jurriaan Rot and Damien Pous. Companions, causality and codensity. *Logical Methods in Computer Science*, 15, 2019.
- 15 Jan JMM Rutten. A tutorial on coinductive stream calculus and signal flow graphs. *Theoretical Computer Science*, 343(3):443–481, 2005.
- 16 David Sprunger and Bart Jacobs. The differential calculus of causal functions. *arXiv preprint arXiv:1904.10611*, 2019.



# Aczel-Mendler Bisimulations in a Regular Category

Jérémy Dubut  

National Institute of Advanced Science and Technology, Tokyo, Japan

---

## Abstract

Aczel-Mendler bisimulations are a coalgebraic extension of a variety of computational relations between systems. It is usual to assume that the underlying category satisfies some form of axiom of choice, so that the theory enjoys desirable properties, such as closure under composition. In this paper, we accommodate the definition in a general regular category – which does not necessarily satisfy any form of axiom of choice. We show that this general definition 1) is closed under composition without using the axiom of choice, 2) coincides with other types of coalgebraic formulations under milder conditions, 3) coincides with the usual definition when the category has the regular axiom of choice. We then develop the particular case of toposes, where the formulation becomes nicer thanks to the power-object monad, and extend the formalism to simulations. Finally, we describe several examples in Stone spaces, toposes for name-passing, and modules over a ring.

**2012 ACM Subject Classification** Theory of computation → Models of computation; Theory of computation → Semantics and reasoning

**Keywords and phrases** Regular Categories, Toposes, Bisimulations, Coalgebra

**Digital Object Identifier** 10.4230/LIPIcs.CALCO.2023.19

**Related Version** *Full Version:* <https://arxiv.org/abs/2303.04442>

**Funding** The author is supported by JST, CREST Grant Number JPMJCR22M1, Japan.

## 1 Introduction

Bisimulations and coalgebra have a rich literature and theory (see for example the textbook [13]). They cover a large variety of systems: non-deterministic, probabilistic [8, 7], quantum [1], name-passing [22], Kripke models [5], and so on. The reason for this success is that, if the underlying notions are on very different types of systems, those share common grounds: relations with logic, games, fixpoints, or even some form of decidability that have the same flavour. This suggested that those theories could be abstracted away into a meta-theory that would witness the essence of these common grounds.

In the present paper, we are interested in Aczel-Mendler bisimilarity [2], which defines a bisimulation as an abstract relation (that is, a subobject of a product) which itself carries a structure of coalgebra, and from which we can recover the coalgebra structures of the systems we are comparing by projections. This abstract notion has the privilege to be very close to usual notions of bisimulations in terms of relations, but this comes with the cost that they are too set-flavoured. For example, some basic properties (such as closure under composition, or their relation to bisimulation maps) only hold when the underlying category has some form of axiom of choice.

These issues prevent the usage of Aczel-Mendler bisimulations in some interesting categories. Regular categories, and particularly toposes, are a class of categories which enjoy very nice properties, and particularly that they have a very convenient theory of relations, crucial for abstract bisimulations. However, they do not satisfy the axiom of choice. This is the case for example of the effective topos [12], which encompass in a category concepts such as decidable sets and computable functions, or the topos of nominal sets [18] which models



© Jérémy Dubut;

licensed under Creative Commons License CC-BY 4.0

10th Conference on Algebra and Coalgebra in Computer Science (CALCO 2023).

Editors: Paolo Baldan and Valeria de Paiva; Article No. 19; pp. 19:1–19:18

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

name-passing, or more generally, infinite systems that have some form of decidability. Being able to abstract bisimulations in such categories then becomes crucial, as a possible way to obtain some general decidability results.

The rest of the paper is organised as follows. In Section 2, we recall some necessary knowledge about relations in a general category and allegories, and particularly maps. In Section 3, we recall the definition of Aczel-Mendler bisimulations, and some of their properties that only hold under some form of the axiom of choice. We then extend them to *regular AM-bisimulations* that work nicely in any regular category. In Section 4, we describe a nicer reformulation of regular AM-bisimulations in toposes, thanks to the power-object monad. In Section 5, we extend this nicer formulation to simulations. Finally, in Section 6, we investigate examples of regular AM-bisimulations, for Stone spaces, toposes of name-passing, and for linear weighted systems.

## Contributions

Our contributions can be summarised as follows: 1) An extension of the theory of Aczel-Mendler bisimulations that works in any regular category, without any usage of the axiom of choice. In particular, we prove that the closure under composition (Proposition 27) and the coincidence with other notions of coalgebraic bisimulations (Theorem 29) does not utilise the axiom of choice. 2) A nicer formulation in the case of toposes thanks to the power-object monads, whose connection to tabulations of coalgebra homomorphisms can be proved (Corollary 38), again without the axiom of choice. 3) We extend this nicer formulation to simulations in a topos (Section 5).

## Background and Related Work

Section 2 is a summary of what is needed from the textbook [9] about allegories and particularly allegories of relations. Applications of allegories, and their extensions, to computer science cover fuzzy logic [25], compilation of logic programs [3], and generic programming [4]. Topos theory has a rich literature on different aspects. We recommend [14, 15] for a thorough reference on the matter. Coalgebra theory, and particularly bisimulations for them, also has a recent rich literature. Most of the development in this paper about bisimulations relies on concepts that can be found in the textbook [13]. A careful comparison between various notion of coalgebraic bisimilarities has been done in [23]. Aczel-Mendler bisimulations can be traced back to [2]. Simulations has been studied in the coalgebraic language in [11] for example. The connection with bisimulation and simulation maps in a categorical framework is also the core of the theory of open maps [16, 26].

## Notations

Given two morphisms  $f : X \rightarrow Y$  and  $g : X' \rightarrow Y'$  in a category with binary products, we denote the pairing by  $\langle f, g \rangle : X \rightarrow Y \times Y'$  (if  $X = X'$ ), and the product by  $f \times g : X \times X' \rightarrow Y \times Y'$ .

## 2 Allegory of Relations

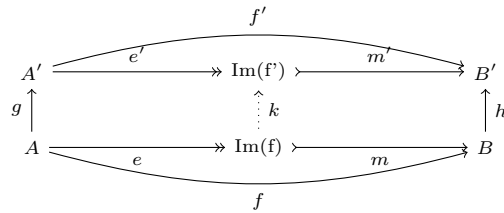
In this section, we cover the general notion of relations in a category, and in particular that they form a tabular allegory. Definitions, propositions, and proofs can be found in [9].

## 2.1 Subobjects and Factorisations

In this paper, subobjects will play a crucial role throughout. Let us then spend some time on their definition. Fix an object  $A$  of  $\mathcal{C}$ . There is a preorder on the class of monos of the form  $m : X \twoheadrightarrow A$  defined by  $m : X \twoheadrightarrow A \sqsubseteq m' : X' \twoheadrightarrow A$  if and only if there is a morphism  $u : X \rightarrow X'$  such that  $m' \cdot u = m$ . In this case,  $u$  is unique and is a mono. A *subobject of  $A$*  is then an equivalence class of monos with  $m : X \twoheadrightarrow A \equiv m' : X' \twoheadrightarrow A$  if  $m \sqsubseteq m'$  and  $m \sqsupseteq m'$ , that is, there are  $u$  and  $u'$  such that  $m' \cdot u = m$  and  $m \cdot u' = m'$ . In this case,  $u$  and  $u'$  are inverse of each other. The preorder on the monos becomes a partial order on subobjects, also denoted by  $\sqsubseteq$ . Throughout the paper, when reasoning on subobjects, we will instead reason on a representing mono. This is harmless when dealing with notions such as pullbacks and factorisations that are unique only up to isomorphisms.

► **Example 1.** In **Set**, since monos are injective functions, subobjects of a set are in bijection with its subsets. The order  $\sqsubseteq$  then corresponds to the usual inclusion  $\subseteq$  of sets.

Given a morphism  $f : A \rightarrow B$ , there is a particular subobject of  $B$  called the *image of  $f$* . In general, it is defined as the smallest (for  $\sqsubseteq$ ) subobject  $\text{Im}(f)$  of  $B$  such that  $f$  can be factorised as  $m \cdot e$ , where  $m$  is any representing mono. The existence of the image is not guaranteed in general. It is however when the category  $\mathcal{C}$  has a nice (epi,mono)-factorisation system, as it is the case for regular categories (and so for toposes). In a regular category, every morphism  $f$  can be uniquely (up to unique isomorphism) factorised as  $m \cdot e$ , where  $m$  is a mono and  $e$  is a regular epi, and furthermore, this factorisation is the image factorisation. In addition, this factorisation is functorial and is preserved by pullbacks, meaning that if we have a commutative diagram of the following form (outer rectangle):



there is a (dotted) morphism that makes the two squares commute, and if the outer rectangle is a pullback, then the rightmost square is also a pullback.

► **Remark 2 (Pullbacks vs. weak-pullbacks).** The preservation of images by pullbacks and the functoriality also imply the preservation of images by weak pullbacks, in the sense that, if the outer rectangle is a weak pullback, then the rightmost square is also a weak pullback.

► **Example 3.** In **Set**, the image of a function is the usual notion of image, that is, the subset  $\{f(a) \mid a \in A\}$  of  $B$ . Since **Set** is regular, and regular epis are surjective functions, the image factorisation is given by the (surjection,injection)-factorisation of the function  $f$ .

## 2.2 Relations in a Regular Category

From now on, let us assume that the category  $\mathcal{C}$  is regular, that is, it has finite limits and a pullback-stable (regular epi,mono)-factorisation as described in the previous section. Everything in this section can be done in locally regular category, but less conveniently. In general:

► **Definition 4.** A relation from  $X$  to  $Y$  is a subobject of  $X \times Y$ .

## 19:4 Aczel-Mendler Bisimulations in a Regular Category

Objects of  $\mathcal{C}$  and relations between them form a category, denoted by  $\mathbf{Rel}(\mathcal{C})$ . The composition is defined as follows. Let  $m_r : R \multimap X \times Y$  and  $m_s : S \multimap Y \times Z$  be two monos, representing two relations,  $r$  from  $X$  to  $Y$  and  $s$  from  $Y$  to  $Z$ . Form the following pullback and (regular epi,mono)-factorisation:

$$\begin{array}{ccc}
 R \star S & \xrightarrow{\mu_2} & S \\
 \mu_1 \downarrow \lrcorner & & \downarrow \pi_1 \cdot m_s \\
 R & \xrightarrow{\pi_2 \cdot m_r} & Y
 \end{array}
 \qquad
 \begin{array}{ccc}
 R \star S & \xrightarrow{\langle \pi_1 \cdot m_r \cdot \mu_1, \pi_2 \cdot m_s \cdot \mu_2 \rangle} & X \times Z \\
 \searrow e_{r;s} & & \nearrow m_{r;s} \\
 & R; S &
 \end{array}$$

The composition  $r; s$  from  $X$  to  $Z$  is then the subobject represented by the mono part  $m_{r;s}$ .

► **Remark 5 (Pullbacks vs. weak pullbacks, continued).** In the definition of the composition, we chose to form a pullback, because we know it exists. However, the definition is unchanged if we take any weak pullback instead.

The *identity relation*  $\Delta_X$  is represented by the diagonal  $\langle \text{id}, \text{id} \rangle : X \multimap X \times X$ .

► **Proposition 6.**  $\mathbf{Rel}(\mathcal{C})$  is a category.

► **Example 7.** In  $\mathbf{Set}$ , the composition of relations is the usual one:

$$R; S = \{(x, z) \in X \times Z \mid \exists y \in Y. (x, y) \in R \wedge (y, z) \in S\},$$

while the identity relation is the usual diagonal  $\Delta_X = \{(x, x) \mid x \in X\}$ .

Of course,  $\mathbf{Rel}(\mathcal{C})$  has much more structure. First, since subobjects are naturally ordered by  $\sqsubseteq$ , and that this order is compatible with the composition,  $\mathbf{Rel}(\mathcal{C})$  has a structure of locally ordered 2-category. Furthermore, it comes equipped with an anti-involution which makes it into a *dagger 2-poset*. This means there is a functor  $(\_)^\dagger : \mathbf{Rel}(\mathcal{C})^{op} \rightarrow \mathbf{Rel}(\mathcal{C})$  such that for every relation  $R$ ,  $R^{\dagger\dagger} = R$  and for every other relation  $S$  with  $R \sqsubseteq S$ ,  $R^\dagger \sqsubseteq S^\dagger$ . This involution is given by the inverse of a relation, as follows. If the relation  $r$  is represented by the mono  $m_r : R \multimap X \times Y$ , then  $r^\dagger$  is represented by  $m_{r^\dagger} = \langle \pi_2, \pi_1 \rangle \cdot m_r : R \multimap Y \times X$ . Finally, the meet of two relations for the partial order  $\sqsubseteq$  is defined and is called the intersection. Given  $m_r : R \multimap X \times Y$  and  $m_s : S \multimap X \times Y$  representing  $r$  and  $s$  respectively, the intersection  $r \cap s$  is then represented by the pullback of  $m_r$  and  $m_s$ . Altogether:

► **Theorem 8.**  $\mathbf{Rel}(\mathcal{C})$  is an allegory, meaning that all this data satisfies the modular law:

$$(R; S) \cap T \sqsubseteq (R \cap (T; S^\dagger)); S.$$

► **Example 9.** In  $\mathbf{Set}$ ,  $R^\dagger$  is the usual inverse of the relation  $R$ :  $R^\dagger = \{(y, x) \mid (x, y) \in R\}$ . The intersection  $\cap$  is the intersection of relations as sets. Finally, the modular law is trivial in  $\mathbf{Rel}(\mathbf{Set})$ . More generally, this law is crucial to make adjoints in an allegory behave like direct/inverse images, (see next section, and the Frobenius reciprocity [17]).

### 2.3 Maps in Allegories

From an allegory (intuitively of relations), it is possible to recover the morphisms of the original category through the notion of *maps*. In a general allegory  $\mathcal{A}$ , a map is a morphism which is a left adjoint (in the 2-categorical sense). Maps form a subcategory of  $\mathcal{A}$  denoted by  $\mathbf{Map}(\mathcal{A})$ . In the case of an allegory of relations:

► **Theorem 10.**  $\mathbf{Map}(\mathbf{Rel}(\mathcal{C}))$  is isomorphic to  $\mathcal{C}$ .



The reason for it is that maps (left adjoints) in  $\mathbf{Rel}(\mathcal{C})$  are precisely the relations represented by a mono of the form  $\langle \text{id}, f \rangle$  for some morphism  $f$  of  $\mathcal{C}$ , justifying the remark from Example 9 that left adjoints in an allegory behave like direct images. Similarly, their right adjoints are relations represented by  $\langle f, \text{id} \rangle$ , corresponding to inverse images. This also implies that  $\mathbf{Rel}(\mathcal{C})$  is *tabular*, that is, it is generated by maps in the following sense. A *tabulation* of a morphism  $\phi : X \rightarrow Y$  in an allegory is a pair of maps  $\psi : Z \rightarrow X$  and  $\xi : Z \rightarrow Y$  such that  $\phi = \psi^\dagger; \xi$  and  $\psi; \psi^\dagger \cap \xi; \xi^\dagger = \text{id}_Z$ .

► **Theorem 11.** *In an allegory of relations, the tabulations of a relation  $R$  are exactly those pairs of relations  $(S, T)$  represented by monos of the form  $\langle \text{id}, f \rangle$  and  $\langle \text{id}, g \rangle$  respectively, with  $f$  and  $g$  jointly monic, and such that  $R = S^\dagger; T$ . In particular, every relation has a tabulation, that is,  $\mathbf{Rel}(\mathcal{C})$  is tabular.*

The intuition of this theorem is that relations are precisely jointly monic spans.

► **Example 12.** In  $\mathbf{Set}$ , maps are graphs of functions, that is, relations of the form  $\{(x, f(x)) \mid x \in X\}$  for some function  $f : X \rightarrow Y$ . Consequently, every relation  $R$  is the same as the span of  $f : R \rightarrow X (x, y) \mapsto x$  and  $g : R \rightarrow Y (x, y) \mapsto y$ , that is,  $R = \{(f(r), g(r)) \mid r \in R\}$ .

### 3 Aczel-Mendler Bisimulations, in Regular Categories

We now start investigating our original problem: a nice general theory of bisimulations in terms of relations. The development of this section will start with the notion of Aczel-Mendler bisimulations [2], where systems are described as coalgebras. We will witness that one bottleneck of this theory is the role of the axiom of choice that is necessary to prove even some basic properties of this notion of bisimulation. This prevents to use this notion in most regular categories. We will then show that we can fix this issue by a careful use of relations.

#### 3.1 Systems as Coalgebras

In this section, we briefly recall coalgebras, and how to model systems with them. For a more complete introduction, see for example [13].

Coalgebras require two ingredients: a category  $\mathcal{C}$  that describes the type of state spaces of our systems and an endofunctor  $F$  on  $\mathcal{C}$  that describes the type of allowed transitions. A *coalgebra* is then a morphism of type  $\alpha : X \rightarrow FX$ . Intuitively,  $X$  is the state space of the system and  $\alpha$  maps a state to the collection of transitions from this state.

► **Example 13.** For example, deterministic transition systems labelled in the alphabet  $\Sigma$  can be modelled with the  $\mathbf{Set}$ -functor  $X \mapsto \Sigma \Rightarrow X$ , mapping  $X$  to the set of functions from  $\Sigma$  to  $X$ . A coalgebra for this functor is a function  $X \rightarrow \Sigma \Rightarrow X$ . It maps a state to a function from  $\Sigma$  to  $X$ , describing what is the next state after reading a particular letter. Non-deterministic labelled transition systems can be described using the functor  $X \mapsto \mathcal{P}(\Sigma \times X)$ . A coalgebra then maps a state to a set of transitions, given by a letter and a state, describing the states we can reach from another state reading a particular letter. Another typical example are probabilistic systems, that can be described using the distribution functor  $\mathcal{D}$ . A transition for those systems is then a distribution on the states, describing what is the probability to reach a state in the next step.

A *morphism of coalgebras* from  $\alpha : X \rightarrow FX$  to  $\beta : Y \rightarrow FY$  is a morphism  $f : X \rightarrow Y$  of  $\mathcal{C}$  such that  $\beta \cdot f = Ff \cdot \alpha$ . Coalgebras on  $F$  and morphisms of coalgebras form a category, which we denote by  $\mathbf{CoAlg}(F)$ .

### 3.2 Aczel-Mendler Bisimulations of Coalgebras

In this section, we follow closely the development of [13]. We recall the definition of Aczel-Mendler bisimulations and give some of their properties.

► **Definition 14.** We say that a relation is an Aczel-Mendler bisimulation (AM-bisimulation for short) from the coalgebra  $\alpha : X \rightarrow FX$  to  $\beta : Y \rightarrow FY$ , if for any mono  $r : R \rightarrow X \times Y$  representing it, there is a morphism  $W : R \rightarrow FR$ , called witness, such that:

$$\begin{array}{ccccc}
 & & X \times Y & \xrightarrow{\alpha \times \beta} & F(X) \times F(Y) \\
 R & \xrightarrow{r} & & & \uparrow \langle F\pi_1, F\pi_2 \rangle \\
 & \searrow W & FR & \xrightarrow{Fr} & F(X \times Y)
 \end{array}$$

► **Example 15.** In the case of non-deterministic labelled transition systems, AM-bisimulations correspond to usual strong bisimulations. The function  $W$  maps a pair  $(x, y)$  of states of  $\alpha$  and  $\beta$  to a subset of triples  $(a, x', y')$  such that  $(a, x') \in \alpha(x)$ ,  $(a, y') \in \beta(y)$ , and  $(x', y') \in R$ . The commutation means that the set of transitions  $\alpha(x)$  from  $x$  exactly corresponds to the set  $\{(a, x') \mid \exists y'. (a, x', y') \in W(x, y)\}$ , and similarly for  $y$ . This implies the property of a bisimulation: if there is a transition  $(a, x')$  from  $x$ , then there is a transition  $(a, y')$  from  $y$  with  $(x', y') \in R$ ; and vice versa.

We show now that AM-bisimulations behave well under the regular axiom of choice:

► **Proposition 16.** Assume that  $\mathcal{C}$  has the regular axiom of choice, that is, every regular epi is split, and that  $F$  preserves weak pullbacks. Then the following is a dagger  $\mathcal{Q}$ -poset, denoted by  $\mathbf{Bis}(F)$ : objects are coalgebras on  $F$ , morphisms are AM-bisimulations,  $\sqsubseteq$ , identities, composition, and  $(\_)\dagger$  are defined as in  $\mathbf{Rel}(\mathcal{C})$ . That is, diagonals are AM-bisimulations, and AM-bisimulations are closed under composition and inverse.

**Proof.** Let us focus on proving that Aczel-bisimulations are closed under composition. We then have two witnesses:

$$\begin{array}{ccc}
 R_1 & \begin{array}{ccc} \xrightarrow{r_1} & X \times Y & \xrightarrow{\alpha \times \beta} \\ & \searrow W_1 & \downarrow \langle F\pi_1, F\pi_2 \rangle \\ & FR_1 & \xrightarrow{Fr_1} \end{array} & F(X) \times F(Y) \\
 & & & \uparrow \langle F\pi_1, F\pi_2 \rangle \\
 & & & F(X \times Y)
 \end{array}
 \qquad
 \begin{array}{ccc}
 R_2 & \begin{array}{ccc} \xrightarrow{r_2} & Y \times Z & \xrightarrow{\beta \times \gamma} \\ & \searrow W_2 & \downarrow \langle F\pi_1, F\pi_2 \rangle \\ & FR_2 & \xrightarrow{Fr_2} \end{array} & F(Y) \times F(Z) \\
 & & & \uparrow \langle F\pi_1, F\pi_2 \rangle \\
 & & & F(Y \times Z)
 \end{array}$$

We then want to construct a morphism  $W : R_1; R_2 \rightarrow F(R_1; R_2)$  such that

$$\begin{array}{ccccc}
 & & X \times Z & \xrightarrow{\alpha \times \gamma} & F(X) \times F(Z) \\
 R_1; R_2 & \xrightarrow{r_1; r_2} & & & \uparrow \langle F\pi_1, F\pi_2 \rangle \\
 & \searrow W & F(R_1; R_2) & \xrightarrow{F(r_1; r_2)} & F(X \times Z)
 \end{array}$$

Since  $F$  preserves weak pullbacks and by definition of composition, we have the following weak pullback and (regular epi,mono)-factorisation:

$$\begin{array}{ccc}
 F(R_1 \star R_2) & \xrightarrow{F\mu_2} & FR_2 \\
 F\mu_1 \downarrow \dashv \lrcorner & & \downarrow F(\pi_1 \cdot r_2) \\
 FR_1 & \xrightarrow{F(\pi_2 \cdot r_1)} & FY
 \end{array}
 \qquad
 \begin{array}{ccc}
 R_1 \star R_2 & \xrightarrow{\langle \pi_1 \cdot r_1 \cdot \mu_1, \pi_2 \cdot r_2 \cdot \mu_2 \rangle} & X \times Z \\
 e_{r_1; r_2} \searrow & & \nearrow r_1; r_2 \\
 & R_1; R_2 &
 \end{array}$$

Denote by  $s$  the section of  $e_{r_1; r_2}$ , which exists by the regular axiom of choice. By the universal property of weak pullbacks, we have  $\phi : R_1; R_2 \longrightarrow F(R_1 \star R_2)$ , such that

$$\begin{array}{ccc}
 R_1; R_2 & \xrightarrow{W_2 \cdot \mu_2 \cdot s} & F R_2 \\
 \downarrow \phi & \searrow F \mu_2 & \downarrow F(\pi_1 \cdot r_2) \\
 F(R_1 \star R_2) & \xrightarrow{F \mu_2} & F R_2 \\
 \downarrow F \mu_1 & & \downarrow F(\pi_1 \cdot r_2) \\
 F R_1 & \xrightarrow{F(\pi_2 \cdot r_1)} & F Y
 \end{array}$$

Now  $W = F e_{r_1; r_2} \cdot \phi$  is the expected witness:

$$\begin{aligned}
 \langle F \pi_1, F \pi_2 \rangle \cdot F(r_1; r_2) \cdot W &= \langle F \pi_1, F \pi_2 \rangle \cdot F(r_1; r_2) \cdot F e_{r_1; r_2} \cdot \phi && \text{(definition of } W) \\
 &= \langle F \pi_1, F \pi_2 \rangle \cdot F \langle \pi_1 \cdot r_1 \cdot \mu_1, \pi_2 \cdot r_2 \cdot \mu_2 \rangle \cdot \phi && \text{(definition of } r_1; r_2) \\
 &= F(\pi_1 \cdot r_1) \times F(\pi_2 \cdot r_2) \cdot \langle F(\mu_1) \cdot \phi, F(\mu_2) \cdot \phi \rangle && \text{(computation on products)} \\
 &= F(\pi_1 \cdot r_1) \times F(\pi_2 \cdot r_2) \cdot \langle W_1 \cdot \mu_1 \cdot s, W_2 \cdot \mu_2 \cdot s \rangle && \text{(definition of } \phi) \\
 &= \alpha \times \gamma \cdot \langle \pi_1 \cdot r_1 \cdot \mu_1, \pi_2 \cdot r_2 \cdot \mu_2 \rangle \cdot s && \text{(definition of the } W_i \text{ and computation on products)} \\
 &= \alpha \times \gamma \cdot (r_1; r_2) && \text{(definition of } s)
 \end{aligned}$$

◀

► **Remark 17.** The preservation of weak pullbacks is a crucial property for a functor related to relations. More surprisingly, the dependence on the axiom of choice is necessary for proving the closure under composition. This was already observed in [13, 23].

In the proof we make the following usage of the regular axiom of choice: we need that the epi part  $e_{r_1; r_2} : R_1 \star R_2 \twoheadrightarrow R_1; R_2$  of a (regular epi,mono)-factorisation to be split, that is, has a section  $s : R_1; R_2 \succrightarrow R_1 \star R_2$ . In **Set**,  $R_1 \star R_2$  is given by triples  $(x, y, z)$  such that  $(x, y) \in R_1$  and  $(y, z) \in R_2$ , so this section is then a choice of such a  $y$  for every  $(x, z)$  in the composition. This kind of choice is usual for example to prove that strong bisimulations are closed under composition: assuming that one has a transition  $(a, x')$  from  $x$ , to prove that one also has such a transition from  $z$ , one should pick an intermediate  $y$ , prove that there is such a transition for  $y$  using that  $R_1$  is a bisimulation, then concluding using the fact that  $R_2$  is a bisimulation.

In this dagger 2-poset of bisimulations, we can also talk about maps and tabulations, as we did for relations. Furthermore, since the 2-categorical structure of  $\mathbf{Bis}(F)$  is given by that of  $\mathbf{Rel}(\mathcal{C})$ , and particularly that the local posets of bisimulations are embedded in the corresponding local poset of relations, results from Section 2.3 can be used here. In particular, we can prove the following:

► **Theorem 18.**  $\mathbf{Map}(\mathbf{Bis}(F))$  is isomorphic to  $\mathbf{CoAlg}(F)$ .

Using results from Section 2.3, proving this theorem boils down to proving that bisimulations that are maps are precisely graphs of coalgebra morphisms:

► **Proposition 19.** A morphism  $h : X \longrightarrow Y$  of  $\mathcal{C}$  is a coalgebra morphism from  $\alpha$  to  $\beta$  if and only if the mono  $\langle id, h \rangle : X \succrightarrow X \times Y$  represents an AM-bisimulation from  $\alpha$  to  $\beta$ .

Using this characterisation of maps for AM-bisimulations, and using the tabularity of the allegory of relations, we can prove that an AM-bisimulation can be described as a span of

morphism of coalgebras, under some form of axiom of choice (see [13]). We can formulate this in terms of tabulations:

► **Proposition 20.** *If  $U$  is an AM-bisimulation from  $\alpha$  to  $\beta$ , and if  $f : Z \rightarrow X$ ,  $g : Z \rightarrow Y$  is a tabulation of  $U$ , then there is a coalgebra structure  $\gamma$  on  $Z$  such that  $f$  is a coalgebra morphism from  $\gamma$  to  $\alpha$  and  $g$  is a coalgebra morphism from  $\gamma$  to  $\beta$ .*

► **Corollary 21.** *Assume  $\mathcal{C}$  has the regular axiom of choice. Assume given two coalgebras  $\alpha : X \rightarrow F(X)$  and  $\beta : Y \rightarrow F(Y)$ , and two points  $p : * \rightarrow X$  and  $q : * \rightarrow Y$ . There is an AM-bisimulation  $r : R \rightarrow X \times Y$  from  $\alpha$  to  $\beta$ , and a point  $c : * \rightarrow R$  such that  $r \cdot c = \langle p, q \rangle$  if and only if there is a span  $X \xleftarrow{f} Z \xrightarrow{g} Y$ , an  $F$ -coalgebra structure  $\gamma$  on  $Z$  such that  $f$  is a coalgebra morphism from  $\gamma$  to  $\alpha$  and  $g$  from  $\gamma$  to  $\beta$ , and a point  $w : * \rightarrow Z$  such that  $f \cdot w = p$  and  $g \cdot w = q$ .*

► **Remark 22.** Here  $*$  is usually the final object of  $\mathcal{C}$ , but it can be any object used to describe initial states in the systems under consideration.

### 3.3 Picking vs. Collecting: AM-Bisimulations for Regular Categories

We have seen that several results about AM-bisimulations depend on the regular axiom of choice, preventing its use in more exotic toposes and regular categories. Actually, the only occurrences are of similar flavour: one wants to prove some property of elements  $(x, z)$  in a composition of relations, and for that, one has to pick a witness  $y$  in between. The main idea of our proposal is that, instead of picking a witness (which would require the axiom of choice), it is enough to collect all the witnesses, prove properties about all of them, and make sure that there is enough of them. This can be done in any regular category as follows:

► **Definition 23.** *We say that a relation is a regular AM-bisimulation from the coalgebra  $\alpha : X \rightarrow FX$  to  $\beta : Y \rightarrow FY$ , if for any mono  $r : R \rightarrow X \times Y$  representing it, there is another relation represented by  $w : W \rightarrow FR \times R$  such that  $\pi_2 \circ w$  is a regular epi and:*

$$\begin{array}{ccccc}
 \pi_2 \circ w & \rightarrow & R & \xrightarrow{r} & X \times Y & \xrightarrow{\alpha \times \beta} & F(X) \times F(Y) \\
 & \searrow & & & & & \\
 W & & & & & & \\
 & \swarrow & & & & & \\
 \pi_1 \circ w & \rightarrow & FR & \xrightarrow{Fr} & F(X \times Y) & \xrightarrow{\langle F\pi_1, F\pi_2 \rangle} & F(X) \times F(Y)
 \end{array}$$

The intuition is as follows:  $W$  collects all the witnesses that  $R$  is a bisimulation. In particular, for a given pair  $(x, y)$  in  $R$ , there might be several witnesses. The fact  $\pi_2 \circ w$  is a regular epi guarantees that every pair of  $R$  has at least one witness. Of course, we have to prove that this extends plain AM-bisimulations:

► **Proposition 24.** *If  $\mathcal{C}$  is a regular category with the regular axiom of choice, then a relation is a regular AM-bisimulation if and only if it is a AM-bisimulation.*

Also, regular bisimulations are closed under composition. This requires a milder condition on  $F$  as already observed in [23].

► **Definition 25.** *We say that  $F$  covers pullbacks if for every pair of pullbacks:*

$$\begin{array}{ccc}
 R & \xrightarrow{v} & Y \\
 u \downarrow \lrcorner & & \downarrow g \\
 X & \xrightarrow{f} & Z
 \end{array}
 \qquad
 \begin{array}{ccc}
 R' & \xrightarrow{v'} & FY \\
 u' \downarrow \lrcorner & & \downarrow Fg \\
 FX & \xrightarrow{Ff} & FZ
 \end{array}$$

the unique morphism  $\gamma : FR \rightarrow R'$  such that  $u' \circ \gamma = Fu$  and  $v' \circ \gamma = Fv$  is a regular epi.

► **Remark 26.** When  $F$  preserves weak pullbacks, then  $F$  covers pullbacks. When  $\mathcal{C}$  has the regular axiom of choice, then both notions coincide.

► **Proposition 27.** When  $F$  covers pullbacks, then regular AM-bisimulations are closed under composition.

In [23], Staton described conditions for several coalgebraic notions of bisimulations to coincide. In this picture, AM-bisimulations were quite weak, as they would coincide with other notions only under some form of axiom of choice (again). Here, we will show that the picture is much nicer with regular AM-bisimulations.

► **Definition 28.** A relation from  $X$  to  $Y$  is a Hermida-Jacobs bisimulation (HJ-bisimulation for short) from  $\alpha : X \rightarrow FX$  to  $\beta : Y \rightarrow FY$  if there is a mono  $r : R \rightarrow X \times Y$  representing it and a morphism  $w : R \rightarrow \overline{FR}$  where  $\overline{FR}$  is obtained by the (epi,mono)-factorisation on the left, and such that the square on the right commutes:

$$\begin{array}{ccc}
 FR & \xrightarrow{\langle F\pi_1, F\pi_2 \rangle \cdot Fr} & FX \times FY \\
 & \searrow e_r & \nearrow m_r \\
 & & \overline{FR}
 \end{array}
 \qquad
 \begin{array}{ccc}
 R & \xrightarrow{w} & \overline{FR} \\
 r \downarrow & & \downarrow m_r \\
 X \times Y & \xrightarrow{\alpha \times \beta} & FX \times FY
 \end{array}$$

A relation is a behavioural equivalence from  $\alpha : X \rightarrow FX$  to  $\beta : Y \rightarrow FY$  if it is represented by a pullback of coalgebra homomorphisms, that is, if there are a coalgebra  $\gamma : Z \rightarrow FZ$  and two coalgebra homomorphisms  $f : \alpha \rightarrow \gamma$  and  $g : \beta \rightarrow \gamma$  such that the mono  $\langle u, v \rangle : R \rightarrow X \times Y$  obtained from their pullback in  $\mathcal{C}$  represents it.

$$\begin{array}{ccc}
 R & \xrightarrow{v} & Y \\
 u \downarrow & \lrcorner & \downarrow g \\
 X & \xrightarrow{f} & Z
 \end{array}$$

- **Theorem 29.** Assume that  $\mathcal{C}$  is a regular category. Then:
- a relation is a regular AM-bisimulation if and only if it is a HJ-bisimulation,
  - if  $\mathcal{C}$  has pushouts, then a regular AM-bisimulation is included in a behavioural equivalence,
  - if  $F$  covers pullbacks, then a behavioural equivalence is a regular AM-bisimulation.

The last two bullets are a consequence of the first bullet and [23]. In Section 3.2, we described that AM-bisimilarity coincides with the existence of a span of coalgebra homomorphisms. This can also be formulated in the context of regular AM-bisimulations. The witness  $w : W \rightarrow FR \times R$  can be seen as a coalgebra in  $\mathbf{Rel}(\mathcal{C})$  (although  $F$  is technically not a functor on it). The coalgebra  $\alpha : X \rightarrow FX$  can also be seen as a coalgebra in  $\mathbf{Rel}(\mathcal{C})$  as  $\langle \alpha, \text{id} \rangle : X \rightarrow FX \times X$ . Then  $\pi_1 \circ r$  can be seen as a coalgebra homomorphism from  $w$  to  $\alpha$ , since the following diagram commutes

$$\begin{array}{ccc}
 W & \xrightarrow{w} & FR \times R \\
 \pi_1 \cdot r \cdot \pi_2 \cdot w \downarrow & & \downarrow F(\pi_1 \circ r) \times \pi_1 \circ r \\
 X & \xrightarrow{\langle \alpha, \text{id} \rangle} & FX \times X
 \end{array}$$

## 4 The Case of Toposes

Here, we investigate the particular case of toposes. The first part of this section recalls folklore about toposes and particularly power-objects, namely, that they form a commutative monad whose Kleisli category is isomorphic to the category of relations. Finally, we will show that regular AM-bisimulations can be formulated much more nicely in this context.

### 4.1 Toposes, as Relation Classifiers

► **Definition 30.** *A topos is a finitely complete category with power objects. The latter condition means that for every object  $X$ , there is a mono  $\in_X: E_X \rightarrow X \times \mathcal{P}X$  such that for every mono of the form  $m: R \rightarrow X \times Y$  there is a unique morphism  $\xi_m: Y \rightarrow \mathcal{P}X$  such that there is a pullback diagram of the form:*

$$\begin{array}{ccc} R & \xrightarrow{\theta_m} & E_X \\ m \downarrow \lrcorner & & \downarrow \in_X \\ X \times Y & \xrightarrow{id \times \xi_m} & X \times \mathcal{P}X \end{array}$$

This formulation passes to relations since  $\xi_m = \xi_{m'}$  if and only if  $m$  and  $m'$  represent the same relation  $r$ . In that case, we will write  $\xi_r$  for  $\xi_m = \xi_{m'}$ . Another formulation of toposes uses sub-object classifiers which can be recovered as  $\mathbb{T} = \in_{\mathbf{1}}: \mathbf{1} \simeq E_{\mathbf{1}} \rightarrow \mathbf{1} \times \mathcal{P}\mathbf{1} \simeq \mathcal{P}\mathbf{1} = \Omega$ . The formulation by power-objects implies that a topos is cartesian closed, which is not the case of the sub-object classifier alone. Conversely,  $\mathcal{P}X$  is equal to  $\Omega^X$  and  $\in_X$  is any mono corresponding to the evaluation morphism  $X \times \Omega^X \rightarrow \Omega$  of the cartesian-closed structure.

► **Example 31.** In **Set**,  $\mathcal{P}X$  is given by the usual power-set and  $E_X$  is the subset of  $X \times \mathcal{P}X$  consisting of pairs  $(x, U)$  such that  $x \in U$ . In **Scha** – the Schanuel topos **Scha** [18], equivalent to the category of nominal sets and equivariant functions –  $\mathcal{P}X$  is the nominal set of finitely supported subsets of  $X$ . In **Eff** – the effective topos [12], intuitively, the category of effective set and computable functions –  $\mathcal{P}X$  is intuitively given by the set of decidable subsets of  $X$  (although the formal description is much harder).

### 4.2 The Power-Object Monad

The following is a folklore result about power-objects, that can be proved for example by noticing that the proof in **Set** does not use either the law of excluded-middle nor the axiom of choice and the fact that any such statement is true in any topos:

► **Theorem 32.** *In a topos  $\mathcal{C}$ ,  $\mathcal{P}$  extends to a commutative monad whose Kleisli category is isomorphic to the category of relations  $\mathbf{Rel}(\mathcal{C})$ .*

Let us describe some parts of this statement that will be useful in the following discussion. First, the structure of *covariant* functor (not to be confused with the more traditional contravariant structure) is given as follows. Given a morphism  $f: X \rightarrow Y$ ,  $\mathcal{P}f: \mathcal{P}X \rightarrow \mathcal{P}Y$  is defined as follows. Consider first the following (epi,mono)-factorisation:

$$\begin{array}{ccc} E_X & \xrightarrow{(f \times id) \cdot \in_X} & Y \times \mathcal{P}X \\ & \searrow e_f & \nearrow m_f \\ & E_f & \end{array}$$

Then  $\mathcal{P}f: \mathcal{P}X \rightarrow \mathcal{P}Y$  is the unique morphism corresponding to  $m_f$ .

The unit  $\eta_X : X \rightarrow \mathcal{P}X$  is defined as  $\xi_{\Delta_X}$ , that is, the unique morphism such that there is a pullback of the form:

$$\begin{array}{ccc} X & \xrightarrow{\theta_X} & X \\ \langle \text{id}, \text{id} \rangle \downarrow \lrcorner & & \downarrow \in_X \\ X \times X & \xrightarrow{\text{id} \times \eta_X} & X \times \mathcal{P}X \end{array}$$

for some  $\theta_X$ . The multiplication  $\mu_X : \mathcal{P}\mathcal{P}X \rightarrow \mathcal{P}X$  is defined as the unique morphism associated with the composition of relations  $\in_X; \in_{\mathcal{P}X}$ .

### 4.3 AM-Bisimulations in a Topos

Since toposes are regular categories, the notion of regular AM-bisimulation makes sense. We show here that it can be reformulated as follows.

► **Definition 33.** *We say that a relation is a toposal AM-bisimulation from the coalgebra  $\alpha : X \rightarrow FX$  to  $\beta : Y \rightarrow FY$ , if for any mono  $r : R \rightarrow X \times Y$  representing it, there is a morphism  $W : R \rightarrow \mathcal{P}FR$  such that:*

$$\begin{array}{ccccc} & & X \times Y & \xrightarrow{\alpha \times \beta} & F(X) \times F(Y) \\ & \nearrow r & & & \searrow \eta_{F(X)} \times \eta_{F(Y)} \\ R & & & & \mathcal{P}F(X) \times \mathcal{P}F(Y) \\ & \searrow W & & & \nearrow (\mathcal{P}F\pi_1, \mathcal{P}F\pi_2) \\ & & \mathcal{P}FR & \xrightarrow{\mathcal{P}Fr} & \mathcal{P}F(X \times Y) \end{array}$$

In other words, an  $F$ -toposal AM-bisimulation between  $\alpha$  and  $\beta$  is a  $\mathcal{P}F$ -AM-bisimulation between  $\eta \cdot \alpha$  and  $\eta \cdot \beta$ . Intuitively, this means that toposal bisimulations look at systems as non-deterministic. This allows to *collect* witnesses as a morphism  $W : R \rightarrow \mathcal{P}FR$  instead of picking some, very much like regular AM-bisimulations.

We have to make sure that toposal and regular AM-bisimulations coincide.

► **Proposition 34.** *Assume that  $\mathcal{C}$  is a topos. Then for every relation  $U$  from  $X$  to  $Y$ , every coalgebra  $\alpha : X \rightarrow FX$  and  $\beta : Y \rightarrow FY$ ,  $U$  is a toposal AM-bisimulation from  $\alpha$  to  $\beta$  if and only if it is a regular AM-bisimulation between them.*

This nicer formulation allows us to prove a much nicer tabularity property, which could only be informally described for regular AM-bisimulations:

► **Proposition 35.** *Assume that  $\mathcal{C}$  is a topos and that  $F$  covers pullbacks. Then the following is a dagger 2-poset: objects are coalgebras on  $F$ , morphisms are toposal AM-bisimulations,  $\sqsubseteq$ , identities, composition, and  $(\_)^\dagger$  are defined as in **Rel**( $\mathcal{C}$ ).*

► **Remark 36.** This Proposition is similar to Proposition 16, without the axiom of choice and assuming only that  $F$  covers pullbacks, but by replacing plain AM-bisimulations by toposal AM-bisimulations.

Obviously, the category of maps of the dagger 2-poset of toposal bisimulations is then not isomorphic to **CoAlg**( $F$ ), but to the category of  $F$ -coalgebras with  $\mathcal{P}F$ -coalgebra morphisms between them. Then tabularity can be formulated as follows:

► **Proposition 37.** *If  $U$  is a toposal bisimulation from the  $F$ -coalgebra  $\alpha$  to the  $F$ -coalgebra  $\beta$ , and if  $f : Z \rightarrow X$ ,  $g : Z \rightarrow Y$  is a tabulation of  $U$ , then there is a  $\mathcal{P}F$ -coalgebra structure  $\gamma$  on  $Z$  such that  $f$  is a  $\mathcal{P}F$ -coalgebra morphism from  $\gamma$  to  $\eta_X \cdot \alpha$  and  $g$  is a  $\mathcal{P}F$ -coalgebra morphism from  $\gamma$  to  $\eta_Y \cdot \beta$ .*

## 19:12 Aczel-Mendler Bisimulations in a Regular Category

► **Corollary 38.** *Assume given two coalgebras  $\alpha : X \rightarrow F(X)$  and  $\beta : Y \rightarrow F(Y)$ , and two points  $p : * \rightarrow X$  and  $q : * \rightarrow Y$ . There is a topological bisimulation  $r : R \rightarrow X \times Y$  from  $\alpha$  to  $\beta$ , and a point  $c : * \rightarrow R$  such that  $r \cdot c = \langle p, q \rangle$  if and only if there is a span  $X \xleftarrow{f} Z \xrightarrow{g} Y$ , a  $\mathcal{P}F$ -coalgebra structure  $\gamma$  on  $Z$ , and a point  $w : * \rightarrow Z$  such that  $f$  is a  $\mathcal{P}F$ -coalgebra morphism from  $\gamma$  to  $\eta_X \cdot \alpha$ ,  $g$  from  $\gamma$  to  $\eta_Y \cdot \beta$ ,  $f \cdot w = p$ , and  $g \cdot w = q$ .*

### 5 From Bisimulations to Simulations

In this section, we extend the analysis of the previous sections to deal with *simulations*. Classically, simulations for coalgebras require a notion of order on morphisms of the form  $X \rightarrow FY$ , to allow one to define that there is fewer transitions coming out of a state than another. This allows to easily modify the definition of AM-bisimulations to obtain *AM-simulations*. We will show that topological bisimulations can also be extended to simulations in a nice way to mitigate those issues. The only reason we chose to stay in a topos and not in a general regular category is because theorems have a nicer formulation there, but most of the discussion here can be done in a regular category.

#### 5.1 Order-Structure on Functors, and Lax Coalgebra Morphisms

We want to be able to compare two morphisms of the form  $X \rightarrow FY$ . So assuming a preorder  $\leq$  on each Hom-set  $\mathcal{C}(X, FY)$ , we can define *lax morphisms of coalgebras*, as follows:

► **Definition 39.** *A lax morphism of coalgebras from  $\alpha : X \rightarrow FX$  to  $\beta : Y \rightarrow FY$  is a morphism  $f : X \rightarrow Y$  of  $\mathcal{C}$  such that  $Ff \cdot \alpha \leq \beta \cdot f$  in  $\mathcal{C}(X, FY)$ .*

Unfortunately, coalgebras and lax morphisms of coalgebras do not form a category in general, and some axioms are required for the interaction of  $\leq$  with the composition.

► **Definition 40.** *A good order structure on  $F$  is a preorder  $\leq$  on each  $\mathcal{C}(X, FY)$  such that: 1) if  $\alpha \leq \beta$  in  $\mathcal{C}(X, FY)$ ,  $f : X' \rightarrow X$ , and  $g : Y \rightarrow Y'$ , then  $Fg \cdot \alpha \cdot f \leq Fg \cdot \beta \cdot f$  in  $\mathcal{C}(X', FY')$ ; 2) if  $h : X \rightarrow FZ$ ,  $k : X \rightarrow FY$ ,  $g : Y \rightarrow Z$ , and  $h \leq Fg \cdot k$  in  $\mathcal{C}(X, FZ)$ , then there is  $k' : X \rightarrow FY$  such that  $k' \leq k$  in  $\mathcal{C}(X, FY)$  and  $h = Fg \cdot k'$ .*

► **Lemma 41.** *When  $\leq$  is a good order structure on  $F$ , then coalgebras and lax morphisms of coalgebras form a category, denoted by  $\mathbf{CoAlg}_{\text{lax}}(F)$ .*

► **Example 42.** When  $F$  is the functor modelling non-deterministic labelled systems and  $\leq$  is given by point-wise inclusion, lax morphisms are exactly morphisms in the sense of [16]. Those morphisms are intuitively morphisms whose graphs are simulations. More generally, we will see that lax morphisms are simulation maps.

#### 5.2 AM-Simulations

► **Definition 43.** *We say that a relation is an AM-simulation from the coalgebra  $\alpha : X \rightarrow FX$  to  $\beta : Y \rightarrow FY$ , if for any mono  $r : R \rightarrow X \times Y$  representing it, there is a morphism  $W : R \rightarrow FR$  such that:*

$$\begin{array}{ccccc}
 & & X \times Y & \xrightarrow{\alpha \times \beta} & F(X) \times F(Y) \\
 & \nearrow r & & & \uparrow \langle F\pi_1, F\pi_2 \rangle \\
 R & & & \leq \times \geq & \\
 & \searrow W & FR & \xrightarrow{Fr} & F(X \times Y)
 \end{array}$$

meaning that  $\alpha \cdot \pi_1 \cdot r \leq F\pi_1 \cdot Fr \cdot W$  and  $\beta \cdot \pi_2 \cdot r \geq F\pi_2 \cdot Fr \cdot W$ .



► **Proposition 44.** *When  $\leq$  is a good order structure, it is equivalent to require that the left inequality is actually an equality  $\alpha \cdot \pi_1 \cdot r = F\pi_1 \cdot Fr \cdot W$ .*

► **Example 45.** When  $F : X \mapsto \mathcal{P}(\Sigma \times X)$ , AM-simulations correspond to strong simulations. The left part of the commutativity means that for every  $(x, y) \in R$  and  $(a, x') \in \alpha(x)$ , there is  $y'$  such that  $(a, (x', y')) \in W(x, y)$ . The right part then implies that necessarily  $(a, y') \in \beta(y)$ .

Much as in the case of AM-bisimulations, diagonals (and actually all AM-bisimulations) are AM-simulations and AM-simulations are closed under composition, only under some conditions. However, they are not closed under inverse. These observations can be encompassed as follows:

► **Proposition 46.** *When  $\mathcal{C}$  has the regular axiom of choice and  $F$  preserves weak pullbacks, then the following is a locally ordered 2-category: objects are  $F$ -coalgebras, morphisms are AM-simulations, identities, compositions, and  $\sqsubseteq$  are given by  $\mathbf{Rel}(\mathcal{C})$ . We denote this category by  $\mathbf{Sim}(F)$ .*

We can formalise the relationship between lax coalgebra morphisms and simulation maps:

► **Theorem 47.** *Maps in  $\mathbf{Rel}(\mathcal{C})$  that are AM-simulations are precisely lax morphisms of coalgebra.*

Note that this theorem cannot have a form as nice as Theorem 18 because AM-simulations are not closed under inverse, and the right adjoint of a map has to be its inverse. At this point, we can also describe the tabulations of AM-simulations:

► **Proposition 48.** *If  $U$  is an AM-simulation from  $\alpha$  to  $\beta$ , and if  $f : Z \rightarrow X$ ,  $g : Z \rightarrow Y$  is a tabulation of  $U$  then, there is a coalgebra structure  $\gamma$  on  $Z$  such that  $f$  is a coalgebra morphism from  $\gamma$  to  $\alpha$  and  $g$  is a lax coalgebra morphism from  $\gamma$  to  $\beta$ .*

► **Corollary 49.** *Assume  $\mathcal{C}$  has the regular axiom of choice. Assume given two coalgebras  $\alpha : X \rightarrow F(X)$  and  $\beta : Y \rightarrow F(Y)$ , and two points  $p : * \rightarrow X$  and  $q : * \rightarrow Y$ . There is an AM-simulation  $r : R \rightarrow X \times Y$  from  $\alpha$  to  $\beta$ , and a point  $c : * \rightarrow R$  with  $r \cdot c = \langle p, q \rangle$  if and only if there is a span  $X \xleftarrow{f} Z \xrightarrow{g} Y$ , an  $F$ -coalgebra structure  $\gamma$  on  $Z$  such that  $f$  is a coalgebra morphism from  $\gamma$  to  $\alpha$  and  $g$  is a lax coalgebra morphism from  $\gamma$  to  $\beta$ , and a point  $w : * \rightarrow Z$  such that  $f \cdot w = p$  and  $g \cdot w = q$ .*

This formalises some observations that simulations are spans of a bisimulation map and a simulation map (see [24] for examples of this fact in the context of open maps).

### 5.3 Extending the Order-Structure

In Section 5.1, we started by assuming a relation  $\leq$  on the Hom-sets of the form  $\mathcal{C}(X, FY)$  satisfying some properties. This good order structure was necessary to prove the properties of Section 5.2. In the coming section, we will pass again from plain to toposal, by considering  $F$ -coalgebras as  $\mathcal{P}F$ -coalgebras. It is then needed to extend good order structures on  $F$  to good order structures on  $\mathcal{P}F$ .

Assume given a relation  $\leq$  on all Hom-sets of the form  $\mathcal{C}(X, FY)$ . We define  $\leq_{\mathcal{P}}$  on  $\mathcal{C}(X, \mathcal{P}FY)$  as follows. A morphism  $f : X \rightarrow \mathcal{P}FY$  uniquely (up to isos) corresponds to a mono of the form  $m_f : R_f \rightarrow FY \times X$  by definition of  $\mathcal{P}$ . Then given two morphisms  $f, g : X \rightarrow \mathcal{P}FY$ ,  $f \leq_{\mathcal{P}} g$  if there exist a morphism  $u : Z \rightarrow R_g$  and an epi  $e : Z \twoheadrightarrow R_f$  such that:  $\pi_1 \cdot m_f \cdot e \leq \pi_1 \cdot m_g \cdot u$  and  $\pi_2 \cdot m_f \cdot e = \pi_2 \cdot m_g \cdot u$ .

## 19:14 Aczel-Mendler Bisimulations in a Regular Category

► **Example 50.** The order  $\leq_{\mathcal{P}}$  seems complicated but can be interpreted easily in **Set**, when the order structure on  $\mathcal{C}(X, FY)$  is a point-wise order, assuming that  $FY$  itself is preordered. Indeed, given two functions  $f, g : X \rightarrow \mathcal{P}FY$ ,  $f \leq_{\mathcal{P}} g$  if and only if for every  $x \in X$ , and every  $a \in f(x) \subseteq FY$  there is  $b \in g(x)$  such that  $a \leq b$  in  $F(Y)$ .

To make it consistent with the previous section, we show that this preserves goodness:

► **Proposition 51.**  $\leq_{\mathcal{P}}$  is a good order structure if  $\leq$  is.

### 5.4 Toposal AM-Simulations

With all those ingredients, we can easily deduce the right notion of *AM toposal-simulations*:

► **Definition 52.** We say that a relation is a toposal AM-simulation from the coalgebra  $\alpha : X \rightarrow FX$  to  $\beta : Y \rightarrow FY$ , if for any mono  $r : R \rightarrow X \times Y$  representing it, there is a morphism  $W : R \rightarrow \mathcal{P}FR$  such that:

$$\begin{array}{ccccc}
 & & X \times Y & \xrightarrow{\alpha \times \beta} & F(X) \times F(Y) \\
 R & \nearrow r & & & \eta_{F(X)} \times \eta_{F(Y)} \\
 & & & \leq_{\mathcal{P}} \times \geq_{\mathcal{P}} & \mathcal{P}F(X) \times \mathcal{P}F(Y) \\
 & \searrow W & \mathcal{P}FR & \xrightarrow{\mathcal{P}Fr} & \mathcal{P}F(X \times Y) \\
 & & & & \langle \mathcal{P}F\pi_1, \mathcal{P}F\pi_2 \rangle
 \end{array}$$

Plain and toposal AM-simulations also coincide under the axiom of choice:

► **Proposition 53.** Assume that  $\mathcal{C}$  has the regular axiom of choice. Then for every relation  $U$  from  $X$  to  $Y$ , every coalgebra  $\alpha : X \rightarrow FX$  and  $\beta : Y \rightarrow FY$ ,  $U$  is an AM-simulation from  $\alpha$  to  $\beta$  if and only if it is a toposal AM-simulation between them.

Finally, we can prove the closure under composition and the characterisation with spans without the axiom of choice:

► **Proposition 54.** Proposition 46 holds without regular axiom of choice when replacing AM-simulations by toposal AM-simulations.

► **Theorem 55.** Assume given two coalgebras  $\alpha : X \rightarrow F(X)$  and  $\beta : Y \rightarrow F(Y)$ , and two points  $p : * \rightarrow X$  and  $q : * \rightarrow Y$ . There is a toposal AM-simulation  $r : R \rightarrow X \times Y$  from  $\alpha$  to  $\beta$ , and a point  $c : * \rightarrow R$  such that  $r \cdot c = \langle p, q \rangle$  if and only if there is a span  $X \xleftarrow{f} Z \xrightarrow{g} Y$ , a  $\mathcal{P}F$ -coalgebra structure  $\gamma$  on  $Z$  such that  $f$  is a  $\mathcal{P}F$ -coalgebra morphism from  $\gamma$  to  $\eta_X \cdot \alpha$  and  $g$  a lax  $\mathcal{P}F$ -coalgebra morphism from  $\gamma$  to  $\eta_Y \cdot \beta$ , and a point  $w : * \rightarrow Z$  such that  $f \cdot w = p$  and  $g \cdot w = q$ .

## 6 Examples

In this section, let us develop some examples in different regular categories.

### 6.1 Vietoris Bisimulations

In [5], Bezhanishvili et al. are studying bisimulations for the Vietoris functor – the functor mapping a topological space to its set of closed subspaces equipped with a suitable topology – in the category **Stone** of Stone spaces and continuous functions. More concretely, they show that so-called descriptive models coincide with coalgebras of the form  $X \rightarrow \mathcal{V}(X) \times A$  where  $\mathcal{V}$  is the Vietoris functor and  $A$  is some fixed Stone space (usually,  $A = \mathcal{P}S = \prod_{s \in S} \{0, 1\}$

equipped with the product topology and  $\{0, 1\}$  equipped with the discrete topology). They are interested in describing relation liftings (much as the one defining HJ-bisimulations) that coincide with behavioural equivalences. They actually proved that in this case AM-bisimilarity does not coincide with behavioural equivalence, and that the main reason is because the Vietoris functor does not preserve weak-pullbacks. In [23], Staton proved that the Vietoris functor is a so-called  $\mathcal{S}$ -powerset functor, and that in particular it covers pullbacks. Together with the (well-known) fact that the category of Stone spaces is regular and has pushouts, Theorem 29 holds in this case, and all three notions – regular AM-bisimulations, HJ-bisimulations, and behavioural equivalences – coincide.

Let us develop the counter-examples described in [5]. Let  $\overline{\mathbb{N}}$  being  $\mathbb{N} \cup \{\infty\}$ , obtained as the Alexandroff-compactification of  $\mathbb{N}$  equipped with the discrete topology. Concretely, the open sets of  $\overline{\mathbb{N}}$  are  $\{U \subseteq \mathbb{N}\} \cup \{U \cup \{\infty\} \mid U \subseteq \mathbb{N} \wedge \exists n \in U. \forall m \geq n. m \in U\}$ . Denote  $\overline{\mathbb{N}} \oplus \overline{\mathbb{N}} \oplus \overline{\mathbb{N}}$ , the coproduct of three copies of  $\overline{\mathbb{N}}$ , by  $3\overline{\mathbb{N}}$ . Let us also consider  $A = \mathcal{P}(\mathbb{N} \times \{+, -\})$  as above. Define the continuous function  $\tau : 3\overline{\mathbb{N}} \rightarrow \mathcal{V}(3\overline{\mathbb{N}})$  as follows:  $\tau(i_1) = \{i_2, i_3\}$  and  $\tau(i_2) = \tau(i_3) = \emptyset$ , where  $i_j$  denotes the  $j$ -th copy of  $i \in \overline{\mathbb{N}}$ . Define two continuous functions  $\lambda, \lambda' : 3\overline{\mathbb{N}} \rightarrow A$   $\lambda(i_1) = \lambda'(i_1) = \{\}$  for all  $i \in \overline{\mathbb{N}}$ ;  $\lambda(\infty_j) = \lambda'(\infty_j) = \{\}$  for  $j \in \{2, 3\}$ ;  $\lambda(i_2) = \lambda'(i_2) = \{i+\}$ ,  $\lambda(i_3) = \lambda'(i_3) = \{i-\}$ , for  $i$  odd;  $\lambda(i_2) = \lambda'(i_3) = \{i+\}$ ,  $\lambda(i_3) = \lambda'(i_2) = \{i-\}$  for  $i$  even. Altogether, this defines two coalgebras  $\alpha = \langle \tau, \lambda \rangle$  and  $\beta = \langle \tau, \lambda' \rangle$ . In [5], they proved that the following relation (for Stone spaces, relations coincide with closed subspaces of a product):

$$R = \{(i_1, i_1) \mid i \in \overline{\mathbb{N}}\} \cup \{(i_2, i_2), (i_3, i_3) \mid i \in \mathbb{N} \text{ odd}\} \cup \{(i_2, i_3), (i_3, i_2) \mid i \in \mathbb{N} \text{ even}\} \\ \cup \{(\infty_j, \infty_k) \mid j, k \in \{2, 3\}\}$$

is a Vietoris bisimulation but not an AM-bisimulation. We can reformulate this as:

► **Theorem 56.**  *$R$  is a regular AM-bisimulation but not an AM-bisimulation.*

For the second part of this statement, this means that there is no continuous function  $W : R \rightarrow \mathcal{V}(R) \times A$  satisfying the requirement of an AM-bisimulation. However, there is a relation  $W \subseteq R \times \mathcal{V}(R) \times A$  that satisfies the requirement of a regular AM-bisimulation as:

$$W = \{((i_1, i_1), \{(i_2, i_2), (i_3, i_3)\}, \{\}) \mid i \in \mathbb{N} \text{ odd}\} \\ \cup \{((i_1, i_1), \{(i_2, i_3), (i_3, i_2)\}, \{\}) \mid i \in \mathbb{N} \text{ even}\} \\ \cup \{((\infty_1, \infty_1), \{(\infty_2, \infty_2), (\infty_3, \infty_3)\}, \{\}), ((\infty_1, \infty_1), \{(\infty_2, \infty_3), (\infty_3, \infty_2)\}, \{\})\} \\ \cup \{((i_j, i_k), \emptyset, \lambda(i_j)) \mid i \in \overline{\mathbb{N}} \wedge (i_j, i_k) \in R\}$$

The interesting part is that  $(\infty_1, \infty_1)$  is related to two elements, and that if one of them is removed, then  $W$  is not closed anymore, and so not a relation in **Stone**. This explains why this relation cannot be restricted to the graph of a continuous function.

## 6.2 Toposes for Name-Passing

In [23], Staton studies models of name-passing and their bisimulations. Three toposes and functors are presented to model different parts of the theory. The first topos is the category of name substitution, which is the category of presheaves over non-empty finite subsets of a fixed countable set, together with all functions between them. It comes with a functor combining non-determinism and name-binding. This functor satisfies strong properties: in particular, AM-bisimulations coincide with HJ-bisimulations, and the largest AM-bisimulation coincide with the largest behavioural equivalence. This framework is already nice as AM-bisimulations describe precisely open bisimulations [20].

The second topos is a refinement of the first one, as the category of functors over all finite subsets of the given countable set, together with injections. The proposed functor in this case is less nice: it does not preserve weak-pullbacks and AM-bisimulations do not coincide with HJ-bisimulations anymore. However, it is nice enough in our theory: it covers pullbacks, and the category is a topos, so regular and with pushouts, then HJ-bisimulations coincide with regular AM-bisimulations, and their existence coincides with the existence of a behavioural equivalence.

For this topos, it is remarked in [23] that if a relation is a HJ-bisimulation (so a regular/toposal AM-bisimulation), then its  $\neg\neg$ -completion is an AM-bisimulation, which means in particular that this framework for name-passing is much nicer when restricting to  $\neg\neg$ -sheaves. One main reason for that is that the sheaf topos for the  $\neg\neg$ -topology satisfies the axiom of choice when the base topos is a presheaf topos over a poset [19], which is the case here.

### 6.3 Weighted Linear Systems

In [6], Bonchi et al. are studying linear weighted systems, that is, coalgebras for the endofunctor  $X \mapsto K \times X^A$  on  $K\mathbf{Vect}$ , in the category of  $K$ -vector spaces, with  $K$  a field, and  $A$  a set. The following discussion can also be made in the category of modules over a ring. The category  $K\mathbf{Vect}$  is abelian, and so is regular and has pushouts. The endofunctor actually preserves pullbacks, so the three notions of bisimilarity coincide by Theorem 29. In this paper, they are interested in linear bisimulations, which coincide with behavioural equivalence, and so to the other two notions of bisimilarities.

In perspective, usual weighted systems are described in the category  $\mathbf{Set}$ , with the functor  $X \mapsto A \Rightarrow K^{(X)}$  where  $K^{(X)}$  is the set of functions from  $X$  to  $K$  which takes finitely many non-zero values. In this context, this functor does not even cover pullbacks in general, and they actually prove that AM-bisimilarity (and so regular AM-bisimilarity since  $\mathbf{Set}$  has the regular axiom of choice) does not coincide with behavioural equivalence.

## 7 Conclusion

This paper introduces some foundations of the theory of bisimulations and simulations in a general regular category, mitigating some known issues about Aczel-Mendler bisimulations. The relations and power objects are the key ingredients for this mitigation: if the axiom of choice allows to pick some witnesses of bisimilarity, the relations and power objects allow to collect them up without need to choose. This paves the way to the study of such bisimulations in more exotic regular categories and toposes.

One direction of future work is to investigate regular AM-bisimulations for probabilistic systems, compared to what is done in [8, 7] for behavioural equivalences. The main challenge is to find a suitable *regular* category of “probabilistic space” and a “probabilistic distribution functor” that *covers pullbacks*. For the first property, the work on Quasi-Borel spaces [10], producing a quasi-topos, is of interest. For the second one, looking at categories of  $\sigma$ -frames (see for example [21]), for which pullbacks do not coincide with pullbacks in the category of measurable spaces is a solution under investigation.

---

### References

- 1 Samson Abramsky. Coalgebras, Chu spaces, and representations of physical systems. *Journal of Philosophical Logic*, 42:551–574, 2013.
- 2 Peter Aczel and Nax Mendler. A final coalgebra theorem. In *Proceedings of Category Theory and Computer Science*, volume 389 of *Lecture Notes in Computer Science*. Springer, 1999.

- 3 Emilio Jesús Gallego Arias and James B. Lipton. Logic Programming in Tabular Alegories. In *Technical Communications of the 28th International Conference on Logic Programming (ICLP'12)*, Leibniz International Proceedings in Informatics, pages 334–347. Schloss Dagstuhl, 2012.
- 4 Roland Backhouse and Paul Hoogendijk. Final dialgebras: from categories to allegories. *Theoretical Informatics and Application*, 33:401–426, 1999.
- 5 Nick Bezhanishvili, Gaelle Fontaine, and Yde Yenema. Vietoris bisimulations. *Journal of Logic and Computation*, 20(5):1017–1040, 2010.
- 6 Filippo Bonchi, Marcello Bonsangue, Michele Boreale, Jan Rutten, and Alexandra Silva. A coalgebraic perspective on linear weighted automata. *Information and Computation*, 211:77–105, 2012.
- 7 Vincent Danos, Josée Desharnais, Francois Laviolette, and Prakash Panangaden. Bisimulation and cocongruence for probabilistic systems. *Information and Computation*, 204(4):503–523, 2006. doi:10.1016/j.ic.2005.02.004.
- 8 Josée Desharnais, Abbas Edalat, and Prakash Panangaden. Bisimulation for Labelled Markov Processes. *Information and Computation*, 179(2):163–193, 2003. doi:10.1006/inco.2001.2962.
- 9 Peter Freyd and Andre Scedrov. *Categories, Allegories*, volume 39 of *Mathematical Library*. North-Holland, 1990.
- 10 Chris Heunen, Ohad Kammar, Sam Staton, and Hongseok Yang. A convenient category for higher-order probability theory. In *Proceedings of the 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS'17)*, 2017.
- 11 Jesse Hughes and Bart Jacobs. Simulations in coalgebra. *Theoretical Computer Science*, 327(1-2):71–108, 2004.
- 12 J. Martin E. Hyland. The effective topos. In *The L.E.J. Brouwer Centenary Symposium*, volume 110 of *Studies in Logic and the Foundations of Mathematics*, pages 165–216, 1982.
- 13 Bart Jacobs. *Introduction to Coalgebra: Towards Mathematics of States and Observation*, volume 29 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2017.
- 14 Peter T. Johnstone. *Sketches of an Elephant: A Topos Theory Compendium*, volume 1. Oxford University Press, 2002.
- 15 Peter T. Johnstone. *Sketches of an Elephant: A Topos Theory Compendium*, volume 2. Oxford University Press, 2002.
- 16 André Joyal, Mogens Nielsen, and Glynn Winskel. Bisimulation from Open maps. *Information and Computation*, 127:164–185, 1996.
- 17 F. William Lawvere. Equality in hyperdoctrines and comprehension schema as an adjoint functor. In *Proceedings of the AMS Symposium on Pure Mathematics XVII*, pages 1–14, 1970.
- 18 F. William Lawvere. Qualitative Distinctions between some Toposes of Generalized Graphs. *Contemporary Mathematics*, 92, 1989.
- 19 Saunders MacLane and Ieke Moerdijk. *Sheaves in Geometry and Logic: A First Introduction to Topos Theory*. Springer New-York, 1992.
- 20 Davide Sangiorgi. A theory of bisimulations for the  $\pi$ -calculus. *Acta Informatica*, 33:69–97, 1996.
- 21 Alex Simpson. Measure, randomness and sublocales. *Annals of Pure and Applied Logic*, 163(11):1642–1659, 2012.
- 22 Sam Staton. *Name-passing process calculi: operational models and structural operational semantics*. PhD thesis, University of Cambridge, 2008.
- 23 Sam Staton. Relating Coalgebraic Notions of Bisimulation. *Logical Methods in Computer Science*, 7:1–21, 2011.
- 24 Paulo Tabuada. Open Maps, Alternating Simulations and Control Synthesis. In *Proceedings of the 15th International Conference in Concurrency Theory (CONCUR'04)*, volume 3170 of *Lecture Notes in Computer Science*, pages 466–480. Springer, 2004.

## 19:18 Aczel-Mendler Bisimulations in a Regular Category

- 25 Michael Winter. *Goguen Categories: A Categorical Approach to L-fuzzy Relations*. Number 25 in Trends in Logic. Springer, 2007.
- 26 Thorsten Wißman, Jérémy Dubut, Shin-ya Katsumata, and Ichiro Hasuo. Path Category for Free: Open Morphisms from Coalgebras with Non-deterministic Branching. In *Proceedings of the 22nd International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'19)*, volume 11425 of *Lecture Notes in Computer Science*. Springer, 2019.

# Completeness for Categories of Generalized Automata

Guido Boccali ✉

University of Torino, Italy

Andrea Laretto ✉

Tallinn University of Technology, Estonia

Fosco Loregian ✉ 

Tallinn University of Technology, Estonia

Stefano Luneia ✉

University of Bologna, Italy

---

## Abstract

We present a slick proof of completeness and cocompleteness for categories of  $F$ -automata, where the span of maps  $E \xleftarrow{d} E \otimes I \xrightarrow{s} O$  that usually defines a deterministic automaton of input  $I$  and output  $O$  in a monoidal category  $(\mathcal{K}, \otimes)$  is replaced by a span  $E \leftarrow FE \rightarrow O$  for a generic endofunctor  $F : \mathcal{K} \rightarrow \mathcal{K}$  of a generic category  $\mathcal{K}$ : these automata exist in their “Mealy” and “Moore” version and form categories  $F$ -Mly and  $F$ -Mre; such categories can be presented as strict 2-pullbacks in  $\mathbf{Cat}$  and whenever  $F$  is a left adjoint, both  $F$ -Mly and  $F$ -Mre admit all limits and colimits that  $\mathcal{K}$  admits. We mechanize our main results using the proof assistant Agda and the library `agda-categories`.

**2012 ACM Subject Classification** Theory of computation  $\rightarrow$  Automata extensions; Theory of computation  $\rightarrow$  Automata over infinite objects; Theory of computation  $\rightarrow$  Formal languages and automata theory; Theory of computation  $\rightarrow$  Formalisms

**Keywords and phrases** Deterministic automata, Moore machines, Mealy machines, coalgebras, cocomplete category

**Digital Object Identifier** 10.4230/LIPIcs.CALCO.2023.20

**Category** (Co)algebraic pearls

**Supplementary Material** *Software (Source Code)*: <https://github.com/iwilare/categorical-automata> archived at `swh:1:dir:5e12eb2d97e15118b9018d5b5f8254fd6f6e24f3`

**Funding** *Fosco Loregian*: The author was supported by the ESF funded Estonian IT Academy research measure (project 2014-2020.4.05.19-0001).

**Acknowledgements** *À René, parce qu’il faut ruser pour te lire.*

## 1 Introduction

One of the most direct representations of *deterministic automata* in the categorical settings consists (cf. [1, 4, 5]) of a span of morphisms  $E \xleftarrow{d} E \times I \xrightarrow{s} O$ , where the left leg provides a notion of *dynamics* or *next state* function, given a current state  $E$  and an input  $I$ , and the right leg provides an *final state* or output  $O$ .

According to whether the output morphism depends on both the current state and an input or just on the state, one can then talk about classes of *Mealy* and *Moore automata*, respectively. This perspective of “automata in a category” naturally captures the idea that morphisms of a category can be interpreted as a general abstraction of processes/sequential operations.



© Guido Boccali, Andrea Laretto, Fosco Loregian, and Stefano Luneia; licensed under Creative Commons License CC-BY 4.0

10th Conference on Algebra and Coalgebra in Computer Science (CALCO 2023).

Editors: Paolo Baldan and Valeria de Paiva; Article No. 20; pp. 20:1–20:14

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The above notion of deterministic automaton carries over to any monoidal category, on which the various classical notions of automata, e.g., minimization, bisimulation, powerset construction, can be equivalently reconstructed; this is studied, to a large extent, in the monograph [5].

In [1, 7], automata are generalized to the case in which, instead of taking spans from the monoidal product of states and inputs  $E \otimes I$ , one considers spans  $E \leftarrow FE \rightarrow O$  for a generic endofunctor  $F : \mathcal{K} \rightarrow \mathcal{K}$ , providing an abstraction for the ambient structure that allows the automata to advance to the “next” state and give an output.

A general theorem asserting that the category of Mealy and Moore automata  $\text{Mly}_{\mathcal{K}}(I, O)$ ,  $\text{Mre}_{\mathcal{K}}(I, O)$  in a monoidal category  $(\mathcal{K}, \otimes)$  are complete and cocomplete whenever  $\mathcal{K}$  is itself complete and cocomplete can be obtained with little conceptual effort, cf. [5, Ch. 11], but the proof given therein is a bit ad-hoc, and provides no intuition for why finite products and terminal objects tend to be so complicated.

With just a little bit more category-theoretic technology, some general considerations can be made about the shape of limits in such settings: colimits and connected limits can be computed as they are computed in  $\mathcal{K}$  (as a consequence of the fact that the forgetful functor from the category of machines *creates* them, cf. [16]), whereas products (and in particular the empty product, the terminal object) have dramatically different shapes than those provided in  $\mathcal{K}$ . The profound reason why this happens is the fact that such a terminal object (which we refer to  $O_{\infty}$ ) coincides with the terminal coalgebra of a specific endofunctor, which, for Moore or Mealy automata, is respectively given by  $A \mapsto O \times RA$  and  $A \mapsto RO \times RA$ . The complicated shape of the terminal object  $O_{\infty}$  in  $\text{Mly}_{\mathcal{K}}(I, O)$  is then explained by Adámek’s theorem, which presents the terminal object  $O_{\infty}$  as an (usually intricate) inverse limit in  $\mathcal{K}$ .

In this paper, we show that under the same assumption of completeness of the underlying category  $\mathcal{K}$ , the completeness of  $F$ -automata can be obtained by requiring that the endofunctor  $F$  admits a right adjoint  $R$ . The proof we provide follows a slick argument proving the existence of (co)limits by fitting each  $\text{Mly}_{\mathcal{K}}(I, O)$  and  $\text{Mre}_{\mathcal{K}}(I, O)$  into a strict 2-pullback in  $\text{Cat}$ , and deriving the result from stability properties of limit-creating functors.

## 1.1 Outline of the paper

The present short note develops as follows:

- First (Section 2) we introduce the language we will employ and the structures we will study:<sup>1</sup> categories of automata valued in a monoidal category  $(\mathcal{K}, \otimes)$  (in two flavours: “Mealy” machines, where one considers spans  $E \leftarrow E \otimes I \rightarrow O$ , and “Moore”, where instead one consider pairs  $E \leftarrow E \otimes I, E \rightarrow O$ ) and of  $F$ -automata, where  $F$  is an endofunctor of  $\mathcal{K}$  (possibly with no monoidal structure). “Mealy” automata are known as “deterministic automata” in today’s parlance, but since we need to distinguish between the two kinds of diagram from time to time, we stick to an older terminology.
- Then (Theorem 3.6), to establish the presence of co/limits of shape  $\mathcal{J}$  in categories of  $F$ -automata, under the two assumptions that  $F : \mathcal{K} \rightarrow \mathcal{K}$  is a left adjoint in an adjunction  $F \overset{\epsilon}{\dashv} \underset{\eta}{R}$ , and that co/limits of shape  $\mathcal{J}$  exist in the base category  $\mathcal{K}$ .
- Last (Subsection 3.1), to address the generalisation to  $F$ -machines of the “behaviour as an adjunction” perspective expounded in [18, 19].

---


<sup>1</sup> An almost identical introductory short section appears in [2], of which the present note is a parallel submission –although related, the two manuscripts are essentially independent, and the purpose of this repetition is the desire for self-containment.



Similarly to the situation for Mealy/Moore machines, where  $F = - \otimes I$ , *discrete* limits in  $F\text{-Mly}$  and  $F\text{-Mre}$  exist but tend to have a shape that is dramatically different than the one in  $\mathcal{K}$ .

A number of examples of endofunctors  $F$  that satisfy the previous assumption come from considering  $F$  as the (underlying endofunctor of the) comonad  $LG$  of an adjunction  $L \dashv G \dashv U$ , since in that case  $LG \dashv UG$ : the shape-flat and flat-sharp adjunctions of a cohesive topos [13, 14], or the base-change adjunction  $\text{Lan}_f \dashv f^* \dashv \text{Ran}_f$  for a morphism of rings, or more generally,  $G$ -modules in representation theory, any essential geometric morphism, or any topological functor  $V : \mathcal{E} \rightarrow \mathcal{B}$  [3, Prop. 7.3.7] with its fully faithful left and right adjoints  $L \dashv V \dashv R$  gives rise to a comodality  $LV$ , left adjoint to a modality  $RV$ .

The results we get are not particularly surprising; we have not, however, been able to trace a reference addressing the *co/completeness* properties of  $F\text{-Mly}$ ,  $F\text{-Mre}$  nor an analogue for the “behaviour as an adjunction” theorems expounded in [18, 19]; in the case  $F = - \otimes I$  *co/completeness* results follows from unwieldy ad-hoc arguments (cf. [5, Ch. 11]), whereas in Theorem 3.6 we provide a clean, synthetic way to derive both results from general principles, starting by describing  $F\text{-Mly}$  and  $F\text{-Mre}$  as suitable pullbacks in  $\text{Cat}$ , in Proposition 3.5.


We provide a mechanisation of our main results using the proof assistant Agda and the library `agda-categories`: we will add a small Agda logo () next to the beginning of a definition or statement whenever it is accompanied by Agda code: this is a hyperlink pointing directly to the formalisation files. The full development is freely available for consultation and is available at <https://github.com/iwilare/categorical-automata>.

## 2 Automata and $F$ -automata

The only purpose of this short section is to fix notation; classical comprehensive references for this material are [1, 5]; in particular, [1, Ch. III] is entirely devoted to the study of what here are called  $F$ -Moore automata, possibly equipped with an “initialization” morphism.

### 2.1 Mealy and Moore automata

For the entire subsection, we fix a monoidal category  $(\mathcal{K}, \otimes, 1)$ .

► **Definition 2.1** (Mealy machine). () A Mealy machine in  $\mathcal{K}$  of input object  $I$  and output object  $O$  consists of a triple  $(E, d, s)$  where  $E$  is an object of  $\mathcal{K}$  and  $d, s$  are morphisms in a span

$$\epsilon := \left( E \xleftarrow{d} E \otimes I \xrightarrow{s} O \right) \quad (2.1)$$

► **Remark 2.2** (The category of Mealy machines). Mealy machines of fixed input and output  $I, O$  form a category, if we define a *morphism of Mealy machines*  $f : (E, d, s) \rightarrow (T, d', s')$  as a morphism  $f : E \rightarrow T$  in  $\mathcal{K}$  such that

$$\begin{array}{ccccc} E & \xleftarrow{d} & E \otimes I & \xrightarrow{s} & O \\ f \downarrow & & \downarrow f \otimes I & & \parallel \\ T & \xleftarrow{d'} & T \otimes I & \xrightarrow{s'} & O \end{array} \quad (2.2)$$

Clearly, composition and identities are performed in  $\mathcal{K}$ .

The category of Mealy machines of input and output  $I, O$  is denoted as  $\text{Mly}_{\mathcal{K}}(I, O)$ .

► **Definition 2.3** (Moore machine). ( $\mathcal{U}$ ) A Moore machine in  $\mathcal{K}$  of input object  $I$  and output object  $O$  is a diagram

$$\mathfrak{m} := \left( E \xleftarrow{d} E \otimes I ; E \xrightarrow{s} O \right) \quad (2.3)$$

► **Remark 2.4** (The category of Moore machines). Moore machines of fixed input and output  $I, O$  form a category, if we define a *morphism of Moore machines*  $f : (E, d, s) \rightarrow (T, d', s')$  as a morphism  $f : E \rightarrow T$  in  $\mathcal{K}$  such that

$$\begin{array}{ccc} E & \xleftarrow{d} & E \otimes I & & E & \xrightarrow{s} & O \\ f \downarrow & & \downarrow f \otimes I & & f \downarrow & & \parallel \\ T & \xleftarrow{d'} & T \otimes I & & T & \xrightarrow{s'} & O \end{array} \quad (2.4)$$

## 2.2 $F$ -Mealy and $F$ -Moore automata

The notion of  $F$ -machine arises by replacing the tensor  $E \otimes I$  in (2.1) with the action  $FE$  of a generic endofunctor  $F : \mathcal{K} \rightarrow \mathcal{K}$  on an object  $E \in \mathcal{K}$ , in such a way that a Mealy/Moore machine is just a  $(- \otimes I)$ -Mealy/Moore machine; cf. [7, ff. 2.1.3<sup>o</sup>], or Chapter III of the monograph [1]. This natural idea acts as an abstraction for the structure that allows the machine to advance to the “next” state and give an output, and it leads to the following two definitions (where we do *not* require  $\mathcal{K}$  to be monoidal).

► **Definition 2.5** ( $F$ -Mealy machine). ( $\mathcal{U}$ ) Let  $O \in \mathcal{K}$  be a fixed object. The objects of the category  $F\text{-Mly}_{/O}$  (or simply  $F\text{-Mly}$  when the object  $O$  is implicitly clear) of  $F$ -Mealy machines of output  $O$  are the triples  $(E, d, s)$  where  $E \in \mathcal{K}$  is an object and  $s, d$  are morphisms in  $\mathcal{K}$  that fit in the span

$$E \xleftarrow{d} FE \xrightarrow{s} O \quad (2.5)$$

A morphism of  $F$ -Mealy machines  $f : (E, d, s) \rightarrow (T, d', s')$  consists of a morphism  $f : E \rightarrow T$  in  $\mathcal{K}$  such that

$$\begin{array}{ccc} E & \xleftarrow{d} & FE & \xrightarrow{s} & O \\ f \downarrow & & \downarrow Ff & & \parallel \\ T & \xleftarrow{d'} & FT & \xrightarrow{s'} & O \end{array} \quad (2.6)$$

Unsurprisingly, we can generalise in the same fashion Definition 2.3 to the case of a generic endofunctor  $F : \mathcal{K} \rightarrow \mathcal{K}$ .

► **Definition 2.6** ( $F$ -Moore machine). ( $\mathcal{U}$ ) Let  $O \in \mathcal{K}$  be a fixed object. The objects of the category  $F\text{-Mre}_{/O}$  (or simply  $F\text{-Mre}$  when the object  $O$  is implicitly clear) of  $F$ -Moore machines of output  $O$  are the triples  $(E, d, s)$  where  $E \in \mathcal{K}$  is an object and  $s, d$  are a pair of morphisms in  $\mathcal{K}$

$$E \xleftarrow{d} FE ; E \xrightarrow{s} O \quad (2.7)$$

A morphism of  $F$ -Moore machines  $f : (E, d, s) \rightarrow (T, d', s')$  consists of a morphism  $f : E \rightarrow T$  in  $\mathcal{K}$  such that

$$\begin{array}{ccc} E & \xleftarrow{d} & FE & & E & \xrightarrow{s} & O \\ f \downarrow & & \downarrow Ff & & f \downarrow & & \parallel \\ T & \xleftarrow{d'} & FT & & T & \xrightarrow{s'} & O \end{array} \quad (2.8)$$

► Remark 2.7 (Interdefinability of notions of machine). All the concepts of machine introduced so far are interdefinable, provided we allow the monoidal base  $\mathcal{K}$  to change (cf. [7, ff. Proposition 30]): a Mealy machine is, obviously, an  $F$ -machine where  $F : \mathcal{K} \rightarrow \mathcal{K}$  is the functor  $- \otimes I : E \mapsto E \otimes I$ ; an  $F$ -machine consists of a Mealy machine in a category of endofunctors: in fact,  $F$ -machines are precisely the Mealy machines of the form  $E \leftarrow F \circ E \rightarrow O$ , where  $E, O$  are constant endofunctors on objects of  $\mathcal{K}$  and  $F$  is the input object: more precisely, the category of  $F$ -machines is contained in the category  $\text{Mly}_{([\mathcal{K}, \mathcal{K}], \circ)}(F, c_O)$ , where  $c_O$  is the constant functor on  $O \in \mathcal{K}$ , as the subcategory of those triples  $(E, d, s)$  where  $E$  is a constant endofunctor.

### 3 Completeness and behaviour in $F$ -Mly and $F$ -Mre

The first result that we want to generalise to  $F$ -machines is the well-known fact that, considering for example Mealy machines, if  $(\mathcal{K}, \otimes)$  has countable coproducts preserved by each  $I \otimes -$ , then the span (2.1) can be “extended” to a span

$$E \xleftarrow{d^+} E \otimes I^+ \xrightarrow{s^+} O \quad (3.1)$$

where  $d^+, s^+$  can be defined inductively from components  $d_n, s_n : E \otimes I^{\otimes n} \rightarrow E, O$ .

Under the same assumptions, each Moore machine (2.3) can be “extended” to a span

$$E \xleftarrow{d^*} E \otimes I^* \xrightarrow{s^*} O \quad (3.2)$$

where  $d^*, s^*$  can be defined inductively from components  $d_n, s_n : E \otimes I^{\otimes n} \rightarrow E, O$ .<sup>2</sup>

► Remark 3.1. In the case of Mealy machines, the object  $I^+$  corresponds to the *free semigroup* on the input object  $I$ , whereas for Moore machines one needs to consider the *free monoid*  $I^*$ : this mirrors the intuition that in the latter case an output can be provided without any previous input. Note that the extension of a Moore machine gives rise to a span of morphisms from the same object  $E \otimes I^*$ , i.e., a Mealy machine that accepts the empty string as input.

A similar construction can be carried over in the category of  $F$ -Mealy machines, using the  $F$ -algebra map  $d : FE \rightarrow E$  to generate iterates  $E \xleftarrow{d_n} F^n E \xrightarrow{s_n} O$ , for  $n \geq 1$ .

From now on, let  $F$  be an endofunctor of a category  $\mathcal{K}$  that has a right adjoint  $R$ . Examples of such arise naturally from the situation where a triple of adjoints  $L \dashv G \dashv R$  is given, since we obtain adjunctions  $LG \dashv RG$  and  $GL \dashv GR$ :

- every homomorphism of rings  $f : A \rightarrow B$  induces a triple of adjoint functor between the categories of  $A$  and  $B$ -modules (cf. [3, 4.7.4]);
- similarly, every homomorphism of monoids  $f : M \rightarrow N$  induces a “base change” functor  $f^* : N\text{-Set} \rightarrow M\text{-Set}$  (this is usually treated as a fact all category theorists know; however, an elementary exposition of this fact can be found in [21, Prop. 4.1.4.11]);
- every essential geometric morphism between topoi  $\mathcal{E} \rightleftarrows \mathcal{F}$ , i.e. every triple of adjoints  $f_! \dashv f^* \dashv f_*$  (cf. [10, 1.16]);
- every topological functor  $V : \mathcal{E} \rightarrow \mathcal{B}$  [3, Prop. 7.3.7] with its fully faithful left and right adjoints  $L \dashv V \dashv R$  (this gives rise to a comodality  $LV$ , left adjoint to a modality  $RV$ ).

<sup>2</sup> Assuming countable coproducts in  $\mathcal{K}$ , the free *monoid*  $I^*$  on  $I$  is the object  $\sum_{n \geq 0} I^n$ ; the free *semigroup*  $I^+$  on  $I$  is the object  $\sum_{n \geq 1} I^n$ ; clearly, if 1 is the monoidal unit of  $\otimes$ ,  $I^* \cong 1 + I^+$ , and the two objects satisfy “recurrence equations”  $I^+ \cong I \otimes I^+$  and  $I^* \cong 1 + I \otimes I^*$ .

## 20:6 Completeness for Categories of Generalized Automata

► **Construction 3.2** (Dynamics of an  $F$ -machine). ( $\mathcal{U}$ ) For any given  $F$ -Mealy machine

$$E \xleftarrow{d} FE \xrightarrow{s} O \quad (3.3)$$

we define the family of morphisms  $s_n : F^n E \rightarrow O$  (for  $n \geq 1$ ) inductively, as the composites

$$\begin{cases} s_1 &= FE \xrightarrow{s} O \\ s_2 &= FFE \xrightarrow{Fd} FE \xrightarrow{s} O \\ s_n &= F^n E \xrightarrow{F^{n-1}d} F^{n-1}E \rightarrow \dots \xrightarrow{FFd} FFE \xrightarrow{Fd} FE \xrightarrow{s} O \end{cases} \quad (3.4)$$

Under our assumption that  $F$  has a right adjoint  $R$ , this is equivalent to the datum of their mates  $\bar{s}_n : E \rightarrow R^n O$  for  $n \geq 1$  under the adjunction  $F^n \dashv_{\eta_n} R^n$  obtained by composition, iterating the structure in  $F \dashv_{\eta} R$ .

Such a  $s_n$  is called the  $n$ th *skip map*. Observe that in case  $\mathcal{K}$  has countable products, the family of all  $n$ th skip maps ( $s_n \mid n \in \mathbb{N}_{\geq 1}$ ) is obviously equivalent to a single map of type  $\bar{s}_\infty : E \rightarrow \prod_{n \geq 1} R^n O$ .

► **Remark 3.3.** Reasoning in a similar fashion, one can define extensions  $s : E \rightarrow O$ ,  $s \circ d : FE \rightarrow E \rightarrow O$ ,  $s \circ d \circ Fd : FFE \rightarrow O$ , etc. for an  $F$ -Moore machine.

This is the first step towards the following statement, which will be substantiated and expanded in Theorem 3.6 below:

▷ **Claim 3.4.** The category  $F\text{-Mre}$  of Definition 2.6 has a terminal object  $\mathfrak{o} = (O_\infty, d_\infty, s_\infty)$  with carrier  $O_\infty = \prod_{n \geq 0} R^n O$ ; similarly, the category  $F\text{-Mly}$  has a terminal object with carrier  $O_\infty = \prod_{n \geq 1} R^n O$ . (Note the shift in the index of the product, motivated by the fact that the skip maps for a Moore machine are indexed on  $\mathbb{N}_{\geq 0}$ , and on  $\mathbb{N}_{\geq 1}$  for Mealy.)

The “modern” way to determine the presence of a terminal object in categories of automata relies on the elegant coalgebraic methods in [9]; the interest in such completeness theorems can be motivated essentially in two ways:

- the terminal object  $O_\infty$  in a category of machines tends to be “big and complex”, as a consequence of the fact that it is often a terminal coalgebra for a suitably defined endofunctor of  $\mathcal{K}$ , so Adámek’s theorem presents it as inverse limit of an op-chain.
- Coalgebra theory allows us to define a *bisimulation* relation between states of different  $F$ -algebras (or, what is equivalent in our blanket assumptions,  $R$ -coalgebras), which in the case of standard Mealy/Moore machines (i.e., when  $F = \_ \otimes I$ ) recovers the notion of bisimulation expounded in [9, Ch. 3].

The following universal characterisation of both categories as pullbacks in  $\text{Cat}$  allows us to reduce the whole problem of completeness to the computation of a terminal object, and thus prove Theorem 3.6.

► **Proposition 3.5.** ( $\mathcal{U}$ )

CX1) the category  $F\text{-Mly}$  of  $F$ -Mealy machines given in Definition 2.5 can be characterised as the top left corner in the pullback square

$$\begin{array}{ccc} F\text{-Mly} & \xrightarrow{U'} & (F/O) \\ V' \downarrow & \lrcorner & \downarrow V \\ \text{Alg}(F) & \xrightarrow{U} & \mathcal{K} \end{array} \quad (3.5)$$

where  $F/O$  is the comma category defined by  $F$  and the constant functor on  $O$ ,  $V$  is the forgetful functor defined by the universal property of comma categories and  $U$  is the canonical forgetful functor of  $F$ -algebras.

cx2) the category  $F\text{-Mre}$  of  $F$ -Moore machines given in Definition 2.6 can be characterised as the top left corner in the pullback square

$$\begin{array}{ccc} F\text{-Mre} & \xrightarrow{U'} & (\mathcal{K}/O) \\ v' \downarrow & \lrcorner & \downarrow V \\ \text{Alg}(F) & \xrightarrow{U} & \mathcal{K} \end{array} \quad (3.6)$$

where  $V$  is the forgetful functor from the slice category  $\mathcal{K}/O$  to  $\mathcal{K}$ , sending an arrow to its domain and  $U$  is the canonical forgetful functor of  $F$ -algebras.

**Proof.** Straightforward inspection of the definition of both pullbacks.  $\blacktriangleleft$

As a consequence of this characterization, by applying [16, V.6, Ex. 3] we can easily show the following completeness result, provided we recall that in both (3.5) and (3.6)  $U$  is monadic, and since  $F$  is a left adjoint,  $V$  preserves connected limits.

► **Theorem 3.6** (Limits and colimits of  $F$ -machines).  $(\hookrightarrow)^3$

- Let  $\mathcal{K}$  be a category admitting colimits of shape  $\mathcal{J}$ ; then,  $F\text{-Mre}$  and  $F\text{-Mly}$  have colimits of shape  $\mathcal{J}$ , and they are computed as in  $\mathcal{K}$ ;
- Equalizers (and more generally, all connected limits) are computed in  $F\text{-Mre}$  and  $F\text{-Mly}$  as they are computed in  $\mathcal{K}$ ; if  $\mathcal{K}$  has countable products and pullbacks,  $F\text{-Mre}$  and  $F\text{-Mly}$  also have products of any finite cardinality (in particular, a terminal object).

**Proof of Theorem 3.6.** It is worth unraveling the content of [16, V.6, Ex. 3], from which the claim gets enormously simplified: the theorem asserts that in any strict pullback square of categories

$$\begin{array}{ccc} \mathcal{A} & \xrightarrow{U'} & \mathcal{B} \\ v' \downarrow & \lrcorner & \downarrow V \\ \mathcal{C} & \xrightarrow{U} & \mathcal{K} \end{array} \quad (3.7)$$

if  $U$  creates, and  $V$  preserves, limits of a given shape  $\mathcal{J}$ , then  $U'$  creates limits of shape  $\mathcal{J}$ . Thus, thanks to Proposition 3.5, all connected limits (in particular, equalizers) are created in the categories of  $F$ -Mealy and  $F$ -Moore machines by the functors  $U' : F\text{-Mly} \rightarrow (F/O)$  and are thus computed as in  $(F/O)$ , i.e. as in  $\mathcal{K}$ ; this result is discussed at length in [5, Ch. 10] in the case of  $(\_ \otimes I)$ -machines, i.e. classical Mealy machines, to prove the following:

- assuming  $\mathcal{K}$  is cocomplete, all colimits are computed in  $F\text{-Mly}$  as they are computed in the base  $\mathcal{K}$ ;
- assuming  $\mathcal{K}$  has connected limits, they are computed in  $F\text{-Mly}$  as they are computed in the base  $\mathcal{K}$ .

Discrete limits have to be treated with additional care: for classical Moore machines (cf. Definition 2.3) the terminal object is the terminal coalgebra of the functor  $A \mapsto A^I \times O$  (cf. [9, 2.3.5]): a swift application of (the analogue of) Adámek's theorem (for a Cartesian category other than  $\text{Set}$ ) yields the object  $[I^*, O]$ ; for classical Mealy machines (cf. Definition 2.1) the terminal object is the terminal coalgebra for  $A \mapsto [I, O] \times [I, A]$ ; similarly, Adámek's theorem yields  $[I^+, O]$ .

<sup>3</sup> We only provide a mechanization of the proof of existence of finite products: binary products, and a terminal object.

20:8 Completeness for Categories of Generalized Automata

Adámek's theorem then yields the terminal object of  $F\text{-Mre}$  as the terminal coalgebra for the functor  $A \mapsto O \times RA$ , which is the  $O_{\infty,0}$  of Claim 3.4, and the terminal object of  $F\text{-Mly}$  as  $O_{\infty,1}$  and for  $A \mapsto RO \times RA$  (in  $F\text{-Mly}$ ). All discrete limits can be computed when pullbacks and a terminal object have been found, but we prefer to offer a more direct argument to build binary products.

Recall from Construction 3.2 the definition of dynamics map associated to an  $F$ -machine  $\mathfrak{c} = (E, d, s)$ .

Now, our claim is two-fold:

TO1) the object  $O_\infty := \prod_{n \geq 1} R^n O$  in  $\mathcal{K}$  carries a canonical structure of an  $F$ -machine  $\mathfrak{o} = (O_\infty, d_\infty, s_\infty)$  such that  $\mathfrak{o}$  is terminal in  $F\text{-Mly}$ ;

TO2) given objects  $(E, d_E, s_E), (T, d_T, s_T)$  of  $F\text{-Mly}$ , the pullback

$$\begin{array}{ccc} P_\infty & \longrightarrow & T \\ \downarrow & \lrcorner & \downarrow \bar{s}_{T,\infty} \\ E & \xrightarrow{\bar{s}_{E,\infty}} & O_\infty \end{array} \quad (3.8)$$

is the carrier of an  $F$ -machine structure that exhibits  $\mathfrak{p} = (P_\infty, d_P, s_P)$  as the product of  $\mathfrak{c} = (E, d_E, s_E), \mathfrak{f} = (T, d_T, s_T)$  in  $F\text{-Mly}$ .

In this way, the category  $F\text{-Mly}$  comes equipped with all finite products; it is easy to prove a similar statement when an infinite number of objects  $(\mathfrak{c}_i \mid i \in I)$  is given by using wide pullbacks whenever they exist in the base category.

Observe that the object  $P_\infty$  can be equivalently characterized as the single wide pullback obtained from the pullback  $P_n$  of  $\bar{s}_{E,n}$  and  $\bar{s}_{T,n}$  (or rather, an intersection, since each  $P_n \rightarrow E \times T$  obtained from the same pullback is a monomorphism):

$$\begin{array}{ccc} P_\infty & \longrightarrow & P_n \\ \downarrow & \ddots & \downarrow \\ P_m & \longrightarrow & E \times T \end{array} \quad (3.9)$$

Showing the universal property of  $P_\infty$  will be more convenient at different times in one or the other definition.

In order to show our first claim in TO1, we have to provide the  $F$ -machine structure on  $O_\infty$ , exhibiting a span

$$O_\infty \xleftarrow{d_\infty} FO_\infty \xrightarrow{s_\infty} O \quad (3.10)$$

On one side,  $s_\infty$  is the adjoint map of the projection  $\pi_1 : O_\infty \rightarrow RO$  on the first factor; the other leg  $d_\infty$  is the adjoint map of the projection deleting the first factor, thanks to the identification  $RO_\infty \cong \prod_{n \geq 2} R^n O$ ; explicitly then, we are considering the following diagram:

$$O_\infty \xleftarrow{\epsilon_{O_\infty}} FRO_\infty \xleftarrow{F\pi_{\geq 2}} FO_\infty \xrightarrow{F\pi_1} FRO \xrightarrow{\epsilon_O} O \quad (3.11)$$

To prove the first claim, let's consider a generic object  $(E, d, s)$  of  $F\text{-Mly}$ , i.e. a span

$$E \xleftarrow{d} FE \xrightarrow{s} O \quad (3.12)$$

and let's build a commutative diagram

$$\begin{array}{ccccccc} E & \xleftarrow{d} & FE & \xrightarrow{s} & O \\ \downarrow u & & \downarrow Fu & & \parallel \\ O_\infty & \xleftarrow{\epsilon_{O_\infty}} & FRO_\infty & \xleftarrow{F\pi_{\geq 2}} & FO_\infty & \xrightarrow{F\pi_1} & FRO & \xrightarrow{\epsilon_O} & O \end{array} \quad (3.13)$$

for a unique morphism  $u : E \rightarrow O_\infty = \prod_{n \geq 1} R^n O$  that we take exactly equal to  $\bar{s}_\infty$ . The argument that  $u$  makes diagram (3.13) commutative, and that it is unique with this property, is now a completely straightforward diagram chasing.

Now let's turn to the proof that the tip of the pullback in (3.8) exhibits the product of  $(E, d_E, s_E), (T, d_T, s_T)$  in  $F\text{-Mly}$ ; first, we build the structure morphisms  $s_P, d_P$  as follows:

- $d_P$  is the dotted map obtained thanks to the universal property of  $P_\infty$  from the commutative diagram

$$\begin{array}{ccccc}
 FP_\infty & \xrightarrow{\quad} & FE & & \\
 \downarrow & \searrow^{d_P} & \downarrow & \searrow & \\
 & & P_\infty & \xrightarrow{\quad} & E \\
 & & \downarrow & & \downarrow \\
 FT & \xrightarrow{\quad} & FO_\infty & \searrow & O_\infty \\
 & \searrow & \downarrow & & \downarrow \\
 & & T & \xrightarrow{\quad} & O_\infty
 \end{array} \tag{3.14}$$

- $s_P : FP_\infty \rightarrow O$  is obtained as the adjoint map of the diagonal map  $P_\infty \rightarrow O_\infty$  in (3.8) composed with the projection  $\pi_1 : O_\infty \rightarrow RO$ .

Let's now assess the universal property of the object

$$P_\infty \xleftarrow{d_P} FP_\infty \xrightarrow{s_P} O \tag{3.15}$$

We are given an object  $\mathfrak{z} = (Z, d_Z, s_Z)$  of  $F\text{-Mly}$  and a diagram

$$\begin{array}{ccccc}
 O & \xlongequal{\quad} & O & \xlongequal{\quad} & O \\
 s_E \uparrow & & s_Z \uparrow & & s_T \uparrow \\
 FE & \xleftarrow{Fu} & FZ & \xrightarrow{Fv} & FT \\
 d_E \downarrow & & d_Z \downarrow & & d_T \downarrow \\
 E & \xleftarrow{u} & Z & \xrightarrow{v} & T
 \end{array} \tag{3.16}$$

commutative in all its parts. To show that there exists a unique arrow  $[u, v] : Z \rightarrow P_\infty$

$$\begin{array}{ccccc}
 & & Z & & \\
 & \swarrow u & \downarrow [u,v] & \searrow v & \\
 E & \xleftarrow{p_E} & P_\infty & \xrightarrow{p_T} & T
 \end{array} \tag{3.17}$$

we can argue as follows, using the joint injectivity of the projection maps  $\pi_n : O_\infty \rightarrow R^n O$ : first, we show that each square

$$\begin{array}{ccc}
 Z & \xrightarrow{u} & E \\
 v \downarrow & & \downarrow \bar{s}_{E,n} \\
 T & \xrightarrow{\bar{s}_{T,n}} & R^n O
 \end{array} \tag{3.18}$$

is commutative, and in particular that its diagonal is equal to the  $n$ th skip map of  $Z$ ; this can be done by induction, showing that the composition of both edges of the square with the canonical projection  $O_\infty \rightarrow R^n O$  equals  $\bar{s}_{n,Z}$  for all  $n \geq 1$ . From this, we deduce that there exist maps

$$Z \xrightarrow{z_n} P_n \longrightarrow E \times T \tag{3.19}$$

## 20:10 Completeness for Categories of Generalized Automata

(cf. (3.9) for the definition of  $P_n$ ) for every  $n \geq 1$ . But now, the very way in which the  $z_n$ s are defined yields that each such map coincides with  $\langle u, v \rangle : Z \rightarrow E \times T$ , thus  $Z$  must factor through  $P_\infty$ . Now we have to exhibit the commutativity of diagrams

$$\begin{array}{ccccc}
 Z & \xleftarrow{d_Z} & FZ & \xrightarrow{s_Z} & O \\
 \downarrow [u,v] & & \downarrow F[u,v] & & \parallel \\
 P_\infty & \xleftarrow{d_P} & FP_\infty & \xrightarrow{s_P} & O
 \end{array} \tag{3.20}$$

and this follows from a straightforward diagram chasing.

This concludes the proof.  $\blacktriangleleft$

► **Remark 3.7.** Spelled out explicitly, the statement that  $\mathfrak{o} = (O_\infty, d_\infty, s_\infty)$  is a terminal object amounts to the fact that given any other  $F$ -Mealy machine  $\mathfrak{e} = (E, d, s)$ , there is a unique  $u_E : E \rightarrow O_\infty$  with the property that

$$\begin{array}{ccccc}
 E & \xleftarrow{d} & FE & \xrightarrow{s} & O \\
 \downarrow u_E & & \downarrow Fu_E & & \parallel \\
 O_\infty & \xleftarrow{d_\infty} & FO_\infty & \xrightarrow{s_\infty} & O
 \end{array} \tag{3.21}$$

are both commutative diagrams; a similar statement holds for  $F$ -Moore automata.

### 3.1 Adjoints to behaviour functors

In [18, 19] the author concentrates on building an adjunction between a category of machines and a category collecting the *behaviours* of said machines.

Call an endofunctor  $F : \mathcal{K} \rightarrow \mathcal{K}$  an *input process* if the forgetful functor  $U : \mathbf{Alg}(F) \rightarrow \mathcal{K}$  has a left adjoint  $G$ ; in simple terms, an input process allows to define free  $F$ -algebras.<sup>4</sup>

In [18, 19] the author concentrates on proving the existence of an adjunction

$$L : \mathbf{Beh}(F) \xrightleftharpoons[\perp]{} \mathbf{Mach}(F) : E \tag{3.22}$$

where  $\mathbf{Mach}(F)$  is the category obtained from the pullback

$$\begin{array}{ccc}
 \mathbf{Mach}(F) & \longrightarrow & \mathcal{K}^{\rightarrow} \times \mathcal{K}^{\rightarrow} \\
 \downarrow & \lrcorner & \downarrow d_1 \times d_0 \\
 \mathbf{Alg}(F) & \xrightarrow{U} & \mathcal{K} \xrightarrow{\Delta} \mathcal{K} \times \mathcal{K}
 \end{array} \tag{3.23}$$

$\Delta$  is the diagonal functor,  $\mathbf{Beh}(F)$  is a certain comma category on the free  $F$ -algebra functor  $G$  and  $d_0, d_1$  are the domain and codomain functors from the arrow category.

Phrased in this way, the statement is conceptual enough to carry over to  $F$ -Mealy and  $F$ -Moore machines (and by extension, to all settings where a category of automata can be presented through a strict 2-pullback in  $\mathbf{Cat}$  of well-behaved functors – a situation that given (3.5), (3.6), (3.23) arises quite frequently).

<sup>4</sup> Obviously, this is in stark difference with the requirement that  $F$  has an adjoint, and the two requests are independent: if  $F$  is a monad, it is always an input process, regardless of  $F$  admitting an adjoint on either side.



► **Theorem 3.8.** ( $\Upsilon$ ) *There exists a functor  $B : F\text{-Mre} \rightarrow \text{Alg}(F)_{/(O_\infty, d_\infty)}$ , where the codomain is the slice category of  $F$ -algebras and the  $F$ -algebra  $(O_\infty, d_\infty)$  is determined in Claim 3.4. The functor  $B$  has a left adjoint  $L$ .*

**Proof.** An object of  $\text{Alg}(F)_{/O_\infty}$  is a tuple  $((A, a), u)$  where  $a : FA \rightarrow A$  is an  $F$ -algebra with its structure map, and  $u : A \rightarrow O_\infty$  is an  $F$ -algebra homomorphism, i.e. a morphism  $u$  such that  $d_\infty \circ Fu = u \circ a$ .

The functor  $B$  is defined as follows:

- on objects  $\epsilon = (E, d, s)$  in  $F\text{-Mre}$ , as the correspondence sending  $\epsilon$  to the unique map  $u_E : E \rightarrow O_\infty$ , which is an  $F$ -algebra homomorphism by the construction in (3.13);
- on morphisms,  $f : (E, d, s) \rightarrow (F, d', s')$  between  $F$ -Moore machines,  $B$  acts as the identity, ultimately as a consequence of the fact that the terminality of  $O_\infty$  yields at once that  $u_F \circ f = u_E$ .

A putative left adjoint for  $B$  realises a natural bijection

$$F\text{-Mre}_{/O} (L((A, a), u), (E, d, s)) \cong \text{Alg}(F)_{/O_\infty} (((A, a), u), B(E, d, s)) \tag{3.24}$$

between the following two kinds of commutative diagrams:

$$\left\{ \begin{array}{c} \begin{array}{ccccc} A & \xleftarrow{a} & FA & & \\ \varphi \downarrow & & \downarrow F\varphi & & \\ E & \xleftarrow{d} & FE & & \\ \varphi \downarrow & & \downarrow & & \\ A & \xrightarrow{u} & O_\infty & \xrightarrow{s_T} & O \\ \varphi \downarrow & \nearrow u_E & & & \parallel \\ E & \xrightarrow{s} & O & & \end{array} \\ \end{array} \right\} \longleftrightarrow \left\{ \begin{array}{c} \begin{array}{ccccc} & & FA & \xrightarrow{F\varphi} & FE \\ & & \downarrow a & & \downarrow d \\ & & A & \xrightarrow{\varphi} & E \\ Fu \swarrow & & \downarrow u & & \downarrow s \\ FO_\infty & & O_\infty & \xrightarrow{s_T} & O \\ & \searrow d_T & & & \end{array} \\ \end{array} \right\} \tag{3.25}$$

There is a clear way to establish this correspondence. ◀

► **Remark 3.9.** ( $\Upsilon$ ) The adjunction in Theorem 3.8 is actually part of a longer chain of adjoints obtained as follows: recall that every adjunction  $G : \mathcal{K} \rightleftarrows \mathcal{H} : U$  induces a “local” adjunction  $\tilde{G} : \mathcal{K}_{/UA} \rightleftarrows \mathcal{H}_{/A} : \tilde{U}$  where  $\tilde{U}(FA, f : FA \rightarrow A) = Uf$ . Then, if  $F$  is an input process, we get adjunctions

$$\mathcal{K}_{/O_\infty} \xrightarrow{\tilde{G}} \text{Alg}(F)_{/(O_\infty, d_\infty)} \xleftarrow{\frac{1}{B}} F\text{-Mre}. \tag{3.26}$$

## 4 Conclusion and Future Works

Our research is part of a bigger ongoing project [2] aimed to understand automata theory from the point of view of formal category theory [6, 24, 25]. The endeavour has a long history (the work of Naudé that we generalize a bit serves as a remarkable example in this direction), and the technology of category-theoretic approaches is rapidly shifting towards 2-dimensional categories as foundations for complex systems [15, 17, 20]. By leveraging simple universal properties of pullbacks and comma objects in  $\text{Cat}$ , we have established a way for generating “categories of automata and their behaviour”.

In fact, our findings hint at the existence of exciting possibilities for understanding behavior coalgebraically within categories of automata. This approach, well-known and fruitful in the literature, has been extensively studied by Jacobs [9, 8]. We are confident that we can extend this line of research to derive insightful statements in the “internal

language” of the category of automata under consideration. For instance, we can examine bisimilarity as an internal equivalence relation in our categories of generalised automata, utilizing the calculus of relations available in every regular category, and categorical algebra, broadly intended. In our opinion, this exploration holds great potential for deepening our understanding of automata theory and its applications.

In future works, we would like to further explore the properties of the adjunctions sketched in this paper, emphasizing on applications. We also plan to delve deeper into the “coalgebraic behavior” perspective, with particular care for its implications in different aspects of automata theory. In [2] we exploit the fact that Mealy automata form a bicategory, building on prior work [11]: it is a banality that the composition of 1-cells in such a bicategory amounts to the so-called *cascade product* between a Mealy machine and a semiautomaton. Among many different direction for future research, an exciting prospect is to prove the Krohn-Rhodes theorem [12, 22, 23] by resorting to bicategorical properties.

Besides providing a guarantee of correctness, formalizing our results in a proof assistant might also pave the way for “concrete” implementations of our theoretical results, where, for instance, the proofs also act as concrete programs that allow the user to convert between different automata in a provably correct way.

Overall, we believe our research started a foundational look to automata theory by offering a novel perspective on known results. As category theorists, we are confident that approaching familiar concepts from a higher vantage point yields invaluable insights, fostering the advancement of the field and unlocking practical applications.

---

## References

- 1 J. Adámek and V. Trnková. *Automata and Algebras in Categories*. Kluwer, 1990.
- 2 G. Boccali, A. Laretto, F. Loregian, and S. Luneia. Bicategories of automata, automata in bicategories, 2023. [arXiv:2303.03865](https://arxiv.org/abs/2303.03865).
- 3 F. Borceux. *Handbook of categorical algebra. 2*, volume 51 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 1994. Categories and structures.
- 4 M. Droste, W. Kuich, and H. Vogler, editors. *Handbook of Weighted Automata*. Springer Berlin Heidelberg, 2009. doi:10.1007/978-3-642-01492-5.
- 5 H. Ehrig, K.-D. Kiermeier, H.-J. Kreowski, and W. Kühnel. *Universal theory of automata. A categorical approach*. XTeubner Studienbücher Informatik. Vieweg+Teubner Verlag Wiesbaden, 1974. doi:10.1007/978-3-322-96644-5.
- 6 J. W. Gray. *Formal Category Theory: Adjointness for 2-Categories*. Springer Berlin Heidelberg, 1974. doi:10.1007/bfb0061280.
- 7 R. Guitart. Tenseurs et machines. *Cahiers de topologie et géométrie différentielle*, 21(1):5–62, 1980. URL: [http://www.numdam.org/item/CTGDC\\_1980\\_\\_21\\_1\\_5\\_0/](http://www.numdam.org/item/CTGDC_1980__21_1_5_0/).
- 8 B. Jacobs. A bialgebraic review of deterministic automata, regular expressions and languages. In *Algebra, Meaning, and Computation*, pages 375–404. Springer Berlin Heidelberg, 2006.
- 9 B. Jacobs. *Introduction to Coalgebra: Towards Mathematics of States and Observation*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2016.
- 10 P. T. Johnstone. *Topos theory*. London Mathematical Society Monographs. No.10. London - New York -San Francisco: Academic Press. XXIII, 367 p. 5 , 1977.
- 11 P. Katis, N. Sabadini, and R.F.C. Walters. Bicategories of processes. *Journal of Pure and Applied Algebra*, 115(2):141–178, February 1997. doi:10.1016/s0022-4049(96)00012-6.
- 12 Kenneth Krohn and John Rhodes. Algebraic theory of machines. i. prime decomposition theorem for finite semigroups and machines. *Transactions of the American Mathematical Society*, 116(0):450–464, 1965. doi:10.1090/s0002-9947-1965-0188316-1.
- 13 F. W. Lawvere. Categories of spaces may not be generalized spaces as exemplified by directed graphs. *Revista colombiana de matemáticas*, 20(3-4):179–186, 1986.

- 14 F. W. Lawvere. Axiomatic cohesion. *Theory and Applications of Categories*, 19:41–49, 2007.
- 15 A. H. Louie. Categorical system theory. *Bulletin of Mathematical Biology*, 45(6):1047–1072, November 1983. doi:10.1007/bf02458830.
- 16 S. Mac Lane. *Categories for the Working Mathematician*, volume 5 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, second edition, 1998. doi:10.1007/978-1-4757-4721-8.
- 17 D. J. Myers. Double categories of open dynamical systems (extended abstract). *Electronic Proceedings in Theoretical Computer Science*, 333:154–167, February 2021.
- 18 G. Naudé. On the adjoint situations between behaviour and realization. *Quaestiones Mathematicae*, 2:245–267, 1977. doi:10.1080/16073606.1977.9632546.
- 19 G. Naudé. Universal realization. *Journal of Computer and System Sciences*, 19(3):277–289, 1979. doi:10.1016/0022-0000(79)90005-9.
- 20 P. Schultz, D. I. Spivak, and C. Vasilakopoulou. Dynamical systems and sheaves. *Applied Categorical Structures*, 28(1):1–57, April 2019. doi:10.1007/s10485-019-09565-x.
- 21 David I. Spivak. *Category Theory for the Sciences*. The MIT Press, 2014.
- 22 Howard Straubing. *Finite Automata, Formal Logic, and Circuit Complexity*. Birkhäuser Boston, 1994. doi:10.1007/978-1-4612-0289-9.
- 23 Charles Wells. A krohn-rhodes theorem for categories. *Journal of Algebra*, 64(1):37–45, May 1980. doi:10.1016/0021-8693(80)90130-1.
- 24 R. J. Wood. Abstract pro arrows I. *Cahiers de topologie et géométrie différentielle*, 23(3):279–290, 1982. URL: [http://www.numdam.org/item/CTGDC\\_1982\\_\\_23\\_3\\_279\\_0/](http://www.numdam.org/item/CTGDC_1982__23_3_279_0/).
- 25 R. J. Wood. Proarrows II. *Cahiers de Topologie et Géométrie Différentielle Catégoriques*, 26(2):135–168, 1985. URL: [http://www.numdam.org/item/CTGDC\\_1985\\_\\_26\\_2\\_135\\_0/](http://www.numdam.org/item/CTGDC_1985__26_2_135_0/).

## **A** Agda formalization

Here we briefly comment here on our use of the Agda proof assistant to formalize some of the main results of this paper.

In most cases we used it to formalize the most tricky aspects of the proofs, without focusing on providing a complete formalization of all results shown in this work, for which the pen-and-paper approach still has a considerable edge in terms of speed and effort. For example, the proof mechanized for Theorem 3.6 concentrates only on explicitly defining terminal and binary products, thus providing only a general insight on how non-connected limits are computed. Our development consists of around 2000 LoC and, thanks to its reusability, has been employed to formalize results in subsequent papers such as [2].

We use the library `agda-categories` as a starting point from which to build and prove further theorems, without having to formalize basic notions of category theory from scratch. Most of the proofs mechanized for this paper are straightforward and follow directly from the universal properties of the objects under consideration; the most difficult part of our development has been to identify the necessary properties to prove facts about inductively defined objects (e.g., the interdependencies between the different lemma needed in Theorem 3.6) and the lack of automation mechanisms to close the proofs, which can end up in particularly long sequences of hom-reasoning steps.

Other minor issues arise from some architectural choices made in the `agda-categories` library, which, following a well-established practice in formalizations of category theory, defines categories as setoid-enriched, i.e., every category incorporates an internal notion of equality between morphisms. This often results in better-behaved but weaker notions of equalities between morphisms that more closely follow the principle of equivalence; for example, in the large category `Cat`, equality of functors is defined as natural isomorphism between functors, rather than strict equality on objects and arrows. This becomes problematic when defining universal objects in `Cat`, such as the (strict) 2-pullbacks used in Proposition 3.5

## 20:14 Completeness for Categories of Generalized Automata

to characterize the categories  $F\text{-Mly}$  and  $F\text{-Mre}$ , since in this picture limits are actually defined up to equivalence of categories –from the theoretical point of view, they are *bilimits*; from the implementation point of view, the weak universal property is due to the lack of uniqueness of identity proofs for arbitrary hom-equalities.

In practice this has been dealt with by working in the (large) category `StrictCat` where equality of functors is defined strictly, which allows us to recover pullbacks between categories and the characterizations shown in this paper.



# On Kripke, Vietoris and Hausdorff Polynomial Functors

Jiří Adámek 

Czech Technical University in Prague, Czech Republic  
Technische Universität Braunschweig, Germany

Stefan Milius  

Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

Lawrence S. Moss  

Indiana University, Bloomington, IN, USA

---

## Abstract

The Vietoris space of compact subsets of a given Hausdorff space yields an endofunctor  $\mathcal{V}$  on the category of Hausdorff spaces. Vietoris polynomial endofunctors on that category are built from  $\mathcal{V}$ , the identity and constant functors by forming products, coproducts and compositions. These functors are known to have terminal coalgebras and we deduce that they also have initial algebras. We present an analogous class of endofunctors on the category of extended metric spaces, using in lieu of  $\mathcal{V}$  the Hausdorff functor  $\mathcal{H}$ . We prove that the ensuing Hausdorff polynomial functors have terminal coalgebras and initial algebras. Whereas the canonical constructions of terminal coalgebras for Vietoris polynomial functors take  $\omega$  steps, one needs  $\omega + \omega$  steps in general for Hausdorff ones. We also give a new proof that the closed set functor on metric spaces has no fixed points.

**2012 ACM Subject Classification** Theory of computation  $\rightarrow$  Models of computation; Theory of computation  $\rightarrow$  Logic and verification

**Keywords and phrases** Hausdorff functor, Vietoris functor, initial algebra, terminal coalgebra

**Digital Object Identifier** 10.4230/LIPIcs.CALCO.2023.21

**Category** (Co)algebraic pearls

**Related Version** *ArXiv version with omitted proofs*: <https://arxiv.org/abs/2303.11071>

**Funding** *Jiří Adámek*: Supported by the grant No. 22-02964S of the Czech Grant Agency.

*Stefan Milius*: Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project number 470467389.

*Lawrence S. Moss*: Supported by grant #586136 from the Simons Foundation.

**Acknowledgements** We are grateful to Pedro Nora for discussions on the proof of Prop. 4.7.

## 1 Introduction

This paper presents results on terminal coalgebras and initial algebras for certain endofunctors on the categories **Haus** of Hausdorff topological spaces and **Met** of extended metric spaces. These results are based on the terminal coalgebra construction first presented by Adámek [2] (in dual form) and independently by Barr [8]. Given an endofunctor  $F$ , iterate  $F$  on the unique morphism  $! : F1 \rightarrow 1$  to obtain the following  $\omega^{\text{op}}$ -chain

$$1 \xleftarrow{!} F1 \xleftarrow{F!} FF1 \xleftarrow{FF!} FFF1 \xleftarrow{FFF!} \dots \quad (1)$$

Assume that the limit exists, and denote it by  $V_\omega$  and the limit cone by  $\ell_n : V_\omega \rightarrow F^n 1$  ( $n < \omega$ ). We obtain a unique morphism  $m : FV_\omega \rightarrow V_\omega$  such that for all  $n \in \omega^{\text{op}}$  we have

$$F\ell_n = (FV_\omega \xrightarrow{m} V_\omega \xrightarrow{\ell_{n+1}} F^{n+1}1). \quad (2)$$



© Jiří Adámek, Stefan Milius, and Lawrence S. Moss;  
licensed under Creative Commons License CC-BY 4.0

10th Conference on Algebra and Coalgebra in Computer Science (CALCO 2023).

Editors: Paolo Baldan and Valeria de Paiva; Article No. 21; pp. 21:1–21:20

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

If  $F$  preserves the limit  $V_\omega$ , then  $m$  is an isomorphism. Its inverse yields the terminal coalgebra  $m^{-1}: V_\omega \rightarrow FV_\omega$ ; shortly  $\nu F = V_\omega$ , and we say that the terminal coalgebra is *obtained in  $\omega$  steps*.

This technique of *finitary iteration* is the most basic and prominent construction of terminal coalgebras. However finitary iteration requires that the limit in (1) exists and also that it is preserved by the functor. It does *not* apply on  $\mathbf{Set}$  to the finite power-set functor  $\mathcal{P}_f$ . For that functor  $FV_\omega \not\cong V_\omega$ . However, a modification of finitary iteration does apply, as shown by Worrell [23]. One makes a *second infinite iteration*, iterating  $F$  on the morphism  $m: FV_\omega \rightarrow V_\omega$  rather than on  $!: F1 \rightarrow 1$ , obtaining a chain

$$V_\omega \xleftarrow{m} V_{\omega+1} \xleftarrow{Fm} V_{\omega+2} \xleftarrow{FFm} \dots \tag{3}$$

Its limit is denoted by  $V_{\omega+\omega} = \lim_{n < \omega} V_{\omega+n}$  with the limit cone  $\bar{l}_n: V_{\omega+\omega} \rightarrow V_{\omega+n}$ , for  $n < \omega$ . Worrell’s insight was that this second limit,  $V_{\omega+\omega}$ , is preserved by every finitary functor. We prove that this also works for set functors built from  $\mathcal{P}_f$  using product, coproduct, and composition (which may be non-finitary). These are the *Kripke polynomial functors* mentioned in our title.

We are interested in other settings where terminal coalgebras may be built using either the limit of (1) or the limit of (3). We study fixed points of naturally occurring endofunctors on Hausdorff spaces and metric spaces, endofunctors built from the Vietoris and Hausdorff functors and several other natural constructions.

In the category  $\mathbf{Top}$  a good analogy of  $\mathcal{P}_f$  is the *Vietoris functor*  $\mathcal{V}$  assigning to every space  $X$  the space of all compact subsets equipped with the Vietoris topology (Section 4). Hofmann et al. [11] define *Vietoris polynomial functors* as those endofunctors on  $\mathbf{Top}$  built from  $\mathcal{V}$ , the constant functors, and the identity functor, using product, coproduct, and composition. We study this on the subcategory  $\mathbf{Haus}$  of Hausdorff spaces and use that  $\mathcal{V}: \mathbf{Haus} \rightarrow \mathbf{Haus}$  preserves limits of  $\omega^{\text{op}}$ -chains, a fact for which we present a new proof. This implies that for *Vietoris polynomial functors* (defined as above but with  $\mathcal{V}$  in lieu of  $\mathcal{P}_f$ ), the terminal coalgebra exists and is the limit of (1). The original proof [11] uses a result by Zenor [24] whose proof is incomplete. The existence of initial algebras follows.

We also present a result for the category  $\mathbf{Met}$  of metric spaces and nonexpanding maps. The role of the Vietoris functor is played by the *Hausdorff functor*  $\mathcal{H}$  assigning to every space  $X$  the space  $\mathcal{H}X$  of all compact subsets with the Hausdorff metric.

**Other contributions.** In addition to the aforementioned results we show results obtained by either varying the category or the endofunctor. For example, consider again the Hausdorff polynomial functors. Whenever  $F$  is such a functor and the constants involved in its construction are complete spaces,  $\nu F$  again turns out to be complete. Analogous results hold for compact spaces, or ultrametric spaces. Finally, we present a proof of the description of  $\nu \mathcal{P}_f$  and  $V_\omega$  for  $\mathcal{P}_f$  exhibited by Worrell [23] (the latter without a proof).

We simplify a proof of a known negative result: the variation of  $\mathcal{H}$  obtained by moving from compact sets to closed sets has no fixed points.

**Related work.** Our work is more general and hence improves results of Abramsky [1], Hofmann et al. [11], and Worrell [23].

There are numerous results about the existence and construction of terminal coalgebras in the literature. At several places we discuss other possible approaches to our results.

## 2 Preliminaries

We review a few preliminary points. We assume that readers are familiar with basic notions of category theory as well as algebras and coalgebras for an endofunctor. We denote by **Set** the category of sets and functions, **Top** is the category of topological spaces and continuous functions, and **Met** is the category of (*extended*) *metric spaces* (so we might have  $d(x, y) = \infty$ ) and non-expanding maps: the functions  $f: X \rightarrow Y$  where  $d(f(x), f(x')) \leq d(x, x')$  holds for every pair  $x, x' \in X$ . Note that this class of morphisms is smaller than the class of continuous functions between metric spaces.

► **Remark 2.1.** Consider an  $\omega^{\text{op}}$ -chain

$$X_0 \xleftarrow{f_0} X_1 \xleftarrow{f_1} X_2 \xleftarrow{f_2} \dots \tag{4}$$

1. In **Set**, the limit  $L$  consists of all sequences  $(x_n)_{n < \omega}$ ,  $x_n \in X_n$  that are *compatible*:  $f_n(x_{n+1}) = x_n$  for every  $n$ . The limit projections are the functions  $\ell_n: L \rightarrow X_n$  defined by  $\ell_n((x_i)) = x_n$ .
2. In **Top**, the limit is again carried by the same set  $L$  as in **Set**, and the limit projections  $\ell_n$  are also the same. The topology on  $L$  has as a base the sets  $\ell_n^{-1}(U)$ , for  $U$  open in  $X_n$ .
3. In **Met**, the limit is again carried by the same set  $L$ , and the same limit projections  $\ell_n$ . The metric on  $L$  is defined by  $d((x_n), (y_n)) = \sup_{n < \omega} d(x_n, y_n)$ .

**Smooth Monomorphisms.** In addition to terminal coalgebras, we also study initial algebras for the functors of interest in this paper. For this, we call on a general result which allows one to infer the existence of the initial algebra for an endofunctor  $F$  from the existence of a terminal coalgebra for  $F$  (or in fact of any algebra with monic structure).

For a class  $\mathcal{M}$  of monomorphisms we denote by  $\text{Sub}_{\mathcal{M}}(A)$  the collection of subobjects of  $A$  represented by monomorphisms from  $\mathcal{M}$ . To say that this is a dcpo means that it is a set which (when ordered by factorization in the usual way) is a poset having directed joins.

► **Definition 2.2** [4, Def. 3.1]. Let  $\mathcal{M}$  be a class of monomorphisms closed under isomorphisms and composition.

1. We say that an object  $A$  has *smooth  $\mathcal{M}$ -subobjects* provided that  $\text{Sub}_{\mathcal{M}}(A)$  is a dcpo with bottom  $\perp$ , where the least element and directed joins are given by colimits of the corresponding diagrams of subobjects.
2. The class  $\mathcal{M}$  is *smooth* if every object of  $\mathcal{A}$  has smooth  $\mathcal{M}$ -subobjects.

A category has *smooth monomorphisms* if the class of all monomorphisms is smooth.

► **Example 2.3.**

1. The categories **Set** and **Top** have smooth monomorphisms, and so does the full subcategory of Hausdorff spaces. This is easy to see.
2. The category **Met** also has smooth monomorphisms (these are the injective non-expanding maps) [4, Lemma A.1].

The full subcategory **CMS** of complete metric spaces does not have smooth monomorphisms. However, strong monomorphisms (isometric embeddings) are smooth in both **Met** and **CMS** [4, Lemma A.2].

3. Strong monomorphisms (subspace embeddings) in **Top** are not smooth [3, Ex. 3.5].

► **Theorem 2.4** [4, Cor. 4.4]. *Let  $\mathcal{M}$  be a smooth class of monomorphisms. If an endofunctor  $F$  preserving  $\mathcal{M}$  has a terminal coalgebra, then it has an initial algebra.*

Note that loc. cit. states more: given any algebra  $m: FA \rightarrow A$  where  $m$  lies in  $\mathcal{M}$ , the initial algebra exists and is a subalgebra of  $(A, m)$ .

### 3 Kripke Polynomial Functors

We turn to the first collection of functors mentioned in the title of this paper: the Kripke polynomial functors on  $\mathbf{Set}$ . The name stems from *Kripke structures* used in modal logic. Our definition below is a slight generalization of the (finite) Kripke polynomial functors presented by Jacobs [12, Def. 2.2.1]. (Kripke polynomial functors using the full power-set functor were originally introduced by Rößiger [19].) We admit arbitrary products in lieu of just arbitrary exponents.

► **Definition 3.1.** The *Kripke polynomial functors*  $F$  are the set functors built from the finite power-set functor, constant functors and the identity functor, by using product, coproduct and composition. In other words, Kripke polynomial functors are built according to the following grammar:

$$F ::= \mathcal{P}_f \mid A \mid \text{Id} \mid \prod_{i \in I} F_i \mid \coprod_{i \in I} F_i \mid FF,$$

where  $A$  ranges over all sets (and is interpreted as a constant functor) and  $I$  is an arbitrary index set.

► **Remark 3.2.** The constant functors could be omitted from the grammar since they are obtainable from the rest of the grammar. The constant functor with value 1 is the empty product. For each set  $A$ , the constant functor with value  $A$  is then a coproduct:  $A = \coprod_{a \in A} 1$ .

► **Example 3.3.** The Kripke polynomial functor  $FX = \mathcal{P}_f(A \times X)$  is the type functor of finitely branching labelled transition systems with a set  $A$  of actions.

► **Remark 3.4.** An endofunctor is *finitary* if it preserves directed colimits. Worrell [23] proved that for every finitary set functor the terminal coalgebra is obtained in  $\omega + \omega$  steps. We prove a version of Worrell's result but for Kripke polynomial functors.

There are Kripke polynomial set functors which are not finitary. One example of such a functor is  $F(X) = X^{\mathbb{N}}$ , where  $\mathbb{N}$  is the set of natural numbers. There are also finitary set functors which are not Kripke polynomial functors. One example is the functor assigning to a set  $X$  the set of *nonempty* finite subsets of  $X$ .

Our proof below uses ideas from Worrell's work [23].

► **Theorem 3.5.** *Every Kripke polynomial functor  $F$  has a terminal coalgebra obtained in  $\omega + \omega$  steps:  $\nu F = V_{\omega+\omega}$ .*

**Proof.**

1. We first observe that  $F$  preserves monomorphisms and intersections of monomorphisms. This is clear for constant functors and for  $\text{Id}$ , and it is easy to see for  $\mathcal{P}_f$ . Moreover, these properties are clearly preserved by product, coproduct and composition.
2. Let  $(X_n)_{n < \omega}$  be an  $\omega^{\text{op}}$ -chain in  $\mathbf{Set}$ . Then the canonical morphism  $m: F(\lim X_n) \rightarrow \lim FX_n$  is monic. This is obvious for constant functors and  $\text{Id}$ . Let us check it for  $\mathcal{P}_f$ . Denote the limit projections by  $\ell_n: \lim X_n \rightarrow X_n$  and  $p_n: \lim \mathcal{P}_f X_n \rightarrow \mathcal{P}_f X_n$  ( $n < \omega$ ); the canonical morphism  $m$  is unique such that  $p_n \cdot m = \mathcal{P}_f \ell_n$ . Now given  $S \neq T$  in  $\mathcal{P}_f(\lim X_n)$ , without loss of generality we can pick  $x \in T \setminus S$ . Using that the  $\ell_n$  are jointly monic, for every  $s \in S$  we can choose  $n < \omega$  such that  $\ell_n(x) \neq \ell_n(s)$ . Since  $S$  is finite, this choice can be performed independently of  $s \in S$ . Thus  $\ell_n(x) \notin \ell_n[S]$ , and hence  $\mathcal{P}_f \ell_n(T) \neq \mathcal{P}_f(S)$ . Thus,  $\mathcal{P}_f \ell_n$  is a jointly monic family. Since  $p_n \cdot m = \mathcal{P}_f \ell_n$ , we see that  $m$  is monic.



3. An induction on Kripke polynomial functors  $F$  now shows that  $m: V_{\omega+1} \rightarrow V_\omega$  is monic. We have seen this for the base case functors in item 2. The desired property that  $m$  is monic is preserved by products, coproducts and composition. In particular, for a composition  $FG$  note that the canonical morphism for  $FG$  is the composition

$$FG(\lim X_n) \xrightarrow{Fm} F(\lim GX_n) \xrightarrow{m'} \lim FGX_n,$$

where  $m$  is the canonical morphism for  $G$  w.r.t. the given  $\omega^{\text{op}}$ -chain and  $m'$  the one for  $F$  and the  $\omega^{\text{op}}$ -chain  $(GX_n)_{n < \omega}$ . So this morphism  $m' \cdot Fm$  is monic since both  $m$  and  $m'$  are so and  $F$  preserves monomorphisms by item 1.

4. Since  $F$  preserves monomorphisms, we see that  $Fm, FFm$  etc. are monic. We obtain a decreasing chain of subobjects  $V_{\omega+n} \twoheadrightarrow V_\omega$ . Therefore, the limit  $V_{\omega+\omega} = \lim_{n < \omega} V_{\omega+n}$  is simply the intersection of these subobjects. From item 1 we know that  $F$  preserves this limit. It follows that  $\nu F = V_{\omega+\omega}$ , as desired.  $\blacktriangleleft$

► **Corollary 3.6.** *Every Kripke polynomial functor  $F$  on  $\text{Set}$  has an initial algebra.*

This follows from Theorem 3.5, Example 2.3.1, and Theorem 2.4 since  $F$  preserves monomorphisms.

► **Example 3.7** [23]. The functor  $\mathcal{P}_f$  has a terminal coalgebra consisting of all finitely branching strongly extensional trees (up to isomorphism of trees). Moreover, the limit  $V_\omega$  consists of all compactly branching strongly extensional trees. We present a proof of these results in Appendix A (Theorem A.15).

## 4 Vietoris Polynomial Functors

Hofmann et al. [11] proved that Vietoris polynomial functors on the category  $\text{Haus}$  of Hausdorff spaces have terminal coalgebras obtained in  $\omega$  steps. Our proof is slightly different from theirs because we wish to avoid a result stated by Zenor [24] whose proof is incomplete.

Recall that a *base* of a topology is a collection  $\mathcal{B}$  of open sets such that every open set is a union of members of  $\mathcal{B}$ . A *subbase* is a collection of open sets whose finite intersections form a base. For every collection  $\mathcal{B}$  of subsets of the space, there is a smallest topology for which  $\mathcal{B}$  is a (sub)base, the family of unions of finite intersections from  $\mathcal{B}$ .

- **Definition 4.1.** 1. Let  $X$  be a topological space. We denote by  $\mathcal{V}X$  the space of compact subsets of  $X$  equipped with the “hit-and-miss” topology. This topology has as a subbase all sets of the following forms:

$$\begin{aligned} U^\diamond &= \{R \in \mathcal{V}X : R \cap U \neq \emptyset\} && (R \text{ hits } U), \\ U^\square &= \{R \in \mathcal{V}X : R \subseteq U\} && (R \text{ misses } X \setminus U), \end{aligned} \tag{5}$$

where  $U$  ranges over the open sets of  $X$ . We call  $\mathcal{V}X$  the *Vietoris space of  $X$* , also known as the *hyperspace of  $X$* .

2. Recalling that the image of a compact set under a continuous function is also compact, for a continuous function  $f: X \rightarrow Y$  we put  $\mathcal{V}f(A) = f[A]$  for every compact subset  $A$  of  $X$ .

► **Remark 4.2.**

1. For a compact Hausdorff space  $X$ , Vietoris [22] defined  $\mathcal{V}X$  to consist of all *closed* subsets of  $X$ . These are the same as the compact subsets in this case. In the coalgebraic literature,  $\mathcal{V}X$  has also mostly been studied for spaces  $X$  which are compact Hausdorff. However,

the “classic Vietoris space” (using closed subsets) does not yield a functor on  $\mathbf{Top}$  (see Hofmann et al. [11, Rem. 2.28]). Hofmann et al. [11, Def. 2.27] call the functor  $\mathcal{V}$  in Definition 4.1 the *compact Vietoris functor*.

2. Michael [16, Thm. 4.9.8] proved that  $X$  is Hausdorff iff so is  $\mathcal{V}X$ .
3. Vietoris [22] originally proved that for a compact Hausdorff space  $X$  (the classic Vietoris space)  $\mathcal{V}X$  is compact Hausdorff, too.
4. A *Stone space* is a compact Hausdorff space having a base of clopen sets. If  $X$  is a Stone space, so is  $\mathcal{V}X$ ; see [16, Thm. 4.9.9] or [13, Section III.4].

► **Proposition 4.3.** *For every continuous function  $f: X \rightarrow Y$  and every open  $U \subseteq Y$ ,  $(f^{-1}(U))^\diamond = (\mathcal{V}f)^{-1}(U^\diamond)$ , and  $(f^{-1}(U))^\square = (\mathcal{V}f)^{-1}(U^\square)$ .*

**Proof.** Let  $R \in \mathcal{V}X$ . Observe that

$$R \cap f^{-1}(U) \neq \emptyset \iff f[R] \cap U \neq \emptyset \iff f[R] \in U^\diamond \iff R \in (\mathcal{V}f)^{-1}(U^\diamond).$$

This proves our first assertion for all  $R$ . For the second assertion, we have

$$R \subseteq f^{-1}(U) \iff f[R] \subseteq U \iff f[R] \in U^\square \iff R \in (\mathcal{V}f)^{-1}(U^\square). \quad \blacktriangleleft$$

► **Corollary 4.4.** *The mappings  $X \mapsto \mathcal{V}X$  and  $f \mapsto \mathcal{V}f$  form a functor  $\mathcal{V}$  on  $\mathbf{Top}$ .*

Indeed, Proposition 4.3 shows that for every subbasic open set of  $\mathcal{V}Y$  its inverse image under  $\mathcal{V}f$  is open in  $\mathcal{V}X$ . This establishes continuity of  $\mathcal{V}f$ .

► **Notation 4.5.** We denote by **Haus**, **KHaus** and **Stone** the full subcategories of  $\mathbf{Top}$  given by all Hausdorff spaces, all compact Hausdorff spaces and all Stone spaces, respectively. By Remark 4.2.2–4,  $\mathcal{V}$  restricts to these three full subcategories, and we denote the restrictions by  $\mathcal{V}$  as well.

► **Remark 4.6.**

1. The full subcategories **Haus**, **KHaus** and **Stone** are closed under limits in  $\mathbf{Top}$ . In particular, the inclusion functors preserve and reflect limits. In fact, **KHaus** is a full reflective subcategory: the reflection of a space is its Stone-Čech compactification.
2. If an  $\omega^{\text{op}}$ -chain as in (4) consists of surjective continuous maps between compact Hausdorff spaces, then each limit projection  $\ell_n: \lim_{k < \omega} X_k \rightarrow X_n$  is surjective, too. Moreover, Eilenberg and Steenrod [9, Cor. 3.9] prove the surjectivity of projections for all codirected limits of surjections between compact Hausdorff spaces; see also Ribes and Zalesskii [18, Prop. 1.1.10]).
3. If  $X$  has a base  $\mathcal{B}$  which is closed under finite unions, then the sets  $U^\diamond$  and  $U^\square$  for  $U \in \mathcal{B}$  already form a subbase of  $\mathcal{V}X$ . Indeed, given a set  $\mathcal{S}$  of open subsets of  $X$  we have  $(\bigcup \mathcal{S})^\diamond = \bigcup \{U^\diamond : U \in \mathcal{S}\}$ . Moreover, it is easy to see that

$$(\bigcup \mathcal{S})^\square = \bigcup \{(\bigcup \mathcal{F})^\square : \mathcal{F} \subseteq \mathcal{S} \text{ finite}\};$$

“ $\supseteq$ ” is trivial, and for “ $\subseteq$ ” use compactness of  $R \in \mathcal{V}X$ . Hence, if  $\mathcal{S}$  consists of basic open sets from  $\mathcal{B}$ , then  $\bigcup \mathcal{F} \in \mathcal{B}$  due to its closure under finite unions. Thus,  $(\bigcup \mathcal{S})^\square$  is a union of sets of the form  $U^\square$  for  $U \in \mathcal{B}$ .

► **Proposition 4.7.** *The functor  $\mathcal{V}: \mathbf{Haus} \rightarrow \mathbf{Haus}$  preserves limits of  $\omega^{\text{op}}$ -chains.*

**Proof.** Consider an  $\omega^{\text{op}}$ -chain as in (4). Let  $M = \lim \mathcal{V}X_n$ , with limit cone  $r_n: M \rightarrow \mathcal{V}X_n$ . Let  $m: \mathcal{V}L \rightarrow M$  be the unique continuous map such that  $\mathcal{V}\ell_n = r_n \cdot m$  for all  $n < \omega$ . We shall prove that  $m$  is a bijection and then that its inverse is continuous, which proves that  $m$  is an isomorphism.

1. Injectivity of  $m$  follows from the fact that  $\mathcal{V}\ell_n$  ( $n < \omega$ ) forms a jointly monic family, as we will now prove. Suppose that  $A, B \in \mathcal{V}L$  satisfy  $\ell_n[A] = \ell_n[B]$  for every  $n < \omega$ . We prove that  $A \subseteq B$ ; by symmetry  $A = B$  follows. Given  $a \in A$ , we show that every open neighbourhood of  $a$  has a nonempty intersection with  $B$ . Since  $B$  is closed, we then have  $a \in B$  (otherwise  $L \setminus B$  would be an open neighbourhood of  $a$  disjoint from  $B$ ). It suffices to prove the desired property for the basic open neighbourhoods  $\ell_n^{-1}(U)$  of  $a$ , for  $U$  open in  $X_n$  (see Remark 2.1.2). Since  $\ell_n[A] = \ell_n[B]$  we have some  $b \in B$  which satisfies  $\ell_n(a) = \ell_n(b)$ . Then  $b \in \ell_n^{-1}(U) \cap B$ .
2. Surjectivity of  $m$ . An element of  $M$  is a sequence  $(K_n)_{n < \omega}$  of compact (hence closed) subsets  $K_n \subseteq X_n$  such that  $f_n[K_{n+1}] = K_n$  for every  $n < \omega$ . We need to find a compact set  $K \subseteq L$  such that  $\ell_n[K] = K_n$  for every  $n < \omega$ . With the subspace topology,  $K_n$  is itself a compact space. The connecting maps  $f_n: X_{n+1} \rightarrow X_n$  restrict to surjective continuous maps  $K_{n+1} \rightarrow K_n$ . Thus, the spaces  $K_n$  form an  $\omega^{\text{op}}$ -chain of surjections in  $\text{KHaus}$ . Let  $K$  be the limit with projections  $p_n: K \rightarrow K_n$ . Then  $K$  is a subset of  $L$ , and each projection  $p_n$  is the restriction of  $\ell_n$  to  $X_n$ .

Let us check that the topology on  $K$  is the subspace topology inherited from  $L$ . A base of the topology on  $K$  is the family of sets  $p_n^{-1}(U)$  as  $U$  ranges over the open subset of  $K_n$ . Each  $U$  is of the form  $V \cap K_n$  for some open  $V$  of  $X_n$ , and  $p_n^{-1}(U) = \ell_n^{-1}(V) \cap K$ . Thus  $p_n^{-1}(U)$  is open in the subspace topology, and the converse holds as well.

The maps  $p_n$  are surjective by Remark 4.6.2. Moreover,  $K$  is a compact space by Remark 4.6.1. Thus,  $K$  is the desired compact set in  $\mathcal{V}L$  such that  $p_n[K] = K_n$  for all  $n$ .

3. Finally, we prove that the inverse  $k: M \rightarrow \mathcal{V}L$ , say, of  $m$  is continuous. We know that the sets  $\ell_n^{-1}(U)$ , for  $U$  open in  $X_n$ , form a base of  $L$ . Moreover, this base is closed under finite unions. By Remark 4.6.3 and using Proposition 4.3 we obtain that  $\mathcal{V}L$  has a subbase given by the following sets

$$(\mathcal{V}\ell_n)^{-1}(U^\diamond) = (\ell_n^{-1}(U))^\diamond \quad \text{and} \quad (\mathcal{V}\ell_n)^{-1}(U^\square) = (\ell_n^{-1}(U))^\square \quad \text{for } U \text{ open in } X_n.$$

It suffices to show that the inverse images of these subbasic open sets of  $\mathcal{V}L$  are open in  $M$ . For  $\mathcal{V}\ell_n^{-1}(U^\diamond)$  with  $U$  open in  $X_n$  we use that  $\mathcal{V}\ell_n \cdot k = r_n$  clearly holds to obtain

$$k^{-1}(\mathcal{V}\ell_n^{-1}(U^\diamond)) = r_n^{-1}(U^\diamond),$$

which is a basic open set of  $M$  by Remark 2.1.2. For the subbasic open sets  $\mathcal{V}\ell_n^{-1}(U^\square)$  the proof is similar.  $\blacktriangleleft$

► **Corollary 4.8.** *The restrictions of  $\mathcal{V}$  to  $\text{KHaus}$  and  $\text{Stone}$  preserve limits of  $\omega^{\text{op}}$ -chains.*

Indeed, use Remark 4.6.1.

► **Remark 4.9.** A *codirected limit* is the limit of a diagram whose scheme is of the form  $P^{\text{op}}$  for a directed poset  $P$ . Proposition 4.7 and Corollary 4.8 hold more generally for codirected limits. The argument is the same. This proves a result stated in Zenor [24], but with an incomplete proof.

The following definition is due to Kupke et al. [14] for Stone spaces, whereas Hofmann et al. [11, Def. 2.29] use arbitrary topological spaces, but they later essentially restrict constants to be (compact) Hausdorff, stably compact or spectral spaces.

► **Definition 4.10.** The *Vietoris polynomial functors* are the endofunctors on  $\text{Top}$  built from the Vietoris functor  $\mathcal{V}$ , the constant functors, and the identity functor, using product, coproduct, and composition. Thus, the Vietoris polynomial functors are built according to the following grammar

$$F ::= \mathcal{V} \mid A \mid \text{Id} \mid \prod_{i \in I} F_i \mid \coprod_{i \in I} F_i \mid FF,$$

where  $A$  ranges over all topological spaces and  $I$  is an arbitrary index set.

► **Theorem 4.11.** *Let  $F: \mathbf{Top} \rightarrow \mathbf{Top}$  be a Vietoris polynomial functor, and assume that all constants in  $F$  are Hausdorff spaces. Then  $F$  has a terminal coalgebra obtained in  $\omega$  steps, and  $\nu F = V_\omega$  is a Hausdorff space.*

**Proof.** An easy induction on Vietoris polynomial functors  $F$  shows that:

1. The functor  $F$  has a restriction  $F_0: \mathbf{Haus} \rightarrow \mathbf{Haus}$ .
2. The restriction  $F_0$  preserves surjective maps; the most important step being for  $\mathcal{V}$  itself, and this uses the fact when  $f: X \rightarrow Y$  is continuous and  $X$  and  $Y$  are Hausdorff, the inverse images of compact sets are compact.
3. The functor  $F_0$  preserves limits of  $\omega^{\text{op}}$ -chains; the most important step is done in Proposition 4.7.

The terminal coalgebra result for  $F_0$  follows from the fact which we have mentioned in Section 2:  $\nu F$  is the limit of the terminal-coalgebra  $\omega^{\text{op}}$ -chain  $F_0^n 1$  ( $n < \omega$ ). Since  $\mathbf{Haus}$  is closed under limits in  $\mathbf{Top}$  and  $F_0^n 1 = F^n 1$ , the functor  $F$  has the same terminal coalgebra  $\nu F = \lim F^n 1$ . ◀

► **Corollary 4.12.** *Let  $F: \mathbf{Top} \rightarrow \mathbf{Top}$  be a Vietoris polynomial functor, and assume that all constants in  $F$  are Hausdorff spaces. Then  $F$  has an initial algebra.*

This follows from Theorem 4.11, Example 2.3.1 and Theorem 2.4, since an easy induction shows that  $F$  preserves monomorphisms.

► **Corollary 4.13.** *Let  $F: \mathbf{Top} \rightarrow \mathbf{Top}$  be a Vietoris polynomial functor in which all constants are compact Hausdorff spaces and only finite coproducts are used. Then the terminal coalgebra  $\nu F$  is a compact Hausdorff space.*

**Proof.** The functor  $F$  restricts to an endofunctor on  $\mathbf{KHaus}$ . Thus, the terminal-coalgebra  $\omega^{\text{op}}$ -chain  $F^n 1$  lies in  $\mathbf{KHaus}$ . Moreover,  $\mathbf{KHaus}$  is closed under limits in  $\mathbf{Top}$  because it is a full reflective subcategory (Remark 4.6.1). Thus,  $\nu F = \lim_{n < \omega} F^n 1$  is compact Hausdorff. ◀

► **Corollary 4.14.** *Let  $F: \mathbf{Top} \rightarrow \mathbf{Top}$  be a Vietoris polynomial functor in which all constants are Stone spaces and only finite coproducts are used. Then the terminal coalgebra  $\nu F$  is a Stone space.*

The proof is similar.

► **Remark 4.15.** Corollary 4.13 essentially appears in work by Hofmann et al. [11, Thm. 3.42] (except for the convergence ordinal). Corollary 4.14 is due to Kupke et al. [14]. Our proof using convergence of the terminal-coalgebra chain is different than the previous ones.

► **Example 4.16.** The terminal coalgebra for  $\mathcal{V}$  itself was identified by Abramsky [1]. By what we have shown, it is  $V_\omega = \lim \mathcal{V}^n 1$ . An easy induction on  $n$  shows that  $\mathcal{V}^n 1$  is  $\mathcal{P}_f^n 1$  with the discrete topology; the key point is that each set  $\mathcal{P}_f^n 1$  is finite. The topology was described in Remark 2.1.2: it has as a base the sets  $\partial_n^{-1}(U)$  as  $U$  ranges over the subsets of  $\mathcal{P}_f^n 1$ . By Corollary 4.14,  $\nu F$  is a Stone space.

In Appendix A, we present for  $\mathcal{P}_f$  a concrete description of  $V_\omega$  as the set of compactly branching strongly extensional trees.

► **Remark 4.17.** Note that Theorem 4.11 also holds for Vietoris polynomial functors when we take  $\mathbf{Haus}$  as our base category. Hofmann et al. [11] consider other full subcategories of  $\mathbf{Top}$ , and they also study the completeness of the category of coalgebras for Vietoris polynomial functors  $F$  (however, they restrict to using finite products and finite coproducts in their

definition of Vietoris polynomial functors). For a Vietoris polynomial functor  $F$  on  $\mathbf{Haus}$ , the category of coalgebras is complete [11, Cor. 3.41]. Moreover, every subfunctor of  $F$  has a terminal coalgebra [11, Cor. 4.6].

► **Remark 4.18.** Hofmann et al. [11, Ex. 2.27(2)] also consider a related construction called the *lower Vietoris space* of  $X$ . It is the set of all closed subsets of  $X$  with the topology generated by all sets  $U^\diamond$ , cf. (5). This again yields a functor on  $\mathbf{Top}$ : a given continuous function is mapped to  $A \mapsto \overline{f[A]}$ , where  $\overline{f[A]}$  denotes the closure of  $f[A]$ . Furthermore, one has a corresponding notion of lower Vietoris polynomial functors. They prove that for such functors  $F$  on the category of stably compact spaces (defined in [11]),  $\mathbf{Coalg} F$  is complete [11, Thm. 3.35]. Furthermore, if a lower Vietoris polynomial functor  $F$  on  $\mathbf{Top}$  can be restricted to that category, then it has a terminal coalgebra obtained by finite iteration:  $\nu F = V_\omega$  [11, Thm. 3.36]. Similar results hold for the category of spectral spaces and spectral maps.

► **Remark 4.19.** Let us mention a very general result which applies in many situations to deliver a terminal coalgebra: Makkai and Paré's Limit Theorem [15, Thm. 5.1.6]. It implies that every accessible endofunctor  $F: \mathcal{A} \rightarrow \mathcal{A}$  on a locally presentable category has an initial algebra and a terminal coalgebra. (Indeed, the theorem implies that the category of  $F$ -coalgebras is cocomplete.) This result cannot be used here because  $\mathbf{Haus}$  is not locally presentable: it does not have a small set of objects that is colimit-dense [3, Prop. 8.2].

► **Open Problem 4.20.**

1. Does every Vietoris polynomial functor on  $\mathbf{Top}$  have a terminal coalgebra?
2. Does every Vietoris polynomial functor on  $\mathbf{KHaus}$  as in Corollary 4.13 have an initial algebra?

Item 1 above is equivalent to asking whether the result that  $\nu F$  exists for every Vietoris polynomial functor would remain true if we allowed non-Hausdorff constants.

## 5 Hausdorff Polynomial Functors

Analogously to the Vietoris polynomial functors on  $\mathbf{Top}$ , we introduce Hausdorff polynomial functors on  $\mathbf{Met}$ . Closer to the situation of Kripke polynomial functors on  $\mathbf{Set}$  than to Vietoris polynomial functors on  $\mathbf{Top}$ , the Hausdorff polynomial functors on  $\mathbf{Met}$  have terminal coalgebras obtained in  $\omega + \omega$  steps.

► **Notation 5.1.** The Hausdorff functor  $\mathcal{H}: \mathbf{Met} \rightarrow \mathbf{Met}$  maps a metric space  $X$  to the space  $\mathcal{H}X$  of all compact subsets of  $X$  equipped with the Hausdorff distance<sup>1</sup> given by

$$\bar{d}(S, T) = \max \left( \sup_{x \in S} d(x, T), \sup_{y \in T} d(y, S) \right), \quad \text{for } S, T \subseteq X \text{ compact,}$$

where  $d(x, S) = \inf_{y \in S} d(x, y)$ . In particular  $\bar{d}(\emptyset, T) = \infty$  for nonempty compact sets  $T$ . For a non-expanding map  $f: X \rightarrow Y$  we have  $\mathcal{H}f: S \mapsto f[S]$ .

► **Remark 5.2.**

1. The functors  $\mathcal{V}: \mathbf{Top} \rightarrow \mathbf{Top}$  and  $\mathcal{H}: \mathbf{Met} \rightarrow \mathbf{Met}$  are closely related: for compact metric spaces  $X$  the Vietoris space  $\mathcal{V}X$  is precisely the topological space induced by the Hausdorff space  $\mathcal{H}X$ .

<sup>1</sup> The definition goes back to Pompeiu [17] and was popularized by Hausdorff [10, p. 293].

## 21:10 On Kripke, Vietoris and Hausdorff Polynomial Functors

2. Some authors define  $\mathcal{H}X$  to consist of all *nonempty* compact subsets of  $X$ . However, Hausdorff [10] did not exclude  $\emptyset$ , and the above formula works (as already indicated) without such an exclusion.

► **Remark 5.3.**

1. For a complete metric space,  $\mathcal{H}X$  is complete again (see e.g. Barnsley [7, Thm. 7.1]). Thus  $\mathcal{H}$  restricts to a functor on the category CMS of complete metric spaces, which we denote by the same symbol  $\mathcal{H}$ .
2. Let UMet denote the category of (extended) ultrametric spaces: the full subcategory of Met given by spaces satisfying the following stronger version of the triangle inequality:

$$d(x, z) \leq \max\{d(x, y), d(y, z)\}.$$

If  $X$  is an ultrametric space, then so is  $\mathcal{H}X$ . To see this, let  $S, T, U \in \mathcal{H}X$ . Write  $p$  for  $\max\{\bar{d}(S, T), \bar{d}(T, U)\}$ . For each  $x \in S$ , there is some  $y \in T$  such that  $d(x, y) \leq \bar{d}(S, T)$ . For this  $y$ , there is some  $z \in U$  such that  $d(y, z) \leq \bar{d}(T, U)$ . So

$$d(x, z) \leq \max\{d(x, y), d(y, z)\} \leq \max\{\bar{d}(S, T), \bar{d}(T, U)\} = p.$$

It follows that  $d(x, U) \leq p$ . This for all  $x \in X$  shows that  $d(S, U) \leq p$ . Note that  $p = \max\{\bar{d}(U, T), \bar{d}(T, S)\}$ . The same argument shows that  $\sup_{z \in U} d(z, S) \leq p$ . So we have  $\bar{d}(S, U) \leq p$ . This proves the ultrametric inequality.

We again denote the restriction of the Hausdorff functor to UMet is denoted by  $\mathcal{H}$ .

3. For a discrete metric space  $X$  (where all distances are 0 or  $\infty$ ),  $\mathcal{H}X$  is the discrete space formed by all finite subsets of  $X$ .
4. For an arbitrary metric space  $X$ , the finite subsets of  $X$  form a dense set in  $\mathcal{H}X$ . Indeed, given a compact set  $S \subseteq X$ , for every  $\varepsilon > 0$ , there exists a finite set  $T \subseteq S$  such that  $S$  is covered by  $\varepsilon$ -balls around the points in  $T$ . Therefore  $d(x, T) \leq \varepsilon$  for all  $x \in S$ , and we have  $d(y, S) = 0$  for all  $y \in T$ . This implies that  $\bar{d}(S, T) \leq \varepsilon$ .

► **Example 5.4.** For the Hausdorff functor, a terminal coalgebra is carried by the space of all finitely branching strongly extensional trees equipped with the discrete metric. This follows from the finite power-set functor  $\mathcal{P}_f$  having its terminal coalgebra formed by those trees (Example 3.7). Indeed, the terminal-coalgebra chain  $V_i$  ( $i \in \text{Ord}$ ) for  $\mathcal{H}$  is obtained by equipping the sets in the terminal-coalgebra chain for  $\mathcal{P}_f$  with the discrete metric. Furthermore, since limits in Met (or CMS) are set-based, we see that both chains converge in exactly  $\omega + \omega$  steps. Therefore  $\nu\mathcal{H} = V_{\omega+\omega}$ .

It follows that, unlike the Vietoris functor, the Hausdorff functor does not preserve limits of  $\omega^{\text{op}}$ -chains: the terminal-coalgebras chain for  $\mathcal{H}(-)$  does not converge before  $\omega + \omega$  steps (see Example 5.4.5.4). Thus this functor does not preserve the limit  $V_\omega = \lim_{n < \omega} V_n$ .

► **Definition 5.5.** Let  $(X_n)_{n < \omega}$  be an  $\omega^{\text{op}}$ -chain in Met. A cone  $r_n: M \rightarrow X_n$  is *isometric* if for all  $x, y \in M$  we have  $d(x, y) = \sup_{n \in \mathbb{N}} d(r_n(x), r_n(y))$ .

By Remark 2.1.3, limit cones of  $\omega^{\text{op}}$ -chains in Met are isometric.

► **Proposition 5.6.** *The Hausdorff functor preserves isometric cones of  $\omega^{\text{op}}$ -chains.*

**Proof.** Let  $(X_n)_{n < \omega}$  be an  $\omega^{\text{op}}$ -chain with connecting maps  $f_n: X_{n+1} \rightarrow X_n$ . Given an isometric cone  $\ell_n: M \rightarrow X_n$  ( $n < \omega$ ), we prove that the cone  $\mathcal{H}\ell_n: \mathcal{H}M \rightarrow \mathcal{H}X_n$  is also isometric:

$$\bar{d}(S, T) = \sup_{n < \omega} \bar{d}(\ell_n(S), \ell_n(T)) \quad \text{for all compact subset } S, T \subseteq M.$$

We can assume that  $S$  and  $T$  are nonempty and finite: since finite sets are dense in  $\mathcal{H}M$  by Remark 5.3.4, and the maps  $\ell_n$  are (non-expanding whence) continuous, the desired equality then holds for all pairs in  $\mathcal{H}M$ . The case where  $S$  or  $T$  is empty is trivial.

Since every  $\ell_n$  is non-expanding, we only need to prove that  $\bar{d}(S, T) \leq c$  holds for  $c = \sup_{n < \omega} \bar{d}(\ell_n[S], \ell_n[T])$ . For this, we show that for every  $\varepsilon > 0$ ,  $\bar{d}(S, T) \leq c + \varepsilon$ . By the definition of the Hausdorff metric  $\bar{d}$ , it suffices to prove that for every  $x \in S$  we have  $d(x, T) \leq c + \varepsilon$ . By symmetry, we then also have  $d(y, S) \leq c + \varepsilon$  for every  $y \in T$ .

Given  $y \in T$  we have  $d(x, y) = \sup_{n < \omega} d(\ell_n(x), \ell_n(y))$ . Thus, there is a  $k < \omega$  such that

$$d(x, y) \leq d(\ell_k(x), \ell_k(y)) + \varepsilon.$$

Since  $T$  is finite, we can choose  $k$  such that this inequality holds for all  $y \in T$ . By definition,

$$\bar{d}(\ell_k(x), \ell_k[T]) = \inf_{y \in T} d(\ell_k(x), \ell_k(y)) \quad \text{in } X_k.$$

Again using that  $T$  is finite, we can pick some  $y \in T$  such that  $d(\ell_k(x), \ell_k[T]) = d(\ell_k(x), \ell_k(y))$ . With this  $y$  we conclude that

$$d(x, T) \leq d(x, y) \leq d(\ell_k(x), \ell_k(y)) + \varepsilon = d(\ell_k(x), \ell_k[T]) + \varepsilon \leq \bar{d}(\ell_k[S], \ell_k[T]) + \varepsilon \leq c + \varepsilon. \quad \blacktriangleleft$$

► **Remark 5.7.** The Hausdorff functor preserves isometric embeddings and their intersections. Indeed, for every subspace  $X$  of a metric space  $Y$ , a set  $S \subseteq X$  is compact in  $X$  iff it is so in  $Y$ . Moreover, given  $S, T \in \mathcal{H}X$ , their distances in  $\mathcal{H}X$  and  $\mathcal{H}Y$  are the same. Thus,  $\mathcal{H}$  preserves isometric embeddings.

Given a collection  $X_i \subseteq Y$  ( $i \in I$ ) of subspaces, a set  $S \subseteq \bigcap_{i \in I} X_i$  is compact in  $X$  iff it is so in  $Y$  (and therefore in every  $X_i$ ). Thus  $\mathcal{H}$  preserves that intersection.

► **Definition 5.8.** The *Hausdorff polynomial functors* are the endofunctors on  $\text{Met}$  built from the Hausdorff functor, the constant functors, and the identity functor, using product, coproduct, and composition. Thus, the Hausdorff polynomial functors are built according to the following grammar (cf. Definition 3.1):

$$F ::= \mathcal{H} \mid A \mid \text{Id} \mid \prod_{i \in I} F_i \mid \coprod_{i \in I} F_i \mid FF,$$

where  $A$  ranges over all metric spaces and  $I$  is an arbitrary index set.

► **Theorem 5.9.** *Every Hausdorff polynomial functor  $F: \text{Met} \rightarrow \text{Met}$  has a terminal coalgebra obtained in  $\omega + \omega$  steps:  $\nu F = V_{\omega + \omega}$ .*

**Proof.** An easy induction over the structure of Hausdorff polynomial functors shows that each such functor  $F$  preserves:

1. isometric cones of  $\omega^{\text{op}}$ -chains, and
2. isometric embeddings and their intersections.

The most important step is done in Proposition 5.6 and Remark 5.7.

We conclude that in the terminal-coalgebra chain, the map  $m: V_{\omega+1} \rightarrow V_\omega$  from (2) in the Introduction is an isometric embedding by item 1. By item 2, all of the maps  $m, Fm, FFm, \dots$  in the chain  $(V_{\omega+n})_{n < \omega}$  are isometric embeddings. Hence  $F$  preserves the intersection of the ensuing subspaces of  $V_\omega$  viz. the limit  $V_{\omega+\omega} = \lim_{n < \omega} V_{\omega+n}$ . Consequently, we have  $\nu F = V_{\omega+\omega}$ . ◀

► **Remark 5.10.** Note that if a Hausdorff polynomial functor  $F$  uses only constants given by complete metric spaces  $A$ , then it has a restriction to an endofunctor on CMS. Indeed, by an easy induction on the structure of  $F$  one shows that  $FX$  is complete whenever  $X$  is complete. Similarly, when  $F$  uses constants which are ultrametric spaces, then  $F$  has a restriction on UMet.

Since CMS and UMet are closed under limits of  $\omega^{\text{op}}$ -chains in Met, we obtain the following

► **Corollary 5.11.** *Every Hausdorff polynomial functor on CMet or UMet has a terminal coalgebra obtained in  $\omega + \omega$  steps.*

► **Corollary 5.12.** *Every Hausdorff polynomial functor  $F$  on Met or CMS has an initial algebra.*

Indeed, since Hausdorff polynomial functors preserve isometric embeddings, this follows from Theorem 5.9, Example 2.3.2, and Theorem 2.4.

► **Remark 5.13.** We mentioned another possible approach to terminal coalgebras in Remark 4.19. Let us comment on the situation regarding the results on Met here. The category Met is locally presentable (see e.g. [6, Ex. 2.3]). The Limit Theorem does imply that on Met, the Hausdorff polynomial functors have terminal coalgebras. In more detail, the Hausdorff functor is finitary: this was proved for its restriction to 1-bounded metric spaces [5, Sec. 3], and the proof for  $\mathcal{H}$  itself is the same. An easy induction then shows that every Hausdorff polynomial functor is accessible, so that the Limit Theorem can be applied. However, our elementary proof shows that the terminal coalgebra chain converges in  $\omega + \omega$  steps. The proof of Makkai and Paré's Limit Theorem does not yield such a bound.

## 6 Variation: the Closed Subset Functor on Met

We have been concerned with the Hausdorff functor taking a metric space  $M$  to the space of its nonempty compact subsets. For two variations, let us consider  $\mathcal{P}_{\text{cl}}: \text{Met} \rightarrow \text{Met}$  taking  $M$  to the set of its *closed* subsets, and its subfunctor  $\mathcal{P}'_{\text{cl}}: \text{Met} \rightarrow \text{Met}$  taking  $M$  to the set of its nonempty closed subsets. Both  $\mathcal{P}_{\text{cl}}M$  and  $\mathcal{P}'_{\text{cl}}M$  are given the Hausdorff metric. For a non-expanding map  $f: X \rightarrow Y$ , the non-expanding map  $\mathcal{P}_{\text{cl}}f: \mathcal{P}_{\text{cl}}X \rightarrow \mathcal{P}_{\text{cl}}Y$  sends a closed subset  $S$  of  $X$  to the closure of  $f[S]$ . This makes  $\mathcal{P}_{\text{cl}}$  and  $\mathcal{P}'_{\text{cl}}$  functors. Due to the empty set,  $\mathcal{P}_{\text{cl}}$  is a closer analog of  $\mathcal{H}$ . It is natural to ask whether the positive results of Section 5 hold for these functors  $\mathcal{P}_{\text{cl}}$  and  $\mathcal{P}'_{\text{cl}}$ . As proved by van Breugel [20, Prop. 8], the functor  $\mathcal{P}_{\text{cl}}$  has no terminal coalgebra. Turning to  $\mathcal{P}'_{\text{cl}}$ , this functor has an initial algebra given by the empty metric space and a terminal coalgebra carried by a singleton metric space. But  $\mathcal{P}'_{\text{cl}}$  has no other fixed points (see van Breugel et al. [21, Cor. 5]), where an object  $X$  is a *fixed point* of an endofunctor  $F$  if  $FX \cong X$ . We provide below a different, shorter proof.

► **Remark 6.1.**

1. A subset  $X$  of a metric space is  $\delta$ -discrete if whenever  $x \neq y$  are elements of  $X$ ,  $d(x, y) \geq \delta$ . Every subset of a  $\delta$ -discrete set is  $\delta$ -discrete, and every such set is closed. Moreover, if  $C$  and  $D$  are different subsets of a  $\delta$ -discrete set, then  $\bar{d}(C, D) \geq \delta$ .
2. A subset  $S$  of an ordinal  $i$  is *cofinal* if for all  $j < i$  there is some  $k \in S$  with  $j \leq k < i$ . If  $S$  is not cofinal, then its complement  $i \setminus S$  must be so. (But it is possible that both  $S$  and  $i \setminus S$  are cofinal in  $i$ .)



► **Theorem 6.2.** *There is no isometric embedding  $\mathcal{P}'_{\text{cl}}M \rightarrow M$  when  $|M| \geq 2$ .*

**Proof.** Suppose towards a contradiction that  $\iota: \mathcal{P}'_{\text{cl}}M \rightarrow M$  were an isometric embedding where  $|M| \geq 2$ . If all distances in  $M$  are 0 or  $\infty$ , then  $\mathcal{P}'_{\text{cl}}M$  is the nonempty power-set of  $M$ . In this case, our result follows from the fact that for  $|M| \geq 2$ ,  $M$  has more nonempty subsets than elements. Thus we fix distinct points  $a, b \in M$  of finite distance, and put  $\delta = d(a, b)/2$ . Let  $A = \{x \in M : d(x, a) \leq \delta\}$ , and let  $B = M \setminus A$ . (In case  $d(a, b) = \infty$ , we need to adjust this by setting  $\delta = \infty$ , and  $B$  to be the points whose distance to  $a$  is finite. But we shall not present the argument in this case.)

We proceed to define an ordinal-indexed sequence of elements  $x_i \in M$ . We also prove that each set  $S_i = \{x_j : j < i\}$  is  $\delta$ -discrete, and we put

$$X_i = \begin{cases} A & \text{if } \{j < i : x_j \in A\} \text{ is cofinal in } i \\ B & \text{else.} \end{cases}$$

For  $i = 0$ , put  $x_0 = \iota(\{a, b\})$ . Given an ordinal  $i > 0$ , we put

$$x_i = \iota(X_i \cap S_i).$$

Being nonempty (since  $i > 0$ ) and  $\delta$ -discrete,  $X_i \cap S_i$  lies in  $\mathcal{P}'_{\text{cl}}M$ .

The remainder of our proof consists of showing that for every ordinal  $i$ :

$$d(x_j, x_k) \geq \delta \quad \text{for } 0 \leq j < k \leq i.$$

We proceed by transfinite induction. Assume that our claim holds for every  $k < i$  and then prove it for  $i$ . The base case  $i = 0$  is trivial. For  $i > 0$ , note first that it follows from the induction hypothesis that  $S_i$  is  $\delta$ -discrete.

Hence, we need only verify that  $d(x_j, x_i) \geq \delta$  when  $0 \leq j < i$ . We argue the case  $X_i = A$ ; when  $X_i = B$ , the argument is similar, mutatis mutandis. For  $j = 0$ , recall that  $x_0 = \iota(\{a, b\})$  and  $x_i = \iota(A \cap S_i)$ . Since  $b$  has distance at least  $\delta$  from every element of  $A$ , we obtain  $\bar{d}(\{a, b\}, A \cap S_i) \geq \delta$ . As  $\iota$  is an isometric embedding, this distance is also  $d(x_0, x_i)$ . Now let  $j > 0$ . Since we have  $X_i = A$ , let  $k$  be such that  $j \leq k < i$  and  $x_k \in A$ . Now either  $x_j = \iota(A \cap S_j)$  or else  $x_j = \iota(B \cap S_j)$ .

In the first case, note that  $x_k \in A \cap S_i$  since  $k < i$ , and  $x_k \notin S_j$  by the definition of  $S_j$  since  $k \geq j$ . So  $A \cap S_j$  and  $A \cap S_i$  are different nonempty subsets of the  $\delta$ -discrete set  $S_i$ . Hence, the distance between these sets is at least  $\delta$ , and therefore we have  $d(x_j, x_i) \geq \delta$ .

In the second case,  $B \cap S_j$  is a nonempty subset of  $B$ , and thus again it not equal to  $A \cap S_i$ . So again we see that  $d(x_j, x_i) = \bar{d}(B \cap S_j, A \cap S_i) \geq \delta$ .

We now obtain the desired contradiction since  $(x_i)$  is an ordinal-indexed sequence of pairwise distinct elements of  $M$ . ◀

► **Corollary 6.3.**

1. *The functor  $\mathcal{P}'_{\text{cl}}: \text{Met} \rightarrow \text{Met}$  has no fixed points except the empty set and the singletons.*
2. *The functor  $\mathcal{P}_{\text{cl}}: \text{Met} \rightarrow \text{Met}$  admits no isometric embedding  $\mathcal{P}_{\text{cl}}M \rightarrow M$ , whence has no fixed point.*

**Proof.** The first item is immediate from Theorem 6.2. For the second one, observe that the inclusion map  $e: \mathcal{P}'_{\text{cl}}M \rightarrow \mathcal{P}_{\text{cl}}M$  is an isometric embedding. Assuming that there were an isometric embedding  $\iota: \mathcal{P}_{\text{cl}}M \rightarrow M$ , we see that  $M$  cannot be empty (since  $\mathcal{P}_{\text{cl}}M$  is nonempty) or a singleton (since then  $|\mathcal{P}_{\text{cl}}M| = 2$ ). Hence  $|M| \geq 2$ . Moreover, we obtain an isometric embedding  $\iota \cdot e: \mathcal{P}'_{\text{cl}}M \rightarrow M$ , contradicting Theorem 6.2. ◀

## 7 Summary

We have investigated versions of the finite power-set functor for the categories **Haus** and **Met**. Our main results are that the Vietoris functor  $\mathcal{V}$ , and indeed all Vietoris polynomial functors, have terminal coalgebras obtained in  $\omega$  steps of the terminal-coalgebra chain. The same holds for the Hausdorff polynomial functors on **Met**, but the iteration takes  $\omega + \omega$  steps and so the underlying reasons are different.

Our work on the Kripke and Hausdorff polynomial functors highlights a technique which we feel could be of wider interest. To prove that a terminal coalgebra exists in a situation where the limit of the  $\omega^{\text{op}}$ -chain (1) is not preserved by the functor, one could try to find preservation properties which imply that the limit of the  $\omega^{\text{op}}$ -chain  $(V_{\omega+n})_n$  was preserved. In **Set**, we used finitariness and preservation of monomorphisms and intersections, and in **Met** we used preservation of intersections, isometric embeddings, and isometric cones.

We have also seen that for the functor  $\mathcal{P}_{\text{cl}}$  on **Met**, there is no fixed point and hence no terminal coalgebra. We leave open the question of whether every Vietoris polynomial functor on **Top** has a terminal coalgebra.

---

## References

- 1 Samson Abramsky. A Cook's Tour of the finitary non-well-founded sets. In *We Will Show Them: Essays in honour of Dov Gabbay*, volume 1, pages 1–18. College Publications, 2005.
- 2 Jiří Adámek. Free algebras and automata realizations in the language of categories. *Comment. Math. Univ. Carolin.*, 15:589–602, 1974.
- 3 Jiří Adámek, Miroslav Hušek, Jiří Rosický, and Walter Tholen. Smallness in topology. preprint; available at <https://arxiv.org/abs/2302.00050>, accepted for publication in *Questiones Mathematicae*, 2023.
- 4 Jiří Adámek, Stefan Milius, and Lawrence S. Moss. Initial algebras without iteration. In Fabio Gaducci and Alexandra Silva, editors, *9th Conference on Algebra and Coalgebra in Computer Science (CALCO)*, volume 211 of *LIPICs*, pages 5:1–5:20. Schloss Dagstuhl, 2021.
- 5 Jiří Adámek, Stefan Milius, Lawrence S. Moss, and Henning Urbat. On finitary functors and their presentations. *J. Comput. System Sci.*, 81(5):813–833, 2015.
- 6 Jiří Adámek and Jiří Rosický. Approximate injectivity and smallness in metric-enriched categories. *J. Pure Appl. Algebra*, 226(6), 2022. Article 106974.
- 7 Michael F. Barnsley. *Fractals Everywhere*. Morgan Kaufmann, 2nd edition, 1993.
- 8 Michael Barr. Terminal coalgebras in well-founded set theory. *Theoret. Comput. Sci.*, 114(2):299–315, 1993.
- 9 Samuel Eilenberg and Norman Steenrod. *Foundations of Algebraic Topology*. Princeton University Press, 1952.
- 10 Felix Hausdorff. *Grundzüge der Mengenlehre*. Veit & Comp., Leipzig, 1914.
- 11 Dirk Hofmann, Renato Neves, and Pedro Nora. Limits in categories of Vietoris coalgebras. *Math. Structures Comput. Sci.*, 29(4):552–587, 2019.
- 12 Bart Jacobs. *Introduction to Coalgebra. Towards Mathematics of States and Observation*. Cambridge University Press, 2016.
- 13 Peter Johnstone. *Stone Spaces*. Cambridge Univ. Press, 1986.
- 14 Clemens Kupke, Alexander Kurz, and Yde Venema. Stone coalgebras. *Theoret. Comput. Sci.*, 327(1–2):109–132, 2004.
- 15 Michael Makkai and Robert Paré. *Accessible Categories: the Foundation of Categorical Model Theory*, volume 104 of *Contemporary Math*. Amer. Math. Soc., Providence, RI, 1989.
- 16 Ernest Michael. Topologies on spaces of subsets. *Trans. Amer. Math. Soc.*, 71:152–182, 1951.
- 17 Dimitrie Pompeiu. Sur la continuité des fonctions de variables complexes. *Annales de la Faculté des Sciences de la Université de Toulouse pour les Sciences Mathématiques et les Sciences Physiques*, 2ième Série, 7(3):265–315, 1905.

- 18 Luis Ribes and Pavel Zalesskii. *Profinite Groups*, volume 40 of *Ergebnisse der Mathematik und ihrer Grenzgebiete*. Springer, 2nd edition, 2010.
- 19 Martin Rößiger. Coalgebras and modal logic. In Horst Reichel, editor, *Proc. Coalgebraic Methods in Computer Science (CMCS 2000)*, volume 33 of *Electron. Notes Theor. Comput. Sci.*, pages 294–315. Elsevier, 2000.
- 20 Franck van Breugel. De Bakker-Zucker processes revisited. *Inform. and Comput.*, 188(1):68–76, 2004.
- 21 Franck van Breugel, Camillo Costantini, and Stephen Watson. Isometries between a metric space and its hyperspace, function space, and space of measures. *Topology Appl.*, 137(1-3):51–57, 2004.
- 22 Leopold Vietoris. Bereiche zweiter Ordnung. *Monatsh. Math. Phys.*, 32(1):258–280, 1922.
- 23 James Worrell. On the final sequence of a finitary set functor. *Theoret. Comput. Sci.*, 338:184–199, 2005.
- 24 Phillip Zenor. On the completeness of the space of compact subsets. *Proc. Amer. Math. Soc.*, 26(1):190–192, 1970.

## A Trees and the Limit of the Terminal-Coalgebra Chain for $\mathcal{P}_f$

We give the description of  $V_\omega$  for  $\mathcal{P}_f$  due to Worrell [23]. We provide a full exposition to the results which Worrell stated without proof.

A *tree* is a directed graph  $t$  with a distinguished node  $\text{root}(t)$  from which every other node can be reached by a unique directed path. Every tree in our sense must have a root, so there is no empty tree. All of our trees are *unordered*. We always identify isomorphic trees.

### ► Definition A.1.

1. We use the notation  $t_x$  for the subtree of  $t$  rooted in the node  $x$  of  $t$ .
2. A tree  $t$  is *extensional* if for every node  $x$  distinct children  $y$  and  $z$  of  $x$  give different (that is, non-isomorphic) subtrees  $t_y$  and  $t_z$ .
3. A *graph bisimulation* between two trees  $t$  and  $u$  is a relation between the nodes of  $t$  and the nodes of  $u$  with the property that whenever  $x$  and  $y$  are related: (a) every child of  $x$  is related to some child of  $y$ , and (b) every child of  $y$  is related to some child of  $x$ .
4. A *tree bisimulation* between two trees  $t$  and  $u$  is a graph bisimulation  $R$  with the additional properties that that
  - a. The nodes  $\text{root}(t)$  and  $\text{root}(u)$  are related; the roots are not related to other nodes; and
  - b. whenever two nodes are related, their parents are also related.
5. Two trees are *tree bisimilar* if there is a tree bisimulation between them.
6. A tree  $t$  is *strongly extensional* if every tree bisimulation on it is a subset of the diagonal

$$\Delta_t = \{(x, x) : x \in t\}.$$

In other words,  $t$  is strongly extensional iff distinct children  $x$  and  $y$  of the same node define subtrees  $t_x$  and  $t_y$  which are *not* tree bisimilar.

### ► Remark A.2.

1. Every composition and every union of tree bisimulations is again a tree bisimulation. In addition, the opposite relation of every tree bisimulation is a tree bisimulation: if  $R$  is a tree bisimulation from  $t$  to  $u$ , then  $R^{\text{op}}$  is a tree bisimulation from  $u$  to  $t$ . Consequently, the largest tree bisimulation on every tree is an equivalence relation.
2. A subtree  $s$  of a strongly extensional tree  $t$  is strongly extensional. Indeed, if  $R$  is a tree bisimulation on  $s$ , then  $R \cup \Delta_t$  is a tree bisimulation on  $t$ . Since  $R \cup \Delta_t \subseteq \Delta_t$ , we have  $R \subseteq \Delta_s$ .

## 21:16 On Kripke, Vietoris and Hausdorff Polynomial Functors

► **Lemma A.3.** *If  $t$  and  $u$  are strongly extensional and related by a tree bisimulation, then we have  $t = u$ .*

**Proof.** Let  $R$  be a tree bisimulation between  $t$  and  $u$ . By Remark A.2,  $R^{\text{op}} \cdot R$  is a tree bisimulation on  $t$ , whence  $R^{\text{op}} \cdot R \subseteq \Delta_t$  by strong extensionality. But every node of  $t$  is related to at least one node of  $u$  (use induction on the depth of nodes) implying that  $R^{\text{op}} \cdot R = \Delta_t$ . Similarly,  $R \cdot R^{\text{op}} = \Delta_u$ . Thus,  $R$  (is a function and it) is an isomorphism of trees, and we identify such trees. ◀

► **Notation A.4.**

1. Let  $\mathcal{T}$  be the class of trees. We define maps  $\partial_n: \mathcal{T} \rightarrow V_n = \mathcal{P}_f^n 1$  as follows:  $\partial_0$  is the unique map to 1, and given the map  $\partial_n$  and a tree  $t$ , we put

$$\partial_{n+1}(t) = \{\partial_n(t_x) : x \text{ is a child of the root of } t\}.$$

On the right we have a subset of  $\mathcal{P}_f^n 1$ , and this is an element of  $\mathcal{P}_f^{n+1} 1$ .

2. The trees  $t$  and  $u$  are *Barr equivalent* provided that  $\partial_n t = \partial_n u$  for all  $n$ . We write  $t \approx u$  in this case.
3. For every tree  $t$ , we define maps  $\rho_n^t: t \rightarrow V_n = \mathcal{P}_f^n 1$  in the following way:  $\rho_0^t$  is the unique map  $t \rightarrow 1$ , and for all nodes  $x$  of  $t$ ,  $\rho_{n+1}^t(x) = \{\rho_n^t(y) : y \text{ is a child of } x \text{ in } t\}$ . This family of maps  $\rho_n^t$  is a cone: we have  $\rho_n^t = v_{m,n} \cdot \rho_m^t$  for every connecting map  $v_{m,n}: \mathcal{P}_f^m 1 \rightarrow \mathcal{P}_f^n 1$ ,  $m \geq n$ . Hence, there is a unique map  $\rho_\omega^t: t \rightarrow V_\omega$  such that  $\ell_n \cdot \rho_\omega^t = \rho_n^t$  for all  $n$ .

► **Remark A.5.** Note that  $V_n = \mathcal{P}_f^n 1$  may be described as the set of all extensional trees of height at most  $n$ . Indeed, 1 is described as the singleton set consisting of the root-only tree, and every finite set of extensional trees in  $V_{n+1} = \mathcal{P}_f V_n$  is represented by the extensional tree obtained by tree-tupling the trees from the given set.

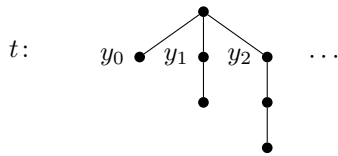
► **Remark A.6.**

1. If  $\rho_{n+1}^t(a) = \rho_{n+1}^t(b)$ , then for all children  $a'$  of  $a$ , there is some child  $b'$  of  $b$  and  $\rho_n^t(a') = \rho_n^t(b')$ . This is easy to see from the definition of  $\rho_{n+1}^t$ .
2. For all trees  $t$ ,  $\rho_i^t(\text{root}(t)) = \partial_i(t)$ . Furthermore, let  $b: t \rightarrow \mathcal{T}$  be given by  $b(x) = t_x$ . Then  $\rho_i^t = \partial_i \cdot b$ .

► **Definition A.7.** Let  $x_0, x_1, \dots$ , be a sequence of nodes in a tree  $t$ , and let  $y$  also be a node in  $t$ . We write  $\lim x_n = y$  to mean that for every  $n$  there is some  $m$  such that  $\rho_n^t(x_p) = \rho_n^t(y)$  whenever  $p \geq m$ .

A tree  $t$  is *compactly branching* if for all nodes  $x$  of  $t$ , the set of children of  $x$  is *sequentially compact*: for every sequence of  $(y_n)$  of children of  $x$  there is a subsequence  $(w_n)$  of  $(y_n)$  and some child  $z$  of  $x$  such that  $\lim w_n = z$ .

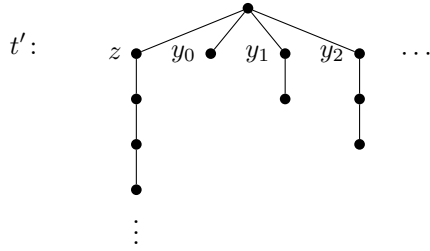
► **Example A.8.** The following tree  $t$  is not compactly branching:



To see this, consider the sequence  $y_0, y_1, \dots$ . Note that for  $n \geq m$ ,  $\rho_n^t(y_n) = \partial_i(t_{y_n}) = t_{y_m}$ . We claim that for every subsequence  $(y_{k_n})$  of this sequence  $(y_n)$  there is no  $y_p$  such that  $\lim_n y_{k_n} = y_p$ . To simplify the notation, we only verify this for the sequence  $(y_n)$  itself. It does not converge to any fixed element  $y_m$  because for  $p > m$ ,

$$\rho_p^t(y_m) = \partial_p(t_{y_m}) \neq \partial_p(t_{y_p}) = \rho_p^t(y_p).$$

In contrast, the following tree is compactly branching (also observe also that  $t \approx t'$ ):



To check the compactness, consider a sequence of children of the root, say  $(x_n)$ . If there is an infinite subsequence which is constant, then of course that sequence converges. If not, then there is a subsequence of  $(x_n)$ , say  $(w_n)$ , where each  $w_n$  is  $y_k$  for some  $k \geq n$ . In this case,  $\lim_n (w_n) = z$ . This is because for all but finitely many  $n$ ,  $\rho_n^t(z) = \partial_n(t_z) = t_{w_n} = \partial_n(t_{w_n}) = \rho_n^t(w_n)$ .

► **Lemma A.9.** *If  $t$  and  $u$  are compactly branching, and if  $\rho_\omega^t(\text{root}(t)) = \rho_\omega^u(\text{root}(u))$ , then there is a tree bisimulation between  $t$  and  $u$  which includes  $\{(x, y) \in t \times u : \rho_\omega^t(x) = \rho_\omega^u(y)\}$ .*

**Proof.** Given compactly branching trees  $t$  and  $u$ , we define a relation  $R \subseteq t \times u$  inductively by

$$x R y \quad \text{iff} \quad (1) \ x = \text{root}(t) \text{ and } y = \text{root}(u), \text{ or } x \text{ and } y \text{ have } R\text{-related parents, and} \\ (2) \ \rho_\omega^t(x) = \rho_\omega^u(y).$$

Let us check that  $R$  is a tree bisimulation. Suppose that  $(x, y)$  are related by  $R$  as above, and let  $x'$  be a child of  $x$  in  $t$ . Using Remark A.6.1 we see that for each  $n$ , there is some child  $y'_n$  of  $y$  in  $u$  with  $\rho_n^t(x') = \rho_n^u(y'_n)$ . Consider the sequence  $y'_0, y'_1, \dots$ . Now  $\rho_n^t(x') = \rho_n^u(y'_m)$  if  $m \geq n$ , since  $\rho_n^t$  and  $\rho_n^u$  form cones:  $\rho_n^t(x') = v_{m,n} \cdot \rho_m^t(x') = v_{m,n} \cdot \rho_m^u(y'_m) = \rho_n^u(y'_m)$ . By sequential compactness, there is a subsequence  $z_0, z_1, \dots$ , and also some child  $z^*$  of  $y$  such that  $\lim z_n = z^*$ . Being a subsequence,  $\rho_n^t(x') = \rho_n^u(z_m)$  whenever  $m \geq n$ . Let us check that for all  $n$ ,  $\rho_n^t(x') = \rho_n^u(z^*)$ . To see this, fix  $n$  and let  $m \geq n$  be large enough so that for  $p \geq m$ ,  $\rho_p^u(z_p) = \rho_p^u(z^*)$ . Thus,  $\rho_n^t(x') = \rho_n^u(z_m) = \rho_n^u(z^*)$ . Thus,  $\rho_\omega^t(x') = \rho_\omega^u(z^*)$ , which shows  $x' R z^*$ , as desired.

The other half of the verification that  $R$  is a tree bisimulation is similar. ◀

► **Notation A.10.** In this section,  $V_\omega$  denotes the limit of (1) for the finite power-set functor.

1. We take the elements of  $V_\omega$  to be compatible sequences  $(x_n)$ . That is,  $x_n \in \mathcal{P}_f^n 1$  and  $\mathcal{P}_f^n(x_{n+1}) = x_n$  for every  $n < \omega$ . To save on notation, we write  $x$  for  $(x_n)$ . We consider the relation  $\rightsquigarrow$  on  $V_\omega$  defined by

$$x \rightsquigarrow y \quad \text{iff} \quad \text{for all } n, y_n \in x_{n+1}. \tag{6}$$

2. Let  $L^+$  be the set of nonempty finite sequences from  $V_\omega$ . We write such a sequence with the notation  $\langle x^1, \dots, x^n \rangle$ . We consider the relation  $\Rightarrow$  on  $L^+$  defined by

$$\langle x^1, \dots, x^n \rangle \Rightarrow \langle y^1, \dots, y^m \rangle \quad \text{iff} \quad m = n + 1, x^1 = y^1, \dots, x^n = y^n, \text{ and } x^n \rightsquigarrow y^{n+1}.$$

In other words,  $m = n + 1$ ,  $\langle y^1, \dots, y^{m-1} \rangle = \langle x^1, \dots, x^n \rangle$ , and  $x^n \rightsquigarrow y^m$ .

## 21:18 On Kripke, Vietoris and Hausdorff Polynomial Functors

3. For each  $x \in V_\omega$ , let  $\text{tr}_x$  be the tree whose nodes are the sequences  $\langle x, x^2, \dots, x^n \rangle \in L^+$  whose first entry is  $x$ , with root the one-point sequence  $\langle x \rangle$ , and with graph relation the restriction of  $\Rightarrow$ . For readers familiar with tree unfoldings of pointed graphs,  $\text{tr}_x$  is the tree unfolding of the graph  $(V_\omega, \rightsquigarrow)$  at the point  $x$ .
4. Finally, let

$$T = \{\text{tr}_x : x \in V_\omega\}. \quad (7)$$

Recall the connecting maps  $\mathcal{P}_f^n!: \mathcal{P}_f^{n+1}1 \rightarrow \mathcal{P}_f^n1$ .

► **Lemma A.11.** *Let  $x \in V_\omega$ .*

1. *For all  $k$  and all  $\langle x, x^2, \dots, x^n \rangle \in \text{tr}_x$ ,  $\rho_k^{\text{tr}_x}(\langle x, x^2, \dots, x^n \rangle) = x_k^n$ .*
2. *Let  $R$  be a tree bisimulation on  $\text{tr}_x$ . If  $\langle x, x^2, \dots, x^n \rangle R \langle x, y^2, \dots, y^n \rangle$ , then for all  $k$ ,*

$$\rho_k^{\text{tr}_x}(\langle x, x^2, \dots, x^n \rangle) = \rho_k^{\text{tr}_x}(\langle x, y^2, \dots, y^n \rangle).$$

3. *The tree  $\text{tr}_x$  is strongly extensional and compactly branching, and  $\partial_\omega(\text{tr}_x) = \rho_\omega^{\text{tr}_x}(\langle x \rangle) = x$ .*

**Proof.**

1. By induction on  $k$ . For  $k = 0$ , our result is clear: the codomain of  $\rho_k$  is 1. Assume our result for  $k$ , fix  $x \in L^+$  and  $\langle x^1, \dots, x^n \rangle \in \text{tr}_x$ . We first prove that

$$\{y_k : x^n \rightsquigarrow y\} = x_{k+1}^n. \quad (8)$$

Indeed, if  $x^n \rightsquigarrow y$ , then  $y_k \in x_{k+1}^n$ . Conversely, if  $a \in x_{k+1}^n$ , we construct  $y \in V_\omega$  such that  $x^n \rightsquigarrow y$  with  $y_k = a$ . Note that

$$x_k^n = \mathcal{P}_f^k!(x_{k+1}^n) = \mathcal{P} \mathcal{P}_f^{k-1}!(x_{k+1}^n) = \mathcal{P}_f^{k-1}![x_{k+1}^n].$$

Since  $a \in x_{k+1}^n$ , we have  $\mathcal{P}_f^{k-1}!(a) \in x_k^n$ . So we let  $y_{k-1} = \mathcal{P}_f^{k-1}!(a)$ . We repeat this argument to define  $y_{k-2}, \dots, y_1, y_0$ ; the point is that  $y_{k-i} \in x_{k-i+1}^n$  for  $i = 0, \dots, k$ . Choices are needed when we go the other way from  $k$ . Note that

$$\mathcal{P}_f^{k+1}![x_{k+2}^n] = \mathcal{P}_f(\mathcal{P}_f^{k+1}!(x_{k+2}^n)) = \mathcal{P}_f^{k+2}!(x_{k+2}^n) = x_{k+1}^n.$$

Every set functor preserves surjective functions, and so  $\mathcal{P}_f^{k+1}!$  is surjective. Thus there is some  $y_{k+1} \in x_{k+2}^n$  such that  $\mathcal{P}_f^{k+1}!(y_{k+1}) = y_{k+1}$ . The same argument enables us to find by recursion on  $i$  a sequence  $y_{k+i+1} \in x_{k+i+2}^n$  such that  $\mathcal{P}_f^{k+i+1}!(y_{k+i+1}) = y_{k+i}$ . This defines  $y$  such that  $x^n \rightsquigarrow y$  according to (6) with  $y_k = a$ .

The induction step is now easy:

$$\begin{aligned} \rho_{k+1}^{\text{tr}_x}(\langle x, x^2, \dots, x^n \rangle) &= \{\rho_k^{\text{tr}_x}(\langle x, x^2, \dots, x^n, y \rangle) : x^n \rightsquigarrow y\} \\ &= \{y_k : x^n \rightsquigarrow y\} && \text{by induction hypothesis} \\ &= x_{k+1}^n && \text{by (8)}. \end{aligned}$$

2. This again is an induction on  $k$ , and the steps are similar to what we have just seen. We also note that tuples in  $\text{tr}_x$  related by a tree bisimulation must have the same length.
3. Note first that by item 1 with  $n = 1$ , we have  $\rho_k^{\text{tr}_x}(\langle x \rangle) = x_k$  for all  $k$ . This implies that  $\rho_\omega^{\text{tr}_x}(\langle x \rangle) = x$ . For the strong extensionality, let  $R$  be a tree bisimulation on  $\text{tr}_x$ . Suppose that  $\langle x, x^2, \dots, x^n \rangle$  and  $\langle x, y^2, \dots, y^n \rangle$  are related by  $R$ . Using items 1 and 2, we see that for all  $k$ , we have  $x_k^n = y_k^n$ . Thus  $x^n = y^n$ . In addition, since  $R$  is a tree bisimulation,

the parents of the two nodes under consideration are also related by  $R$ . So the same argument shows that  $x^{n-1} = y^{n-1}$ . Continuing in this way shows that  $x^{n-2} = y^{n-2}, \dots, x^2 = y^2$ . Hence  $\langle x, x^2, \dots, x^n \rangle = \langle x, y^2, \dots, y^n \rangle$ .

Finally, we verify that  $\text{tr}_x$  is compactly branching. To simplify the notation a little, we shall show this for children of the root  $\langle x \rangle$ . So suppose we have an infinite sequence  $\langle x, y^1 \rangle, \langle x, y^2 \rangle, \dots$ . Recall that each set  $\mathcal{P}_f^n 1$  is finite. By successively thinning the sequence  $y^1, y^2, \dots$ , we may assume that for all  $n \in \omega$  and all  $p, q \geq n$ ,  $y_p^n = y_q^n$ . Let  $z \in V_\omega$  be the ‘diagonal’ sequence  $z_n = y_n^n$ . Since every  $\langle x, y^n \rangle$  is a child of the root  $\langle x \rangle$  (in symbols:  $\langle x \rangle \Rightarrow \langle x, y^n \rangle$ ), we have  $x \rightsquigarrow y^n$ . This implies that for all  $n$ , we have  $z_n = y_n^n \in x_{n+1}$ , whence  $x \rightsquigarrow z$ . Thus,  $\langle x, z \rangle$  is a child of the root of  $\text{tr}_x$ . Recall from item 1 that  $\rho_n^{\text{tr}_x}(\langle x, z \rangle) = z_n$ . So we obtain the desired conclusion:  $\lim \langle x, y^n \rangle = \langle x, z \rangle$ . ◀

► **Lemma A.12.** *For every tree  $t$  there is a Barr-equivalent tree  $t^* \in T$  such that  $t^*$  is strongly extensional and compactly branching.*

**Proof.** Given any tree  $t$ , we have  $x = \partial_\omega(t) \in V_\omega$ . For all  $n$ ,  $x_n = \partial_n(t)$ . The tree  $t^* = \text{tr}_x$  in Lemma A.11.3 is strongly extensional and compactly branching. Recall that the root of  $t^*$  is  $\langle x \rangle$ . By Lemma A.11.1, we have that for all  $n < \omega$ ,

$$\partial_n(t^*) = \rho_n^{t^*}(\text{root}(\text{tr}_x)) = \rho_n^{t^*}(\langle x \rangle) = x_n = \partial_n(t). \quad \blacktriangleleft$$

► **Lemma A.13.** *The set  $T$  defined in (7) is the set of all compactly branching, strongly extensional trees.*

**Proof.** By Lemma A.11.3 we know that every tree in  $T$  is strongly extensional and compactly branching. For the reverse inclusion, let  $t$  be compactly branching and strongly extensional. Let  $t^*$  be as in Lemma A.12 for  $t$ . By Lemmas A.3 and A.9,  $t = t^*$ . Thus  $t \in T$ . ◀

► **Definition A.14.** Let  $D$  be the set of finitely branching strongly extensional trees. Let  $\delta: D \rightarrow \mathcal{P}_f D$  take a strongly extensional tree  $t$  to the (finite) set of its subtrees  $t_x$ .

In this definition, we use Remark A.2.2: a subtree of a strongly extensional tree is strongly extensional.

► **Theorem A.15** [23]. *For the finite power-set functor  $\mathcal{P}_f$  the following hold:*

1. *the maps  $\partial_n: T \rightarrow \mathcal{P}_f^n 1$  given by  $\partial_n(\text{tr}_x) = x_n$  form a limit of (2); thus,  $V_\omega \cong T$ ,*
2. *the coalgebra  $(D, \delta)$  is terminal.*

**Proof.**

1. The map  $\varphi: V_\omega \rightarrow T$  given by  $\varphi(x) = \text{tr}_x$  is obviously surjective. Suppose that  $\text{tr}_x = \text{tr}_y$ . The roots of these trees are  $\langle x \rangle$  and  $\langle y \rangle$ . For all  $n$ , we have that

$$x_n = \rho_n^{\text{tr}_x}(\langle x \rangle) = \rho_n^{\text{tr}_y}(\langle y \rangle) = y_n.$$

Thus  $\partial_\omega(\langle x \rangle) = \partial_\omega(\langle y \rangle)$ . By Lemmas A.3 and A.9,  $x = y$ . So  $\varphi$  is injective. The formula for  $\partial_n$  comes from Lemma A.11.1.

2. We use Theorem 3.5. The map  $m: V_{\omega+1} \rightarrow V_\omega$  in (2) assigns to a finite set of trees in  $V_\omega$  their tree-tupling. Its image is the set of all strongly extensional, compactly branching trees which are finitely branching at the root. An easy induction on  $n$  shows that  $V_{\omega+n}$  is the set of all compactly branching, strongly extensional trees  $t$  with the property that the topmost  $n$  levels of  $t$  are finitely branching. With this description,  $V_{\omega+n} \subseteq D$ , and the limit  $V_{\omega+\omega}$  is simply the intersection  $D = \bigcap_n V_{\omega+n}$ . This shows that the carrier set of  $\nu \mathcal{P}_f$  is  $D$ . For the structure map  $\delta$ , note that  $m: \mathcal{P}_f V_\omega \rightarrow V_\omega$  in (2) is tree-tupling, as are  $\mathcal{P}_f m, \mathcal{P}_f \mathcal{P}_f m$ , etc. It follows that in the intersection,  $D$ , the coalgebra structure is the inverse of tree-tupling. ◀

## 21:20 On Kripke, Vietoris and Hausdorff Polynomial Functors

This concludes our work showing that for the finite power-set functor  $\mathcal{P}_f$ ,  $V_\omega$  is the set  $T$  of strongly extensional, compactly branching trees, and the terminal coalgebra  $\nu\mathcal{P}_f$  is the set  $D$  of finitely branching, strongly extensional trees.



# CRDTs, Coalgebraically

Nathan Liittschwager ✉ 

University of California, Santa Cruz, CA, USA

Stelios Tsampas ✉ 

Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

Jonathan Castello ✉ 

University of California, Santa Cruz, CA, USA

Lindsey Kuper ✉ 

University of California, Santa Cruz, CA, USA

---

## Abstract

We describe ongoing work that models conflict-free replicated data types (CRDTs) from a coalgebraic point of view. CRDTs are data structures designed for replication across multiple physical locations in a distributed system. We show how to model a CRDT at the local replica level using a novel coalgebraic semantics for CRDTs. We believe this is the first step towards presenting a unified theory for specifying and verifying CRDTs and replicated state machines. As a case study, we consider *emulation* of CRDTs in terms of coalgebra.

**2012 ACM Subject Classification** Theory of computation; Theory of computation → Semantics and reasoning; Theory of computation → Distributed computing models; Theory of computation → Concurrency; Theory of computation → Formal languages and automata theory

**Keywords and phrases** Coalgebra, Distributed Systems, Concurrency, Bisimulation

**Digital Object Identifier** 10.4230/LIPIcs.CALCO.2023.22

**Category** Early Ideas

**Funding** This material is based upon work supported by the National Science Foundation under Grant No. CCF-2145367. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

## 1 Introduction

In distributed systems, data replication guards against machine failures and keeps data physically close to clients who require low-latency access, but it introduces the problem of keeping replicas consistent with one another in the face of network partitions and unpredictable message latency. *Conflict-free replicated data types* (CRDTs) [14, 13] are data structures whose operations must satisfy certain mathematical properties that can be leveraged to ensure *strong convergence*, meaning that replicas are guaranteed to have equivalent state given that they have received and applied the same (unordered) set of update operations.

A typical use case for CRDTs is that of a text document being made available to  $n$  concurrent users via an online collaborative editor. We can model such a collaborative editing application as a coalgebra

$$\langle \text{update}, \text{query} \rangle : X \rightarrow X^A \times S,$$

where  $X$  is some black-box state (representing the editor internals),  $A$  a finite set of arguments, e.g. add or delete a character at a particular location via  $\text{update} : X \rightarrow X^A$ , and  $S$  being the



© Nathan Liittschwager, Stelios Tsampas, Jonathan Castello, and Lindsey Kuper; licensed under Creative Commons License CC-BY 4.0

10th Conference on Algebra and Coalgebra in Computer Science (CALCO 2023).

Editors: Paolo Baldan and Valeria de Paiva; Article No. 22; pp. 22:1–22:5

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

*observable* state: the text string itself accessible through  $\text{query} : X \rightarrow S$ . As a shorthand, we use  $u$  and  $q$  for *update* and *query*, respectively.

Viewing each client as acting on the text document via  $u : X \rightarrow X^A$ , we can define  $u^* : X \rightarrow X^{A^*}$  by recursion as an iterated update that consumes a *list* of commands  $\sigma \in A^*$  and gives the current state:  $u^*(x)(\langle \rangle) = x$  and  $u^*(x)(a \cdot \sigma) = u^*(u(x)(a))(\sigma)$ . Hence we may understand that all client interactions are represented as words in  $\sigma \in A^*$ , and the state of the object  $x \in X$  is generated by  $u^*(x)(\sigma)$ . Observe that the coalgebra  $\langle u, q \rangle : X \rightarrow X^A \times S$  is a  $((-)^A \times S)$ -coalgebra, and hence by standard results [4]:

► **Proposition 1.** *The final  $((-)^A \times S)$ -coalgebra is given by*

$$S^{A^*} \xrightarrow{\langle \zeta_1, \zeta_2 \rangle} (S^{A^*})^A \times S,$$

where  $\zeta_1(\varphi)(a) = \lambda(\sigma \in A^*). \varphi(a \cdot \sigma)$  and  $\zeta_2(\varphi) = \varphi(\langle \rangle)$ . That is, for any  $((-)^A \times S)$ -coalgebra  $\langle u, q \rangle : X \rightarrow X^A \times S$ , there is a unique coalgebra homomorphism  $\text{beh} : X \rightarrow S^{A^*}$  that satisfies  $\text{beh}(x) = \lambda(\sigma \in A^*). q(u^*(x)(\sigma))$ .

The final coalgebra  $\langle \zeta_1, \zeta_2 \rangle : S^{A^*} \rightarrow (S^{A^*})^A \times S$  thus defines all possible infinite behaviors of the collaborative text editor, under the assumption that it is implemented by some centralized server that totally orders and executes all client requests. However, such a centralized approach is often infeasible or simply undesirable. Instead, each client may be working on a local copy – a *replica* – of the object and propagating changes between replicas.

To keep replicas consistent, specialized algorithms and communication middleware can be used. For example, the *state machine replication* [11] approach guarantees *strong consistency* (informally, where clients cannot tell that the data is replicated) by ensuring that each replica executes the same sequence of commands. However, this approach requires expensive coordination between replicas. CRDTs take a different approach: they avoid the need for coordination by carefully constraining the state space  $X$  and the implementation of the update and query methods, and sacrifice strong consistency in favor of *strong convergence*. Under strong convergence, replicas that have received the same set of updates (in any order) agree in state, but clients may observe differing intermediate states.

In this short paper, we propose studying CRDTs under a coalgebraic lens. We find that CRDTs lend themselves to a coalgebraic interpretation: they are implemented as replicated objects at multiple locations, where each replica has an opaque internal state, but publicly available methods centered around calls to *update* or *query*; strong convergence and emulation of CRDTs are primarily about *observable behavior* from perspective of a client. By taking the coalgebraic approach, we can use the well-developed theory of universal coalgebra to reason about strong convergence of CRDTs. Moreover, the coalgebraic approach lets us make precise a notion of *emulation* of CRDTs that has until now been known only informally.

## 2 Coalgebraic Semantics of CRDTs

CRDTs may be specified in an *operation-based* (or *op-based*) style or in a *state-based* style. Op-based CRDTs require that replicas transmit messages containing the *effects* of a local update *downstream* to other replicas. Updates are *applied* to replicas in a way that respects causality [5]: when an update is applied to replica  $i$ , any causally preceding updates must have already been applied to  $i$ , but *concurrently* applied updates do not need to be restricted to any particular order; an op-based CRDT converges regardless. State-based CRDTs, on the other hand, converge by restricting their (observable) state to be a join-semilattice  $\langle S, \sqcup \rangle$ , and updates are propagated between replicas by simply passing copies of state  $s \in S$

between replicas, and merging states via a least-upper-bound operator  $\sqcup$ . The two styles are equivalent in the sense that given a state-based CRDT, one can construct a corresponding op-based CRDT that *emulates* it, and vice versa [14].

We show that the semantics of a CRDT can be described as a coalgebra  $c : X \rightarrow F(X)$ , where  $F : \mathbf{Sets} \rightarrow \mathbf{Sets}$  is a Kripke-polynomial functor [4], beginning with the semantics for state-based CRDTs. Similarly to the text-editing example, the CRDT consists of a state space  $X$ , a set  $S$  of observables, an update map  $u : X \rightarrow X^A$  and a query map  $q : X \rightarrow S$ . In addition, we assume an abstract set of *events*  $E$  and equip the CRDT with a “history” morphism  $h : X \rightarrow \mathcal{P}(E)$ , interpreted as a kind of log for events  $e \in E$  that have happened at the replica. Intuitively, events are given by a map that “wraps” interactions (e.g., inputs  $a \in A$ ) with the environment and tag it with meta-data, such as a sequence number.

To model the least-upper-bound operator  $\sqcup$ , we require a map  $\text{merge} : X \rightarrow X^S$  that allows an object with state  $x$  to receive an input state  $s = q(x') \in S$  from some other replica  $x'$ , along with rules requiring that  $\text{merge}$  is inflationary wrt to queries. *Strong convergence* is defined as the property  $\forall x, x' \in X. (h(x) = h(x') \implies q(x) = q(x'))$ , which CRDT coalgebras must satisfy: when two replica states have observed the same set of events, then their query state is the same.

► **Definition 2.** *A state-based CRDT consists of a state space  $X$ , inputs  $A$ , events  $E$ , a payload  $S$  where  $S = \langle S, \sqcup \rangle$  is a join-semilattice, and maps*

$$\langle u, q, h, \xi, \text{merge} \rangle : X \rightarrow X^A \times S \times \mathcal{P}(E) \times \mathcal{P}(E)^{A+S} \times X^S,$$

*s.t. the following hold for all  $x \in X$ ,  $s \in S$ ,  $a \in A$ ,*

- (i)  $q(\text{merge}(x)(s)) = q(x) \sqcup s$ ;
- (ii)  $q(x) \sqcup q(u(x)(a)) = q(u(x)(a))$ ;
- (iii)  $h(u(x)(a)) = h(x) \cup \xi(x)(a)$ ;
- (iv)  $h(\text{merge}(x)(s)) = h(x) \cup \xi(x)(s)$ ;
- (v)  $\forall x' \in X. (h(x) = h(x') \implies q(x) = q(x'))$ .

Op-based CRDTs are similar, except they define local updates  $u : X \rightarrow X^A$  in terms of two other methods: a side-effect free *prepare* method  $\text{prep} : X \rightarrow M^A$  and an effectful *apply* method  $\text{app} : X^M \rightarrow X^M$ , where  $M$  is a set of messages. We assume  $M$  is equipped with a partial order  $\prec_{hb}$ , the so-called *happens-before* relation [5, 12]. This implies that  $M$  is equipped with metadata sufficient for  $\prec_{hb}$  to make sense. Say elements  $m, m'$  are *concurrent* and write  $m \parallel m'$  if  $\neg((m \prec_{hb} m') \vee (m' \prec_{hb} m))$ . Upon a client invoking a local update of type  $a \in A$ , op-based CRDTs first generate a message  $m \in M$  with  $\text{prep}$ , and then send  $m$  downstream to neighboring replicas using some communication middleware that ensures that messages are delivered in an order consistent with causality [12]. The middleware delays delivery of  $m$  to a replica with state  $x$  until a decider (assumed sufficient to ensure causality)  $\text{dlvr?} : X \rightarrow \mathbf{2}^M$  returns “yes”, after which the message is applied with  $\text{app}$ . The decider  $\text{dlvr?}$  can be implemented independently of the CRDT application. A common approach is the vector clock protocol [8, 2].

► **Definition 3.** *An operation-based CRDT consists of a state space  $X$ , inputs  $A$ , events  $E$ , payload  $S$ , and maps*

$$\langle u, q, h, \xi, \text{dlvr?}, \text{prep}, \text{app} \rangle : X \rightarrow X^A \times S \times \mathcal{P}(E)^M \times \mathcal{P}(E)^E \times \mathbf{2}^E \times E^A \times X^E$$

*s.t. the following hold for all  $x \in X$ ,  $a \in A$ ,  $m, m' \in M$*

- (i)  $u(x) = \lambda(a \in A). \text{app}(x)(\text{prep}(x)(a))$ ;
- (ii)  $h(u(x)(a)) = h(x) \cup \xi(x)(\text{prep}(x)(a))$ ;

- (iii)  $h(\text{app}(x)(m)) = h(x) \cup \xi(x)(m)$
- (iv) if  $\text{dlvr?}(x)(m) = \text{dlvr?}(x)(m') = \top$ , then updates  $m, m$  commute. I.e.,  $q(x') = q(x'')$  where  $x' = \text{app}(\text{app}(x)(m))(m')$  and  $x'' = \text{app}(\text{app}(x)(m'))(m)$ .
- (v)  $\forall x' \in X. (h(x) = h(x') \implies q(x) = q(x'))$

Critically, the coalgebra above models a *single* replica, of which there are  $n$  many, initialized from some starting state  $x_0 \in X$ . Communication, from this point of view, is abstracted to the communication middleware.

### 3 Emulation of CRDTs

Much existing work on CRDT semantics (e.g., [1, 3, 9, 7, 6, 10]) has treated op-based and state-based CRDTs as distinct classes of objects, often only considering one class or the other. The justification for this approach is that a state-based CRDT can *emulate* a corresponding op-based CRDT, and vice versa [14]. Despite this commonly cited fact, a notion of emulation is never made precise. Here we aim to fill this gap by showing that emulation of CRDTs may be thought of in terms of *bisimulation* of transition systems.

► **Definition 4.** A transition system on a state space  $X$  with observations  $S$  is a coalgebra  $\langle \text{next}, \text{obs} \rangle : X \rightarrow \mathcal{P}(X) \times S$  s.t.  $x \rightarrow x' \iff x' \in \text{next}(x)$ , and  $x \downarrow s \iff s = \text{obs}(x)$ . Given two transition systems  $\langle \text{next}_1, \text{obs}_1 \rangle : X \rightarrow \mathcal{P}(X) \times S$  and  $\langle \text{next}_2, \text{obs}_2 \rangle : Y \rightarrow \mathcal{P}(Y) \times S$ , a relation  $R \subseteq X \times Y$  is a bisimulation iff  $\forall (x, y) \in X \times Y$ , if  $R(x, y)$ , then

- $x \downarrow s \implies y \downarrow s$ ;
- $x \rightarrow x' \implies \exists y'. y \rightarrow y'$ ;
- $y \rightarrow y' \implies \exists x'. x \rightarrow x'$ .

It can be shown [4] that coalgebras  $c : X \rightarrow F(X)$  may be mapped to transition systems  $d : X \rightarrow \mathcal{P}(X)$ . For the coalgebra of definition 2, define  $x \rightarrow x' \iff (\exists a \in A. u(x)(a) = x') \vee (\exists s \in S. \text{merge}(x)(s) = x')$ , and  $x \downarrow s \iff q(x) = s$ . The construction for definition 3 is similar, with the restriction that  $x \rightarrow x' \iff (\exists a \in A. u(x)(a) = x') \vee (\exists m \in M. \text{app}(x)(m) = x' \wedge \text{dlvr?}(x)(m))$ .

This translation to transition systems exposes the similarity of state-based and operation-based CRDTs by revealing there are really only two “kinds” of transition steps: local steps via  $u : X \rightarrow X^A$  and synchronization steps via  $\text{merge} : X \rightarrow X^S$  for state-based CRDTs, and  $\text{app} : X \rightarrow X^M$  for operation-based CRDTs. For both kinds of CRDTs, the observable payload is given by  $x \downarrow s$ .

► **Proposition 5 (Emulation of state-based CRDTs by op-based CRDTs).** Let  $F, G : \mathbf{Sets} \rightarrow \mathbf{Sets}$  be appropriate functors s.t. given coalgebra  $X \xrightarrow{\langle u, q, h, \xi, \text{merge} \rangle} F(X)$  satisfying definition 2, with transition system semantics  $X \xrightarrow{\langle \text{next}_1, \text{obs}_1 \rangle} \mathcal{P}(X) \times S$ . Then there is a coalgebra  $Y \xrightarrow{\langle u', q', h', \xi', \text{dlvr?}, \text{prep}, \text{app} \rangle} G(Y)$  satisfying definition 3 with transition semantics  $Y \xrightarrow{\langle \text{next}_2, \text{obs}_2 \rangle} \mathcal{P}(Y) \times S$  s.t. there is a bisimulation relation  $R : X \times Y$  between  $X \xrightarrow{\langle \text{next}_1, \text{obs}_1 \rangle} \mathcal{P}(X) \times S$  to  $Y \xrightarrow{\langle \text{next}_2, \text{obs}_2 \rangle} \mathcal{P}(Y) \times S$ .

### 4 Future Work

There are two main directions for future work.

First, the semantics given here can be lifted to consider the semantics of CRDTs (and possible state machine replication in general) from a more *global* point of view, i.e., as interacting asynchronous processes.

Second, the above proposition only gives one direction of the emulation result. The other direction is left to future work. More generally, both operation-based and state-based CRDTs need exhibit strong convergence, which can be thought of as a form of observational equivalence, similar to how emulation is approached here. However, a more interesting approach might be to frame both operation-based and state-based CRDTs as satisfying strong convergence as a universal property, showing that the difference between CRDTs amounts to nothing more than choice of construction.

---

## References

---

- 1 Sebastian Burckhardt, Alexey Gotsman, Hongseok Yang, and Marek Zawirski. Replicated data types: Specification, verification, optimality. In *Proceedings of the 41st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '14, pages 271–284, New York, NY, USA, 2014. Association for Computing Machinery. doi:10.1145/2535838.2535848.
- 2 C. J. Fidge. Timestamps in message-passing systems that preserve the partial ordering. *Proceedings of the 11th Australian Computer Science Conference*, 10(1):56–66, 1988.
- 3 Victor B. F. Gomes, Martin Kleppmann, Dominic P. Mulligan, and Alastair R. Beresford. Verifying strong eventual consistency in distributed systems. *Proc. ACM Program. Lang.*, 1(OOPSLA), October 2017. doi:10.1145/3133933.
- 4 Bart Jacobs. *Introduction to Coalgebra: Towards Mathematics of States and Observation*, volume 59 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2016. doi:10.1017/CB09781316823187.
- 5 Leslie Lamport. Time, clocks, and the ordering of events in a distributed system. *Commun. ACM*, 21(7):558–565, July 1978. doi:10.1145/359545.359563.
- 6 Hongjin Liang and Xinyu Feng. Abstraction for conflict-free replicated data types. In *Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation*, PLDI 2021, pages 636–650, New York, NY, USA, 2021. Association for Computing Machinery. doi:10.1145/3453483.3454067.
- 7 Yiyun Liu, James Parker, Patrick Redmond, Lindsey Kuper, Michael Hicks, and Niki Vazou. Verifying replicated data types with typeclass refinements in Liquid Haskell. *Proc. ACM Program. Lang.*, 4(OOPSLA), November 2020. doi:10.1145/3428284.
- 8 Friedemann Mattern. Virtual time and global states of distributed systems. In *Parallel and Distributed Algorithms*, pages 215–226. North-Holland, 1989.
- 9 Kartik Nagar and Suresh Jagannathan. Automated parameterized verification of CRDTs. In Isil Dillig and Serdar Tasiran, editors, *Computer Aided Verification*, pages 459–477, Cham, 2019. Springer International Publishing.
- 10 Abel Nieto, Léon Gondelman, Alban Reynaud, Amin Timany, and Lars Birkedal. Modular verification of op-based CRDTs in separation logic. *Proc. ACM Program. Lang.*, 6(OOPSLA2), October 2022. doi:10.1145/3563351.
- 11 Fred B. Schneider. Implementing fault-tolerant services using the state machine approach: A tutorial. *ACM Comput. Surv.*, 22(4):299–319, December 1990. doi:10.1145/98163.98167.
- 12 Reinhard Schwarz and Friedemann Mattern. Detecting causal relationships in distributed computations: In search of the holy grail. *Distributed Computing*, 7(3):149–174, March 1994. doi:10.1007/BF02277859.
- 13 Marc Shapiro, Nuno Preguiça, Carlos Baquero, and Marek Zawirski. A comprehensive study of Convergent and Commutative Replicated Data Types. Research Report RR-7506, Inria – Centre Paris-Rocquencourt; INRIA, January 2011. URL: <https://hal.inria.fr/inria-00555588>.
- 14 Marc Shapiro, Nuno Preguiça, Carlos Baquero, and Marek Zawirski. Conflict-free replicated data types. In Xavier Défago, Franck Petit, and Vincent Villain, editors, *Stabilization, Safety, and Security of Distributed Systems*, pages 386–400, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.



# Amortized Analysis via Coinduction

Harrison Grodin   

Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, USA

Robert Harper   

Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, USA

---

## Abstract

Amortized analysis is a program cost analysis technique for data structures in which the cost of operations is specified in aggregate, under the assumption of continued sequential use. Typically, amortized analyses are presented inductively, in terms of finite sequences of operations. We give an alternative coinductive formulation and prove that it is equivalent to the standard inductive definition. We describe a classic amortized data structure, the batched queue, and outline a coinductive proof of its amortized efficiency in **calf**, a dependent type theory for cost analysis.

**2012 ACM Subject Classification** Theory of computation  $\rightarrow$  Type theory; Theory of computation  $\rightarrow$  Logic and verification; Software and its engineering  $\rightarrow$  Functional languages; Theory of computation  $\rightarrow$  Program reasoning; Theory of computation  $\rightarrow$  Categorical semantics

**Keywords and phrases** amortized analysis, coinduction, data structure, mechanized proof

**Digital Object Identifier** 10.4230/LIPIcs.CALCO.2023.23

**Category** Early Ideas

## Supplementary Material

*Software (Source Code)*: <https://github.com/jonsterling/agda-calf> [18], archived at `swh:1:dir:7750187b111d75acca1980e9abffae2d63ffbe69`

**Funding** This material is based upon work supported by the United States Air Force Office of Scientific Research under grant number FA9550-21-0009 (Tristan Nguyen, program manager) and the National Science Foundation under grant number CCF-1901381. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the AFOSR or NSF.

**Acknowledgements** We are grateful to Yue Niu, Max New, and David Spivak for insightful discussions about this research.

## 1 Introduction

The **calf** framework is a dependent type theory that supports verification of both correctness conditions and cost bounds [19], based on call-by-push-value [17]. Amortized analysis is a cost analysis technique for data structures in which the operation costs are specified in aggregate, under the assumption of continued sequential use [23]. In this work, we demonstrate how amortized analysis can be understood as coalgebraic in **calf**.

In call-by-push-value, there are two sorts of types: value types  $A, B, C$  and computation types  $X, Y, Z$ . The type  $FA$  is a computation type classifying computations that result in a value of type  $A$ , and the type  $UX$  is a value type classifying suspended computations of type  $X$ . Computation types beyond  $FA$  will be essential for amortized analysis; in particular, we will make extensive use of products  $X \times Y$ , coproducts  $\Sigma_{a:A} X(a)$ , powers  $A \rightarrow X$ , and coinductive types  $\nu X. Y(X)$  [2], all of which are computation types.

Semantically, we will interpret value types in **Set** and computation types in the category of  $\mathbb{C}$ -sets, where  $\mathbb{C}$  is a monoid representing cost, as is standard for cost analysis of functional



© Harrison Grodin and Robert Harper;

licensed under Creative Commons License CC-BY 4.0

10th Conference on Algebra and Coalgebra in Computer Science (CALCO 2023).

Editors: Paolo Baldan and Valeria de Paiva; Article No. 23; pp. 23:1–23:6

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

## 23:2 Amortized Analysis via Coinduction

programs [7, 8, 15, 6]. This is a simplification of **calf**, avoiding modalities. As in **calf**, we provide a primitive effect  $\text{step}^c(-)$  that incurs  $c$  units of abstract cost, interpreted using the  $\mathbb{C}$ -action. The  $\mathbb{C}$ -action associated to a computation type justifies equations describing how steps are incorporated into its elements:

$$\begin{aligned} \text{step}_{X \times Y}^c(\langle x, y \rangle) &= \langle \text{step}_X^c(x), \text{step}_Y^c(y) \rangle \\ \text{step}_{\Sigma_{a:A} X(a)}^c(\langle a, x \rangle) &= \langle a, \text{step}_X^c(x) \rangle \\ \text{step}_{A \rightarrow X}^c(\lambda a. x) &= \lambda a. \text{step}_X^c(x) \\ \text{step}_{\nu X. Y(X)}^c(\text{gen}(a. y; a_0)) &= \text{gen}(a. \text{step}_Y^c(\nu X. Y(X))(y); a_0) \end{aligned}$$

In other words, cost at a product or power type is incurred pointwise, cost at a coproduct type is pushed into the given summand, and cost at a coinductive type is propagated forward. In this work, we will make use of the  $A$ -wide coproduct of a computation type  $X$ , also known as the *copower* of  $X$  by  $A$  [16, 9], which we write as  $A \times X \triangleq \Sigma_{-:A} X$ . Note that  $\mathbf{1} \times X$  is isomorphic to  $X$ .

### 2 Cofree Comonads for Amortized Abstract Data Types

Throughout this paper, we will use queues as a running example of an abstract data type, although the development generalizes to other sequential-use abstract data types. Queues are an abstract type representing an ordered collection with a first-in-first-out data policy. Let value type  $E$  be the type of elements; the queue operations can be written as follows:

$$\begin{aligned} \text{enqueue}[e] &\sim \mathbf{1} \\ \text{dequeue} &\sim E + \mathbf{1} \end{aligned}$$

This signature describes an operation  $\text{enqueue}[e]$  for each  $e : E$  and an operation  $\text{dequeue}$ .

In a type theory with one sort of type, a machine offering these operations is given via the following cofree comonad [12, 22, 21], interpreted in **Set**:

$$\text{queue}(X) \triangleq \nu Q. (\text{quit} : X) \times (\text{enqueue} : E \rightarrow Q) \times (\text{dequeue} : (E + \mathbf{1}) \times Q)$$

Up to isomorphism, each operation corresponds to a product of its output type and  $Q$ , using a function for an  $E$ -wide product. In call-by-push-value, though, we must distinguish between a product of computation types and a copower of a value type and a computation type. Since the result type of an operation is a value type, such as  $E + \mathbf{1}$  for the  $\text{dequeue}$  operation, we must use the latter. Thus, we may define the type of (amortized) queues as follows, interpreted in the category of  $\mathbb{C}$ -sets:

$$\text{queue}(X) \triangleq \nu Q. (\text{quit} : X) \times (\text{enqueue} : E \rightarrow Q) \times (\text{dequeue} : (E + \mathbf{1}) \times Q)$$

The type  $\text{queue}(X)$  can be understood as “object-oriented” [4, 13, 5], since the use of a queue involves a sequence of  $\text{enqueue}$  and  $\text{dequeue}$  projections terminated by a  $\text{quit}$ . Cost incurred at this type is propagated forward, accumulating at all future  $\text{quit}$  components (of type  $X$ ) for end-of-use accounting.

### 3 Coinductive Amortized Analysis

Let  $\mathbb{C} = (\mathbb{N}, +, 0)$ . We define two queue implementations of type  $\text{queue}(X)$  and prove their amortized equivalence. Here, we let  $X = \mathbf{F1}$ , requiring that the queues terminate with an element of  $\mathbf{F1}$  (i.e., simply a cost in  $\mathbb{C}$ ).



■ **Listing 1** Single-list specification implementation of a queue.

```
spec-queue : list E → queue (F unit)
quit      (spec-queue l) = ret triv
enqueue   (spec-queue l) e = step 1 (spec-queue (l # [ e ]))
dequeue   (spec-queue []) = ret (nothing , spec-queue [])
dequeue   (spec-queue (e :: l)) = ret (just e , spec-queue l)
```

■ **Listing 2** Amortized-efficient batched implementation of a queue.

```
batched-queue : list E → list E → queue (F unit)
quit      (batched-queue bl fl) = step (Φ (bl , fl)) (ret triv)
enqueue   (batched-queue bl fl) e = batched-queue (e :: bl) fl
dequeue   (batched-queue bl []) with reverse bl
... | [] = ret (nothing , batched-queue [] [])
... | e :: fl = step (length bl) (ret (just e , batched-queue [] fl))
dequeue   (batched-queue bl (e :: fl)) =
  ret (just e , batched-queue bl fl)
```

► **Example 1** (Specification Queue). One simple implementation of a queue, called `spec-queue`, is given in Listing 1 by coinduction using copattern matching [1], using a single list as the underlying representation type. The enqueue operation is annotated with one unit of cost; however, this is unrealistic, since a full traversal of the list is performed for each enqueue operation. We will treat this implementation as a client-facing specification, next defining a queue that actually implements this cost model. ┘

► **Example 2** (Batched Queue). Now, we define an amortized-efficient implementation which only incurs one large cost infrequently [10, 11, 3, 20]. This underlying representation type of the implementation is two lists: the “front list”, `fl`, and the “back list”, `bl`. Elements are enqueued to `bl` and dequeued from `fl`; if `fl` is empty when attempting to dequeue, the current `bl` is reversed and used in place of `fl` going forward. The `calf` implementation, called `batched-queue`, is shown in Listing 2. The `quit` case uses a *potential function*  $\Phi(\text{bl}, \text{fl}) = \text{length}(\text{bl})$ , as in the physicist’s method of amortized analysis [23], accounting for elements enqueued on `bl` that were never moved to `fl`. ┘

The amortized analysis is proved via a bisimulation; the theorem statement is analogous to the traditional amortized analysis, using the potential function to accumulate payment [23]. Every enqueue to `spec-queue` pushes one unit of cost forward, while `batched-queue` pushes  $\text{length}(\text{bl})$  units of cost forward only on the occasional dequeue, retroactively using its surplus potential from previous enqueue operations.

► **Theorem 3** (Amortized Analysis of Batched Queue). *For all lists `bl` and `fl`,*

$$\text{batched-queue } \text{bl } \text{fl} = \text{step}^{\Phi(\text{bl}, \text{fl})}(\text{spec-queue } (\text{fl} \# \text{reverse } \text{bl})).$$

**Proof.** By routine coinduction, propagating cost forward over computation types. ◀

## 4 Relation to Inductive Amortized Analysis

Amortized analysis is typically framed algebraically, describing the cost incurred by a finite sequence of operations. In the preceding sections we observed that the analysis is naturally

## 23:4 Amortized Analysis via Coinduction

■ **Listing 3** Program evaluation at a queue.

```
eval : queue-program A → U (queue X) → A × X
eval (return a    ) q = a , Queue.quit q
eval (enqueue e p) q = eval p (Queue.enqueue q e)
eval (dequeue k  ) q =
  bind (k (proj1 (Queue.dequeue q))) λ p →
    eval p (proj2 (Queue.dequeue q))
```

viewed as *coalgebraic*. In fact these perspectives are equivalent. Define the free monad corresponding to the queue operation signature given in Section 2:

$$\mathbf{program}(A) \triangleq \mu P. (\mathbf{return} : A) + (\mathbf{enqueue} : E \times P) + (\mathbf{dequeue} : U(E + 1 \rightarrow FP))$$

An element of  $\mathbf{program}(A)$  is a finite sequence of queue instructions terminated by returning a value of type  $A$ . We may evaluate a program on a queue, by induction on the program:

$$\mathbf{eval} : \mathbf{program}(A) \rightarrow U(\mathbf{queue}(X)) \rightarrow A \times X$$

This expresses the usual notion of running a sequence of operations on a data structure; the code is in Listing 3. Semantically, this definition corresponds to a morphism

$$\mathbf{program}(A) \times \mathbf{queue}(X) \rightarrow A \times X$$

resembling a monad-comonad interaction law [21, 14], here adjusted for call-by-push-value. Using  $\mathbf{eval}$ , we may define an alternative notion of queue equivalence. Let  $q_1, q_2 : \mathbf{queue}(X)$ :

► **Definition 4** (Sequence-of-Operations Queue Equivalence). *Say  $q_1 \approx q_2$  iff for all types  $A$  and programs  $p : \mathbf{program}(A)$ , it is the case that  $\mathbf{eval}(p, q_1) = \mathbf{eval}(p, q_2)$ .*

► **Theorem 5** (Amortizing Sequences of Operations). *It is the case that  $q_1 = q_2$  iff  $q_1 \approx q_2$ .*

**Proof.** By routine ( $\Rightarrow$ ) induction and ( $\Leftarrow$ ) coinduction. ◀

Thus, coalgebraic amortized equivalence coincides with the traditional algebraic notion. Unsurprisingly, a proof that  $q_1 \approx q_2$  shares the same core reasoning as a proof that  $q_1 = q_2$ ; however, it requires the auxiliary definitions of  $\mathbf{program}(A)$  and  $\mathbf{eval}$ .

## 5 Conclusion

Here, we developed a computation type of amortized queues in **calF** as the cofree comonad of a functor based on the product, power, and copower computation type constructors, built to propagate cost forward for end-of-use accounting. We defined specification and amortized queue implementations and stated a theorem relating them via the physicist’s method of amortized analysis. Finally, we observed that coinductive bisimulation coincides with traditional sequence-of-operations reasoning in amortized analysis. Our results for queues and two other simple amortized data structures are formalized in **calF**, which is embedded in Agda [18].



In future work, we hope to extend this approach to support abstract data types with binary and parallel operations, infinite sequences of operations, and situations in which an amortized implementation may be less costly than the specification. Additionally, we hope to better characterize the given constructions, accounting for the asymmetry present in call-by-push-value. As abstract data types are described via a comonad on the category of algebras for a monad, we also hope to connect to bialgebraic presentations of operational semantics [24].

## References

- 1 Andreas Abel, Brigitte Pientka, David Thibodeau, and Anton Setzer. Copatterns: Programming infinite structures by observations. *ACM SIGPLAN Notices*, 48(1):27–38, January 2013. doi:10.1145/2480359.2429075.
- 2 Adriana Balan and Alexander Kurz. On Coalgebras over Algebras. *Electronic Notes in Theoretical Computer Science*, 264(2):47–62, August 2010. doi:10.1016/j.entcs.2010.07.013.
- 3 F. Warren Burton. An efficient functional implementation of FIFO queues. *Information Processing Letters*, 14(5):205–206, July 1982. doi:10.1016/0020-0190(82)90015-1.
- 4 William R. Cook. Object-oriented programming versus abstract data types. In J. W. de Bakker, W. P. de Roever, and G. Rozenberg, editors, *Foundations of Object-Oriented Languages*, Lecture Notes in Computer Science, pages 151–178, Berlin, Heidelberg, 1991. Springer. doi:10.1007/BFb0019443.
- 5 William R. Cook. On understanding data abstraction, revisited. In *Proceedings of the 24th ACM SIGPLAN Conference on Object Oriented Programming Systems Languages and Applications*, OOPSLA '09, pages 557–572, New York, NY, USA, October 2009. Association for Computing Machinery. doi:10.1145/1640089.1640133.
- 6 Joseph W. Cutler, Daniel R. Licata, and Norman Danner. Denotational recurrence extraction for amortized analysis. *Proceedings of the ACM on Programming Languages*, 4(ICFP):97:1–97:29, August 2020. doi:10.1145/3408979.
- 7 Nils Anders Danielsson. Lightweight semiformal time complexity analysis for purely functional data structures. *ACM SIGPLAN Notices*, 43(1):133–144, January 2008. doi:10.1145/1328897.1328457.
- 8 Norman Danner, Daniel R. Licata, and Ramyaa Ramyaa. Denotational cost semantics for functional languages with inductive types. In *Proceedings of the 20th ACM SIGPLAN International Conference on Functional Programming*, ICFP 2015, pages 140–151, New York, NY, USA, August 2015. Association for Computing Machinery. doi:10.1145/2784731.2784749.
- 9 Jeff Egger, Rasmus Ejlers Møgelberg, and Alex Simpson. Enriching an Effect Calculus with Linear Types. In Erich Grädel and Reinhard Kahle, editors, *Computer Science Logic*, Lecture Notes in Computer Science, pages 240–254, Berlin, Heidelberg, 2009. Springer. doi:10.1007/978-3-642-04027-6\_19.
- 10 David Gries. *The Science of Programming*. Springer New York, April 1989.
- 11 Robert Hood and Robert Melville. Real-time queue operations in pure LISP. *Information Processing Letters*, 13(2):50–54, November 1981. doi:10.1016/0020-0190(81)90030-2.
- 12 Bart Jacobs. Mongruences and cofree coalgebras. In V. S. Alagar and Maurice Nivat, editors, *Algebraic Methodology and Software Technology*, Lecture Notes in Computer Science, pages 245–260, Berlin, Heidelberg, 1995. Springer. doi:10.1007/3-540-60043-4\_57.
- 13 Bart Jacobs. Objects And Classes, Co-Algebraically. In Burkhard Freitag, Cliff B. Jones, Christian Lengauer, and Hans-Jörg Schek, editors, *Object Orientation with Parallelism and Persistence*, The Kluwer International Series in Engineering and Computer Science, pages 83–103. Springer US, Boston, MA, 1996. doi:10.1007/978-1-4613-1437-0\_5.
- 14 Shin-ya Katsumata, Exequiel Rivas, and Tarmo Uustalu. Interaction Laws of Monads and Comonads. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '20, pages 604–618, New York, NY, USA, July 2020. Association for Computing Machinery. doi:10.1145/3373718.3394808.
- 15 G. A. Kavvos, Edward Morehouse, Daniel R. Licata, and Norman Danner. Recurrence extraction for functional programs through call-by-push-value. *Proceedings of the ACM on Programming Languages*, 4(POPL):15:1–15:31, December 2019. doi:10.1145/3371083.
- 16 Gregory Maxwell Kelly. *Basic Concepts of Enriched Category Theory*. CUP Archive, February 1982.

- 17 Paul Blain Levy. *Call-By-Push-Value: A Functional/Imperative Synthesis*. Springer Netherlands, Dordrecht, 2003. URL: <http://link.springer.com/10.1007/978-94-007-0954-6>, doi:10.1007/978-94-007-0954-6.
- 18 Yue Niu, Jon Sterling, Harrison Grodin, and Robert Harper. **calf**: A Cost-Aware Logical Framework. URL: <https://github.com/jonsterling/agda-calf>.
- 19 Yue Niu, Jonathan Sterling, Harrison Grodin, and Robert Harper. A cost-aware logical framework. *Proceedings of the ACM on Programming Languages*, 6(POPL):9:1–9:31, January 2022. doi:10.1145/3498670.
- 20 Chris Okasaki. *Purely Functional Data Structures*. PhD thesis, Carnegie Mellon University, 1996. doi:10.1007/3-540-61628-4\_5.
- 21 Gordon Plotkin and John Power. Tensors of Comodels and Models for Operational Semantics. *Electronic Notes in Theoretical Computer Science*, 218:295–311, October 2008. doi:10.1016/j.entcs.2008.10.018.
- 22 John Power and Olha Shkaravska. From Comodels to Coalgebras: State and Arrays. *Electronic Notes in Theoretical Computer Science*, 106:297–314, December 2004. doi:10.1016/j.entcs.2004.02.041.
- 23 Robert Endre Tarjan. Amortized Computational Complexity. *SIAM Journal on Algebraic Discrete Methods*, 6(2):306–318, April 1985. doi:10.1137/0606031.
- 24 D. Turi and G. Plotkin. Towards a mathematical operational semantics. In *Proceedings of Twelfth Annual IEEE Symposium on Logic in Computer Science*, pages 280–291, June 1997. doi:10.1109/LICS.1997.614955.

# Higher-Order Mathematical Operational Semantics

Sergey Goncharov  

Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

Stefan Milius  

Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

Lutz Schröder  

Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

Stelios Tsampas  

Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

Henning Urbat  

Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

---

## Abstract

We present a higher-order extension of Turi and Plotkin’s abstract GSOS framework that retains the key feature of the latter: for every language whose operational rules are represented by a higher-order GSOS law, strong bisimilarity on the canonical operational model is a congruence with respect to the operations of the language. We further extend this result to weak (bi-)similarity, for which a categorical account of Howe’s method is developed. It encompasses, for instance, Abramsky’s classical compositionality theorem for applicative similarity in the untyped  $\lambda$ -calculus. In addition, we give first steps of a theory of logical relations at the level of higher-order abstract GSOS.

**2012 ACM Subject Classification** Theory of computation  $\rightarrow$  Categorical semantics

**Keywords and phrases** Abstract GSOS, lambda-calculus, applicative bisimilarity, bialgebra

**Digital Object Identifier** 10.4230/LIPIcs.CALCO.2023.24

**Category** Early Ideas

**Abstract GSOS.** Turi and Plotkin’s *Mathematical Operational Semantics* [6] provides an elegant categorical approach to modelling the operational semantics of process and programming languages, and elucidates when and why such semantics are well-behaved. In this framework the operational rules of a language are represented as a distributive law of a monad over a comonad in a suitable category. An important example is that of *GSOS laws*, viz. natural transformations of the form

$$\rho_X : \Sigma(X \times BX) \rightarrow B\Sigma^*X,$$

for endofunctors  $\Sigma, B : \mathcal{C} \rightarrow \mathcal{C}$  determining the *syntax* and *behaviour* of the language at hand and the free (term) monad  $\Sigma^*$  generated by  $\Sigma$ . A GSOS law is thought of representing a set of inductive transition rules that specify how programs are run. For example, the choice of  $\mathcal{C} = \mathbf{Set}$  and  $BX = (\mathcal{P}_f X)^L$ , where  $\mathcal{P}_f$  is the finite powerset functor and  $L$  a set of transition labels, leads to the well-known GSOS rule format for specifying labelled transition systems. To every GSOS law  $\rho$  one can canonically associate an *operational model* given by a coalgebra  $\gamma : \mu\Sigma \rightarrow B(\mu\Sigma)$  on the initial algebra  $\mu\Sigma$  (the object of programs), and dually a *denotational model* given by an algebra  $\alpha : \Sigma(\nu B) \rightarrow \nu B$  on the final coalgebra  $\nu B$  (the object of abstract behaviours). In fact,  $\gamma$  and  $\alpha$  respectively extend to an *initial* and *final*  $\rho$ -bialgebra. These universal properties entail an important well-behavedness feature: every language modelled by a GSOS law is *compositional*, that is, strong bisimilarity on its



© Sergey Goncharov, Stefan Milius, Lutz Schröder, Stelios Tsampas, and Henning Urbat; licensed under Creative Commons License CC-BY 4.0

10th Conference on Algebra and Coalgebra in Computer Science (CALCO 2023).

Editors: Paolo Baldan and Valeria de Paiva; Article No. 24; pp. 24:1–24:3



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

operational model is a congruence w.r.t. the operations of the language. Compositionality greatly simplifies reasoning, as it means that the behaviour of a program is determined by the behaviour of its subprograms.

**Higher-Order Abstract GSOS.** In the past 25 years the abstract GSOS framework has been successfully instantiated to a wealth of first-order languages, even fairly complex ones such as CCS or the  $\pi$ -calculus. However, its application to *higher-order* languages, including the  $\lambda$ -calculus, has been a long-standing open problem. The difficulty lies in the phenomenon that higher-order programs represent both computation and data, which means that the behaviour of a generic set  $X$  of programs should be described by a function space like  $BX = X^X$ . Since  $X$  occurs both co- and contravariantly, the map  $X \mapsto BX$  is not functorial, hence abstract GSOS does not apply.

In recent work [5] we introduced *higher-order abstract GSOS*, an extension of Turi and Plotkin's original framework designed to overcome this issue. The core idea is to replace the behaviour endofunctor  $B: \mathcal{C} \rightarrow \mathcal{C}$  by a behaviour *bifunctor*  $B: \mathcal{C}^{\text{op}} \times \mathcal{C} \rightarrow \mathcal{C}$  of mixed variance, and GSOS laws by *higher-order GSOS laws*, given by families of morphisms

$$\rho_{X,Y}: \Sigma(X \times B(X,Y)) \rightarrow B(X, \Sigma^*(X+Y))$$

natural in  $Y \in \mathcal{C}$  and *dinatural* in  $X \in \mathcal{C}$ . For example, combinatory logics like Curry's SKI-calculus can be modelled by a higher-order GSOS law of a polynomial functor  $\Sigma$  over the behaviour bifunctor  $B(X,Y) = Y + Y^X$  on  $\mathcal{C} = \mathbf{Set}$ . For the untyped  $\lambda$ -calculus, we take the presheaf category  $\mathcal{C} = \mathbf{Set}^{\mathbb{F}}$  (where  $\mathbb{F}$  is the category of finite sets), a syntax endofunctor  $\Sigma$  whose initial algebra is the presheaf of  $\lambda$ -terms, and a bifunctor  $B$  modelling  $\beta$ -reductions and the substitution structure of  $\lambda$ -terms, building on earlier ideas by Fiore et al. [3].

Generalizing the first-order case, every higher-order GSOS law  $\rho$  induces an operational model  $\gamma: \mu\Sigma \rightarrow B(\mu\Sigma, \mu\Sigma)$ , which extends to an initial higher-order  $\rho$ -bialgebra. However, in sharp contrast to the first-order case, a final bialgebra usually fails to exist. Nonetheless, higher-order GSOS laws admit a compositional semantics: under mild assumptions on  $\mathcal{C}$ ,  $\Sigma$  and  $B$ , strong bisimilarity on the operational model is a congruence. For example, for the  $\lambda$ -calculus, the operational model extends the transition system on  $\lambda$ -terms given by  $\beta$ -reductions, and strong coalgebraic bisimilarity amounts to strong applicative bisimilarity.

With these foundations at hand, a number of interesting directions open up. In the following we outline a few results, insights, and goals of our ongoing research.

**Weak Bisimilarity.** While the above compositionality result applies to strong bisimilarity, notions of behavioural equivalence for higher-order languages are typically forms of *weak* bisimilarity where computation steps (e.g.  $\beta$ -reductions) are unobservable and only function applications are deemed relevant. A prime example is Abramsky's *applicative bisimilarity* [1] for the  $\lambda$ -calculus. Proving congruence results for weak bisimilarity is known to be challenging for first-order languages and even more so for higher-order ones, where tailor-made proof techniques such as Howe's method are needed. We have recently established such a result in the generality of higher-order abstract GSOS [7]: weak (bi-)similarity on the operational model of a higher-order GSOS law is a congruence provided that its associated weak model forms a *lax* higher-order bialgebra. This generalizes a corresponding result for first-order abstract GSOS [2]. Our theorem holds in all categories  $\mathcal{C}$  where relations are sufficiently well-behaved, e.g. in all (co-)complete, well-powered and locally distributive categories. Its proof is substantially more complex than in the strong case and requires the development of several new techniques of independent interest, including an abstract categorical version

of Howe’s method and the construction of relation liftings of behaviour bifunctors. As an instance of the theorem we recover, e.g., an important property of the  $\lambda$ -calculus originally proved by Abramsky [1]: applicative bisimilarity is a congruence, and hence provides a sound and complete coinductive proof method for contextual equivalence of  $\lambda$ -terms.

A current aim is generalizing the above results from bisimilarity to behavioural distances, e.g. for probabilistic  $\lambda$ -calculi [4], using liftings of bifunctors to quantale-valued relations.

**Logical Relations.** Besides Howe’s method, another important operational technique for reasoning about higher-order languages is given by *logical relations*; for instance, they yield an efficient proof of strong normalization for the simply typed  $\lambda$ -calculus. The idea is as follows: on the set  $\mu\Sigma$  of programs one forms a predicate (or multi-ary relation)  $P$  that implies the property of interest, e.g. normalization, and is compatible with function application, that is, if a program computes a function  $f: \mu\Sigma \rightarrow \mu\Sigma$ , then  $f$  respects  $P$ . One then shows by structural induction that every program lies in  $P$ , whence  $P = \mu\Sigma$ . In practice logical relations are usually invented ad hoc, but their generic flavour allows for a more systematic approach based on higher-order abstract GSOS. In fact, the “ $f$  respects  $P$ ” assertion may be neatly explained using bifunctorial relation liftings, and the generic parts of the structural induction come for free for languages modelled by a higher-order GSOS law. We expect that this approach can greatly reduce the proof obligations for arguments using logical relations.

---

#### References

- 1 S. Abramsky. The lazy  $\lambda$ -calculus. In *Research topics in Functional Programming*, pages 65–117. Addison Wesley, 1990.
- 2 Filippo Bonchi, Daniela Petrişan, Damien Pous, and Jurriaan Rot. Lax bialgebras and up-to techniques for weak bisimulations. In *CONCUR’15*, pages 240–253, 2015.
- 3 Marcelo P. Fiore, Gordon D. Plotkin, and Daniele Turi. Abstract syntax and variable binding. In *LICS’99*, pages 193–202. IEEE, 1999.
- 4 Francesco Gavazzo. Quantitative behavioural reasoning for higher-order effectful programs: Applicative distances. In *LICS’18*, pages 452–461. ACM, 2018.
- 5 Sergey Goncharov, Stefan Milius, Lutz Schröder, Stelios Tsampas, and Henning Urbat. Towards a higher-order mathematical operational semantics. In *POPL’23*, volume 7, pages 632–658. ACM, 2023. [arXiv:2210.13387](#).
- 6 Daniele Turi and Gordon D. Plotkin. Towards a mathematical operational semantics. In *LICS’97*, pages 280–291. IEEE, 1997.
- 7 Henning Urbat, Stelios Tsampas, Sergey Goncharov, Stefan Milius, and Lutz Schröder. Weak similarity in higher-order mathematical operational semantics. In *LICS’23*. IEEE, 2023. [arXiv:2302.08200](#).

