Jouko Väänänen · Åsa Hirvonen
Ruy de Queiroz (Eds.)

LNCS 9803

Logic, Language, Information, and Computation

23rd International Workshop, WoLLIC 2016
Puebla, Mexico, August 16–19th, 2016
Proceedings

Springer

Lecture Notes in Computer Science 9803

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

FoLLI Publications on Logic, Language and Information

Subline of Lectures Notes in Computer Science

More information about this series at http://www.springer.com/series/7407

Jouko Väänänen · Åsa Hirvonen
Ruy de Queiroz (Eds.)

Logic, Language, Information, and Computation

23rd International Workshop, WoLLIC 2016
Puebla, Mexico, August 16–19th, 2016
Proceedings

Springer

Editors
Jouko Väänänen
Department of Mathematics and Statistics
University of Helsinki
Helsinki
Finland

Åsa Hirvonen
Department of Mathematics and Statistics
University of Helsinki
Helsinki
Finland

Ruy de Queiroz
Centro de Informática
Recife, Pernambuco
Brazil

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-662-52920-1 ISBN 978-3-662-52921-8 (eBook)
DOI 10.1007/978-3-662-52921-8

Library of Congress Control Number: 2016944475

LNCS Sublibrary: SL1 – Theoretical Computer Science and General Issues

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer-Verlag GmbH Berlin Heidelberg

Preface

This volume contains the papers presented at the 23rd Workshop on Logic, Language, Information and Computation (WoLLIC 2016) held during August 16–19, 2016, at the Department of Computer Science, Benemérita Universidad Autónoma de Puebla, Puebla, Mexico. The WoLLIC series of workshops started in 1994 with the aim of fostering interdisciplinary research in pure and applied logic. The idea is to have a forum that is large enough in the number of possible interactions between logic and the sciences related to information and computation, and yet is small enough to allow for concrete and useful interaction among participants.

There were 41 submissions this year. Each submission was reviewed by at least three Program Committee members. The committee decided to accept 23 papers. The program also included six invited lectures by Pablo Barceló (Universidad de Chile, Chile), Dana Bartošová (University of Sáo Paulo, Brazil), Johann A. Makowsky (Technion - Israel Institute of Technology, Israel), Alessandra Palmigiano (TU Delft, The Netherlands), Sonja Smets (University of Amsterdam, The Netherlands), and Andrés Villaveces (Universidad Nacional de Colombia, Colombia). There were also five tutorials given by Barceló, Makowsky, Palmigiano, Smets, and Villaveces.

As a tribute to a recent breakthrough in mathematics, there was also a screening of Csicsery's "Counting from Infinity: Yitang Zhang and the Twin Prime Conjecture" (2015), which centers on the life and work of Yitang Zhang in the celebrated twin prime conjecture, his result being that there are infinitely many pairs of primes separated by at most 70 million.

We would very much like to thank all Program Committee members and external reviewers for the work they put into reviewing the submissions. The help provided by the EasyChair system created by Andrei Vorokonkov is gratefully acknowledged. Finally, we would like to acknowledge the generous financial support by the Benemérita Universidad Autónoma de Puebla's Department of Computer Science, and the scientific sponsorship of the following organizations: Interest Group in Pure and Applied Logics (IGPL), The Association for Logic, Language and Information (FoLLI), Association for Symbolic Logic (ASL), European Association for Theoretical Computer Science (EATCS), European Association for Computer Science Logic (EACSL), Sociedade Brasileira de Computação (SBC), and Sociedade Brasileira de Lógica (SBL).

May 2016

Åsa Hirvonen
Ruy de Queiroz
Jouko Väänänen

Organization

Program Committee

Samson Abramsky	Oxford University, UK
Dietmar Berwanger	CNRS and Université Paris-Saclay, France
Guram Bezhanishvili	New Mexico State University, USA
Ruy de Queiroz	Centro de Informatica, Universidade Federal de Pernambuco, Brazil
Arnaud Durand	Université Paris 7, France
Pietro Galliani	University of Sussex, UK
Nina Gierasimczuk	University of Amsterdam, The Netherlands
Jeroen Groenendijk	University of Amsterdam, The Netherlands
Lauri Hella	University of Tampere, Finland
Wesley Holliday	University of California, USA
Juha Kontinen	University of Helsinki, Finland
Larry Moss	Indiana University, USA
Andre Nies	University of Auckland, New Zealand
Aarne Ranta	University of Gothenburg, Sweden
Mehrnoosh Sadrzadeh	Queen Mary University of London, UK
Norma Short	Aix en Provence, France
Rineke Verbrugge	University of Groningen, The Netherlands
Heribert Vollmer	Leibniz Universität Hannover, Germany
Jouko Väänänen	University of Helsinki and University of Amsterdam, Finland and The Netherlands
Dag Westerståhl	University of Stockholm, Sweden

Tutorial/Abstracts

Ultrafilters in Dynamics and Ramsey Theory

Dana Bartošová

Department of Mathematics, University of Toronto, Toronto, Canada
dana.bartosova@mail.utoronto.ca

Abstract. I will recall some famous Ramsey-type statements that admit a simple proof with the use of ultrafilter on discrete semigroups. Gowers' Ramsey theorem will be an example that up-to-date does not posses an ultrafilter-free proof. Stepping up from discrete (semi)groups to groups of automorphisms of homogeneous structures, I will show how their dynamics connects with structural Ramsey theory and how combinatorics on ultrafilters is relevant to dynamical problems. This is partially a joint work with Andrew Zucker (Carnegie Mellon University).

When is $\mathbf{P}_{\mathfrak{A}} = \mathbf{NP}_{\mathfrak{A}}$ over Arbitrary Structures $\mathfrak{A}$? (A tutorial)

J.A. Makowsky

Department of Computer Science, Technion - Israel Institute of Technology, Haifa, Israel
janos@cs.technion.ac.il

Abstract. In a series of lectures we review the complexity theory for computations over arbitrary relational and algebraic structures $\mathfrak{A}$.

We will cover the following topics:

(i) Register machines over arbitrary relational and algebraic structures $\mathfrak{A}$. Some history, H. Friedman's work of the 1970 ties, [FM92]. The Blum-Shub-Smale approach to complexity, [BCSS96, BCSS98].
(ii) What do we expect from a theory of computability over the reals? Critical evaluations, [Fef15, BC06, Mam14].
(iii) The role of quantifier elimination: B. Poizat's characterization of P = NP over $\mathfrak{A}$, [Poi95, Pru06].
(iv) Proving quantifier elimination. Presburger arithmetic and the field of complex numbers. Shoenfield's quantifier elimination theorem, [KK67, Hod93].
(v) Disproving quantifier elimination. The missing predicates.
(vi) For which structures $\mathfrak{A}$ can we prove $\mathbf{P}_{\mathfrak{A}} \neq \mathbf{NP}_{\mathfrak{A}}$? Abelian groups and boolean algebras, [Pru02, Pru03]
(vii) The logical content of the P = NP problem. Fast quantifier elimination vs. descriptive complexity, [Lib04].

Similar courses were given:

2013: At the Computer Science Department of the Technion–Israel Institute of Technology as Graduate Seminar 238900 under the title *The millennium question* **P** = **NP** *over the real numbers.*
2014: At the 5th Indian School of Logic and Applications (ISLA-2014) at Tezpur University, Assam, India, under the title **P** $=_?$ **NP** *over arbitrary structures*.
2014: At the 26th European Summer School in Logic, Language and Information (ESSLLI 2014) in an enlarged form together with K. Meer, also under the title **P** $=_?$ **NP** *over arbitrary structures*.

See www.cs.technion.ac.il/~janos/#invitations.

Partially supported by a grant of Technion Research Authority. Work done in part while the author was visiting the Simons Institute for the Theory of Computing in Spring 2016.

References

[BC06] Braverman, M., Cook, S.: Computing over the reals: foundations for scientific computing. Not. AMS **53**(3), 318–329 (2006)

[BCSS96] Blum, L., Cucker, F., Shub, M., Smale, S.: Algebraic settings for the problem "P $\neq$ NP?". In: The Mathematics of Numerical Analysis, Number 32 in Lectures in Applied Mathematics, pp. 125–144. Amer. Math. Soc. (1996)

[BCSS98] Blum, L., Cucker, F., Shub, M., Smale, S.: Complexity and Real Computation. Springer (1998)

[Fef15] Feferman, S.: Theses for computation and recursion on concrete and abstract structures. In: Turing's Revolution, pp. 105–126. Springer (2015)

[FM92] Friedman, H., Mansfield, R.: Algorithmic procedures. Trans. Am. Math. Soc. 297–312 (1992)

[Hod93] Hodges, W.: Model theory, vol. 42. In: Encyclopedia of Mathematics and its Applications. Cambridge University Press (1993)

[KK67] Kreisel, G., Krivine, J.L.: Elements of Mathematical Logic: Model Theory. North Holland (1967)

[Lib04] Libkin, L.: Elements of Finite Model Theory. Springer (2004)

[Mam14] Mamino, M.: On the computing power of +, −, and $\times$. In: Proceedings of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), p. 68. ACM (2014)

[Poi95] Poizat, B.: Les Petits Cailloux: Une Approche Modèle-Théorique De L'algorithmie. Aléas, Paris (1995)

[Pru02] Prunescu, M.: A model-theoretic proof for p $\neq$ np over all infinite abelian group. J. Symbolic Logic **67**(01), 235–238 (2002)

[Pru03] Prunescu, M.: P $\neq$ np for all infinite Boolean algebras. Math. Logic Q. **49**(2), 210–213 (2003)

[Pru06] Prunescu, M.: Fast quantifier elimination means p = np. In: Logical Approaches to Computational Barriers, pp. 459–470. Springer (2006)

Proof Systems for the Logics for Social Behaviour

Alessandra Palmigiano

Technical University of Delft, Delft, The Netherlands

The range of 'logics for social behaviour' (by which I mean those logics aimed at capturing aspects such as agency and information flow) is rapidly expanding, and their theory is being intensively investigated, especially w.r.t. their semantic aspects. However, these logics typically lack a comparable proof-theoretic development. More often than not, the hurdles preventing their standard proof-theoretic development are due to the very features which make them capture essential aspects of the real world, such as their not being closed under uniform substitution, or the presence of certain extralinguistic labels and devices encoding key interactions between logical connectives [5].

In this talk I will focus on *multi-type calculi*, a methodology introduced in [3, 4, 7] to provide DEL and PDL with analytic calculi, and pursued also in [1, 2, 6].

Multi-type languages allow the upgrade of actions, agents, coalitions, etc. from *parameters* in the generation of formulas, to *terms*. Like formulas, they thus become first-class citizens of the framework, endowed with their corresponding structural connectives and rules. In this richer environment, many features which were insurmountable hurdles to the standard treatment can be understood as symptoms of the original languages of these logics lacking the necessary expressivity to encode certain key interactions *within the language*. The success of the multi-type methodology in defining analytic calculi for logics as proof-theoretically impervious as DEL lies in its providing a mathematical environment in which the expressivity problems can be clearly identified.

I will argue that multi-type calculi can provide a platform for a uniform proof-theoretic account of the logics for social behaviour.

References

1. Bilkova, M., Greco, G., Palmigiano, A., Tzimoulis, A., Wijnberg, N.: Logic of resources and capabilities (In preparation, 2016)
2. Frittella, S., Greco, G., Palmigiano, A., Yang, F.: Structural multi-type sequent calculus for inquisitive logic. In: Proceedings of the WoLLIC 2016 (2016). arXiv:1604.00936v1
3. Frittella, S., Greco, G., Kurz, A., Palmigiano, A.: Multi-type display calculus for propositional dynamic logic. J. Logic Comput. (2014). Special Issue on Substructural Logic and Information Dynamics

4. Frittella, S., Greco, G., Kurz, A., Palmigiano, A., Sikimić, V.: A multi-type display calculus for dynamic epistemic logic. J. Logic Comput. (2014). Special Issue on Substructural Logic and Information Dynamics
5. Frittella, S., Greco, G., Kurz, A., Palmigiano, A., Sikimić, V.: A proof-theoretic semantic analysis of dynamic epistemic logic. J. Logic Comput. (2014). Special Issue on Substructural Logic and Information Dynamics
6. Frittella, S., Greco, G., Kurz, A., Palmigiano, A., Sikimić, V.: Multi-type sequent calculi. In: Proceedings of the Trends in Logic, vol. XIII, pp. 81–93 (2014)
7. Greco, G., Kurz, A., Palmigiano, A.: Dynamic epistemic logic displayed. Logic, rationality and interaction. In: Proceedings of the Fourth International Workshop. LORI 2013

Sahlqvist Correspondence via Duality and Its Applications

Alessandra Palmigiano

Technical University of Delft, Delft, The Netherlands

Since the 1970s, correspondence theory has been one of the most important items in the toolkit of modal logicians. Unified correspondence [6] is a very recent approach, which has imported techniques from duality, algebra and formal topology [10] and exported the state of the art of correspondence theory well beyond normal modal logic, to a wide range of logics including, among others, intuitionistic and distributive lattice-based (normal modal) logics [8], non-normal (regular) modal logics [18], substructural logics [5, 7, 9], hybrid logics [13], and mu-calculus [2, 3, 4].

The breadth of this work has stimulated many and varied applications. Some are closely related to the core concerns of the theory itself, such as the understanding of the relationship between different methodologies for obtaining canonicity results [7, 17], or of the phenomenon of pseudo-correspondence [11]. Other, possibly surprising applications include the dual characterizations of classes of finite lattices [14], the identification of the syntactic shape of axioms which can be translated into analytic rules of proper display and Gentzen calculi [15, 16], and the design of display-type calculi for the logics of resources and capabilities, and their applications to the logical modelling of business organizations [1]. Finally, the insights of unified correspondence theory have made it possible to determine the extent to which the Sahlqvist theory of classes of normal DLEs can be reduced to the Sahlqvist theory of normal Boolean expansions, by means of Gödel-type translations [12].

The most important technical tools in unified correspondence are: (a) a very general syntactic definition of the class of Sahlqvist formulas, which applies uniformly to each logical signature and is given purely in terms of the order-theoretic properties of the algebraic interpretations of the logical connectives; (b) the algorithm ALBA, which effectively computes first-order correspondents of input term-inequalities, and is guaranteed to succeed on a wide class of inequalities (the so-called *inductive* inequalities) which, like the Sahlqvist class, can be defined uniformly in each mentioned signature, and which properly and significantly extends the Sahlqvist class.

In this tutorial, the fundamental principles and conceptual insights underlying these developments will be illustrated in the setting of Boolean algebras with operators [10].

References

1. Bilkova, M., Greco, G., Palmigiano, A., Tzimoulis, A., Wijnberg, N.: The logic of resources and capabilities (In preparation, 2016)
2. Conradie, W., Craig, A.: Canonicity results for mu-calculi: an algorithmic approach. J. Logic Comput. (forthcoming). arXiv:1408.6367 (arXiv Preprint)

3. Conradie, W., Craig, A., Palmigiano, A., Zhao, Z.: Constructive canonicity for lattice-based fixed point logics (Submitted). arXiv:1603.06547 (arXiv preprint)
4. Conradie, W., Fomatati, Y., Palmigiano, A., Sourabh, S.: Algorithmic correspondence for intuitionistic modal mu-calculus. Theoret. Comput. Sci. **564**, 30–62 (2015)
5. Conradie, W., Frittella, S., Palmigiano, A., Piazzai, M., Tzimoulis, A., Wijnberg, N.: Categories: how I learned to stop worrying and love two sorts (Submitted). arXiv:1604.00777 (arXiv preprint)
6. Conradie, W., Ghilardi, S., Palmigiano, A.: Unified correspondence. In: Baltag, A., Smets, S. (eds.) Johan van Benthem on Logic and Information Dynamics. Outstanding Contributions to Logic, vol. 5, pp. 933–975. Springer International Publishing (2014)
7. Conradie, W., Palmigiano, A.: Constructive canonicity of inductive inequalities (Submitted). arXiv:1603.08341 (arXiv preprint)
8. Conradie, W., Palmigiano, A.: Algorithmic correspondence and canonicity for distributive modal logic. Annals Pure Applied Logic **163**(3), 338–376 (2012)
9. Conradie, W., Palmigiano, A.: Algorithmic correspondence and canonicity for non-distributive logics. J. Logic Comput. (forthcoming). arXiv:1603.08515 (arXiv preprint)
10. Conradie, W., Palmigiano, A., Sourabh, S.: Algebraic modal correspondence: Sahlqvist and beyond (Submitted)
11. Conradie, W., Palmigiano, A., Sourabh, S., Zhao, Z.: Canonicity and relativized canonicity via pseudo-correspondence: an application of ALBA (Submitted). arXiv:1511.04271 (arxiv preprint)
12. Conradie, W., Palmigiano, A., Zhao, Z.: Sahlqvist via translation (Submitted). arXiv:1603.08220 (arXiv preprint)
13. Conradie, W., Robinson, C.: On Sahlqvist theory for hybrid logic. J. Logic Comput. doi:10.1093/logcom/exv045
14. Frittella, S., Palmigiano, A., Santocanale, L.: Dual characterizations for finite lattices via correspondence theory for monotone modal logic. J. Logic Comput. (forthcoming). arXiv:1408.1843 (arXiv preprint)
15. Greco, G., Ma, M., Palmigiano, A., Tzimoulis, A., Zhao, Z.: Unified correspondence as a proof-theoretic tool. J. Logic Comput. (forthcoming). arXiv:1603.08204 (arXiv preprint)
16. Ma, M., Zhao, Z.: Unified correspondence and proof theory for strict implication. J. Logic Comput. (forthcoming). arXiv:1604.08822 (arXiv preprint)
17. Palmigiano, A., Sourabh, S., Zhao, Z.: Jónsson-style canonicity for ALBA-inequalities. J. Logic Comput. doi:10.1093/logcom/exv041
18. Palmigiano, A., Sourabh, S., Zhao, Z.: Sahlqvist theory for impossible worlds. J. Logic Comput. (forthcoming). arXiv:1603.08202 (arXiv preprint)

Informational Cascades: A Test for Rationality?

Sonja Smets

Institute for Logic, Language and Computation,
University of Amsterdam, Amsterdam, The Netherlands

Abstract. I report on joint work with A. Baltag, Z. Christoff and J.U. Hansen in [3], based on our investigation of the decision processes of individuals that lead to the social herding phenomenon known as informational cascades. The question we address in our paper deals with whether rational agents who use their higher-order reasoning powers and who can reflect on the fact that they are part of an informational cascade, can ultimately stop the cascade from happening. To answer this question we use dynamic epistemic logic to give a complete analysis of the information flow in an informational cascade, capturing the agent's observations, their communication and their higher-order reasoning power. Our models show that individual rationality isn't always a cure that can help us to stop a cascade. However, other factors that deal with the underlying communication protocol or that focus on the reliability of agents in the group, give rise to conditions that can be imposed to prevent or stop an informational cascade from happening in certain scenarios.

Informational cascades are social herding phenomena in which individual agents in a sequence decide to follow the decisions of their predecessors while simply ignoring their own private evidence. In such situations, individuals are given information about their predecessors' decisions but not about the reasons or the evidence on which these decisions are based. So when the first agents in the sequence made a correct decision, their followers will all get it right. However, the opposite can easily happen and when everyone gets it wrong we end up with a potential social-epistemic catastrophe. Such phenomena can illustrate a clear case of when social features interfere with agent's truth-tracking abilities. Hence not all situations involving communication and rational deliberation seem to be epistemically beneficial at the group level.

In this context we study the logical mechanism behind such informational cascades. It is important to note that we are looking at situations in which the total sum of private information should in principle be enough for the group to track the truth, yet in an informational cascade the group fails to do so. To gain a better understanding of this phenomenon, it is our aim to check whether this failure to track the truth can be due to any form of irrationality present when agents form or aggregate their beliefs. Our investigation is driven by questions such as: are rational and introspective agents, who reflect upon their own knowledge and beliefs and who can reason about the knowledge and beliefs of their predecessors, able to stop or prevent a cascade? Even more, are agents with unboundedly rational powers, and who are aware of the dangers of the sequential deliberation protocol that they are part of, able to block a cascade? Indeed, in some cases a cascade can be prevented by making agents aware of it. However, as is shown in [3] this is not always the case.

There are examples of informational cascades in which no amount of higher-order reasoning is enough to stop an informational cascade. Our argument is based on a model of examples of informational cascades in [3], allowing us to represent the individual reasoning of each agent involved. Formally, we use the tools of dynamic epistemic logic [4, 5, 6, 9]. On the one hand we use a probabilistic dynamic epistemic logic to represent agents who apply probabilistic conditioning. On the other hand we also model the situation in which agents do not use sophisticated probabilistic tools but rather apply a simply non-Bayesian form of heuristic reasoning. We note that a full syntactic encoding of an informational cascade in the presence of a common knowledge operator, is offered in [2] based on a logic that combines a variant of the Logic of Communication and Change from [7] and a variant of Probabilistic Dynamic Epistemic Logic in [8].

Based on our logical analysis in [3], we conclude that cascades cannot always be avoided by rational means. Our model of unboundedly rational agents, equipped with full higher-order reasoning powers, shows that these agents (irrespective of whether they adopt Bayesian reasoning or another non-Bayesian heuristic) still end up in a cascade. Even more, the group's inability to track the truth may actually be a direct consequence of each agent's rational attempt to track the truth individually.

Investigations of different cascade scenarios point out that changes in the underlying communication protocol can make a difference. In most cascade scenarios, agents announce their decisions to their followers, i.e. they communicate about their opinions and beliefs but not about the reasons for their beliefs. Following [3], one can argue that exactly the fact that this communication protocol is based on the exchange of partial information, is the problem. Indeed allowing for more communication in which agents can share not only their beliefs but also their justifications, may stop the cascade. In ideal cases, when total communication can be achieved and agents share all their evidence, reasons, beliefs, etc., we can effectively stop a cascade. It is interesting to investigate different types of communication protocols and their effect on the formation of cascades. An analysis in which such protocols are formalised as strategies in a game theoretic setting, is provided in [1]. Further investigations point out that other social factors can similarly affect the outcome of an informational cascade. For instance the level of trust among agents in a group can make a difference. In [10] the results of an experiment are shown which indicates that agent's perceived reliability of their predecessors can affect the formation of a cascade.

Acknowledgement. Sonja Smets' research on this topic is funded by the European Research Council under the European Community's Seventh Framework Programme (FP7/2007-2013)/ERC Grant agreement no. 283963.

References

1. Achimescu, A.: Games and Logics for Informational Cascades. Master's thesis, ILLC University of Amsterdam, master of Logic Thesis, MoL-2014-04 (2014)
2. Achimescu, A., Baltag, A., Sack, J.: The probabilistic logic of communication and change. J. Logic Comput. (2016)
3. Baltag, A., Christoff, Z., Hansen, J.U., Smets, S.: Logical models of informational cascades. In: van Benthem, J., Lui, F. (eds.) Logic across the University: Foundations and Applications, pp. 405–432. Studies in Logic, College Publications (2013)
4. Baltag, A., Moss, L.: Logics for epistemic programs. Synthese **139**, 165–224 (2004)
5. Baltag, A., Moss, L., Solecki, S.: The logic of public announcements, common knowledge and private suspicions. In: Proceedings of TARK 1998 (Seventh Conference on Theoretical Aspects of Rationality and Knowledge), pp. 43–56. Morgan Kaufmann Publishers (1998)
6. van Benthem, J.: Logical Dynamics of Information and Interaction. Cambridge University Press (2011)
7. van Benthem, J., Eijck, J., Kooi, B.: Logics of communication and change. Inf. Commun. **204**, 1620–1662 (2006)
8. van Benthem, J., Gerbrandy, J., Kooi, B.: Dynamic update with probabilities. Stud. Logica. **93**, 67–96 (2009)
9. van Ditmarsch, H., van der Hoek, W., Kooi, B.: Dynamic epistemic logic, vol. 337. In: Synthese Library, Springer, The Netherlands (2008)
10. van Weegen, L.: Informational cascades under variable reliability assessments. A formal and empirical investigation. Master's thesis, ILLC University of Amsterdam, master of Logic Thesis, MoL-2014-21 (2014)

Belief Dynamics in a Social Context

Sonja Smets

University of Amsterdam, Amsterdam, The Netherlands

This tutorial is addressed to researchers and students who are interested in the logical/philosophical study of notions of belief and knowledge, including group beliefs and collective "knowledge". We are interested both in the representation of these different types of attitudes as well as in their dynamics, i.e. how these attitudes change in communities of interconnected agents capable of reflection, communication, reasoning, argumentation etc. I will start by introducing the basic concepts and models, using standard techniques from Dynamic Epistemic Logic and their adaptations for dealing with belief revision. I will further focus on characterizing a group's "epistemic potential" and I touch on cases in which a group's ability to track the truth is higher than that of each of its members. This tutorial paves the way for my invited lecture in which I focus on situations in which the group's dynamics leads to informational distortions (i.e. the "madness of the crowds", in particular the phenomenon of informational cascades). This tutorial is based on a number of recent papers that make use of a variety of formal tools ranging over dynamic epistemic logics, game theory and network theory.

Generalized Amalgamation Classes and Limit Models: Implicit Logics

Andrés Villaveces

Departamento de Matemáticas, Universidad Nacional de Colombia
Bogotá 111321, Colombia
avillavecesn@unal.edu.co

Abstract. This is a two-hour tutorial on two kinds of (generalized) amalgamation classes and the emergence of language (implicit logic) from their semantical properties: abstract elementary classes and sheaves of structures. I will provide definitions, examples and a description of the emergence of logic from their purely semantical properties.

– Amalgamation classes. Ordered and controlled by topologies. Examples and problems.
– Examples: sheaves of structures and abstract elementary classes with amalgamation. Orbital (Galois) types and language.
– Implicit language from semantics. The Presentation Theorem.
– Interpolation in AECs: comparing languages.

Contents

The Useful MAM, a Reasonable Implementation of the Strong λ-Calculus

Beniamino Accattoli(✉)

INRIA and LIX, École Polytechnique, Palaiseau, France
beniamino.accattoli@inria.fr

Abstract. It has been a long-standing open problem whether the strong λ-calculus is a reasonable computational model, i.e. whether it can be implemented within a polynomial overhead with respect to the number of β-steps on models like Turing machines or RAM. Recently, Accattoli and Dal Lago solved the problem by means of a new form of sharing, called *useful sharing*, and realised via a calculus with explicit substitutions. This paper presents a new abstract machine for the strong λ-calculus based on useful sharing, the *Useful Milner Abstract Machine*, and proves that it reasonably implements leftmost-outermost evaluation. It provides both an alternative proof that the λ-calculus is reasonable and an improvement on the technology for implementing strong evaluation.

1 Introduction

The higher-order computational model of reference is the λ-calculus, that comes in two variants, *weak* or *strong*. Introduced at the inception of computer science as a mathematical approach to computation, it later found applications in the theoretical modelling of programming languages and, more recently, proof assistants. The weak λ-calculus is the backbone of functional languages such as LISP, Scheme, OCAML, or Haskell. It is *weak* because evaluation does not enter function bodies and, usually, terms are assumed to be closed. By removing these restrictions one obtains the *strong* λ-calculus, that underlies proof assistants like Coq, Isabelle, and Twelf, or higher-order logic programming languages such as λ-prolog or the Edinburgh Logical Framework. Higher-order features nowadays are also part of mainstream programming languages like Java or Python.

The abstract, mathematical character is both the advantage and the drawback of the higher-order approach. The advantage is that it enhances the modularity and the conciseness of the code, allowing to forget about low-level details at the same time. The drawback is that the distance from low-level details makes its complexity harder to analyse, in particular its main computational rule, called β-*reduction*, at first sight is not an atomic operation. In particular, β can be nasty, and make the program grow at an exponential rate. The number of β-steps, then, does not even account for the time to write down the result, suggesting that it is not a reasonable cost model. This is the *size-explosion problem* [6], and affects both the weak and the strong λ-calculus.

J. Väänänen et al. (Eds.): WoLLIC 2016, LNCS 9803, pp. 1–21, 2016.
DOI: 10.1007/978-3-662-52921-8_1

The λ-Calculus is Reasonable, Indeed. A cornerstone of the theory is that, nonetheless, in the weak λ-calculus the number of β-steps *is* a reasonable cost model for time complexity analyses [9,14,25], where *reasonable* formally means that it is polynomially related to the cost model of RAM or Turing machines.

For the strong λ-calculus, the techniques developed for the weak one do not work, as wilder forms of size-explosion are possible. A natural candidate cost model from the theory of λ-calculus is the number of (Lévy) optimal parallel steps, but it has been shown by Asperti and Mairson that such a cost model is not reasonable [8].

It is only very recently that the strong case has been solved by Accattoli and Dal Lago [6], who showed that the number of leftmost-outermost β-steps to full normal form is a reasonable cost model. The proof of this result relies on two theoretical tools. First, the *Linear Substitution Calculus* (LSC), an expressive and simple decomposition of the λ-calculus via linear logic and rewriting theory, developed by Accattoli and Kesner [3] as a variation over a calculus by Robin Milner [24]. Second, *useful sharing*, a new form of shared evaluation introduced by Accattoli and Dal Lago on top of the LSC. Roughly, the LSC is a calculus where the meta-level operation of substitution used by β-reduction is internalised and decomposed in micro steps, *i.e.* it is what is usually called a calculus with *explicit substitutions*. The further step is to realise that some of these micro substitution steps are useless: they do not lead to the creation of other β-redexes, their only aim is to unshare the result and provide the full normal form. Useful evaluation then performs only those substitution steps that are useful, *i.e.* not useless. By avoiding useless unsharing steps, it computes a shared representation of the normal form of size linear in the number of steps, whose unsharing may cause an exponential blow up in size. This is how the size-explosion problem is circumvented, see [6] for more explanations.

This Paper. In this paper we provide an alternative proof that the strong λ-calculus is reasonable (actually only of the hard half, that is the simulation of λ-calculus on RAM, the other half being much easier, see [5]), by replacing the LSC with the *Useful Milner Abstract Machine*. The aim of the paper is threefold:

1. *Getting Closer To Implementations*: the LSC decomposes β-reduction in micro-steps but omits details about the search for the next redex to reduce. Moreover, in [6] useful sharing is used as a sort of black box on top of the LSC. Switching to abstract machines provides a solution closer to implementations and internalises useful sharing.
2. *The First Reasonable Strong Abstract Machine*: the literature on abstract machines for strong evaluation is scarce (see below) and none of the machines in the literature is reasonable. This work thus provides an improvement of the technology for implementing strong evaluation.
3. *Alternative Proof*: the technical development in [6] is sophisticated, because a second aim of that paper is to connect some of the used tools (namely useful sharing and the subterm property) with the seemingly unrelated notion of *standardisation* from rewriting theory. Here we provide a more basic, down-to-earth approach, not relying on advanced rewriting theory.

The Useful MAM. The Milner Abstract Machine (MAM) is a variant with just one *global environment* of the Krivine Abstract Machine (KAM), introduced in [1] by Accattoli, Barenbaum, and Mazza. The same authors introduce in [2] the Strong MAM, *i.e.* the extension of the MAM to strong evaluation, that is a version with just one global environment of Cregut's Strong KAM [13], essentially the only other abstract machine for strong (call-by-name) evaluation in the literature. Both are not reasonable. The problem is that these machines do not distinguish between useful and useless steps.

The Useful MAM introduced in this paper improves the situation, by refining the Strong MAM. The principle is quite basic, let us sketch it. Whenever a β-redex $(\lambda x.t)u$ is encountered, the Strong MAM adds an entry $[x{\leftarrow}u]$ to the environment E. The Useful MAM, additionally, executes an auxiliary machine on u—the Checking Abstract Machine (Checking AM)—to establish its usefulness. The result of this check is a label l that is attached to the entry $[x{\leftarrow}u]^l$. Later on, when an occurrence of x is found, the Useful MAM replaces x with u only if the label on $[x{\leftarrow}u]^l$ says that it is useful. Otherwise the machine backtracks, to search for the next redex to reduce.

The two results of the paper are:

1. *Qualitative* (Theorem 2): the Useful MAM correctly and completely implements leftmost-outermost (LO for short) β-evaluation—formally, the two are weakly bisimilar.
2. *Quantitative* (Theorem 5): the Useful MAM is a reasonable implementation, *i.e.* the work done by both the Useful MAM and the Checking AM is polynomial in the number of LO β-steps and in the size of the initial term.

Related Work. Beyond Crégut's [12,13] and Accattoli, Barenbaum, and Mazza's [2], we are aware of only two other works on strong abstract machines, García-Pérez, Nogueira and Moreno-Navarro's [22] (2013), and Smith's [27] (unpublished, 2014). Two further studies, de Carvalho's [11] and Ehrhard and Regnier's [19], introduce strong versions of the KAM but for theoretical purposes; in particular, their design choices are not tuned towards implementations (*e.g.* rely on a naïve parallel exploration of the term). Semi-strong machines for call-by-value (*i.e.* dealing with weak evaluation but on open terms) are studied by Grégoire and Leroy [23] and in a recent work by Accattoli and Sacerdoti Coen [4] (see [4] for a comparison with [23]). More recent work by Dénès [18] and Boutiller [10] appeared in the context of term evaluation in Coq. None of the machines for strong evaluation in the literature is reasonable, in the sense of being polynomial in the number of β-steps. The machines developed by Accattoli and Sacerdoti Coen in [4] are reasonable, but they are developed in a semi-strong setting only. Another difference between [4] and this work is that call-by-value simplifies the treatment of usefulness because it allows to compute the labels for usefulness while evaluating the term, that is not possible in call-by-name.

Global environments are explored by Fernández and Siafakas in [20], and used in a minority of works, *e.g.* [17,25]. Here we use the terminology for abstract machines coming from the distillation technique in [1], related to the *refocusing* *semantics* of Danvy and Nielsen [16] and introduced to revisit the relationship between the KAM and weak linear head reduction pointed out by Danos and Regnier [15]. We do not, however, employ the distillation technique itself.

Proofs. All proofs have been omitted. Those of the main lemmas and theorems concerning the Useful MAM can be found in the appendix. The other ones can be found in the longer version on the author's web page.

2 λ-Calculus and Leftmost-Outermost Evaluation

The syntax of the λ-calculus is given by the following grammar for terms:

$$\lambda - \textsc{Terms} \qquad t, u, w, r ::= x \mid \lambda x.t \mid tu.$$

We use $t\{x{\leftarrow}u\}$ for the usual (meta-level) notion of substitution. An abstraction $\lambda x.t$ binds x in t, and we silently work modulo α-equivalence of bound variables, *e.g.* $(\lambda y.(xy))\{x{\leftarrow}y\} = \lambda z.(yz)$. We use $\mathtt{fv}(t)$ for the set of free variables of t.

Contexts. One-hole contexts C and the *plugging* $C\langle t\rangle$ of a term t into a context C are defined by:

Contexts	Plugging	
$C ::= \langle\cdot\rangle \mid \lambda x.C \mid Ct \mid tC$	$\langle\cdot\rangle\langle t\rangle := t$	$(Cu)\langle t\rangle := C\langle t\rangle u$
	$(\lambda x.C)\langle t\rangle := \lambda x.C\langle t\rangle$	$(uC)\langle t\rangle := uC\langle t\rangle$

As usual, plugging in a context can capture variables, *e.g.* $(\lambda y.(\langle\cdot\rangle y))\langle y\rangle = \lambda y.(yy)$. The plugging $C\langle C'\rangle$ of a context C' into a context C is defined analogously. A context C is *applicative* if $C = C'\langle\langle\cdot\rangle\overline{u}\rangle$ for some C' and $\overline{u}$.

We define β-reduction $\to_\beta$ as follows:

Rule at Top Level	Contextual closure
$(\lambda x.t)u \mapsto_\beta t\{x{\leftarrow}u\}$	$C\langle t\rangle \to_\beta C\langle u\rangle \quad \text{if } t \mapsto_\beta u$

A term t is a *normal form*, or simply *normal*, if there is no u such that $t \to_\beta u$, and it is *neutral* if it is normal and it is not of the form $\lambda x.u$ (*i.e.* it is not an abstraction). The *position* of a β-redex $C\langle t\rangle \to_\beta C\langle u\rangle$ is the context C in which it takes place. To ease the language, we will identify a redex with its position. A *derivation* $d : t \to^k u$ is a finite, possibly empty, sequence of reduction steps. We write $|t|$ for the size of t and $|d|$ for the length of d.

Leftmost-Outermost Derivations. The left-to-right outside-in order on redexes is expressed as an order on positions, *i.e.* contexts.

Definition 1 (Left-to-Right Outside-In Order).

1. *The* outside-in order*:*
   (a) Root*: $\langle\cdot\rangle \prec_O C$ for every context $C \neq \langle\cdot\rangle$;*
   (b) Contextual closure*: If $C \prec_O C'$ then $C''\langle C\rangle \prec_O C''\langle C'\rangle$ for any C''.*

2. *The* left-to-right order*: $C \prec_L C'$ is defined by:*
   (a) Application*: If $C \prec_p t$ and $C' \prec_p u$ then $Cu \prec_L tC'$;*
   (b) Contextual closure*: If $C \prec_L C'$ then $C''\langle C\rangle \prec_L C''\langle C'\rangle$ for any C''.*
3. *The* left-to-right outside-in order*: $C \prec_{LO} C'$ if $C \prec_O C'$ or $C \prec_L C'$:*

The following are a few examples. For every context C, it holds that $\langle\cdot\rangle \not\prec_L C$. Moreover $(\lambda x.\langle\cdot\rangle)t \prec_O (\lambda x.(\langle\cdot\rangle u))t$ and $(\langle\cdot\rangle t)u \prec_L (wt)\langle\cdot\rangle$.

Definition 2 (LO β-Reduction). *Let t be a λ-term and C a redex of t. C is the* leftmost-outermost *β-redex (LO β for short) of t if $C \prec_{LO} C'$ for every other β-redex C' of t. We write $t \to_{\mathtt{LO}\beta} u$ if a step reduces the LO β-redex.*

The next immediate lemma guarantees that we defined a total order.

Lemma 1 (Totality of $\prec_{LO}$). *If $C \prec_p t$ and $C' \prec_p t$ then either $C \prec_{LO} C'$ or $C' \prec_{LO} C$ or $C = C'$. Therefore, $\to_{\mathtt{LO}\beta}$ is deterministic.*

LO Contexts. For the technical development of the paper we need two characterisations of when a context is the position of the LO β-redex. The first, following one, is used in the proofs of Lemma 5.2 and Lemma 6.4.

Definition 3 (LO Contexts). *A context C is* LO *if*

1. Right Application*: whenever $C = C'\langle tC''\rangle$ then t is neutral, and*
2. Left Application*: whenever $C = C'\langle C''t\rangle$ then $C'' \neq \lambda x.C'''$.*

The second characterisation is inductive, and it used to prove Lemma 10.3

Definition 4 (iLO Context). *Inductive LO β (or iLO) contexts are defined by induction as follows:*

$$\frac{}{\langle\cdot\rangle \text{ is iLO}}\ (\textit{ax-iLO}) \qquad \frac{C \text{ is iLO} \qquad C \neq \lambda x.C'}{Ct \text{ is iLO}}\ (\textit{@l-iLO})$$

$$\frac{C \text{ is iLO}}{\lambda x.C \text{ is iLO}}\ (\lambda\textit{-iLO}) \qquad \frac{t \text{ is neutral} \qquad C \text{ is iLO}}{tC \text{ is iLO}}\ (\textit{@r-iLO})$$

As expected,

Lemma 2 ($\to_{\mathtt{LO}\beta}$-steps and Contexts). *Let t be a λ-term and C a redex in t. C is the LO β redex in t iff C is LO iff C is iLO.*

3 Preliminaries on Abstract Machines

We study two abstract machines, the Useful MAM (Fig. 4) and an auxiliary machine called the Checking AM (Fig. 2).

The Useful MAM is meant to implement LO β-reduction strategy via a decoding function $\underline{\cdot}$ mapping machine states to λ-terms. Machine states s are given by a *code* $\overline{t}$, that is a λ-term t *not considered up to α-equivalence* (which is why it is over-lined), and some data-structures like stacks, frames, and environments. The data-structures are used to implement the search for the next LO-redex and a form of micro-steps substitution, and they decode to evaluation contexts for $\rightarrow_{\mathtt{LO}\beta}$. Every state s decodes to a term $\underline{s}$, having the shape $C_s\langle\overline{t}\rangle$, where $\overline{t}$ is the code currently under evaluation and C_s is the evaluation context given by the data-structures.

The Checking AM tests the usefulness of a term (with respect to a given environment) and outputs a label with the result of the test. It uses the same states and data-structures of the Useful MAM.

The Data-Structures. First of all, our machines are executed on *well-named* terms, that are those α-representants where all variables (both bound and free) have distinct names. Then, the data-structures used by the machines are defined in Fig. 1, namely:

- *Stack* π: it contains the arguments of the current code;
- *Frame* F: a second stack, that together with π is used to walk through the term and search for the next redex to reduce. The items ϕ of a frame are of two kinds. A variable x is pushed on the frame F whenever the machines starts evaluating under an abstraction λx. A *head argument context* $\overline{t}\Diamond\pi$ is pushed every time evaluation enters in the right subterm $\overline{u}$ of an application $\overline{t}\overline{u}$. The entry saves the left part $\overline{t}$ of the application and the current stack π, to restore them when the evaluation of the right subterm $\overline{u}$ is over.
- *Global Environment* E: it is used to implement micro-step evaluation (*i.e.* the substitution on a variable occurrence at the time), storing the arguments of β-redexes that have been encountered so far. Most of the literature on abstract machines uses *local environments* and *closures*. Having just one global environment E removes the need for closures and simplifies the machine. On the other hand, it forces to use explicit α-renamings (the operation $\overline{t}^{\alpha}$ in $\rightsquigarrow_{\mathtt{e}_{red}}$ and $\rightsquigarrow_{\mathtt{e}_{abs}}$ in Fig. 4), but this does not affect the overall complexity, as it speeds up other operations, see [1]. The entries of E are of the form $[x{\leftarrow}\overline{t}]^l$, *i.e.* they carry a label l used to implement usefulness, to be explained later on in this section. We write $E(x) = [x{\leftarrow}\overline{t}]^l$ when E contains $[x{\leftarrow}\overline{t}]^l$ and $E(x) = \perp$ when in E there are no entries of the form $[x{\leftarrow}\overline{t}]^l$.

The Decoding. Every state s decodes to a term $\underline{s}$ (see Fig. 3), having the shape $C_s\langle\overline{t}{\downarrow_E}\rangle$, where

- $\overline{t}{\downarrow_E}$ is a λ-term, roughly obtained by applying to the code the substitution induced by the global environment E. More precisely, the operation $\overline{t}{\downarrow_E}$ is called *unfolding* and it is properly defined at the end of this section.
- C_s is a context, that will be shown to be a LO context, obtained by decoding the stack π and the dump F and applying the unfolding. Note that, to improve readability, π is decoded in postfix notation for plugging.

Frames	$F ::= \epsilon \mid F : \phi$	Stacks	$\pi ::= \epsilon \mid \overline{t} : \pi$
Frame Items	$\phi ::= \overline{t}\Diamond\pi \mid x$	Phases	$\varphi ::= \blacktriangledown \mid \blacktriangle$
Labels	$l ::= abs \mid (red, n \in \mathbb{N}) \mid neu$	Environments	$E ::= \epsilon \mid [x{\leftarrow}\overline{t}]^{l} : E$

Fig. 1. Grammars.

Frame	Code	Stack	Env	Ph		Frame	Code	Stack	Env	Ph
F	$\overline{t}\overline{u}$	π	E	$\blacktriangledown$	$\rightharpoonup_{\blacktriangledown c_1}$	F	$\overline{t}$	$\overline{u} : \pi$	E	$\blacktriangledown$
F	$\lambda x.\overline{t}$	$\overline{u} : \pi$	E	$\blacktriangledown$	$\rightharpoonup_{o_1}$	output $(red, 1)$				
F	$\lambda x.\overline{t}$	ϵ	E	$\blacktriangledown$	$\rightharpoonup_{\blacktriangledown c_2}$	$F : x$	$\overline{t}$	ϵ	E	$\blacktriangledown$
F	x	π	E	$\blacktriangledown$	$\rightharpoonup_{o_2}$	output $(red, n+1)$				
						if $E(x) = [x{\leftarrow}\overline{t}]^{(red,n)}$				
F	x	$\overline{u} : \pi$	E	$\blacktriangledown$	$\rightharpoonup_{o_3}$	output $(red, 2)$				
						if $E(x) = [x{\leftarrow}\overline{t}]^{abs}$				
F	x	π	E	$\blacktriangledown$	$\rightharpoonup_{\blacktriangledown c_3}$	F	x	π	E	$\blacktriangle$
		if $E(x) = \bot$ or $E(x) = [x{\leftarrow}\overline{t}]^{neu}$ or $(E(x) = [x{\leftarrow}\overline{t}]^{abs}$ and $\pi = \epsilon)$								
$F : x$	$\overline{t}$	ϵ	E	$\blacktriangle$	$\rightharpoonup_{\blacktriangle c_4}$	F	$\lambda x.\overline{t}$	ϵ	E	$\blacktriangle$
$F : \overline{t}\Diamond\pi$	$\overline{u}$	ϵ	E	$\blacktriangle$	$\rightharpoonup_{\blacktriangle c_5}$	F	$\overline{t}\overline{u}$	π	E	$\blacktriangle$
F	$\overline{t}$	$\overline{u} : \pi$	E	$\blacktriangle$	$\rightharpoonup_{\blacktriangle c_6}$	$F : \overline{t}\Diamond\pi$	$\overline{u}$	ϵ	E	$\blacktriangledown$
ϵ	$\overline{t}\overline{u}$	ϵ	E	$\blacktriangle$	$\rightharpoonup_{o_4}$	output neu				
ϵ	$\lambda x.\overline{t}$	ϵ	E	$\blacktriangle$	$\rightharpoonup_{o_5}$	output abs				

Fig. 2. The Checking Abstract Machine (Checking AM).

$$\begin{aligned} \underline{\epsilon} &:= \langle\cdot\rangle & \underline{C_s} &:= \underline{F}\langle\underline{\pi}\rangle{\downarrow_E} \\ \underline{\overline{u} : \pi} &:= \langle\langle\cdot\rangle\underline{\overline{u}}\rangle\underline{\pi} & \underline{s} &:= \underline{F}\langle\langle\underline{\overline{t}}\rangle\underline{\pi}\rangle{\downarrow_E} = \underline{C_s}\langle\underline{\overline{t}}{\downarrow_E}\rangle \\ \underline{F : \overline{t}\Diamond\pi} &:= \underline{F}\langle\langle\underline{\overline{t}}\langle\cdot\rangle\rangle\underline{\pi}\rangle & & \\ \underline{F : x} &:= \underline{F}\langle\lambda x.\langle\cdot\rangle\rangle & & \text{where } s = (F, \overline{t}, \pi, E) \end{aligned}$$

Fig. 3. Decoding.

The Transitions. According to the distillation approach of [1] we distinguish different kinds of transitions, whose names reflect a proof-theoretical view, as machine transitions can be seen as cut-elimination steps [1,7]:

- *Multiplicatives* $\rightsquigarrow_{\mathtt{m}}$: they fire a β-redex, except that if the argument is not a variable then it is not substituted but added to the environment;
- *Exponentials* $\rightsquigarrow_{\mathtt{e}}$: they perform a clashing-avoiding substitution from the environment on the single variable occurrence represented by the current code. They implement micro-step substitution.
- *Commutatives* $\rightsquigarrow_{\mathtt{c}}$: they locate and expose the next redex according to the LO evaluation strategy, by rearranging the data-structures.

Both exponential and commutative transitions are invisible on the λ-calculus. Garbage collection is here simply ignored, or, more precisely, it is encapsulated at the meta-level, in the decoding function.

Labels for Useful Sharing. A label l for a code in the environment can be of three kinds. Roughly, they are:

- *Neutral*, or $l = \mathit{neu}$: it marks a neutral term, that is always useless as it is β-normal and its substitution cannot create a redex, because it is not an abstraction;
- *Abstraction*, or $l = \mathit{abs}$: it marks an abstraction, that is a term that is at times useful to substitute. If the variable that it is meant to replace is applied, indeed, the substitution of the abstraction creates a β-redex. But if it is not applied, it is useless.
- *Redex*, or $l = \mathit{red}$: it marks a term that contains a β-redex. It is always useful to substitute these terms.

Actually, the explanation we just gave is oversimplified, but it provides a first intuition about labels. In fact in an environment $[x{\leftarrow}\overline{t}]^l : E$ it is not really $\overline{t}$ that has the property mentioned by its label, rather the term $\overline{t}{\downarrow}_E$ obtained by unfolding the rest of the environment on $\overline{t}$. The idea is that $[x{\leftarrow}\overline{t}]^{red}$ states that it is useful to substitute $\overline{t}$ to *later on* obtain a redex inside it (by potential further substitutions on its variables coming from E). The precise meaning of the labels will be given by Definition 6, and the properties they encode will be made explicit by Lemma 11.

A further subtlety is that the label red for redexes is refined as a pair (red, n), where n is the number of substitutions in E that are needed to obtain the LO redex in $\overline{t}{\downarrow}_E$. Our machines never inspect these numbers, they are only used for the complexity analysis of Sect. 5.2.

Grafting and Unfoldings. The unfolding of the environment E on a code $\overline{t}$ is defined as the recursive *capture-allowing* substitution (called *grafting*) of the entries of E on $\overline{t}$.

Definition 5 (Grafting and Environment Unfolding). *The operation of grafting $\overline{t}\{\{x{\leftarrow}\overline{u}\}\}$ is defined by*

$$(\overline{w}\overline{r})\{\{x{\leftarrow}\overline{u}\}\} := \overline{w}\{\{x{\leftarrow}\overline{u}\}\}\overline{r}\{\{x{\leftarrow}\overline{u}\}\} \qquad (\lambda y.\overline{w})\{\{x{\leftarrow}\overline{u}\}\} := \lambda y.\overline{w}\{\{x{\leftarrow}\overline{u}\}\}$$
$$x\{\{x{\leftarrow}\overline{u}\}\} := \overline{u} \qquad y\{\{x{\leftarrow}\overline{u}\}\} := y$$

Given an environment E we define the unfolding of E on a code $\overline{t}$ as follows:

$$\overline{t}{\downarrow}_{\epsilon} := \overline{t} \qquad \overline{t}{\downarrow}_{[x\leftarrow\overline{u}]^l:E} := \overline{t}\{\{x{\leftarrow}\overline{u}\}\}{\downarrow}_E$$

or equivalently as:

$$(\overline{u}\overline{w}){\downarrow}_E := \overline{u}{\downarrow}_E\overline{w}{\downarrow}_E \qquad x{\downarrow}_{[x\leftarrow\overline{u}]^l:E'} := \overline{u}{\downarrow}_{E'}$$
$$(\lambda x.\overline{u}){\downarrow}_E := \lambda x.\overline{u}{\downarrow}_E \qquad x{\downarrow}_{[y\leftarrow\overline{u}]^l:E'} := x{\downarrow}_{E'} \qquad x{\downarrow}_{\epsilon} := x$$

For instance, $(\lambda x.y){\downarrow}_{[y\leftarrow xx]^{neu}} = \lambda x.(xx)$. The unfolding is extended to contexts as expected (*i.e.* recursively propagating the unfolding and setting $\langle\cdot\rangle{\downarrow}_E = E$).

Let us explain the need for grafting. In [2], the Strong MAM is decoded to the LSC, that is a calculus with explicit substitutions, *i.e.* a calculus able to represent the environment of the Strong MAM. Matching the representation of the environment on the Strong MAM and on the LSC does not need grafting but it is, however, a quite technical affair. Useful sharing adds many further complications in establishing such a matching, because useful evaluation computes a shared representation of the normal form and forces some of the explicit substitutions to stay under abstractions. The difficulty is such, in fact, that we found much easier to decode directly to the λ-calculus rather than to the LSC. Such an alternative solution, however, has to push the substitution induced by the environment through abstractions, which is why we use grafting.

Lemma 3 (Properties of Grafting and Unfolding).

1. *If the bound names of t do not appear free in u then $t\{x{\leftarrow}u\} = t\{\{x{\leftarrow}u\}\}$.*
2. *If moreover they do not appear free in E then $t{\downarrow_E}\{x{\leftarrow}u{\downarrow_E}\} = t\{x{\leftarrow}u\}{\downarrow_E}$.*

4 The Checking Abstract Machine

The Checking Abstract Machine (Checking AM) is defined in Fig. 2. It starts executions on states of the form $(\epsilon, \overline{t}, \epsilon, E, \blacktriangledown)$, with the aim of checking the usefulness of $\overline{t}$ with respect to the environment E, *i.e.* it walks through $\overline{t}$ and whenever it encounters a variable x it looks up its usefulness in E.

The Checking AM has six commutative transitions, noted $\rightharpoonup_{\mathtt{c}_i}$ with $i = 1,..,6$, used to walk through the term, and five output transitions, noted $\rightharpoonup_{\mathtt{o}_j}$ with $j = 1,..,5$, that produce the value of the test for usefulness, to be later used by the Useful MAM. The exploration is done in two alternating phases, evaluation $\blacktriangledown$ and backtracking $\blacktriangle$. Evaluation explores the current code towards the head, storing in the stack and in the frame the parts of the code that it leaves behind. Backtracking comes back to an argument that was stored in the frame, when the current head has already been checked. Note that the Checking AM never modifies the environment, it only looks it up.

Let us explain the transitions. First the commutative ones:

- $\rightharpoonup_{\blacktriangledown\mathtt{c}_1}$: the code is an application $\overline{t}\overline{u}$ and the machine starts exploring the left subterm $\overline{t}$, storing $\overline{u}$ on top of the stack π.
- $\rightharpoonup_{\blacktriangledown\mathtt{c}_2}$: the code is an abstraction $\lambda x.\overline{t}$ and the machine goes under the abstraction, storing x on top of the frame F.
- $\rightharpoonup_{\blacktriangledown\mathtt{c}_3}$: the machine finds a variable x that either has no associated entry in the environment (if $E(x) = \bot$) or its associated entry $[x{\leftarrow}\overline{t}]^l$ in the environment is useless. This can happen if either $l = \mathit{neu}$, *i.e.* substituting $\overline{t}$ would only lead to a neutral term, or $l = \mathit{abs}$, *i.e.* substituting $\overline{t}$ would provide an abstraction, but the stack is empty, and so it is useless to substitute the abstraction because no β-redexes will be obtained. Thus the machine switches to the backtracking phase ($\blacktriangle$), whose aim is to undo the frame to obtain a new subterm to explore.

– $\rightharpoonup_{\blacktriangle \mathtt{c}_4}$: it is the inverse of $\rightharpoonup_{\blacktriangledown \mathtt{c}_2}$, it puts back on the code an abstraction that was previously stored in the frame.
– $\rightharpoonup_{\blacktriangle \mathtt{c}_5}$: backtracking from the evaluation of an argument $\overline{u}$, it restores the application $\overline{t}\overline{u}$ and the stack π that were previously stored in the frame.
– $\rightharpoonup_{\blacktriangle \mathtt{c}_6}$: backtracking from the evaluation of the left subterm $\overline{t}$ of an application $\overline{t}\overline{u}$, the machine starts evaluating the right subterm (by switching to the evaluation phase $\blacktriangledown$) with an empty stack ϵ, storing on the frame the pair $\overline{t}\Diamond\pi$ of the left subterm and the previous stack π.

Then the output transitions:

– $\rightharpoonup_{\mathtt{o}_1}$: the machine finds a β-redex, namely $(\lambda x.\overline{t})\overline{u}$ and thus outputs a label saying that it requires only one substitution step (namely substituting the term the machine was executed on) to eventually find a β-redex.
– $\rightharpoonup_{\mathtt{o}_2}$: the machine finds a variable x whose associated entry $[x{\leftarrow}\overline{t}]^{(red,n)}$ in the environment is labeled with (red, n), and so outputs a label saying that it takes $n+1$ substitution steps to eventually find a β-redex (n plus 1 for the term the machine was executed on).
– $\rightharpoonup_{\mathtt{o}_3}$: the machine finds a variable x whose associated entry $[x{\leftarrow}\overline{t}]^{abs}$ in the environment is labeled with abs, so $\overline{t}$ is an abstraction, and the stack is non-empty. Since substituting the abstraction will create a β-redex, the machine outputs a label saying that it takes two substitution steps to obtain a β-redex, one for the term the machine was executed on and one for the abstraction $\overline{t}$.
– $\rightharpoonup_{\mathtt{o}_4}$: the machine went through the whole term, that is an application, and found no redex, nor any redex that can be obtained by substituting from the environment. Thus that term is neutral and so the machine outputs the corresponding label.
– $\rightharpoonup_{\mathtt{o}_5}$: as for the previous transition, except that the term is an abstraction, and so the output is the abs label.

The fact that commutative transitions only walk through the code, without changing anything, is formalised by the following lemma, that is crucial for the proof of correctness of the Checking AM (forthcoming Theorem 1).

Lemma 4 (Commutative Transparency).
Let $s = (F, \overline{u}, \pi, E, \varphi) \rightsquigarrow_{\mathtt{c}_{1,2,3,4,5,6}} (F', \overline{u}', \pi', E, \varphi') = s'$. *Then*

1. Decoding Without Unfolding: $\underline{F}\langle\langle\overline{u}\rangle\underline{\pi}\rangle = \underline{F'}\langle\langle\overline{u}'\rangle\underline{\pi'}\rangle$, *and*
2. Decoding With Unfolding: $\underline{s} = \underline{s'}$.

For the analysis of the properties of the Checking AM we need a notion of well-labeled environment, *i.e.* of environment where the labels are consistent with their intended meaning. It is a technical notion also providing enough information to perform the complexity analysis, later on. Moreover, it includes two structural properties of environments: (1) in $[x{\leftarrow}\overline{t}]^l$ the code $\overline{t}$ cannot be a variable, and (2) there cannot be two entries associated to the same variables.

Definition 6 (Well-Labeled Environments). Well-labeled global environments E *are defined by*

1. Empty*: ϵ is well-labeled;*
2. Inductive*: $[x{\leftarrow}\overline{t}]^l : E'$ is well-labeled if E' is well-labeled, x is fresh with respect to $\overline{t}$ and E', and*
   (a) Abstractions*: if $l = abs$ then $\overline{t}$ and $\overline{t}{\downarrow}_{E'}$ are normal abstractions;*
   (b) Neutral Terms*: if $l = neu$ then $\overline{t}$ is an application and $\overline{t}{\downarrow}_{E'}$ is neutral.*
   (c) Redexes*: if $l = (red, n)$ then $\overline{t}$ is not a variable, $\overline{t}{\downarrow}_{E'}$ contains a β-redex. Moreover, $\overline{t} = C\langle\overline{u}\rangle$ with C a LO context and*
      – *if $n = 1$ then $\overline{u}$ is a β-redex,*
      – *if $n > 1$ then $\overline{u} = x$ and $E' = E'' : [y{\leftarrow}\overline{u}]^l : E'''$ with*
         • *if $n > 2$ then $l = (red, n-1)$*
         • *if $n = 2$ then $l = (red, 1)$ or ($l = abs$ and C is applicative).*

Remark 1. Note that by the definition it immediately follows that if $E = E' : [x{\leftarrow}\overline{t}]^{(red,n)} : E''$ is well-labeled then the length of E'', and thus of E, is at least n. This fact is used in the proof of Theorem 3.1

The study of the Checking AM requires some terminology and two invariants. A state s is *initial* if it is of the form $(\epsilon, \overline{t}, \epsilon, E, \varphi)$ with E well-labeled and it is *reachable* if there are an initial state s' and a Checking AM execution $\rho : s' \rightharpoonup^* s$. Both invariants are used to prove the correctness of the Checking AM: the *normal form invariant* to guarantee that codes labeled with *neu* and *abs* are indeed normal or neutral, while the *decoding invariant* is used for the redex labels.

Lemma 5 (Checking AM Invariants). *Let $s = F \mid \overline{u} \mid \pi \mid E \mid \varphi$ be a Checking AM reachable state and E be a well-labeled environment.*

1. Normal Form*:*
   (a) Backtracking Code*: if $\varphi = \blacktriangle$, then $\overline{u}{\downarrow}_E$ is normal, and if π is non-empty, then $\overline{u}{\downarrow}_E$ is neutral;*
   (b) Frame*: if $F = F' : \overline{w}\Diamond\pi' : F''$, then $\overline{w}{\downarrow}_E$ is neutral.*
2. Decoding*: C_s is a LO context.*

Finally, we can prove the main properties of the Checking AM, *i.e.* that when executed on $\overline{t}$ and E it provides a label l to extend E with a consistent entry for $\overline{t}$ (*i.e.* such that $[x{\leftarrow}\overline{t}]^l : E$ is well-labeled), and that such an execution takes time linear in the size of $\overline{t}$.

Theorem 1 (Checking AM Properties). *Let $\overline{t}$ be a code and E a global environment.*

1. Determinism and Progress*: the Checking AM is deterministic and there always is a transition that applies;*
2. Termination and Complexity*: the execution of the Checking AM on $\overline{t}$ and E always terminates, taking $O(|\overline{t}|)$ steps, moreover*
3. Correctness*: if E is well-labeled, x is fresh with respect to E and $\overline{t}$, and l is the output then $[x{\leftarrow}\overline{t}]^l : E$ is well-labeled.*

5 The Useful Milner Abstract Machine

The Useful MAM is defined in Fig. 4. It is very similar to the Checking AM, in particular it has exactly the same commutative transitions, and the same organisation in evaluating and backtracking phases. The difference with respect to the Useful MAM is that the output transitions are replaced by micro-step computational rules that reduce β-redexes and implement useful substitutions. Let us explain them:

- *Multiplicative Transition* $\rightsquigarrow_{\mathtt{m}_1}$: when the argument of the β-redex $(\lambda x.\overline{t})y$ is a variable y then it is immediately substituted in $\overline{t}$. This happens because (1) such substitution are not costly and (2) because in this way the environment stays compact, see also Remark 2 at the end of the paper.
- *Multiplicative Transition* $\rightsquigarrow_{\mathtt{m}_2}$: if the argument $\overline{u}$ is not a variable then the entry $[x{\leftarrow}\overline{u}]^l$ is added to the environment. The label l is obtained by running the Checking AM on $\overline{u}$ and E.
- *Exponential Transition* $\rightsquigarrow_{\mathtt{e}_{red}}$: the environment entry associated to x is labeled with (red, n) thus it is useful to substitute $\overline{t}$. The idea is that in at most n additional substitution steps (shuffled with commutative steps) a β-redex will be obtained. To avoid variable clashes the substitution α-renames $\overline{t}$.
- *Exponential Transition* $\rightsquigarrow_{\mathtt{e}_{abs}}$: the environment associates an abstraction to x and the stack is non empty, so it is useful to substitute the abstraction (again, α-renaming to avoid variable clashes). Note that if the stack is empty the machine rather backtracks using $\rightsquigarrow_{\blacktriangledown \mathtt{c}_3}$.

Frame	Code	Stack	Env	Ph		Frame	Code	Stack	Env	Ph
F	$\overline{t}\overline{u}$	π	E	$\blacktriangledown$	$\rightsquigarrow_{\blacktriangledown \mathtt{c}_1}$	F	$\overline{t}$	$\overline{u} : \pi$	E	$\blacktriangledown$
F	$\lambda x.\overline{t}$	$y : \pi$	E	$\blacktriangledown$	$\rightsquigarrow_{\mathtt{m}_1}$	F	$\overline{t}\{x{\leftarrow}y\}$	π	E	$\blacktriangledown$
F	$\lambda x.\overline{t}$	$\overline{u} : \pi$	E	$\blacktriangledown$	$\rightsquigarrow_{\mathtt{m}_2}$	F	$\overline{t}$	π	$[x{\leftarrow}\overline{u}]^l : E$	$\blacktriangledown$
					if $\overline{u}$ is not a variable and l is the output of the Checking AM on $\overline{u}$ and E					
F	$\lambda x.\overline{t}$	ϵ	E	$\blacktriangledown$	$\rightsquigarrow_{\blacktriangledown \mathtt{c}_2}$	$F : x$	$\overline{t}$	ϵ	E	$\blacktriangledown$
F	x	π	E	$\blacktriangledown$	$\rightsquigarrow_{\mathtt{e}_{red}}$	F	$\overline{t}^{\alpha}$	π	E	$\blacktriangledown$
									if $E(x) = [x{\leftarrow}\overline{t}]^{(red,n)}$	
F	x	$\overline{u} : \pi$	E	$\blacktriangledown$	$\rightsquigarrow_{\mathtt{e}_{abs}}$	F	$\overline{t}^{\alpha}$	$\overline{u} : \pi$	E	$\blacktriangledown$
									if $E(x) = [x{\leftarrow}\overline{t}]^{abs}$	
F	x	π	E	$\blacktriangledown$	$\rightsquigarrow_{\blacktriangledown \mathtt{c}_3}$	F	x	π	E	$\blacktriangle$
					if $E(x) = \bot$ or $E(x) = [x{\leftarrow}\overline{t}]^{neu}$ or $(E(x) = [x{\leftarrow}\overline{t}]^{abs}$ and $\pi = \epsilon)$					
$F : x$	$\overline{t}$	ϵ	E	$\blacktriangle$	$\rightsquigarrow_{\blacktriangle \mathtt{c}_4}$	F	$\lambda x.\overline{t}$	ϵ	E	$\blacktriangle$
$F : \overline{t}\lozenge\pi$	$\overline{u}$	ϵ	E	$\blacktriangle$	$\rightsquigarrow_{\blacktriangle \mathtt{c}_5}$	F	$\overline{t}\overline{u}$	π	E	$\blacktriangle$
F	$\overline{t}$	$\overline{u} : \pi$	E	$\blacktriangle$	$\rightsquigarrow_{\blacktriangle \mathtt{c}_6}$	$F : \overline{t}\lozenge\pi$	$\overline{u}$	ϵ	E	$\blacktriangledown$

$\overline{t}^{\alpha}$ is any code α-equivalent to $\overline{t}$ such that it is well-named and its bound names are fresh with respect to those in the other machine components.

Fig. 4. The Useful Milner Abstract Machine (Useful MAM).

The Useful MAM starts executions on *initial states* of the form $(\epsilon, \overline{t}, \epsilon, \epsilon)$, where $\overline{t}$ is such that any two variables (bound or free) have distinct names, and any other component is empty. A state s is *reachable* if there are an initial state s' and a Useful MAM execution $\rho : s' \rightsquigarrow^* s$, and it is *final* if no transitions apply.

5.1 Qualitative Analysis

The results of this subsection are the correctness and completeness of the Useful MAM. Four invariants are required. The *normal form* and *decoding invariants* are exactly those of the Checking AM (and the proof for the commutative transitions is the same). The *environment labels invariant* follows from the correctness of the Checking AM (Theorem 1.2. The *name invariant* is used in the proof of Lemma 7.

Lemma 6 (Useful MAM Qualitative Invariants). *Let $s = F \mid \overline{u} \mid \pi \mid E \mid \varphi$ be a state reachable from an initial term $\overline{t}_0$. Then:*

1. Environment Labels*: E is well-labeled.*
2. Normal Form*:*
   (a) Backtracking Code*: if $\varphi = \blacktriangle$, then $\overline{u}\downarrow_E$ is normal, and if π is non-empty, then $\overline{u}\downarrow_E$ is neutral;*
   (b) Frame*: if $F = F' : \overline{w}\Diamond\pi' : F''$, then $\overline{w}\downarrow_E$ is neutral.*
3. Name:
   (a) Substitutions*: if $E = E' : [x{\leftarrow}\overline{t}] : E''$ then x is fresh wrt $\overline{t}$ and E'';*
   (b) Abstractions and Evaluation*: if $\varphi = \blacktriangledown$ and $\lambda x.\overline{t}$ is a subterm of $\overline{u}$, π, or π' (if $F = F' : \overline{w}\Diamond\pi' : F''$) then x may occur only in $\overline{t}$;*
   (c) Abstractions and Backtracking*: if $\varphi = \blacktriangle$ and $\lambda x.\overline{t}$ is a subterm of π or π' (if $F = F' : \overline{w}\Diamond\pi' : F''$) then x may occur only in $\overline{t}$.*
4. Decoding*: C_s is a LO context.*

We can now show how every single transition projects on the λ-calculus, and in particular that multiplicative transitions project to LO β-steps.

Lemma 7 (One-Step Weak Simulation, Proof at Page 17). *Let s be a reachable state.*

1. Commutative*: if $s \rightsquigarrow_{\mathtt{c}_{1,2,3,4,5,6}} s'$ then $\underline{s} = \underline{s'}$;*
2. Exponential*: if $s \rightsquigarrow_{\mathtt{e}_{red},\mathtt{e}_{abs}} s'$ then $\underline{s} = \underline{s'}$;*
3. Multiplicative*: if $s \rightsquigarrow_{\mathtt{m}_1,\mathtt{m}_2} s'$ then $\underline{s} \rightarrow_{\mathtt{LO}\beta} \underline{s'}$.*

We also need to show that the Useful MAM computes β-normal forms.

Lemma 8 (Progress, Proof at Page 18). *Let s be a reachable final state. Then $\underline{s}$ is β-normal.*

The theorem of correctness and completeness of the machine with respect to $\rightarrow_{\mathtt{LO}\beta}$ follows. The bisimulation is *weak* because transitions other than $\rightsquigarrow_{\mathtt{m}}$ are invisible on the λ-calculus. For a machine execution ρ we denote with $|\rho|$ (resp. $|\rho|_{\mathtt{x}}$) the number of transitions (resp. $\mathtt{x}$-transitions for $\mathtt{x} \in \{\mathtt{m}, \mathtt{e}, \mathtt{c}, \ldots\}$) in ρ.

Theorem 2 (Weak Bisimulation, Proof at Page 18). *Let s be an initial Useful MAM state of code $\bar{t}$.*

1. Simulation: *for every execution $\rho : s \rightsquigarrow^* s'$ there exists a derivation $d\colon \underline{s} \rightarrow^*_{\mathtt{LO}\beta} \underline{s'}$ such that $|d| = |\rho|_{\mathtt{m}}$;*
2. Reverse Simulation: *for every derivation $d\colon \bar{t} \rightarrow^*_{\mathtt{LO}\beta} u$ there is an execution $\rho : s \rightsquigarrow^* s'$ such that $\underline{s'} = u$ and $|d| = |\rho|_{\mathtt{m}}$.*

5.2 Quantitative Analysis

The complexity analyses of this section rely on two additional invariants of the Useful MAM, the subterm and the environment size invariants.

The subterm invariant bounds the size of the duplicated subterms and it is crucial. For us, $\overline{u}$ is a subterm of $\bar{t}$ if it does so up to variable names, both free and bound. More precisely: define t^- as t in which all variables (including those appearing in binders) are replaced by a fixed symbol $*$. Then, we will consider u to be a subterm of t whenever u^- is a subterm of t^- in the usual sense. The key property ensured by this definition is that the size $|\overline{u}|$ of $\overline{u}$ is bounded by $|\bar{t}|$.

Lemma 9 (Useful MAM Quantitative Invariants). *Let $s = F \mid \overline{u} \mid \pi \mid E \mid \varphi$ be a state reachable by the execution ρ from the initial code $\bar{t}_0$.*

1. Subterm*: environment, which is a subterm of the initial term by*
   (a) Evaluating Code*: if $\varphi = \blacktriangledown$, then $\overline{u}$ is a subterm of $\bar{t}_0$;*
   (b) Stack*: any code in the stack π is a subterm of $\bar{t}_0$;*
   (c) Frame*: if $F = F' : \overline{w}\Diamond\pi' : F''$, then any code in π' is a subterm of $\bar{t}_0$;*
   (d) Global Environment*: if $E = E' : [x{\leftarrow}\overline{w}]^l : E''$, then $\overline{w}$ is a subterm of $\bar{t}_0$;*
2. Environment Size*: the length of the global environment E is bound by $|\rho|_{\mathtt{m}}$.*

The proof of the polynomial bound of the overhead is in three steps. First, we bound the number $|\rho|_{\mathtt{e}}$ of exponential transitions of an execution ρ using the number $|\rho|_{\mathtt{m}}$ of multiplicative transitions of ρ, that by Theorem 2 corresponds to the number of LO β-steps on the λ-calculus. Second, we bound the number $|\rho|_c$ of commutative transitions of ρ by using the number of exponential transitions and the size of the initial term. Third, we put everything together.

Multiplicative vs Exponential Analysis. This step requires two auxiliary lemmas. The first one essentially states that commutative transitions *eat* normal and neutral terms, as well as LO contexts.

Lemma 10. *Let $s = F \mid \bar{t} \mid \pi \mid E \mid \blacktriangledown$ be a state and E be well-labeled. Then*

1. *If $\bar{t}{\downarrow_E}$ is a normal term and $\pi = \epsilon$ then $s \rightsquigarrow^*_{\mathtt{c}} F \mid \bar{t} \mid \pi \mid E \mid \blacktriangle$.*
2. *If $\bar{t}{\downarrow_E}$ is a neutral term then $s \rightsquigarrow^*_{\mathtt{c}} F \mid \bar{t} \mid \pi \mid E \mid \blacktriangle$.*
3. *If $\bar{t} = C\langle\overline{u}\rangle$ with $C{\downarrow_E}$ a LO context then there exist F' and π' such that $s \rightsquigarrow^*_{\mathtt{c}} F' \mid \overline{u} \mid \pi' \mid E \mid \blacktriangledown$;*

The second lemma uses Lemma 10 and the environment labels invariant (Lemma 6.1 to show that the exponential transitions of the Useful MAM are indeed useful, as they head towards a multiplicative transition, that is towards β-redexes.

Lemma 11 (Useful Exponentials Lead to Multiplicatives). *Let s be a reachable state such that $s \leadsto_{\mathtt{e}_{(red,n)}} s'$.*

1. *If $n = 1$ then $s' \leadsto_{\mathtt{c}}^{*} \leadsto_{\mathtt{m}} s''$;*
2. *If $n = 2$ then $s' \leadsto_{\mathtt{c}}^{*} \leadsto_{\mathtt{e}_{abs}} \leadsto_{\mathtt{m}} s''$ or $s' \leadsto_{\mathtt{c}}^{*} \leadsto_{\mathtt{e}_{(red,1)}} s''$;*
3. *If $n > 1$ then $s' \leadsto_{\mathtt{c}}^{*} \leadsto_{\mathtt{e}_{(red,n-1)}} s''$.*

Finally, using the environment size invariant (Lemma 9.2) we obtain the local boundedness property, that is used to infer a quadratic bound via a standard reasoning (already employed in [6]).

Theorem 3 (Exponentials vs Multiplicatives, Proof at Page 19). *Let s be an initial Useful MAM state and $\rho : s \leadsto^{*} s'$.*

1. Local Boundedness*: if $\sigma : s' \leadsto^{*} s''$ and $|\sigma|_{\mathtt{m}} = 0$ then $|\sigma|_{\mathtt{e}} \leq |\rho|_{\mathtt{m}}$;*
2. Exponentials are Quadratic in the Multiplicatives*: $|\rho|_{\mathtt{e}} \in O(|\rho|_{\mathtt{m}}^{2})$.*

Commutative vs Exponential Analysis. The second step is to bound the number of commutative transitions. Since the commutative part of the Useful MAM is essentially the same as the commutative part of the Strong MAM of [2], the proof of such bound is essentially the same as in [2]. It relies on the subterm invariant (Lemma 9.1).

Theorem 4 (Commutatives vs Exponentials, Proof at Page 20). *Let $\rho : s \leadsto^{*} s'$ be a Useful MAM execution from an initial state of code t. Then:*

1. Commutative Evaluation Steps are Bilinear*: $|\rho|_{\blacktriangledown\mathtt{c}} \leq (1 + |\rho|_{\mathtt{e}}) \cdot |t|$.*
2. Commutative Evaluation Bounds Backtracking*: $|\rho|_{\blacktriangle\mathtt{c}} \leq 2 \cdot |\rho|_{\blacktriangledown\mathtt{c}}$.*
3. Commutative Transitions are Bilinear*: $|\rho|_{c} \leq 3 \cdot (1 + |\rho|_{\mathtt{e}}) \cdot |t|$.*

The Main Theorem. Putting together the matching between LO β-steps and multiplicative transitions (Theorem 2), the quadratic bound on the exponentials via the multiplicatives (Theorem 3.2) and the bilinear bound on the commutatives (Theorem 4.3) we obtain that the number of the Useful MAM transitions to implement a LO β-derivation d is at most quadratic in the length of d and linear in the size of t. Moreover, the subterm invariant (Lemma 9.1) and the analysis of the Checking AM (Theorem 1.2) allow to bound the cost of implementing the execution on RAM.

Theorem 5 (Useful MAM Overhead Bound, Proof at Page 20). *Let $d : t \to_{\mathtt{LO}\beta}^{*} u$ be a leftmost-outermost derivation and ρ be the Useful MAM execution simulating d given by Theorem 2.2. Then:*

1. Length*:* $|\rho| = O((1+|d|^2)\cdot|t|)$*.*
2. Cost*: ρ is implementable on RAM in* $O((1+|d|^2)\cdot|t|)$ *steps.*

Remark 2. Our bound is quadratic in the number of the LO β-steps but we believe that it is not tight. In fact, our transition $\rightsquigarrow_{\mathtt{m}_1}$ is a standard optimisation, used for instance in Wand's [28] (Sect. 2), Friedman et al.'s [21] (Sect. 4), and Sestoft's [26] (Sect. 4), and motivated as an optimization about *space*. In Sands, Gustavsson, and Moran's [25], however, it is shown that it lowers the overhead for *time* from quadratic to linear (with respect to the number of β-steps) for call-by-name evaluation in a weak setting. Unfortunately, the simple proof used in [25] does not scale up to our setting, nor we have an alternative proof that the overhead is linear. We conjecture, however, that it does.

Proofs of the Main Lemmas and Theorems

Proof of One-Step Weak Bisimulation Lemma (Lemma 7, p. 13)

1. *Commutative*: the proof is exactly as the one for the Checking AM (Lemma 4.2), that can be found in the longer version of this paper on the author's webpage.
2. *Exponential*:

   – **Case**$s = (F, x, \pi, E, \blacktriangledown) \rightsquigarrow_{\mathtt{e}_{red}} (F, \bar{t}^{\alpha}, \pi, E, \blacktriangledown) = s'$ **with** $E(x) = [x{\leftarrow}\bar{t}]^{(red,n)}$ Then $E = E' : [x{\leftarrow}\bar{t}]^{(red,n)} : E''$ for some environments E', and E''. Remember that terms are considered up to α-equivalence.

   $$\underline{s} = C_{s'}\langle x{\downarrow_E}\rangle = C_{s'}\langle \bar{t}{\downarrow_{E''}}\rangle = C_{s'}\langle \bar{t}{\downarrow_E}\rangle = \underline{s'}$$

   In the chain of equalities we can replace $\bar{t}{\downarrow_{E''}}$ with $\bar{t}{\downarrow_E}$ because by well-labeledness the variables bound by E' are fresh with respect to $\bar{t}$.
   – **Case** $s = (F, x, \bar{u} : \pi, E, \blacktriangledown) \rightsquigarrow_{\mathtt{e}_{abs}} (F, \bar{t}^{\alpha}, \bar{u} : \pi, E, \blacktriangledown) = s'$ **with** $E(x) = [x{\leftarrow}\bar{t}]^{abs}$ The proof that $\underline{s} = \underline{s'}$ is exactly as in the previous case.
3. *Multiplicative*:

   – **Case** $s = (F, \lambda x.\bar{t}, y : \pi, E, \blacktriangledown) \rightsquigarrow_{\mathtt{m}_1} (F, \bar{t}\{x{\leftarrow}y\}, \pi, E, \blacktriangledown) = s'$ Note that $\underline{C_s} = \underline{F}\langle\underline{\pi}\rangle{\downarrow_E}$ is LO by the decoding invariant (Lemma 6.4). Note also that by the name invariant (Lemma 6.3b) x can only occur in $\bar{t}$. Then:

   $$\begin{array}{lll}
   \underline{(F, \lambda x.\bar{t}, y : \pi, E, \blacktriangledown)} & = & \underline{F}\langle\langle\lambda x.\bar{t}\rangle\underline{y : \pi}\rangle{\downarrow_E} \\
   & = & \underline{F}\langle\langle(\lambda x.\bar{t})y\rangle\underline{\pi}\rangle{\downarrow_E} \\
   & = & C_{s'}\langle(\lambda x.\bar{t}{\downarrow_E})y{\downarrow_E}\rangle \\
   & \rightarrow_{\mathtt{LO}\beta} & C_{s'}\langle\bar{t}{\downarrow_E}\{x{\leftarrow}y{\downarrow_E}\}\rangle \\
   & =_{L.\,6.3b\&L.\,3.2} & C_{s'}\langle\bar{t}\{x{\leftarrow}y\}{\downarrow_E}\rangle \\
   & = & \underline{(F, \bar{t}\{x{\leftarrow}y\}, \pi, E, \blacktriangledown)}
   \end{array}$$

   – **Case** $s = (F, \lambda x.\bar{t}, \bar{u} : \pi, E, \blacktriangledown) \rightsquigarrow_{\mathtt{m}_2} (F, \bar{t}, \pi, [x{\leftarrow}\bar{u}]^l : E, \blacktriangledown) = s'$ **with** $\bar{u}$ **not a variable.** Note that $\underline{C_{s'}} = \underline{F}\langle\langle\cdot\rangle\underline{\pi}\rangle{\downarrow_E} = \underline{F{\downarrow_E}}\langle\langle\cdot\rangle\underline{\pi{\downarrow_E}}\rangle$ is LO by the decoding invariant (Lemma 6.4). Note also that by the name invariant (Lemma 6.3b) x can only occur in $\bar{t}$. Then:

$$\begin{array}{lll}
\underline{(F, \lambda x.\bar{t}, \bar{u} : \pi, E, \blacktriangledown)} & = & \underline{F}\langle\langle\lambda x.\bar{t}\rangle\underline{\bar{u} : \pi}\rangle\downarrow_E \\
& = & \underline{F}\langle\langle(\lambda x.\bar{t})\bar{u}\rangle\underline{\pi}\rangle\downarrow_E \\
& = & \underline{F\downarrow_E}\langle\langle(\lambda x.\bar{t}\downarrow_E)\bar{u}\downarrow_E\rangle\underline{\pi\downarrow_E}\rangle \\
& \to_{\mathtt{LO}\beta} & \underline{F\downarrow_E}\langle\langle\bar{t}\downarrow_E\{x\leftarrow\bar{u}\downarrow_E\}\rangle\underline{\pi\downarrow_E}\rangle \\
& =_{L.\,6.3b\&L.\,3.2} & \underline{F\downarrow_E}\langle\langle\bar{t}\{x\leftarrow\bar{u}\}\downarrow_E\rangle\underline{\pi\downarrow_E}\rangle \\
& = & \underline{F}\langle\langle\bar{t}\{x\leftarrow\bar{u}\}\rangle\underline{\pi}\rangle\downarrow_E \\
& =_{L.\,6.3b\&L.\,3.1} & \underline{F}\langle\langle\bar{t}\{\{x\leftarrow\bar{u}\}\}\rangle\underline{\pi}\rangle\downarrow_E \\
& =_{L.\,6.3b} & \underline{F}\langle\langle\bar{t}\rangle\underline{\pi}\rangle\{\{x\leftarrow\bar{u}\}\}\downarrow_E \\
& = & \underline{F}\langle\langle\bar{t}\rangle\underline{\pi}\rangle\downarrow_{[x\leftarrow\bar{u}]^l : E} \\
& = & \underline{(F, \bar{t}, \pi, [x\leftarrow\bar{u}]^l : E, \blacktriangledown)}
\end{array}$$ □

Proof of the Progress Lemma (Lemma 8, p. 13)

A simple inspection of the machine transitions shows that final states have the form $(\epsilon, \bar{t}, \epsilon, E, \blacktriangle)$. Then by the normal form invariant (Lemma 6.2a) $\underline{s} = \bar{t}\downarrow_E$ is β-normal. □

Proof of the Weak Bisimulation Theorem (Theorem 2, p. 13)

1. By induction on the length $|\rho|$ of ρ, using the one-step weak simulation lemma (Lemma 7). If ρ is empty then the empty derivation satisfies the statement. If ρ is given by $\sigma : s \rightsquigarrow^* s''$ followed by $s'' \rightsquigarrow s'$ then by *i.h.* there exists $e : \underline{s} \to^*_{\mathtt{LO}\beta} \underline{s''}$ s.t. $|e| = |\sigma|_{\mathtt{m}}$. Cases of $s'' \rightsquigarrow s'$:
   (a) *Commutative or Exponential*. Then $\underline{s''} = \underline{s'}$ by Lemmas 7.1 and 7.2, and the statement holds taking $d := e$ because $|d| = |e| =_{i.h.} |\sigma|_{\mathtt{m}} = |\rho|_{\mathtt{m}}$.
   (b) *Multiplicative*. Then $\underline{s''} \to_{\mathtt{LO}\beta} \underline{s'}$ by Lemma 7.3 and defining d as e followed by such a step we obtain $|d| = |e| + 1 =_{i.h.} |\sigma|_{\mathtt{m}} + 1 = |\rho|_{\mathtt{m}}$.
2. We use $\mathtt{nf}_{\mathtt{ec}}(s)$ to denote the normal form of s with respect to exponential and commutative transitions, that exists and is unique because $\rightsquigarrow_{\mathtt{c}} \cup \rightsquigarrow_{\mathtt{e}}$ terminates (termination is given by forthcoming Theorems 3 and 4, that are postponed because they actually give precise complexity bounds, not just termination) and the machine is deterministic (as it can be seen by an easy inspection of the transitions). The proof is by induction on the length of d. If d is empty then the empty execution satisfies the statement.
   If d is given by $e : \bar{t} \to^*_{\mathtt{LO}\beta} w$ followed by $w \to_{\mathtt{LO}\beta} u$ then by *i.h.* there is an execution $\sigma : s \rightsquigarrow^* s''$ s.t. $w = \underline{s''}$ and $|\sigma|_{\mathtt{m}} = |e|$. Note that since exponential and commutative transitions are mapped on equalities, σ can be extended as $\sigma' : s \rightsquigarrow^* s'' \rightsquigarrow^*_{\mathtt{e}_{red}, \mathtt{e}_{abs}, \mathtt{c}_{1,2,3,4,5,6}} \mathtt{nf}_{\mathtt{ec}}(s'')$ with $\underline{\mathtt{nf}_{\mathtt{ec}}(s'')} = w$ and $|\sigma'|_{\mathtt{m}} = |e|$. By the progress property (Lemma 8) $\mathtt{nf}_{\mathtt{ec}}(s'')$ cannot be a final state, otherwise $w = \underline{\mathtt{nf}_{\mathtt{ec}}(s'')}$ could not reduce. Then $\mathtt{nf}_{\mathtt{ec}}(s'') \rightsquigarrow_{\mathtt{m}} s'$ (the transition is necessarily multiplicative because $\mathtt{nf}_{\mathtt{ec}}(s'')$ is normal with respect to the other transitions). By the one-step weak simulation lemma

(Lemma 7.3) $\underline{\mathtt{nf}_{\mathtt{ec}}(s'')} = w \rightarrow_{\mathtt{LO}\beta} \underline{s'}$ and by determinism of $\rightarrow_{\mathtt{LO}\beta}$ (Lemma 1) $\underline{s'} = u$. Then the execution ρ defined as σ' followed by $\mathtt{nf}_{\mathtt{ec}}(s'') \rightsquigarrow_{\mathtt{m}} s'$ satisfy the statement, as $|\rho|_{\mathtt{m}} = |\sigma'|_{\mathtt{m}} + 1 = |\sigma|_{\mathtt{m}} + 1 = |e| + 1 = |d|$. □

Proof of the Exponentials vs Multiplicatives Theorem (Theorem 3, p. 15)

1. We prove that $|\sigma|_{\mathtt{e}} \leq |E|$. The statement follows from the environment size invariant (Lemma 9.2), for which $|E| \leq |\rho|_{\mathtt{m}}$.

   If $|\sigma|_{\mathtt{e}} = 0$ it is immediate. Then assume $|\sigma|_{\mathtt{e}} > 0$, so that there is a first exponential transition in σ, *i.e.* σ has a prefix $s' \rightsquigarrow_{\mathtt{c}}^{*}\rightsquigarrow_{\mathtt{e}} s'''$ followed by an execution $\tau : s''' \rightsquigarrow^{*} s''$ such that $|\tau|_{\mathtt{m}} = 0$. Cases of the first exponential transition $\rightsquigarrow_{\mathtt{e}}$:
   - Case $\rightsquigarrow_{\mathtt{e}_{abs}}$: the next transition is necessarily multiplicative, and so τ is empty. Then $|\sigma|_{\mathtt{e}} = 1$. Since the environment is non-empty (otherwise $\rightsquigarrow_{\mathtt{e}_{abs}}$ could not apply), $|\sigma|_{\mathtt{e}} \leq |E|$ holds.
   - Case $\rightsquigarrow_{\mathtt{e}_{(red,n)}}$: we prove by induction on n that $|\sigma|_{\mathtt{e}} \leq n$, that gives what we want because $n \leq |E|$ by Remark 1. Cases:
     - $n = 1$) Then τ has the form $s''' \rightsquigarrow_{\mathtt{c}}^{*} s''$ by Lemma 11.1, and so $|\sigma|_{\mathtt{e}} = 1$.
     - $n = 2$) Then τ is a prefix of $\rightsquigarrow_{\mathtt{c}}^{*}\rightsquigarrow_{\mathtt{e}_{abs}}$ or $\rightsquigarrow_{\mathtt{c}}^{*}\rightsquigarrow_{\mathtt{e}_{(red,1)}}$ by Lemma 11.2. In both cases $|\sigma|_{\mathtt{e}} \leq 2$.
     - $n > 2$) Then by Lemma 11.3 τ is either shorter or equal to $\rightsquigarrow_{\mathtt{c}}^{*}\rightsquigarrow_{\mathtt{e}_{(red,n-1)}}$, and so $|\sigma|_{\mathtt{e}} \leq 2$, or it is longer than $\rightsquigarrow_{\mathtt{c}}^{*}\rightsquigarrow_{\mathtt{e}_{(red,n-1)}}$, *i.e.* it writes as $\rightsquigarrow_{\mathtt{c}}^{*}$ followed by an execution τ' starting with $\rightsquigarrow_{\mathtt{e}_{(red,n-1)}}$. By *i.h.* $|\tau'| \leq n-1$ and so $|\sigma| \leq n$.
2. This is a standard reasoning: since by local boundedness (the previous point) $\mathtt{m}$-free sequences have a number of $\mathtt{e}$-transitions that are bound by the number of preceding $\mathtt{m}$-transitions, the sum of all $\mathtt{e}$-transitions is bound by the square of $\mathtt{m}$-transitions. It is analogous to the proof of Theorem 7.2.3 in [6]. □

Proof of Commutatives vs Exponentials Theorem (Theorem 4, p. 15)

1. We prove a slightly stronger statement, namely $|\rho|_{\blacktriangledown\mathtt{c}} + |\rho|_{\mathtt{m}} \leq (1 + |\rho|_{\mathtt{e}}) \cdot |t|$, by means of the following notion of size for stacks/frames/states:

$$\begin{array}{rclcrcl} |\epsilon| &:=& 0 & & |x : F| &:=& |F| \\ |\bar{t} : \pi| &:=& |\bar{t}| + |\pi| & & |\bar{t}\Diamond\pi : F| &:=& |\pi| + |F| \\ |(F, \bar{t}, \pi, E, \blacktriangledown)| &:=& |F| + |\pi| + |\bar{t}| & & |(F, \bar{t}, \pi, E, \blacktriangle)| &:=& |F| + |\pi| \end{array}$$

   By direct inspection of the rules of the machine it can be checked that:
   - *Exponentials Increase the Size*: if $s \rightsquigarrow_{\mathtt{e}} s'$ is an exponential transition, then $|s'| \leq |s| + |t|$ where $|t|$ is the size of the initial term; this is a consequence of the fact that exponential steps retrieve a piece of code from the environment, which is a subterm of the initial term by Lemma 9.1;

– *Non-Exponential Evaluation Transitions Decrease the Size*: if $s \leadsto_a s'$ with $a \in \{\mathtt{m}_1, \mathtt{m}_2, \blacktriangledown\mathtt{c}_1, \blacktriangledown\mathtt{c}_2, \blacktriangledown\mathtt{c}_3\}$ then $|s'| < |s|$ (for $\blacktriangledown\mathtt{c}_3$ because the transition switches to backtracking, and thus the size of the code is no longer taken into account);
– *Backtracking Transitions do not Change the Size*: if $s \leadsto_a s'$ with $a \in \{\blacktriangle\mathtt{c}_4, \blacktriangle\mathtt{c}_5, \blacktriangle\mathtt{c}_6\}$ then $|s'| = |s|$.

Then a straightforward induction on $|\rho|$ shows that

$$|s'| \leq |s| + |\rho|_{\mathtt{e}} \cdot |t| - |\rho|_{\blacktriangledown\mathtt{c}} - |\rho|_{\mathtt{m}}$$

i.e. that $|\rho|_{\blacktriangledown\mathtt{c}} + |\rho|_{\mathtt{m}} \leq |s| + |\rho|_{\mathtt{e}} \cdot |t| - |s'|$.
Now note that $|\cdot|$ is always non-negative and that since s is initial we have $|s| = |t|$. We can then conclude with

$$\begin{aligned} |\rho|_{\blacktriangledown\mathtt{c}} + |\rho|_{\mathtt{m}} &\leq |s| + |\rho|_{\mathtt{e}} \cdot |t| - |s'| \\ &\leq |s| + |\rho|_{\mathtt{e}} \cdot |t| \qquad = |t| + |\rho|_{\mathtt{e}} \cdot |t| = (1 + |\rho|_{\mathtt{e}}) \cdot |t| \end{aligned}$$

2. We have to estimate $|\rho|_{\blacktriangle\mathtt{c}} = |\rho|_{\blacktriangle\mathtt{c}_4} + |\rho|_{\blacktriangle\mathtt{c}_5} + |\rho|_{\blacktriangle\mathtt{c}_6}$. Note that
   (a) $|\rho|_{\blacktriangle\mathtt{c}_4} \leq |\rho|_{\blacktriangledown\mathtt{c}_2}$, as $\leadsto_{\blacktriangle\mathtt{c}_4}$ pops variables from F, pushed only by $\leadsto_{\blacktriangledown\mathtt{c}_2}$;
   (b) $|\rho|_{\blacktriangle\mathtt{c}_5} \leq |\rho|_{\blacktriangle\mathtt{c}_6}$, as $\leadsto_{\blacktriangle\mathtt{c}_5}$ pops pairs $\bar{t}\Diamond\pi$ from F, pushed only by $\leadsto_{\blacktriangle\mathtt{c}_6}$;
   (c) $|\rho|_{\blacktriangle\mathtt{c}_6} \leq |\rho|_{\blacktriangledown\mathtt{c}_3}$, as $\leadsto_{\blacktriangle\mathtt{c}_6}$ ends backtracking phases, started only by $\leadsto_{\blacktriangledown\mathtt{c}_3}$.
   Then $|\rho|_{\blacktriangle\mathtt{c}} \leq |\rho|_{\blacktriangledown\mathtt{c}_2} + 2|\rho|_{\blacktriangledown\mathtt{c}_3} \leq 2|\rho|_{\blacktriangledown\mathtt{c}}$.
3. We have $|\rho|_c = |\rho|_{\blacktriangledown\mathtt{c}} + |\rho|_{\blacktriangle\mathtt{c}} \leq_{P.2} |\rho|_{\blacktriangledown\mathtt{c}} + 2|\rho|_{\blacktriangledown\mathtt{c}} \leq_{P.1} 3 \cdot (1 + |\rho|_{\mathtt{e}}) \cdot |t|$. □

Proof of the Useful MAM Overhead Bound Theorem (Theorem 5, p. 15)

1. By definition, the length of the execution ρ simulating d is given by $|\rho| = |\rho|_{\mathtt{m}} + |\rho|_{\mathtt{e}} + |\rho|_c$. Now, by Theorem 3.2 we have $|\rho|_{\mathtt{e}} = O(|\rho|_{\mathtt{m}}^2)$ and by Theorem 4.3 we have $|\rho|_c = O((1 + |\rho|_{\mathtt{e}}) \cdot |t|) = O((1 + |\rho|_{\mathtt{m}}^2) \cdot |t|)$. Therefore, $|\rho| = O((1 + |\rho|_{\mathtt{e}}) \cdot |t|) = O((1 + |\rho|_{\mathtt{m}}^2) \cdot |t|)$. By Theorem 2.2 $|\rho|_{\mathtt{m}} = |d|$, and so $|\rho| = O((1 + |d|^2) \cdot |t|)$.
2. The cost of implementing ρ is the sum of the costs of implementing the multiplicative, exponential, and commutative transitions. Remember that the idea is that variables are implemented as references, so that environment can be accessed in constant time (*i.e.* they do not need to be accessed sequentially):
   (a) *Commutative*: every commutative transition evidently takes constant time. At the previous point we bounded their number with $O((1+|d|^2) \cdot |t|)$, which is then also the cost of all the commutative transitions together.
   (b) *Multiplicative*: a $\leadsto_{\mathtt{m}_1}$ transition costs $O(|t|)$ because it requires to rename the current code, whose size is bound by the size of the initial term by the subterm invariant (Lemma 9.1a). A $\leadsto_{\mathtt{m}_2}$ transition also costs $O(|t|)$ because executing the Checking AM on $\overline{u}$ takes $O(|\overline{u}|)$ commutative steps (Theorem 1.2), commutative steps take constant time, and the size of $\overline{u}$ is bound by $|t|$ by the subterm invariant (Lemma 9.1b). Therefore, all together the multiplicative transitions cost $O(|d| \cdot |t|)$.

(c) *Exponential*: At the previous point we bounded their number with $|\rho|_{\mathtt{e}} = O(|d|^2)$. Each exponential step copies a term from the environment, that by the subterm invariant (Lemma 9.1d) costs at most $O(|t|)$, and so their full cost is $O((1 + |d|) \cdot |t|^2)$ (note that this is exactly the cost of the commutative transitions, but it is obtained in a different way).

Then implementing ρ on RAM takes $O((1 + |d|) \cdot |t|^2)$ steps. $\square$

References

1. Accattoli, B., Barenbaum, P., Mazza, D.: Distilling abstract machines. In: ICFP 2014, pp. 363–376 (2014)
2. Accattoli, B., Barenbaum, P., Mazza, D.: A strong distillery. In: Feng, X., Park, S. (eds.) APLAS 2015. LNCS, vol. 9458, pp. 231–250. Springer, Heidelberg (2015). doi:10.1007/978-3-319-26529-2_13
3. Accattoli, B., Bonelli, E., Kesner, D., Lombardi, C.: A nonstandard standardization theorem. In: POPL, pp. 659–670 (2014)
4. Accattoli, B., Coen, C.S.: On the relative usefulness of fireballs. In: LICS 2015, pp. 141–155 (2015)
5. Accattoli, B., Dal Lago, U.: On the invariance of the unitary cost model for head reduction. In: RTA, pp. 22–37 (2012)
6. Accattoli, B., Lago, U.D.: (Leftmost-Outermost) Beta reduction is invariant, indeed. Logical Methods Comput. Sci. **12**(1), 1–46 (2016)
7. Ariola, Z.M., Bohannon, A., Sabry, A.: Sequent calculi and abstract machines. ACM Trans. Program. Lang. Syst. **31**(4), 13:1–13:48 (2009)
8. Asperti, A., Mairson, H.G.: Parallel beta reduction is not elementary recursive. In: POPL, pp. 303–315 (1998)
9. Blelloch, G.E., Greiner, J.: Parallelism in sequential functional languages. In: FPCA, pp. 226–237 (1995)
10. Boutiller, P.: De nouveaus outils pour manipuler les inductif en Coq. Ph.D. thesis, Université Paris Diderot, Paris 7 (2014)
11. de Carvalho, D.: Execution time of lambda-terms via denotational semantics and intersection types (2009). CoRR abs/0905.4251
12. Crégut, P.: An abstract machine for lambda-terms normalization. In: LISP and Functional Programming, pp. 333–340 (1990)
13. Crégut, P.: Strongly reducing variants of the Krivine abstract machine. Higher Order Symbol. Comput. **20**(3), 209–230 (2007)
14. Dal Lago, U., Martini, S.: The weak lambda calculus as a reasonable machine. Theoret. Comput. Sci. **398**(1–3), 32–50 (2008)
15. Danos, V., Regnier, L.: Head linear reduction. Technical report (2004)
16. Danvy, O., Nielsen, L.R.: Refocusing in reduction semantics. Technical report RS-04-26, BRICS (2004)
17. Danvy, O., Zerny, I.: A synthetic operational account of call-by-need evaluation. In: PPDP, pp. 97–108 (2013)
18. Dénès, M.: Étude formelle d'algorithmes efficaces en algèbre linéaire. Ph.D. thesis, Université de Nice - Sophia Antipolis (2013)
19. Ehrhard, T., Regnier, L.: Böhm trees, Krivine's machine and the Taylor expansion of lambda-terms. In: Beckmann, A., Berger, U., Löwe, B., Tucker, J.V. (eds.) CiE 2006. LNCS, vol. 3988, pp. 186–197. Springer, Heidelberg (2006)

20. Fernández, M., Siafakas, N.: New developments in environment machines. Electron. Notes Theoret. Comput. Sci. **237**, 57–73 (2009)
21. Friedman, D.P., Ghuloum, A., Siek, J.G., Winebarger, O.L.: Improving the lazy Krivine machine. Higher Order Symbol. Comput. **20**(3), 271–293 (2007)
22. García-Pérez, Á., Nogueira, P., Moreno-Navarro, J.J.: Deriving the full-reducing Krivine machine from the small-step operational semantics of normal order. In: PPDP, pp. 85–96 (2013)
23. Grégoire, B., Leroy, X.: A compiled implementation of strong reduction. In: ICFP 2002, pp. 235–246 (2002)
24. Milner, R.: Local bigraphs and confluence: two conjectures. Electron. Notes Theoret. Comput. Sci. **175**(3), 65–73 (2007)
25. Sands, D., Gustavsson, J., Moran, A.: Lambda calculi and linear speedups. In: Mogensen, T.Æ., Schmidt, D.A., Sudborough, I.H. (eds.) The Essence of Computation. LNCS, vol. 2566, pp. 60–82. Springer, Heidelberg (2002)
26. Sestoft, P.: Deriving a lazy abstract machine. J. Funct. Program. **7**(3), 231–264 (1997)
27. Smith, C.: Abstract machines for higher-order term sharing. Presented at IFL 2014 (2014)
28. Wand, M.: On the correctness of the Krivine machine. Higher Order Symbol. Comput. **20**(3), 231–235 (2007)

Compactness in Infinitary Gödel Logics

Juan P. Aguilera(✉)

Vienna University of Technology, 1040 Vienna, Austria
aguilera@logic.at

Abstract. We outline some model-building procedures for infinitary Gödel logics, including a suitable ultrapower construction. As an application, we provide two proofs of the fact that the usual characterizations of cardinals κ such that the Compactness and Weak Compactness Theorems hold for the infinitary language $\mathcal{L}_{\kappa,\kappa}$ are also valid for the corresponding Gödel logics.

Keywords: Gödel logic · Infinitary logic · Compactness

1 Introduction

Infinitary logics, or logics with infinitely long expressions, were first studied by Scott and Tarski [7,8]. Specifically, let κ and λ be cardinal numbers and consider a language $\mathcal{L}_{\kappa,\lambda}$ consisting of the following *non-logical symbols*:

1. finitary predicate symbols,
2. finitary function symbols,
3. constants,

and the following *logical symbols*:

4. a set of variables of size κ,
5. conjunctions $\bigwedge_{\iota<\delta} A_\iota$ and disjunctions $\bigvee_{\iota<\delta} A_\iota$ for $\delta < \kappa$,
6. implication and negation,
7. quantifier chains $\forall_{\iota<\delta} x_\iota$ and $\exists_{\iota<\delta} x_\iota$, for $\delta < \lambda$.

Note, in particular, that we do not necessarily include equality in the language. We give ourselves as much notational freedom as the context allows. For example, we might write $\forall \vec{x}$ or $\bigwedge A_\iota$ if the precise length of the connective is not important.

Infinitary languages quickly gathered interest due to their rich model-theoretic properties and expressive power. For example, the following formula separates the standard model of arithmetic from non-standard models:

$$\forall x \bigvee_{n<\omega} n > x.$$

Partially supported by FWF grants P-26976-N25, I-1897-N25, I-2671-N35, and W1255-N23.

J. Väänänen et al. (Eds.): WoLLIC 2016, LNCS 9803, pp. 22–35, 2016.
DOI: 10.1007/978-3-662-52921-8_2

As is well known, the usual finitary logic ($\mathcal{L}_{\omega,\omega}$ in this notation) is compact. The natural question arose as to whether the languages $\mathcal{L}_{\kappa,\lambda}$ could satisfy suitable analogs of compactness. Recall that a cardinal κ is *weakly compact* if, and only if, it is inaccessible and satisfies the *tree property*, i.e., any tree of size κ such that every level has $< \kappa$ nodes has a branch B of length κ. If so, we say B is a branch *through* the tree. A filter[1] U on some set is κ*-complete* if the intersection of less than κ-many sets in U is also in U. A cardinal κ is *strongly compact* if any κ-complete filter on any set can be extended to a κ-complete ultrafilter. If U is an ultrafilter and $X \in U$, we say X has *measure one with respect to* U (and X has measure zero if $X \notin U$). Let

$$\mathcal{P}_\kappa A = \{S \subset A \colon |S| < \kappa\}.$$

We say an ultrafilter on $\mathcal{P}_\kappa A$ is a *fine measure* if it contains all sets of the form

$$A\uparrow := \{S \in \mathcal{P}_\kappa A \colon A \subset S\}.$$

It is well known (see, for example, [4,5]) that a cardinal κ is strongly compact if, and only if, for every cardinal λ, there exists a fine measure on $\mathcal{P}_\kappa \lambda$. By results of Keisler and Tarski [6] and Hanf [3], the languages $\mathcal{L}_{\kappa,\omega}$ and $\mathcal{L}_{\kappa,\kappa}$ satisfy a strong (resp. weak) analog of the usual compactness theory for classical logic if, and only if, κ is a strongly (resp. weakly) compact cardinal. Specifically, whenever Σ is an arbitrary set (resp. a set where at most κ-many non-logical symbols appear) of formulae such that every subset of Σ of cardinality $< \kappa$ has a model, then Σ has a model. We show that, in a sense made precise below, this is also true when the underlying logic is replaced by any first-order Gödel logic. As we will see, although the proofs are essentially as in the classical case, we need to circumvent a few minor technicalities that arise. In particular, we will need to introduce the notion of *coherent models* for Gödel logics and prove Łoś's Theorem for a suitable ultrapower construction. It has a similar flavor to the analog in continuous model theory (for example, see [2]). An important difference is that, of course, not all logical connectives in Gödel logics are continuous.

2 Gödel Logics

Definition 1. *Let* U *be a set and*[2] $V \subset [0, 1]$ *be closed and containing* 0 *and* 1. *A* valuation $[\![\cdot]\!]$ *of* $\mathcal{L}_{\kappa,\lambda}$ *for* U *and* V *consists of*

1. *For each variable* v, *a value* $[\![v]\!] \in U$;
2. *For each function symbol* f *of arity* n, *a function* $[\![f]\!] \colon U^n \to U$
3. *Similarly, for each predicate symbol, a function* $[\![P]\!] \colon U^n \to V$;

A model *(or* V*-model, if we want to be precise) is a structure* $(U, [\![\cdot]\!])$.

[1] Recall that a (proper) filter $U \neq \wp(X)$ on a set X is a collection of subsets of X that is closed under binary intersections and supersets.

[2] As unfortunate as it is, 'V' is the usual notation for this.

In this paper, the term'model' is used both as in Definition 1 and in the classical sense. The meaning shall always be clear from the context. Also, V will always denote a closed subset of $[0,1]$ containing 0 and 1. Valuations are naturally extended to map any term t to an element $[\![t]\!] \in U$ and any $\mathcal{L}_{\kappa,\lambda}$-formula to a truth value $r \in V$:

$$\begin{aligned}
[\![\bigwedge_{\iota<\delta} A_\iota]\!] &= \inf\{[\![A_\iota]\!] \colon \iota < \delta\};\\
[\![\bigvee_{\iota<\delta} A_\iota]\!] &= \sup\{[\![A_\iota]\!] \colon \iota < \delta\};\\
[\![A \to B]\!] &= \begin{cases} [\![B]\!] & \text{if } [\![A]\!] > [\![B]\!],\\ 1 & \text{if } [\![A]\!] \leq [\![B]\!]; \end{cases}\\
[\![\forall_{\iota<\delta} x_\iota\, A(\vec{x})]\!] &= \inf\{[\![A(\vec{u})]\!] \colon u_\iota \in U \text{ for each } \iota < \delta\};\\
[\![\exists_{\iota<\delta} x_\iota\, A(\vec{x})]\!] &= \sup\{[\![A(\vec{u})]\!] \colon u_\iota \in U \text{ for each } \iota < \delta\}.
\end{aligned}$$

We will also sometimes abuse terminology by making statements about 'all $\vec{u} \subset U$,' when in reality we mean 'all $\vec{u} \subset U$ of the appropriate length.' Hence, the last line of the above definition could have been written as

$$[\![\exists_{\iota<\delta} x_\iota\, A(\vec{x})]\!] = \sup\{[\![A(\vec{u})]\!] \colon \vec{u} \subset U\}.$$

Negation is defined by $\neg A = A \to \bot$, so that

$$[\![\neg A]\!] = \begin{cases} 0 & \text{if } [\![A]\!] > 0,\\ 1 & \text{if } [\![A]\!] = 0; \end{cases} \tag{1}$$

in particular:

$$[\![\neg\neg A]\!] = \begin{cases} 0 & \text{if } [\![A]\!] = 0,\\ 1 & \text{if } [\![A]\!] > 0. \end{cases} \tag{2}$$

If Γ is a set of formulae, we define $[\![\Gamma]\!] = \inf\{[\![B]\!] \colon B \in \Gamma\}$. We say that a set Γ of $\mathcal{L}_{\kappa,\lambda}$-formulae 1-*entails* A, and write $\Gamma \models A$, if $1 = [\![\Gamma]\!]$ implies $1 = [\![A]\!]$ for any valuation $[\![\cdot]\!]$. Given a language $\mathcal{L}_{\kappa,\lambda}$ and a truth-value set V, we can formally define the *Gödel logic* G_V as the set of pairs (Γ, A) such that $\Gamma \models A$.

Indeed, a notion of entailment is usually taken as the central semantic notion for Gödel logics, instead of that of satisfiability. This is due to the fact that satisfiability can in general be defined from entailment, but not conversely (for a general treatment of first-order Gödel logics, see [1]).

Suppose Γ is a set of $\mathcal{L}_{\kappa,\lambda}$-sentences. We say that a set $S \subset \Gamma$ of cardinality $< \kappa$ is a κ-*reduction* for (Γ, A) if $\Gamma \models A$ implies $S \models A$. The following is the main definition:

Definition 2.

- *We say that* $\mathcal{L}_{\kappa,\lambda}$ satisfies the Weak Compactness Theorem for G_V *if every pair* (Γ, A) *where at most* κ*-many non-logical symbols appear has a* κ*-reduction.*

– *We say that* $\mathcal{L}_{\kappa,\lambda}$ satisfies the Compactness Theorem for G_V *if every pair* (Γ, A) *has a* κ*-reduction.*

The first-order language $\mathcal{L}$ under consideration is not important for the previous definition. It should rather be regarded as a statement about κ, λ, and/or V.

2.1 Models Coherent with an Enumeration

Note that our valuations include both interpretations and variable assignments. Hence, we might find two morally equal models that differ only in this regard. To remedy this, we consider the following notion:

Definition 3. *Let* $\mathfrak{U} = (U, [\![\cdot]\!])$ *and* $\mathfrak{W} = (U, (\!|\cdot|\!))$ *be models over the same language. We say* $\mathfrak{U}$ *and* $\mathfrak{W}$ *are* equivalent *if they coincide except perhaps for the values of variables, i.e.,* $[\![P(\vec{u})]\!] = (\!|P(\vec{u})|\!)$ *and* $[\![f(\vec{u})]\!] = (\!|f(\vec{u})|\!)$ *for each* $\vec{u} \subset U$*, each predicate symbol* P *and each function symbol* f*.*

We denote by $\mathcal{T}(\mathcal{L}_{\kappa,\kappa})$ the set of all terms in the language $\mathcal{L}_{\kappa,\kappa}$. In the future, we might be tempted to assume that the set of $\mathcal{L}_{\kappa,\kappa}$-formulae has cardinality κ. This occurs, e.g., if $\kappa = \kappa^{<\kappa}$ and only κ-many non-logical symbols appear in $\mathcal{L}_{\kappa,\kappa}$, as this implies that the set of $\mathcal{L}_{\kappa,\kappa}$-formulae has cardinality $\kappa^{<\kappa}$.

Under this assumption, we shall describe a procedure to replace a G_V-model by an equivalent one where quantified formulae are nicely witnessed. Although it is tailored for our purposes, it can easily be adapted to different contexts. This procedure and its kin will usually be used as Skolemnization supplements for Gödel logics. Let $\mathcal{F}(\mathcal{L}_{\kappa,\kappa}) = \{F_\iota \colon \iota < \kappa\}$ be an enumeration of all $\mathcal{L}_{\kappa,\kappa}$-formulae and $\{y_\iota^{\xi,i} \colon \xi, \iota < \kappa, i < \omega\}$ be a set of distinguished variables whose complement has size κ.

We say an occurrence of a formula F_ι in $\mathcal{F}(\mathcal{L}_{\kappa,\kappa})$ is *irregular* if ι is of the form $\gamma + k$ with γ limit, $0 < k < \omega$, $\vec{x}$ are free variables in F_ι and $F_\gamma = \forall \vec{x}\, F_\iota$ or $F_\gamma = \exists \vec{x}\, F_\iota$. We say an occurrence of a formula is *regular* if it is not irregular.

Lemma 4. *If* κ *is uncountable and the set of* $\mathcal{L}_{\kappa,\kappa}$*-formulae has cardinality* κ*, then there is an enumeration* $\mathcal{F}(\mathcal{L}_{\kappa,\kappa})$ *of* $\mathcal{L}_{\kappa,\kappa}$ *such that:*

1. *each formula appears unboundedly often;*
2. *each formula appears regularly at least once;*
3. $y_\iota^{\xi,i}$ *does not appear in* $\{F_\gamma \colon \gamma < \iota\}$ *for any* ι, ξ, i*;*
4. *whenever* $F_\iota = \forall_{\xi<\delta} x_\xi\, F(x_\xi)_{\xi<\delta}$ *or* $F_\iota = \exists_{\xi<\delta} x_\xi\, F(x_\xi)_{\xi<\delta}$ *appears regularly for the first time in the sequence, then* $F_{\iota+i} = F(y_\iota^{\xi,i})_{\xi<\delta}$ *for each* $0 < i < \omega$*.*

Proof. Assign a formula to each limit ordinal $< \kappa$ in such a way that conditions 1 and 3 are verified. Condition 2 is verified automatically, as a formula can only be irregular at a successor stage. If F_ι is a regular-for-the-first-time occurrence of a formula whose outermost symbol is a chain of quantifiers, define $F_{\iota+i}$ for $i < \omega$ in such a way that condition 4 is witnessed to hold; otherwise, set $F_{\iota+i} = F_\iota$ for $i < \omega$. $\square$

We say an enumeration $\mathcal{F}(\mathcal{L}_{\kappa,\kappa}) = \{\mathcal{F}_\iota \colon \iota < \kappa\}$ is *suitable* if either $\kappa = \aleph_0$ or $\mathcal{F}(\mathcal{L}_{\kappa,\kappa})$ satisfies conditions 1–4 in the statement of Lemma 4.

Definition 5. *We say a model $(U, [\![\cdot]\!])$ is $\mathcal{F}(\mathcal{L}_{\kappa,\kappa})$-*coherent *if $\mathcal{F}(\mathcal{L}_{\kappa,\kappa})$ is suitable and whenever a formula $F_\iota = \forall_{\xi<\delta} x_\xi\, F(x_\xi)_{\xi<\delta}$ or $F_\iota = \exists_{\xi<\delta} x_\xi\, F(x_\xi)_{\xi<\delta}$ appears regularly for the first time in the sequence, then*

$$[\![F_\iota]\!] = \lim_{i<\omega} [\![F_{\iota+i}]\!] \tag{3}$$

Proposition 6. *Suppose the set of $\mathcal{L}_{\kappa,\kappa}$-formulae has cardinality κ. Let $\mathcal{F} = \mathcal{F}(\mathcal{L}_{\kappa,\kappa})$ be a suitable enumeration and $\mathfrak{U} = (U, [\![\cdot]\!])$. Then, there exists an $\mathcal{F}$-coherent model $\mathfrak{W} = (U, (\!|\cdot|\!))$ equivalent to $\mathfrak{U}$.*

Proof. This is clear if $\kappa = \aleph_0$. Suppose $\aleph_1 \leq \kappa$ and partition the set of variables in the language into $Y = \{y_\iota^{\xi,i} \colon \xi, \iota < \kappa, i < \omega\}$ and its complement, Y' and fix a bijection g from Y' onto the set of all variables. We define the valuation $(\!|\cdot|\!)$ to be equal to $[\![\cdot]\!]$ except for the values of variables. Set $(\!|v|\!) = [\![g(v)]\!]$ whenever $v \in Y'$. It remains to define $(\!|\cdot|\!)$ at Y. Let u_0 be an arbitrary, fixed element of U such that $[\![v]\!] = u_0$ for some variable v.

Suppose A is a formula with a chain (or a block of chains) of quantifiers as outermost symbol, e.g., $A = \forall \vec{x}\, F(\vec{x})$. We have that $[\![\forall \vec{x}\, F(\vec{x})]\!] = \inf\{F(\vec{t}) \colon \vec{t} \subset U\}$. Let $\eta = lh(\vec{t})$. Fix an ω-sequence of η-sequences $\{\vec{t_i} \subset U \colon i < \omega\}$ such that $\lim_{i<\omega} F(\vec{t_i}) = [\![\forall \vec{x}\, F(\vec{x})]\!]$. Let F_ι be the first regular occurrence of $\forall \vec{x}\, F(\vec{x})$ in $\mathcal{F}$. We define

$$(\!|y_\iota^{\xi,i}|\!) = \begin{cases} (t_\xi)_i & \text{if } \xi < \eta \\ u_0 & \text{otherwise.} \end{cases}$$

By construction, clearly (3) holds whenever F_ι has a chain (or a block of chains) of quantifiers as outermost symbol and appears regularly for the first time. Moreover, $[\![B(\vec{u})]\!]$ and $(\!|B(\vec{u})|\!)$ coincide for every formula B and every $\vec{u} \subset U$. □

2.2 Ultraproducts

Let U be an ultrafilter on some set I and let $\{\mathfrak{U}_\iota \colon \iota \in I\}$ be a family of models in the language $\mathcal{L}_{\kappa,\lambda}$. We define the ultraproduct of $\{\mathfrak{U}_\iota \colon \iota \in I\}$ in the obvious way, namely, by setting $U = \prod_{\iota\in I} U_\iota / \equiv$, where

$$f \equiv g \text{ if, and only if, } \{\iota \colon f(\iota) = g(\iota)\} \in U.$$

For a function symbol F, we set

$$F[f] = [g] \text{ if, and only if, } \{\iota \colon F(f(\iota)) = g(\iota)\} \in U.$$

For a predicate symbol P, we define $[\![P[f]]\!] = r$ if, and only if,

$$\text{for every } \varepsilon > 0,\ \{\iota \colon |P(f(\iota)) - r| < \varepsilon\} \in U.$$

The ultraproduct is well-defined:

Lemma 7. *Assume P is atomic. Then $\{\iota\colon |P(f(\iota)) - r| < \varepsilon\} \in U$ for exactly one $r \in [0,1]$, so that the ultraproduct is well-defined. Moreover, if U is $(2^{\aleph_0})^+$-complete, then $[\![P[f]]\!] = r$ if, and only if, $\{\iota\colon P(f(\iota)) = r\} \in U$.*

Proof. Suppose that for no r is it the case that $\{\iota\colon |P(f(\iota)) - r| < \varepsilon\} \in U$ for every ε. For each r, choose $\varepsilon_r > 0$ witnessing this. By (topological) compactness of V, finitely-many intervals $(r-\varepsilon_r, r+\varepsilon_r)$ cover V. However, by finite additivity of the ultrafilter, not all of the sets

$$\{\iota\colon |P(f(\iota)) - r| < \varepsilon_r\}$$

can have measure zero—a contradiction. Similarly, let r_0 and r_1 be distinct and $\varepsilon < |r_2 - r_1|/2$. Then $A_i = \{\iota\colon |P(f(\iota)) - r_i| < \varepsilon\}$ cannot have measure one for both $i = 0$ and $i = 1$, as $A_0 \cap A_1 = \varnothing$. A similar argument shows that if U is $(2^{\aleph_0})^+$-complete, then

$$\{\iota\colon P(f(\iota)) = r\} \in U$$

for exactly one $r \in V$. □

We now show that Łoś's Theorem holds in most cases of interest:

Proposition 8. *Assume U is a $(\kappa + \aleph_1)$-complete ultrafilter on I. Let $\mathfrak{U} = (W, [\![\cdot]\!])$ be the ultraproduct of $\{\mathfrak{U}_\iota\colon \iota \in I\}$ by U. Then, Łoś's Theorem holds for $\mathcal{L}_{\kappa,\lambda}$, i.e., for any formula $\varphi \in \mathcal{L}_{\kappa,\lambda}$,*

$$[\![\varphi[f]_{\xi<\delta}]\!] = r \textit{ if, and only if, for every } \varepsilon > 0,\ \{\iota\colon |[\![\varphi(f(\iota))_{\xi<\delta}]\!] - r| < \varepsilon\} \in U. \quad (4)$$

Moreover, if $2^{\aleph_0} < \kappa$, then

$$[\![\varphi[f]_{\xi<\delta}]\!] = r \textit{ if, and only if, } \{\iota\colon [\![\varphi(f(\iota))_{\xi<\delta}]\!] = r\} \in U. \quad (5)$$

Proof. To spare the reader from an otherwise unreadable proof, we will sometimes identify formulae with their truth values and assume predicates are monadic. The proof is by a straightforward induction as usual.

($\bigwedge$) Let $\varphi[f] = \bigwedge_\gamma \varphi_\gamma[f]$. Write $r = [\![\bigwedge_\gamma \varphi_\gamma[f]]\!] = \inf_\gamma [\![\varphi_\gamma[f]]\!]$ and $[\![\varphi_\gamma[f]]\!] = r_\gamma$. Let $\varepsilon > 0$. The induction hypothesis gives that for every γ,

$$A_\gamma := \{\iota\colon |\varphi_\gamma(f(\iota)) - r_\gamma| < \varepsilon/3\} \in U.$$

By κ-completeness, $A := \bigcap_\gamma A_\gamma \in U$. Pick γ_0 such that $r_{\gamma_0} - r < \varepsilon/3$. Since

$$|\varphi(f(\iota)) - r| \leq |\varphi(f(\iota)) - \varphi_{\gamma_0}(f(\iota))| + |\varphi_{\gamma_0}(f(\iota)) - r_{\gamma_0}| + |r_{\gamma_0} - r|,$$

it suffices to show that $|\varphi(f(\iota)) - \varphi_{\gamma_0}(f(\iota))| \leq \varepsilon/3$ in some measure-one set. Suppose not, so that for every ι in some $A' \in U$, there is some $\gamma(\iota)$ such that $\varphi_{\gamma_0}(f(\iota)) > \varphi_{\gamma(\iota)}(f(\iota)) + \varepsilon/3$. Since U is κ-complete and the set of all possible γ has cardinality $< \kappa$, then $\gamma(\iota)$ must take a constant value, say γ^*, in a measure-one subset of A'. We apply the induction hypothesis to φ_{γ_0} to obtain a refinement A'' of A' such that

$$A'' := A' \cap \{\iota\colon |\varphi_{\gamma_0}(f(\iota)) - r_{\gamma_0}| < \varepsilon/6\} \in U, \tag{6}$$

and once more to obtain a further refinement of A'' that witnesses the analog of (6) for γ^*. From this follows that $\varphi_{\gamma^*}[f] \leq \varphi_{\gamma_0}[f] - \varepsilon/3$. Hence, $r = \inf_\gamma r_\gamma \leq r_{\gamma^*} \leq r_{\gamma_0} - \varepsilon/3$; a contradiction.
Conversely, if $\varphi[f] = r' \neq r$, then by the argument above,

$$\{\iota\colon |\varphi(f(\iota)) - r| < |r' - r|/2\} \notin U.$$

($\forall$) Let $r = [\![\forall_{\xi<\delta} x_\xi\, \varphi(x_\xi)_{\xi<\delta}]\!] = [\![\forall \vec{x}\, \varphi(\vec{x})]\!] = \inf_{\vec{f}}\{[\![\varphi[\vec{f}]]\!]\}$. Choose a sequence of (sequences of) terms $\{\vec{f}_i\colon i < \omega\}$ such that $\varphi(\vec{f}_i)$ converges to r and let $r_i = [\![\varphi[\vec{f}_i]]\!]$. From the induction hypothesis follows that for any $i < \omega$, and any $\varepsilon > 0$,

$$\{\iota\colon |\varphi(\vec{f}_i(\iota)) - r_i| < \varepsilon\} \in U.$$

In fact, $\aleph_1$-completeness gives that for any $\varepsilon > 0$,

$$A := \{\iota\colon |\varphi(\vec{f}_i(\iota)) - r_i| < \varepsilon \text{ for every } i\} \in U.$$

Hence, $\forall \vec{x}\, \varphi(\vec{x}) \leq r$ in a measure-one set. We show that for every $\varepsilon > 0$,

$$\{\iota\colon r - \forall\vec{x}\varphi\vec{x}(\iota) \leq \varepsilon\} \in U.$$

Suppose towards a contradiction that for some $0 < \varepsilon^* < 1/2$, we have $\forall \vec{x}\, \varphi(\vec{x}) + \varepsilon^* < r$ in a measure-one subset of A. Define $\vec{g} \in \left(\prod_{\iota \in I} U_\iota\right)^\delta$ by setting

$$\vec{g}(\iota) = \begin{cases} \text{some sequence of terms } \vec{t} \text{ such that } [\![\varphi(\vec{t})]\!]_\iota + \varepsilon^*/2 < r & \text{if it exists} \\ \text{some arbitrary term} & \text{otherwise.} \end{cases}$$

We claim that $\vec{g}(\iota)$ is defined using the first clause in a measure-one set. This follows from the fact that $A' := \{\iota\colon [\![\forall \vec{x}\, \varphi(\vec{x})]\!]_\iota + \varepsilon^* < r\} \in U$. Indeed, for each $\iota \in A'$, there must exist some sequence of terms $\vec{t}_\iota$ such that $0 \leq [\![\varphi(\vec{t}_\iota)]\!]_\iota - [\![\forall\vec{x}\varphi(\vec{x})]\!]_\iota < \varepsilon^*/2$. But then $[\![\varphi(\vec{t}_\iota)]\!]_\iota + \varepsilon^*/2 < r$. Hence the claim follows.
Let $r' < r$ be such that for all $\varepsilon > 0$, $\{\iota\colon |\varphi(\vec{g}(\iota)) - r'| < \varepsilon\} \in U$. We must necessarily have $r' + \varepsilon^*/2 \leq r$. We apply the induction hypothesis to obtain, say, $\varphi[\vec{g}] + \varepsilon^*/3 < r$, which contradicts $r = \inf_{\vec{f}}\{\varphi[\vec{f}]\}$.
To obtain the converse implication, we use the one we just proved as in the first case to show that if $\forall \vec{x}\, \varphi(\vec{x}) = r' \neq r$, then

$$\{\iota\colon |\forall \vec{x}\, \varphi(\vec{x}(\iota)) - r| < |r' - r|/2\} \notin U.$$

($\to$) Let $r = A[f] \to B[f]$, $s = A[f]$, and $t = B[f]$. First suppose $s \le t$ so that $r = 1$, and let $0 < \varepsilon < (t-s)/2$. By induction hypothesis,

$$\{\iota \colon |A(f(\iota)) - s| < \varepsilon \text{ and } |B(f(\iota)) - t| < \varepsilon\} \in U, \tag{7}$$

so that $A(f(\iota)) \to B(f(\iota)) = 1$ on a measure-one set. Now suppose $s > t$, so that $r = t$ and $0 < \varepsilon < (t-s)/2$. As above, the induction hypothesis gives (7) and so $A(f(\iota)) \to B(f(\iota)) = t$ on a measure-one set. The converse is obtained as before.

The remaining cases are similar. Finally, if $2^{\aleph_0} < \kappa$, then (5) holds for atomic formulae by Lemma 7 and the same inductive argument goes through. □

Corollary 9. *Łoś's Theorem holds for the language $\mathcal{L}_{\omega,\omega}$ and the logic G_V for ultraproducts by countably complete ultrafilters.*

Also, from the proof of Proposition 8 follows that:

Corollary 10. *Łoś's Theorem holds for the language $\mathcal{L}_{\omega,\omega}$ and the logic G_V whenever V is finite.*

3 Compactness Theorems

3.1 Weak Compactness

Theorem 11. *Let κ be an uncountable cardinal.*

1. *If $\mathcal{L}_{\kappa,\kappa}$ satisfies the Weak Compactness Theorem for G_V, then κ is weakly compact;*
2. *If κ is weakly compact, then $\mathcal{L}_{\kappa,\kappa}$ satisfies the Weak Compactness Theorem for G_V.*

Proof. 1. We only need the seemingly weaker assumption that $\mathcal{L}_{\kappa,\omega}$ satisfies the Weak Compactness Theorem. Assume $\mathcal{L}_{\kappa,\omega}$ contains a unary predicate symbol $\mathtt{P}$ and a set of constant symbols $\{c_\alpha \colon \alpha < \kappa\}$. To see that κ is inaccessible, note that if $\{\kappa_\alpha \colon \alpha < \lambda\}$ were a sequence of length $\lambda < \kappa$ cofinal in κ, then there would be no κ-reduction for $(\Gamma, \bot)$, where Γ is the set consisting of the sentences

- $\bigvee_{\alpha<\lambda} \bigvee_{\iota<\kappa_\alpha} \mathtt{P}(c_\iota)$,
- $\neg\mathtt{P}(c_\iota)$ for $\iota < \kappa$.

Clearly $S \not\models \bot$ for any proper subset S of Γ—a model witnessing this is provided by interpreting c_ι as ι and setting $[\![\mathtt{P}]\!](\iota) = 1$ for each $\iota \in \{\xi < \kappa \colon \neg\mathtt{P}(c_\xi) \notin \Gamma\}$ and $[\![\mathtt{P}]\!](\iota) = 0$ for all other ι (if it is in Γ, $\bigvee_{\alpha<\lambda} \bigvee_{\iota<\kappa_\alpha} \mathtt{P}(c_\iota)$ is witnessed to be true by any ι such that $\neg\mathtt{P}(c_\iota) \notin \Gamma$); while $\Gamma \models \bot$ vacuously. Hence, κ is regular.

If κ were not a strong limit, so that $2^\lambda \ge \kappa$ for some $\lambda < \kappa$, then there would be no κ-reduction for $(\Gamma, \bot)$ if Γ is the set consisting of the formulae

$$\neg \bigwedge_{\alpha<\lambda} \neg^{1+f(\alpha)}\mathtt{P}(c_\alpha), \text{ for } f \colon \lambda \to 2, \tag{8}$$

where $\neg^n$ has the obvious meaning. Indeed, if S is a proper subset of Γ, then let $g\colon \lambda \to 2$ be such that the corresponding instance of (8) does not belong to S. Interpret each c_α as α and set $[\![\mathtt{P}]\!](\alpha)$ to be 0 or 1 according as $g(\alpha)$ equals 0 or 1. Then $[\![\neg^{1+f(\alpha)}\mathtt{P}(c_\alpha)]\!] = 1$ for each $\alpha < \lambda$ if, and only if, $f = g$, and $[\![\neg^{1+f(\alpha)}\mathtt{P}(c_\alpha)]\!] = 0$ for some α otherwise (negated formulae only take values 0 and 1 by (1)), so that (8) takes value 1 if, and only if, $f \neq g$; in particular, $[\![S]\!] = 1$. However, $\Gamma \models \bot$ vacuously as $[\![\Gamma]\!] = 1$ is impossible, for the function g on λ defined by $g(\alpha) = [\![\neg\neg\mathtt{P}(c_\alpha)]\!]$ must be distinct from each $f\colon \lambda \to 2$. To see this, notice that for any such f, we must have by (8) that $[\![\neg^{1+f(\alpha)}\mathtt{P}(c_\alpha)]\!] = 0$ for some α, but

$$[\![\neg^{1+[\![\neg\neg\mathtt{P}(c_\alpha)]\!]}\mathtt{P}(c_\alpha)]\!] = 1$$

for each $\alpha < \lambda$. To see this, notice that it follows by (2) we have:

$$[\![\neg^{1+[\![\neg\neg\mathtt{P}(c_\alpha)]\!]}\mathtt{P}(c_\alpha)]\!] = \begin{cases} [\![\neg P(c_\alpha)]\!] & \text{if } [\![\mathtt{P}(c_\alpha)]\!] = 0, \\ [\![\neg\neg P(c_\alpha)]\!] & \text{if } [\![\mathtt{P}(c_\alpha)]\!] > 0. \end{cases}$$

The claim then follows by (1) and (2). Hence, κ is inaccessible.

It remains to show κ has the tree property. Let T be a tree of size κ such that each level has cardinality $< \kappa$. Denote by $l(\alpha)$ the αth level of T. We consider the set of sentences Γ consisting of

– $\neg(\mathtt{P}(c_\alpha) \wedge \mathtt{P}(c_\beta))$, for every α and β that are T-incomparable, and
– $\bigvee_{\xi \in l(\alpha)} \mathtt{P}(c_\xi)$, for every α.

For any subset S of Γ of cardinality $< \kappa$, there is a model witnessing $S \not\models \bot$; namely, choose a large-enough downwards-closed fragment of T as universe, assign α to the constant c_α and have $\mathtt{P}$ take value 1 along a sufficiently-large well-ordered set and 0 everywhere else. By the Weak Compactness Theorem, there is also a model witnessing $\Gamma \not\models \bot$. For each T-incomparable α and β,

$$[\![\neg(\mathtt{P}(c_\alpha) \wedge \mathtt{P}(c_\beta))]\!] = 1$$

i.e.,

$$\text{either } [\![\mathtt{P}(c_\alpha)]\!] = 0 \text{ or } [\![\mathtt{P}(c_\beta)]\!] = 0.$$

In particular, all points lying on the same level are incomparable, so that

$$[\![\bigvee_{\xi \in l(\alpha)} \mathtt{P}(c_\xi)]\!] = 1$$

implies that $\mathtt{P}$ must evaluate to 1 on one point in each level. The ordinals α such that $[\![\mathtt{P}(c_\alpha)]\!] = 1$ determine a branch through T. Therefore, T has the tree property.

2. Let Γ be a set of $\mathcal{L}_{\kappa,\kappa}$-formulae of cardinality κ. Suppose κ is a weakly compact cardinal and $S \not\models A$ for every $S \subset \Gamma$ of cardinality $< \kappa$. We will assume all symbols in $\mathcal{L}_{\kappa,\kappa}$ appear in Γ, so that there are only κ-many $\mathcal{L}_{\kappa,\kappa}$-formulae,

and construct a model $(U, [\![\cdot]\!])$ such that $[\![B]\!] = 1$ for each $B \in \Gamma$ and $[\![A]\!] < 1$. The assumption that all symbols in $\mathcal{L}_{\kappa,\kappa}$ appear in Γ results in no loss of generality, for any symbol not appearing in Γ can be evaluated arbitrarily by the model while preserving the conclusion. Fix some $\mathcal{T} = \mathcal{T}(\mathcal{L}_{\kappa,\kappa})$, and some suitable (in the sense of Sect. 2.1) enumeration $\mathcal{F} = \mathcal{F}(\mathcal{L}_{\kappa,\kappa})$.

Let T be the subtree of $V^{<\kappa}$ consisting of all $t\colon \gamma \to V$ such that $\gamma < \kappa$ and there exists an $\mathcal{F}$-coherent model $(W, (\!|\cdot|\!))$ fulfilling the following three conditions:

1. $(\!|F_\iota|\!) = t(\iota)$ for all $\iota < \gamma$;
2. $t(\iota) = 1$ if $F_\iota \in \Gamma$;
3. $t(\iota) < 1$ if $F_\iota = A$;

By hypothesis and Proposition 6, there is one such t for each subset of Γ of cardinality $< \kappa$. Additionally, each level of T has size $|V|$ and κ has the tree property, whereby there exists a branch $\mathcal{B}$ through T. This branch assigns a unique value in V to each formula in $\mathcal{F}$. For each initial segment t of $\mathcal{B}$, there exists a model agreeing with t on all valuations.

Define a relation $\equiv$ to hold between two terms $r, s \in \mathcal{T}$ whenever for each atomic $P(x)$, there exists $\iota < \kappa$ such that $P(r)$ and $P(s)$ appear before F_ι in $\mathcal{F}$ and are assigned the same value by the branch. We let the universe U of the model to be equal to $\mathcal{T}/\equiv$. For each atomic formula $F_\iota \in \mathcal{F}$, we set

$$[\![F_\iota]\!] = r \text{ if, and only if, } t(\iota) = r \text{ for some } t \in \mathcal{B}. \tag{9}$$

This is well-defined, by the definition of $\equiv$. In order to finish the proof, it remains to check that Eq. (9) holds true for arbitrary formulae. If so, then we will have a model where $[\![\Gamma]\!] = 1$ and $[\![A]\!] < 1$. We will check that the following properties hold:

1. $[\![B \to C]\!] = [\![C]\!]$ if $[\![B]\!] > [\![C]\!]$, and $[\![B \to C]\!] = 1$ otherwise.
2. $[\![\bigvee_{\iota<\delta} B_\iota]\!] = \sup\{[\![B]\!]_\iota \colon \iota < \delta\}$.
3. $[\![\bigwedge_{\iota<\delta} B_\iota]\!] = \inf\{[\![B]\!]_\iota \colon \iota < \delta\}$.
4. $[\![\forall_{\iota<\delta} x_\iota\, B(\vec{x})]\!] = \inf\{[\![B(\vec{u})]\!] \colon u_\iota \in \mathcal{T} \text{ for each } \iota < \delta\}$.
5. $[\![\exists_{\iota<\delta} x_\iota\, B(\vec{x})]\!] = \sup\{[\![B(\vec{u})]\!] \colon u_\iota \in \mathcal{T} \text{ for each } \iota < \delta\}$.

Notice that $\mathcal{B}$ evaluates all validities to 1 and respects entailment: if $t(\iota) = 1$ and $F_\iota \models F_\xi$, then $t(\xi) = 1$. The following observation will be used repeatedly: if $t(\iota) = 1$ and $F_\iota = B \to C$, then $[\![B]\!] \leq [\![C]\!]$. This follows from the fact that in every model where $(\!|B \to C|\!) = 1$, we must have $(\!|B|\!) \leq (\!|C|\!)$. This already gives one half of property (1). Conversely, assume $t(\iota) < 1$ and $F_\iota = B \to C$. Let ξ be large enough so that both B and C appear before F_ξ. In any model agreeing with $\mathcal{B}$ up to ξ, necessarily $(\!|B \to C|\!) < 1$, whence $(\!|B \to C|\!) = (\!|C|\!)$ and so $[\![B \to C]\!] = [\![C]\!]$.

For property (2), notice that $B_\xi \to \bigvee_{\iota<\delta} B_\iota$ is valid and thus $[\![\bigvee_{\iota<\delta} B_\iota]\!] \geq \sup\{[\![B]\!]_\iota \colon \iota < \delta\}$. Conversely, let ι^* be an ordinal such that all B_ι appear before F_{ι^*}. Since any model must evaluate $\bigvee_\iota B_\iota$ to the infimum of the values of the B_ι and there exists a model agreeing with $\mathcal{B}$ up to ι^*, it follows that $[\![\bigvee_{\iota<\delta} B_\iota]\!] = \sup\{[\![B]\!]_\iota \colon \iota < \delta\}$. Property (3) is proved analogously.

We show (4): clearly $[\![\forall_{\iota<\delta}x_\iota\, B(\vec{x})]\!] \leq [\![B(\vec{u})]\!]$ for any sequence of terms $\vec{u}$ in $\mathcal{T}$, as $\forall_{\iota<\delta}x_\iota\, B(\vec{x}) \to B(\vec{u})$ is valid. To see that equality holds, it suffices to notice that, if F_ι is the first regular occurrence of $\forall_{\iota<\delta}x_\iota\, B(\vec{x})$, since there exists an $\mathcal{F}$-coherent model agreeing with $\mathcal{B}$ up to level $\iota+\omega$, then $[\![F_\iota]\!] = \lim_{i<\omega} [\![F_{\iota+i}]\!]$.

As for property (5), we clearly have $[\![\exists_{\iota<\delta}x_\iota\, B(\vec{x})]\!] \geq [\![B(\vec{u})]\!]$ for any sequence of terms $\vec{u}$. Suppose $[\![D]\!] \geq [\![B(\vec{u})]\!]$ for any sequence of terms $\vec{u}$ and some formula D. Then we have $[\![B(\vec{u}) \to D]\!] = 1$ by property (1). This implies $[\![\forall\vec{x}(B(\vec{x}) \to D)]\!] = 1$ by property (4), whereby also $[\![\exists\vec{x}\, B(\vec{x}) \to D]\!] = 1$, for

$$\forall\vec{x}(B(\vec{x}) \to D) \models \exists\vec{x}\, B(\vec{x}) \to D.$$

This yields $[\![\exists\vec{x}\, B(\vec{x})]\!] \leq [\![D]\!]$ as desired and finishes the proof. $\square$

3.2 Strong Compactness

Theorem 12. *Let κ be a cardinal.*

1. *If $\mathcal{L}_{\kappa,\kappa}$ satisfies the Compactness Theorem for G_V, then κ is strongly compact;*
2. *If κ is strongly compact, then $\mathcal{L}_{\kappa,\kappa}$ satisfies the Compactness Theorem for G_V.*

Proof. 1. The classical proof goes through. As before, we only suppose for the first claim that $\mathcal{L}_{\kappa,\omega}$ satisfies the Compactness Theorem for G_V. Let F be a κ-complete filter on some set I. Assume $\mathcal{L}_{\kappa,\omega}$ contains a unary predicate $\mathtt{S}$ for every subset S of I and a constant c. Let Γ be the set of

- (extension) sentences $\mathtt{S}(c)$ for every $S \in F$;
- all sentences true in the (classical) structure $(I, \{S\}_{S\subset I})$, in particular:
- (monotonicity) $\mathtt{S}(c) \to \mathtt{S}'(c)$ for every $S \subset S' \subset I$,
- (κ-completeness) $\bigwedge_{\iota<\delta} \mathtt{S}_\iota(c) \to \mathtt{S}(c)$, for $\delta < \kappa$ and $S = \bigcup_{\iota<\delta} S_\iota$,
- (maximality) $\mathtt{S}(c) \vee \neg\mathtt{S}(c)$ for every $S \subset I$.

For every subset Δ of Γ of cardinality $< \kappa$, there is a model witnessing $\Delta \not\models \bot$. In fact, there is a model that takes only values 0 and 1 obtained by taking I as universe and interpreting $\mathtt{S}$ as S for each predicate $\mathtt{S}$ appearing in Δ and c as some element belonging to $\bigcap_{\mathtt{S}\in\Delta} S$, which exists by κ-completeness. By the Compactness Theorem, there is a model $(U, [\![\cdot]\!], \{S^*\}_{S\subset I}, c)$ witnessing $\Gamma \not\models \bot$. Define

$$S \in F^* \text{ if, and only if, } [\![\mathtt{S}(c)]\!] = 1.$$

Clearly, F^* extends F, as $\mathtt{S}(c) \in \Gamma$ for every $S \in F$, whence $[\![\mathtt{S}(c)]\!] = 1$. Also, F^* is a κ-complete filter: suppose $S \in F^*$, so that $[\![\mathtt{S}(c)]\!] = 1$, and $S' \supset S$. Since $\mathtt{S}(c) \to \mathtt{S}'(c) \in \Gamma$, then $[\![\mathtt{S}(c) \to \mathtt{S}'(c)]\!] = 1$, which implies $[\![\mathtt{S}(c)]\!] = [\![\mathtt{S}'(c)]\!] = 1$.

Suppose $S_\iota \in F^*$ for every $\iota < \delta$ and $\delta < \kappa$. It follows that $[\![\mathtt{S}_\iota(c)]\!] = 1$ for each $\iota < \delta$. Letting $S = \bigcap_{\iota<\delta} S_\iota$, we have that $\bigwedge_{\iota<\delta} \mathtt{S}_\iota(c) \to \mathtt{S}(c) \in \Gamma$, whence $[\![\mathtt{S}(c)]\!] = 1$. Hence, F^* is a κ-complete filter. In fact, F^* is an ultrafilter, for $\mathtt{S}(c) \vee \neg\mathtt{S}(c) \in \Gamma$, so that if $S \notin F^*$, then $[\![\mathtt{S}(c)]\!] < 1$, and so the fact that $[\![\mathtt{S}(c) \vee \neg\mathtt{S}(c)]\!] = 1$ implies that $[\![\neg\mathtt{S}(c)]\!] = 1$.

2. Conversely, suppose that κ is a strongly compact cardinal and that for any $S \subset \Gamma$ of cardinality Γ, we have $S \not\models A$, as witnessed by a model $\mathfrak{U}_S = (U_S, [\![\cdot]\!]_S)$. Consider the ultraproduct $\mathfrak{U} = (U, [\![\cdot]\!])$ by a fine measure on $\mathcal{P}_\kappa \Gamma$. By Proposition 8 and the fact that $\kappa > (2^{\aleph_0})^+$ (κ is inaccessible), the ultraproduct satisfies (5), i.e., for any formula $\varphi(x_\xi)_{\xi<\delta}$,

$$[\![\varphi[f]_{\xi<\delta}]\!] = r \text{ if, and only if, } \{S \in \mathcal{P}_\kappa \Gamma \colon [\![\varphi(f(S))_{\xi<\delta}]\!] = r\} \text{ has measure one.}$$

Fineness of the measure implies that $\{\varphi\}\uparrow = \{S \in \mathcal{P}_\kappa \Gamma \colon \{\varphi\} \subset S\}$ has measure one for any $\varphi \in \Gamma$. Moreover, $[\![\varphi]\!]_S = 1$ for any $S \in \{\varphi\}\uparrow$, and so $[\![\varphi]\!] = 1$. Similarly, $[\![A]\!] < 1$, because $[\![A]\!]_S < 1$ for all $S \in \mathcal{P}_\kappa \Gamma$. □

4 An Alternative Proof

(The proofs of) Theorems 11 and 12 are evidence that, sufficiently high up Cantor's realm, the influence of logics' size on their behavior becomes progressively more prevalent, and, that of other traits, progressively less. An example of this is the fact that, for Gödel logics, the truth-value set V seems to play no role whatsoever, in clear contrast to usual finitary first-order logics.

This should not be surprising. Indeed, large cardinalities allow us to diffuse otherwise-characteristic properties of logics by means of codings. Herein, a key ingredient is the regularity of the models Proposition 6 yields. This provides us with alternative proofs of 11.2 and 12.2. These proofs are somewhat more extensive than the ones provided originally, although they do have the clear advantage that with little or no effort, they can be adapted into other contexts. For definiteness, we focus on weak compactness in the following.

Another proof of 11.2 Suppose κ is a weakly compact cardinal and $S \not\models A$ for every $S \subset \Gamma$ of cardinality $< \kappa$. As before, without loss of generality, we assume all symbols in $\mathcal{L}_{\kappa,\kappa}$ appear in Γ. Define a first-order infinitary language $\mathcal{L}'_{\kappa,\kappa}$ consisting of

- the same set of variables `Var` as $\mathcal{L}_{\kappa,\kappa}$,
- the same set of function and constant symbols as $\mathcal{L}_{\kappa,\kappa}$,
- a predicate $P_r^C(\vec{x})$ for every $r \in V$ whenever $C(\vec{x})$ is a $\mathcal{L}_{\kappa,\kappa}$-formula,
- predicates $S^{C,B}(\vec{x}, \vec{y})$ and $W^{C,B}(\vec{x}, \vec{y})$ whenever $C(\vec{x})$ and $B(\vec{y})$ are $\mathcal{L}_{\kappa,\kappa}$-formulae.

Only κ-many non-logical symbols appear in $\mathcal{L}_{\kappa,\kappa}$; thus, the set of $\mathcal{L}_{\kappa,\kappa}$-formulae has cardinality κ. Consequently, only κ-many non-logical symbols appear in $\mathcal{L}'_{\kappa,\kappa}$. We will interpret the infinitary G_V-logic over $\mathcal{L}_{\kappa,\kappa}$ in classical logic. The intended interpretation of $P_r^C(\vec{x})$ is '$C(\vec{x})$ has truth value r.' Similarly, the intended interpretations of $S^{C,B}(\vec{x}, \vec{y})$ and $W^{C,B}(\vec{x}, \vec{y})$ are, respectively, '$C(\vec{x})$ has a (strictly) smaller truth value than $B(\vec{y})$.' Let $\mathcal{F}(\mathcal{L}_{\kappa,\kappa}) = \{F_\iota \colon \iota < \kappa\}$ be a suitable enumeration of all $\mathcal{L}_{\kappa,\kappa}$-formulae with distinguished set of variables $\{y_\iota^{\xi,i} \colon \xi, \iota < \kappa, i < \omega\}$. If $C = \forall_{\xi<\delta} x_\xi F(x_\xi)_{\xi<\delta}$ or $C = \exists_{\xi<\delta} x_\xi F(x_\xi)_{\xi<\delta}$ and F_ι

is the first regular appearance of C in $\mathcal{F}$, we denote by $\mathtt{Var}(F,(x_\xi)_{\xi<\delta})$ the set $\{y^{\xi,i}_\iota : \xi < \delta, i < \omega\}$.

We use the fact that if κ is weakly compact, then $\mathcal{L}_{\kappa,\kappa}$ satisfies the Weak Compactness Theorem for classical logic, as recalled in Sect. 1. Let Σ consist of all sentences of one of the following forms:

1. $\bigvee_{r\in V} P^C_r(\vec{x})$, for each $C(\vec{x}) \in \mathcal{L}_{\kappa,\kappa}$;
2. $P^C_r(\vec{x}) \to \neg P^C_s(\vec{x})$, for each $C(\vec{x}) \in \mathcal{L}_{\kappa,\kappa}$ and each $r \neq s$ in V;
3. $S^{C,B}(\vec{x},\vec{y}) \leftrightarrow \bigvee_{r\in V}\bigvee_{s\in V\cap(r,1]}\big(P^C_r(\vec{x}) \wedge P^B_s(\vec{y})\big)$, for each $C(\vec{x})$, $B(\vec{y}) \in \mathcal{L}_{\kappa,\kappa}$;
4. $W^{C,B}(\vec{x},\vec{y}) \leftrightarrow \bigvee_{r\in V}\bigvee_{s\in V\cap[r,1]}\big(P^C_r(\vec{x}) \wedge P^B_s(\vec{y})\big)$, for each $C(\vec{x})$, $B(\vec{y}) \in \mathcal{L}_{\kappa,\kappa}$;
5. $W^{\bigwedge_{\iota<\delta} C_\iota, C_\xi}((\vec{x}_\iota)_{\iota<\delta}, \vec{x}_\xi)$ for each $\xi < \delta < \kappa$ and each $C(\vec{x}) \in \mathcal{L}_{\kappa,\kappa}$;
6. $P^{\bigwedge_{\iota<\delta} C_\iota}_r(\vec{y}_\iota)_{\iota<\delta} \leftrightarrow \bigwedge_{\epsilon>0}\bigvee_{\xi<\delta}\bigvee_{t\in V\cap[r,r+\epsilon]} P^{C_\xi}_t(\vec{y}_\xi)$, for each sequence of formulae $C_\iota(\vec{y}_\iota) \in \mathcal{L}_{\kappa,\kappa}$;
7. $W^{C_\xi, \bigvee_{\iota<\delta} C_\iota}(\vec{x}_\xi, (\vec{x}_\iota)_{\iota<\delta})$ for each $\xi < \delta < \kappa$ and each $C(\vec{x}) \in \mathcal{L}_{\kappa,\kappa}$;
8. $P^{\bigvee_{\iota<\delta} C_\iota}_r(\vec{y}_\iota)_{\iota<\delta} \leftrightarrow \bigwedge_{\epsilon>0}\bigvee_{\xi<\delta}\bigvee_{t\in V\cap[r-\epsilon,r]} P^{C_\xi}_t(\vec{y}_\xi)$, for each sequence of formulae $C_\iota(\vec{y}_\iota) \in \mathcal{L}_{\kappa,\kappa}$;
9. $W^{\forall\vec{x}C, C}(\vec{y})$ for each $C(\vec{y}) \in \mathcal{L}_{\kappa,\kappa}$;
10. $P^{\forall\vec{x}C(\vec{x})}_r(\vec{y}) \leftrightarrow \bigwedge_{\epsilon>0}\bigvee_{\vec{z}\in\mathtt{Var}(C,\vec{x})}\bigvee_{t\in V\cap[r,r+\epsilon]} P^C_t(\vec{z},\vec{y})$, for each $C(\vec{z},\vec{y}) \in \mathcal{L}_{\kappa,\kappa}$;
11. $W^{C,\exists\vec{x}C}(\vec{y})$ for each $C(\vec{y}) \in \mathcal{L}_{\kappa,\kappa}$;
12. $P^{\exists\vec{x}C(\vec{x})}_r(\vec{y}) \leftrightarrow \bigwedge_{\epsilon>0}\bigvee_{\vec{z}\in\mathtt{Var}(C,\vec{x})}\bigvee_{t\in V\cap[r-\epsilon,r]} P^C_t(\vec{z},\vec{y})$, for each $C(\vec{z},\vec{y}) \in \mathcal{L}_{\kappa,\kappa}$;
13. $W^{C,B}(\vec{x},\vec{y}) \to P^{C\to B}_1(\vec{x},\vec{y})$, for every $C(\vec{x}), B(\vec{y}) \in \mathcal{L}_{\kappa,\kappa}$;
14. $\big(S^{B,C}(\vec{x},\vec{y}) \wedge P^B_r(\vec{x})\big) \to P^{C\to B}_r(\vec{x},\vec{y})$, for every $C(\vec{x}), B(\vec{y}) \in \mathcal{L}_{\kappa,\kappa}$;
15. $\bigwedge_{r\in V\cap[0,1)} P^A_r$;
16. $P^B_1(\vec{x})$, for every $B(\vec{x}) \in \Gamma$.

The first two conditions above state that each formula has exactly one truth value. Conditions 3 and 4 define the predicates $S^{C,B}(\vec{x},\vec{y})$ and $W^{C,B}(\vec{x},\vec{y})$. Conditions 5–14 define how truth values should behave in nonatomic formulae. Specifically, conditions 5–8 define conjunctions and disjunctions, conditions 9–12 define quantifiers, and conditions 13 and 14 define implication.

The restriction of the domain of the conjunction in conditions 10 and 12 is necessary in order to avoid a conjunction of length κ. The last two conditions state that any formula in Γ must have truth value 1 and A must not.

Let Δ be a subset of Σ of cardinality $< \kappa$ and

$$\Delta_0 = \{B \in \mathcal{L}_{\kappa,\kappa} : P^B_1 \in \Delta\}.$$

By hypothesis, there is a G_V-model witnessing $\Delta_0 \not\models A$. The key point is that, by Proposition 6, we can find an $\mathcal{F}$-coherent model $\mathfrak{W} = (U, [\![\cdot]\!])$ witnessing $\Delta_0 \not\models A$. We define a (classical) model $\mathfrak{U}$ for Δ with the same universe:

- For any function symbol f, we set

$$\mathfrak{U} \models f(\vec{x}) = y \text{ if, and only if, } [\![f]\!](\vec{x}) = y. \tag{10}$$

– For any atomic formula C, we set

$$\mathfrak{U} \models P_r^C(\vec{a}) \text{ if, and only if, } [\![C(\vec{a})]\!] = r. \tag{11}$$

Any $F \in \Delta$ of the form 1–16 is satisfied: one verifies by induction that (11) holds for arbitrary formulae $C(\vec{a})$. For example, if F is of the form 10 or 12, then $\mathfrak{U} \models F$ because $\mathfrak{W}$ is $\mathcal{F}$-coherent.

Hence, any subset of Σ of cardinality $< \kappa$ has a classical model, whereby the Weak Compactness Theorem for classical logic yields a model of Σ, say, $\mathfrak{U}$. Let U be the universe of this model. We define a G_V-model with universe U via (10) and (11).

The classical model $\mathfrak{U}$ satisfies sentences 1–16. Since it satisfies 1 and 2, each formula is assigned exactly one truth value in the G_V-model. One verifies—once more by induction—that (11) holds for arbitrary formulae $C(\vec{a})$. For example, suppose $C = \forall x F(x)$. Let $r = [\![\forall x F(x)]\!]$ and $r_u = [\![F(u)]\!]$, so that $r = \inf_{u \in \mathcal{T}} r_u$. By induction hypothesis, $\mathfrak{U} \models P_{r_u}^{F(u)}$ for each u. By Eqs. 1, 2, 4, and 9, $\mathfrak{U} \models P_s^{\forall x F(x)}$ for some $s \leq r$. But necessarily $s = r$, for $\{r_u \colon u \in \mathtt{Var}(C, \vec{x})\}$ converges to r by 10. The other cases are treated similarly. Finally, the model witnesses $\Gamma \not\models A$ by 15 and 16. $\square$

References

1. Baaz, M., Preining, N., Zach, R.: First-order Gödel logics. Ann. Pure Appl. Logic **147**, 23–47 (2008)
2. Yaacov, I.B., Berenstein, A., Henson, C.W., Usvyatsov, A.: Model theory for metric structures. In: Lecture Notes Series of the London Mathematical Society, vol. 350, pp. 315–427 (2008)
3. Hanf, W.P.: On a problem of Erdös and Tarski. Fundamenta Mathematicae **53**, 325–334 (1964)
4. Jech, T.: Set Theory. Springer, New York (2003)
5. Kanamori, A.: The Higher Infinite. Springer, New York (2009)
6. Keisler, H.J., Tarski, A.: From accessible to inaccessible cardinals. Fundamenta Mathematicae **53**, 225–308 (1964)
7. Scott, D., Tarski, A.: The sentential calculus with infinitely long expressions. Colloquium Mathematicum **16**, 166–170 (1958)
8. Tarski, A.: Remarks on predicate logic with infinitely long expressions. Colloquium Mathematicum **16**, 171–176 (1958)

Cut Elimination for Gödel Logic with an Operator Adding a Constant

Juan P. Aguilera(✉) and Matthias Baaz

Vienna University of Technology, 1040 Vienna, Austria
aguilera@logic.at

Abstract. We consider an extension of propositional Gödel logic by an unary operator that enables the addition of a positive real to truth values. We provide a suitable calculus of relations and show completeness and cut elimination.

Keywords: Gödel logic · Cut elimination · Calculus of relations

1 Introduction

Propositional Gödel logic is an extension of intuitionistic logic that takes truth values in the set $[0,1]$. We consider an extension of Gödel logic by a unary operator $\circ$ that adds a positive constant to truth values. This logic can be considered as a logic extending Gödel logic by properties of Łukasiewicz logic that themselves imply the non-recursive-enumerability of the first-order analog. The propositional fragment of this extension can be axiomatized by adding to an axiomatization of Gödel logic the following two simple formulae [2]:

1. $A \prec \circ A$, and
2. $\circ(A \to B) \leftrightarrow (\circ A \to \circ B)$.

We construct an analytic sequents-of-relations calculus based on the relations $<$ and $\leq$, where $\leq$ corresponds to implication ($A \to B$) and $<$ corresponds to the connective $\prec$, where $A \prec B$ is defined as $(B \to A) \to B$. In Sect. 4, we prove cut elimination of the calculus using a Gentzen-style argument based on inductive decomposition of formulae. This calculus is surprisingly more closely related to usual sequent calculi than to the only known analytic calculus for Łukasiewicz propositional logic (see [3,4]). Although it is very simple, its cut elimination is not that straightforward due to the asymmetry of the new operator $\circ$. Indeed, we make use of two technical tools that are not otherwise required: the first one is Avron's *communication* rule; the second one is the following artificial-looking cut:

$$\frac{A < A}{1 < A}$$

This rule is eliminated together with the other cuts.

Partially supported by FWF grants P-26976-N25, I-1897-N25, I-2671-N35, and W1255-N23.

J. Väänänen et al. (Eds.): WoLLIC 2016, LNCS 9803, pp. 36–51, 2016.
DOI: 10.1007/978-3-662-52921-8_3

2 Preliminaries

Definition 1. *We consider the language $\mathcal{L}$ of propositional logic, augmented with a unary operator $\circ$. A propositional Gödel* $\circ$-valuation $\mathfrak{I}$ *is a function from the set of propositional variables into* $[0, 1]$ *with* $\mathfrak{I}(\bot) = 0$ *and* $\mathfrak{I}(\top) = 1$*, together with a real number* $c \in (0, 1]$*. This valuation can be extended to a function mapping formulas from* $\mathcal{L}$ *into* $[0, 1]$ *as follows:*

$$\mathfrak{I}(A \wedge B) = \min\{\mathfrak{I}(A), \mathfrak{I}(B)\},$$
$$\mathfrak{I}(A \vee B) = \max\{\mathfrak{I}(A), \mathfrak{I}(B)\},$$
$$\mathfrak{I}(A \to B) = \begin{cases} \mathfrak{I}(B) & \textit{if } \mathfrak{I}(A) > \mathfrak{I}(B), \\ 1 & \textit{if } \mathfrak{I}(A) \leq \mathfrak{I}(B), \end{cases}$$
$$\mathfrak{I}(\circ A) = \min\{\mathfrak{I}(A) + c, 1\}.$$

We define $\neg A$ by $A \to \bot$ and $A \prec B$ by $(B \to A) \to B$. Thus, we get

$$\mathfrak{I}(\neg A) = \begin{cases} 0 & \text{if } \mathfrak{I}(A) > 0, \\ 1 & \text{otherwise}, \end{cases}$$
$$\mathfrak{I}(A \prec B) = \begin{cases} 1 & \text{if } \mathfrak{I}(A) < \mathfrak{I}(B), \\ \mathfrak{I}(B) & \text{if } \mathfrak{I}(A) \geq \mathfrak{I}(B). \end{cases}$$

Note, in particular, that $\mathfrak{I}(A \prec B) = 1$ if $\mathfrak{I}(A) = \mathfrak{I}(B) = 1$. A formula is called *valid* if it is mapped to 1 for all valuations. The set of all formulas which are valid is called the $\circ$-*propositional Gödel logic* and will be denoted by $G_\circ$.

Proposition 2. *A Hilbert-type axiom system for $G_\circ$ is given by the following axioms and rules:*

I1	$\bot \to A$	I8	$(A \to B) \to [(C \to A) \to (C \to B)]$
I2	$A \to (B \to A)$	I9	$[A \to (C \to B)] \to [C \to (A \to B)]$
I3	$(A \wedge B) \to A$	I10	$(A \to C) \wedge (B \to C) \to ((A \vee B) \to C)$
I4	$(A \wedge B) \to B$	I11	$(C \to A) \wedge (C \to B) \to (C \to (A \wedge B))$
I5	$A \to (B \to (A \wedge B))$	I12	$(A \to (B \to C)) \to (A \wedge B \to C)$
I6	$A \to (A \vee B)$	I13	$[A \to (A \to B)] \to (A \to B)$
I7	$B \to (A \vee B)$	I14	$A \prec \top$
R1	$A \prec \circ A$	R2	$\circ (A \to B) \leftrightarrow (\circ A \to \circ B)$
G1	$(A \to B) \vee (B \to A)$	MP	$\dfrac{A \quad A \to B}{B}$

Proof (*Soundness*). The axioms (I1)–(I14), as well as G1 and MP, are well known to be sound for any extension of Gödel Logic. If $\mathfrak{I}(\circ A) = 1$, then $A \prec \circ A$ holds. If $\mathfrak{I}(\circ A) < 1$, then $\mathfrak{I}(A) < \mathfrak{I}(\circ A)$, whence $A \prec \circ A$ holds as well. Hence, R1 is valid.

To show validity of R2, we distinguish two cases: (i) if $\mathfrak{I}(A) \le \mathfrak{I}(B)$, then $\mathfrak{I}(\circ A) \le \mathfrak{I}(\circ B)$, whereby $1 = \mathfrak{I}(\circ(A \to B)) = \mathfrak{I}(\circ B)$. Hence, R2 holds. (ii) If $\mathfrak{I}(A) > \mathfrak{I}(B)$, then $\mathfrak{I}(A \to B) = \mathfrak{I}(B)$, and so $\mathfrak{I}(\circ(A \to B)) = \mathfrak{I}(\circ B)$. Thus, $\circ(A \to B) \to (\circ A \to \circ B)$. Now, either $\mathfrak{I}(\circ A) \le \mathfrak{I}(\circ B)$ holds, whence $1 = \mathfrak{I}(\circ B) = \mathfrak{I}(\circ A)$ follows, or $\mathfrak{I}(\circ A) > \mathfrak{I}(\circ B)$ holds, whence $\mathfrak{I}(\circ A \to \circ B) = \mathfrak{I}(\circ B)$. In any case, R2 holds.

(*Completeness*). In [2, Theorem 3, (c)], it was shown that the axiom system obtained by replacing R1 by the two axioms

1. $(\bot \prec \circ\bot) \to (A \prec \circ A)$,
2. $(\bot \leftrightarrow \circ\bot) \to (A \leftrightarrow \circ A)$,

is complete for ${G_\circ}^+$, a variation of $G_\circ$ where c could be taken to be zero. $(\bot \prec \circ\bot)$ is an instance of R1 and therefore R1 and R2 are sufficient to derive 1 and 2 above if c is not zero. □

We remark that the deduction theorem holds for the axiom system given by Proposition 2 because it holds for its restriction without the operator $\circ$.

3 The Calculi $\mathsf{RG}_\circ^-$ and $\mathsf{RG}_\circ$

We will define a sequents-of-relations calculus $\mathsf{RG}_\circ$, as well as a fragment thereof, called $\mathsf{RG}_\circ^-$. As we show, the calculus $\mathsf{RG}_\circ^-$ is already sound and complete (Proposition 4). Moreover, $\mathsf{RG}_\circ$ admits cut elimination. This is proved in Sect. 4.

Herein a sequent is a finite set of components of the form $A < B$ or $A \le B$ for formulae A, B. We denote sequents by expressions of the form

$$A_1 \lhd_1 B_1 | \ldots | A_n \lhd_n B_n,$$

where the sign $\lhd_i$ is either $<$ or $\le$ and plays a role similar to the sequent arrow in traditional sequent calculi. By 'component,' we always mean 'an occurrence of the component,' e.g., the sequent $A < B | A < B$ has two components.

We say a component $A < B$ is *satisfied* by an interpretation $\mathfrak{I}$ if $\mathfrak{I}(A \prec B) = 1$ and a component $A \le B$ is satisfied by an interpretation $\mathfrak{I}$ if $\mathfrak{I}(A \to B) = 1$. A sequent Σ is satisfied by $\mathfrak{I}$ if $\mathfrak{I}$ satisfies at least one of its components. Thus, the separation sign "|" is interpreted as disjunction at the meta-level. A sequent Σ is *valid* if it is satisfied by all interpretations.

The axioms of $\mathsf{RG}_\circ^-$ are:

A1. $A \le A$ A2. $0 \le A$ A3. $A < 1$.

The external structural rules are[1]:

$$\frac{\mathcal{H}|A < B|A < B}{\mathcal{H}|A < B}c_1 \qquad \frac{\mathcal{H}|A \le B|A \le B}{\mathcal{H}|A \le B}c_2$$

[1] c stands for 'contraction'; w stands for 'weakening'; *com* stands for 'communication.'

$$\frac{\mathcal{H}}{\mathcal{H}|A < B}\, w_1 \qquad \frac{\mathcal{H}|A \lhd_1 B \qquad \mathcal{H}|C \lhd_3 D}{\mathcal{H}|A \lhd_3 D|C \lhd_4 B}\, com$$

where either $\lhd_1 = \lhd_2 = \leq$ and $\{\lhd_3, \lhd_4\} = \{<, \leq\}$, or $< \in \{\lhd_1, \lhd_2\}$ and $\lhd_3 = \lhd_4 = <$. The internal structural rules are

$$\frac{\mathcal{H}|A < B}{\mathcal{H}|A \leq B}\, w_2 \qquad \frac{\mathcal{H}|A \leq B}{\mathcal{H}|A < C|C \leq B}\, w_3$$

$$\frac{\mathcal{H}|A \leq B}{\mathcal{H}|A \leq C|C < B}\, w_4 \qquad \frac{\mathcal{H}|A < B}{\mathcal{H}|A < C|C < B}\, w_5$$

$$\frac{\mathcal{H}|A < B \qquad \mathcal{H}|B < C}{\mathcal{H}|A < C}\, cut_1 \qquad \frac{\mathcal{H}|A < B \qquad \mathcal{H}|B \leq C}{\mathcal{H}|A < C}\, cut_2$$

$$\frac{\mathcal{H}|A \leq B \qquad \mathcal{H}|B < C}{\mathcal{H}|A < C}\, cut_3 \qquad \frac{\mathcal{H}|A \leq B \qquad \mathcal{H}|B \leq C}{\mathcal{H}|A \leq C}\, cut_4$$

We proceed to logical inferences. The rules for conjunction and disjunction are obtained by replacing $\lhd$ by $<$ or $\leq$ in the following rules:

$$\frac{\mathcal{H}|C \lhd A \quad \mathcal{H}|C \lhd B}{\mathcal{H}|C \lhd (A \wedge B)}\, \wedge_1^{\lhd} \qquad \frac{\mathcal{H}|A \lhd C|B \lhd C}{\mathcal{H}|(A \wedge B) \lhd C}\, \wedge_2^{\lhd}$$

$$\frac{\mathcal{H}|C \lhd A|C \lhd B}{\mathcal{H}|C \lhd (A \vee B)}\, \vee_1^{\lhd} \qquad \frac{\mathcal{H}|A \lhd C \qquad \mathcal{H}|B \lhd C}{\mathcal{H}|(A \vee B) \lhd C}\, \vee_2^{\lhd}$$

The rules for implication are:

$$\frac{\mathcal{H}|A \leq B \mid C < B}{\mathcal{H}|C < (A \to B)}\, \to_1 \qquad \frac{\mathcal{H}|1 < C|B < A \qquad \mathcal{H}|B < C}{\mathcal{H}|(A \to B) < C}\, \to_2$$

$$\frac{\mathcal{H}|A \leq B \mid C \leq B}{\mathcal{H}|C \leq (A \to B)}\, \to_3 \qquad \frac{\mathcal{H}|1 \leq C \mid B < A \qquad \mathcal{H}|B \leq C}{\mathcal{H}|(A \to B) \leq C}\, \to_4$$

Finally, the rules for the operator $\circ$ are as follows:

$$\frac{\mathcal{H}|A \leq B}{\mathcal{H}|A < \circ B}\, \circ_1 \qquad \frac{\mathcal{H}|A \leq B}{\mathcal{H}| \circ A \leq \circ B}\, \circ_2$$

The rule w_1 is an *internal weakening*. By *external weakening* we mean one of w_2–w_5. The *critical components* of an inference are those displayed above, i.e., all components not in $\mathcal{H}$. We say a component is *introduced* by an inference if it appears in its conclusion but is not among its premises. The concept of a formula being *introduced* by an inference is defined analogously. An *end-segment* of a proof π is a downwards-closed subset of π taken as a tree.

Lemma 3

1. *Modus ponens, i.e., the sequent* $A \leq B|A \to B \leq B$, *is derivable in* $\mathsf{RG}_\circ^-$.
2. $\mathsf{RG}_\circ^-$ *derives* $1 \leq A \to B$ *if, and only if, it derives* $A \leq B$.
3. $\mathsf{RG}_\circ^-$ *derives* $1 < A \prec B$ *if, and only if, it derives* $A < B$.

4. $\mathsf{RG}_\circ^-$ derives $1 \le A \vee B$ if, and only if, it derives $1 \le A | 1 \le B$. $\mathsf{RG}_\circ^-$ derives $1 < A \vee B$ if, and only if, it derives $1 < A | 1 < B$.

Proof. 1. Modus ponens is derived as follows:

$$\dfrac{\dfrac{\dfrac{A \le A}{A \le B | B < A}\,w_3}{A \le B | 1 \le B | B < A}\,w_1 + w_2}{A \le B | A \to B \le B}\,\to_4$$

2. From $A \le B$, we derive $1 \le A \to B$ by w_1 and $\to_3$. Assume $1 \le A \to B$ is derivable. The following computation, starting from modus ponens, shows $A \le B$ is derivable:

$$\dfrac{\dfrac{A < 1}{A \le 1}\,w_2 \qquad \dfrac{1 \le A \to B \qquad A \to B \le B | A \le B}{1 \le B | A \le B}\,cut_4}{A \le B}\,cut_4$$

3. Proceed as follows, where (*) as is obtained from $1 \le (B \to A) \to B$ as in 2:

$$\dfrac{\dfrac{A < 1 \qquad \dfrac{(*) B \to A \le B \qquad \dfrac{\dfrac{\dfrac{B \le B}{B \le A | A < B}\,w_4}{1 \le A | B \le A | A < B}\,w_1 + w_2}{1 \le B \to A | A < B}\,\to_3}{1 \le B | A < B}\,cut_4}{A < B | A < B}\,cut_2}{A < B}\,c_2$$

4. We deal only with $\le$. The other case is analogous. One implication is obtained immediately by applying $\vee_1^{\lhd}$. For the converse:

$$\dfrac{\dfrac{1 \le A \vee B \qquad \dfrac{\dfrac{A \le A \qquad \dfrac{B \le B}{B \le A | A \le B}\,w_3 + w_2}{A \vee B \le A | A \le B}\,\vee_2 \qquad B \le B}{A \vee B \le A | A \vee B \le B}\,\vee_2}{1 \le A | A \vee B \le B}\,cut_4 \qquad 1 \le A \vee B}{1 \le A | 1 \le B}\,cut_4$$

□

Proposition 4. *The calculus $\mathsf{RG}_\circ^-$ is sound and complete for the intended interpretation.*

Proof (*Soundness*). The proof relies on a sequents-of-relations calculus for Gödel Logic formulated in [1]. Therein $<$ is interpreted in such a way that $A < B$ is satisfied if and only if $\mathfrak{I}(A) < \mathfrak{I}(B)$. All axioms and rules coincide under both interpretations of $<$ except for rule $(\rightarrow_2)$ and axiom $A3$. Axiom $A3$ is clearly sound. To verify that rule $(\rightarrow_2)$ is valid, note that $A \rightarrow B < C$ is equivalent to $(A \leq B \wedge 1 < C) \vee (B < A \wedge B < C)$. By distributing, we see that it is also equivalent to the formula

$$(A \leq B \vee B < A) \wedge (A \leq B \vee B < C) \wedge (1 < C \vee B < A) \wedge (1 < C \vee B < C).$$

The first conjunct is a tautology. As $1 < C$ implies $B < C$, the fourth conjunct reduces to $B < C$, which subsumes the second one. This gives the validity of the rule. Finally, axioms $\circ_1$ and $\circ_2$ are clearly sound.

(*Completeness*). It suffices to note that any cut-free proof in the complete calculus in [1] can be simulated by the axioms and rules of $\mathsf{RG}_\circ^-$ using axioms and weakening rules to obtain the axioms of the former. Any proof of $1 \leq A$, where A is a formula already valid in Gödel Logic can be simulated in $\mathsf{RG}_\circ^-$. The only rule in [1] which is different to the corresponding rule in $\mathsf{RG}_\circ^-$ has premises which are weakenings of the premises of the original rule. Thus, it suffices to verify that the axioms involving $\circ$ are derivable in $\mathsf{RG}_\circ^-$. Axiom (R1) can be derived directly by rule $\circ_1$. Axiom (R2) can be derived from modus ponens by the following two inferences:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \leq A}{A \leq B|B \leq A}\,w_3+w_2}{A \leq B|1 \leq B|B < A}\,w_1+w_2}{1 \leq A \rightarrow B|B < A}\,\rightarrow_3}{1 \leq \circ(A \rightarrow B)|B < A}\,\circ_1}{1 \leq \circ(A \rightarrow B)|\circ B < \circ A}\,\circ_2 \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{B \leq B}{B \leq A|A \leq B}\,w_3+w_2}{B \leq A \rightarrow B}\,\rightarrow_3}{1 \leq A \rightarrow B|B \leq A \rightarrow B}\,w_1+w_2}{1 \leq \circ(A \rightarrow B)|B \leq A \rightarrow B}\,\circ_1}{1 \leq \circ(A \rightarrow B)|\circ B \leq \circ(A \rightarrow B)}\,\circ_2}{\dfrac{\circ A \rightarrow \circ B \leq \circ(A \rightarrow B)|1 \leq \circ(A \rightarrow B)}{1 \leq (\circ A \rightarrow \circ B) \rightarrow \circ(A \rightarrow B)}\,\rightarrow_3}\,\rightarrow_4$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \leq B|A \rightarrow B \leq B}{A \leq B|\circ(A \rightarrow B) \leq \circ B}\,\circ_2}{\circ A \leq \circ B|\circ(A \rightarrow B) \leq \circ B}\,\circ_2}{\circ(A \rightarrow B) \leq (\circ A \rightarrow \circ B)}\,\rightarrow_3}{\circ(A \rightarrow B) \leq (\circ A \rightarrow \circ B)|1 \leq \circ A \rightarrow \circ B}\,w_1+w_2}{1 \leq \circ(A \rightarrow B) \rightarrow (\circ A \rightarrow \circ B)}\,\rightarrow_3$$

Since we can derive $1 \leq A$ for all instances of any axiom, as well as modus ponens, we can use rule cut_4 to obtain $1 \leq A$ for any formula derivable in the Hilbert-style calculus given by Proposition 2. $\square$

Corollary 5. *All true sequents are derivable in* $\mathsf{RG}_\circ^-$.

Proof. Assume $A_1 \lhd B_1|...|A_n \lhd B_n$ is a true sequent, where each occurrence of $\lhd$ is either $<$ or $\leq$. By Proposition 4, the sequent $1 \leq \bigvee_i A_i \gg B_i$ is derivable, where each occurrence of $\gg$ is either $\rightarrow$ or $\prec$, as appropriate. By Lemma 3.4, the sequent $1 \lhd A_1 \gg B_1|...|1 \lhd A_n \gg B_n$ is derivable. Finally, by Lemmas 3.2 and 3.3, $A_1 \lhd B_1|...|A_n \lhd B_n$ is derivable, as desired. □

Proposition 6. *Compound axioms are derivable in* $\mathsf{RG}_\circ^-$ *from atomic axioms.*

Proof. We consider only the axiom $F \leq F$ for simplicity. The others are similar. Proceed by induction:

1. $F = A \wedge B$:

$$\dfrac{\dfrac{A \leq A}{A \wedge B \leq A}\wedge_1 \quad \dfrac{B \leq B}{A \wedge B \leq B}\wedge_1}{A \wedge B \leq A \wedge B}\wedge_2$$

2. $F = A \vee B$:

$$\dfrac{\dfrac{A \leq A}{B \leq A \vee B}\vee_1 \quad \dfrac{B \leq B}{A \leq A \vee B}\vee_1}{A \vee B \leq A \vee B}\vee_2$$

3. $F = A \to B$:

$$\dfrac{\dfrac{\dfrac{\dfrac{A \leq A}{A \leq B|B < A}w_4}{1 \leq B|A \leq B|B < A}w_1 + w_2}{1 \leq A \to B|B < A}\to_4 \quad \dfrac{\dfrac{B \leq B}{A \leq B|B \leq B}w_1 + w_2}{B \leq A \to B}\to_3}{A \to B \leq A \to B}\to_4$$

4. $F = \circ A$:

$$\dfrac{A \leq A}{\circ A \leq \circ A}\circ_2$$

□

3.1 An Extension

We consider an auxiliary extension of $\mathsf{RG}_\circ^-$ by the following *self-cut* rule:

$$\dfrac{\mathcal{H}|A < A}{\mathcal{H}|1 < A}m$$

It is easy to see that this rule is valid.

Definition 7. *The calculus* $\mathsf{RG}_\circ$ *is the extension of* $\mathsf{RG}_\circ^-$ *resulting by the addition of the self-cut rule.*

In the following section, we show that $\mathsf{RG}_\circ$ admits cut elimination. By a *cut*, we mean either any instance of cut_1–cut_4, or an instance of m. This addition corresponds operationally to the extension of LK or LJ by the mix rule.

4 Cut Elimination

The following is the main theorem:

Theorem 8. $\mathsf{RG}_\circ$ *admits cut elimination; hence, so too does* $\mathsf{RG}_\circ^-$.

To prove Theorem 8, we need a few auxiliary lemmata. We state and prove them now. As the reader will notice, we will sometimes omit cases and/or labeling of rules if we deem it harmless.

Lemma 9. *If there exists a cut-free proof of a sequent $\mathcal{H}$, then there exists a cut-free proof of $\mathcal{H}$ where all instances of w_3–w_5 are such that the formula introduced in the critical components is either atomic or of the form $\circ C$.*

Proof. This can be checked by induction on the size of the introduced formula. We consider the inference w_5. The others are taken care of analogously. For example, a weakening introducing $A \wedge B$, can be replaced as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{C < D}{C < A | A < D}w_5 \quad \dfrac{C < D}{C < B | B < D}w_5}{C < A \wedge B | A < D | B < D}}{C < A \wedge B | A \wedge B < D | B < D}}{C < A \wedge B | A \wedge B < D | A \wedge B < D}}{C < A \wedge B | A \wedge B < D}$$

If the introduced formula is of the form $A \to B$, then consider the following derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{A \leq A}{A \leq B | B < A}w_5}{A \leq B | B < A | 1 \leq D} \quad \dfrac{C < D}{C < B | B < D}w_5}{C < B | A \leq B | A \to B < D}}{C < A \to B | A \to B < D | B < D}$$

The other cases are treated similarly. □

Lemma 10. *For any proof π ending with an instance of m cutting an atomic A and otherwise cut-free, there exists a proof agreeing with π up to that inference, with no instances of m, and such that all cuts have A as cut formula.*

Proof. We proceed by going upwards through π up to the point where $A < A$ was introduced and modifying π as follows:

1. If the inference is some weakening, say w_3, of the form

$$\dfrac{\mathcal{H} | A \leq B}{\mathcal{H} | A < A | A \leq B}w_3$$

then modify π by omitting this inference. At the end of the proof, add an instance of w_1 as follows:

$$\dfrac{\mathcal{H}}{\mathcal{H} | 1 < A}w_1$$

2. If the inference is an instance of *com*, say

$$\frac{\mathcal{H}|A < B \qquad \mathcal{H}|C < A}{\mathcal{H}|A < A|C < B}com$$

   Replace this inference with appropriate instances of cut_1 and w_1.
3. If the inference is a contraction, apply these three steps to each of the two occurrences of $A < A$.

The resulting proof is as required.

□

Lemma 11. *If π is an otherwise-cut-free proof of $\mathcal{H}$ whose last inference is an atomic instance of one of cut_1–cut_4, then there is a cut-free proof of $\mathcal{H}$.*

Proof. Suppose for definiteness that the last inference is an atomic instance of cut_4. Consider the end-segment of the proof of the form

$$\cfrac{\cfrac{\cfrac{\mathcal{G}|C < A}{\vdots\rho}}{\mathcal{H}|C < A} \qquad \mathcal{H}|A < D}{\mathcal{H}|C < D}$$

where $\mathcal{G}|C < A$ is the sequent that introduces the indicated instance of $C < A$. We proceed by cases according to how $C < A$ was inferred. Repeatedly apply any of the following steps until the proof is as desired:

1. If the inference is an instance of w_5 of the form

$$\frac{\mathcal{G}'|C < B}{\mathcal{G}'|C < A|A < B}w_5$$

   we apply an instance of communication as follows:

$$\frac{\mathcal{G}'|C < B \qquad \mathcal{H}|A < D}{\mathcal{G}'|\mathcal{H}|C < D|A < B}com$$

   but then the lower hypersequent is simply $\mathcal{G}|\mathcal{H}|C < D$. Repeat the proof ρ below this hypersequent to obtain $\mathcal{H}|C < D$.
2. If the inference is an instance of w_1, instead, weaken to introduce the sequent $C < D$ and apply ρ to arrive at $\mathcal{H}|C < D$.
3. If the inference is an instance of *com*, say,

$$\frac{\mathcal{G}'|B < A \quad \mathcal{G}'|C < E}{\mathcal{G}'|C < A|B < E}com$$

   we apply the cut rule before this instance of communication as follows:

$$\cfrac{\cfrac{\mathcal{G}'|B < A \quad \mathcal{H}|A < D}{\mathcal{G}'|\mathcal{H}|B < D}cut_1 \qquad \mathcal{G}'|C < E}{\mathcal{G}'|\mathcal{H}|C < D|B < E}com$$

4. If the inference is a contraction, then apply the four steps at the inference where each of the two instances of $C < A$ is introduced.
5. A remaining possibility is that the cut formula A is the constant 1 introduced via an axiom in the left-hand side. In this case, the component $1 < D$ on the right-hand side can only be introduced either via an external weakening, in which case we proceed as in case 1, or via an internal weakening, in which case we replace the inference by an instance of *com*. □

Lemma 12. *Suppose π is a cut-free proof of $\mathcal{H}|\circ A \leq B$. Then there is a cut-free proof of $\mathcal{H}|A < B$.*

Proof. We proceed according as how the sequent $\circ A \leq B$ is inferred. There are three cases. (i) If $\circ A \leq B$ is inferred by an instance of w_1, then simply apply w_1 to infer $A < B$. Else, either (ii) $\circ A \leq B$ is the critical sequent of an inference

$$\frac{\mathcal{H}|A \leq C}{\mathcal{H}| \circ A \leq \circ C}\circ_2 \tag{1}$$

in which case we replace (1) by

$$\frac{\mathcal{H}|A \leq C}{\mathcal{H}|A < \circ C}\circ_1$$

or (iii) the sequent is obtained through a weakening:

$$\frac{\mathcal{H}|C \leq D}{\mathcal{H}|C < \circ A| \circ A \leq D}$$

If so, we replace this inference as follows:

$$\dfrac{\dfrac{\mathcal{H}|C \leq D}{\mathcal{H}|C \leq A|A < D}}{\mathcal{H}|C < \circ A|A < D}\circ_1$$

□

Lemma 13. *For any proof π ending with an instance of m cutting a formula $\circ A$ and otherwise cut-free, there exists a proof agreeing with π up to that inference, with no instances of m, and such that all cuts have A as cut formula.*

Proof. As before, let $\mathcal{G}|\circ A < \circ A$ be the hypersequent where $\circ A < \circ A$ is inferred. If $\mathcal{G}| \circ A < \circ A$ is the lower sequent of an inference $\circ_1$, then apply Lemma 12 to obtain a cut-free proof of $\mathcal{G}|A < A$ and infer $\mathcal{G}|1 < A$ by m. Apply Lemma 10 to obtain a proof agreeing with π up to this point and where all cuts have A as cut formula and infer $\mathcal{G}|1 < \circ A$ using $\circ_1$. Finally, adjoin to the resulting proof the second half of π.

If $\mathcal{G}| \circ A < \circ A$ is the lower sequent of an inference

$$\frac{\mathcal{G}'|C < \circ A \quad \mathcal{G}'| \circ A < D}{\mathcal{G}'| \circ A < \circ A|C < D}com$$

then replace this inference with an instance of cut_1 and w_1 to obtain $\mathcal{G}'|C < D|1 < \circ A$.

Finally, if $\mathcal{G}| \circ A < \circ A$ is the lower sequent of an instance of an internal weakening, say,

$$\frac{\mathcal{G}'| \circ A \leq D}{\mathcal{G}'| \circ A < \circ A| \circ A \leq D}$$

then simply replace this weakening with an external weakening w_1 with critical formula $1 < \circ A$. $\square$

Lemma 14. *If π is an otherwise cut-free proof of $\mathcal{H}$ whose last inference is an instance of cut_1–cut_4 with cut formula $\circ A$, then there is a proof of $\mathcal{H}$ whose only cuts have A as cut formula.*

The proof of Lemma 14 may be found in the Appendix. With this, we can proceed to:

Proof of Theorem 8. We proceed by going downwards through the proof. By induction, assume we are given a proof whose only cut is the last inference I. We proceed by a simultaneous induction on the complexity of the cut formula and the type of cut. Specifically, we successively transform the proof to obtain one of the following:

1. a proof whose only cuts have as cut formula a proper subformula of the initial cut formula, provided I is an instance of one of cut_1–cut_4;
2. if I is an instance of m, then we obtain either a proof whose only cuts are instances of cut_1–cut_4 with the same cut formula as I, or a proof whose only cut is an instance of m and with a proper subformula of the initial cut formula as cut formula.

If the cut formula is atomic (including the case where it is the constant 1), proceed by applying Lemma 10 or Lemma 11, as appropriate. If it is of the form $\circ A$, apply Lemma 13 or Lemma 14. We consider only one more case—implication. For example, suppose there is an end-segment of the proof of the form

$$\dfrac{\dfrac{\dfrac{\dfrac{\mathcal{F}|A < B|C < B}{\mathcal{F}|C < A \to B}}{\vdots \rho_1}}{\mathcal{H}|C < A \to B} \qquad \dfrac{\dfrac{\dfrac{\mathcal{G}|1 < D|B < A \quad \mathcal{G}|B < D}{\mathcal{G}|A \to B < D}}{\vdots \rho_2}}{\mathcal{H}|A \to B < D}}{\mathcal{H}|C < D} \; cut_1$$

Replace the end-segment of the proof with

$$
\dfrac{\dfrac{\dfrac{\mathcal{H}|C<1 \qquad \dfrac{\dfrac{\mathcal{H}|A<B|C<B \quad \mathcal{H}|1<D|B<A}{\mathcal{H}|B<B|C<B|1<D}\,cut_1}{\mathcal{H}|1<B|C<B|1<D}\,m}{\mathcal{H}|C<B|C<B|1<D}\,cut_1 \qquad \mathcal{H}|C<1}{\dfrac{\mathcal{H}|C<B|C<B|C<D}{\mathcal{H}|C<B|C<D}\,c_1}\,cut_1 \qquad \mathcal{H}|B<D}{\dfrac{\mathcal{H}|C<D|C<D}{\mathcal{H}|C<D}\,c_1}\,cut_1
$$

Suppose the last inference is an instance of m with cut formula $A \to B$. Since both the left-hand and right-hand sides of the component must be introduced, we can assume by Lemma 9 that they are introduced by a logical inference. Hence, the proof must have an end-segment with one of the following forms:

1.

$$
\dfrac{\mathcal{F}|A \leq B|1 < A \to B \quad \mathcal{F}|A \leq B|B < B}{\mathcal{F}|A \leq B|A \to B < B}\,\to_2
$$

$$
\vdots\rho_1
$$

$$
\dfrac{\mathcal{G}|A \leq B|A \to B < B}{\mathcal{G}|A \to B < A \to B}\,\to_1
$$

$$
\vdots\rho_2
$$

$$
\dfrac{\mathcal{H}|A \to B < A \to B}{\mathcal{H}|1 < A \to B}\,m
$$

2.

$$
\dfrac{\mathcal{F}|A \leq B|B < B}{\mathcal{F}|B < A \to B}\,\to_1
$$

$$
\vdots\rho_1
$$

$$
\dfrac{\mathcal{G}|1 < A \to B|B < A \qquad \mathcal{G}|B < A \to B}{\mathcal{G}|A \to B < A \to B}\,\to_2
$$

$$
\vdots\rho_2
$$

$$
\dfrac{\mathcal{H}|A \to B < A \to B}{\mathcal{H}|1 < A \to B}\,m
$$

In this case, replace the end-segment with the following:

$$
\mathcal{F}|A \leq B|B < B
$$

$$
\vdots\rho_1, \rho_2
$$

$$
\dfrac{\dfrac{\mathcal{H}|A \leq B|B < B}{\mathcal{H}|A \leq B|1 < B}\,m}{\mathcal{H}|1 < A \to B}\,\to_1
$$

□

As a consequence of the cut-elimination theorem, we obtain the following result. Say a rule A/B is *strongly sound* if under every interpretation $\mathfrak{I}$, $\mathfrak{I}(A) = 1$ implies $\mathfrak{I}(B) = 1$.

Corollary 15. *Every strongly sound rule can be eliminated.*

Proof. As the deduction theorem holds for the Hilbert-style calculus, every strongly sound rule can be eliminated by the addition of a valid formula and cuts. The valid formula can be proved and the cuts eliminated. □

5 Conclusion

It is not clear whether the communication rule is actually essential for the proof. It remains open whether it can be eliminated from the cut-free calculus. If this were the case, then one could arrive at a Maehara-style proof of interpolation, i.e., construct interpolants by induction on the depth of cut-free proofs (see [5] for the classical and intuitionistic formulation of the lemma).

Appendix

Proof of Lemma 14. The end-segment of the proof will be of the form

$$\dfrac{\dfrac{\dfrac{\mathcal{G}|C < \circ A}{\vdots \rho_1}}{\mathcal{H}|C < \circ A} \qquad \dfrac{\dfrac{\mathcal{F}| \circ A < D}{\vdots \rho_2}}{\mathcal{H}| \circ A < D}}{\mathcal{H}|C < D}$$

where $C < \circ A$ and $\circ A < D$ are inferred, respectively, at the hypersequents $\mathcal{G}|C < \circ A$ and $\mathcal{F}| \circ A < D$. We proceed according to which inferences were used above $\mathcal{G}|C < \circ A$ and $\mathcal{F}| \circ A < D$.

1. If the inferences were respectively $\circ_1$ or $\circ_2$, so that the proof is

$$\dfrac{\dfrac{\dfrac{\dfrac{\mathcal{G}|C \leq A}{\mathcal{G}|C < \circ A}}{\vdots \rho_1}}{\mathcal{H}|C < \circ A} \qquad \dfrac{\dfrac{\dfrac{\mathcal{F}|A \leq E}{\mathcal{F}| \circ A \leq \circ E}}{\vdots \rho_2}}{\mathcal{H}| \circ A < \circ E}}{\mathcal{H}|C < \circ E}\ cut$$

then replace it with

$$\dfrac{\dfrac{\dfrac{\dfrac{\mathcal{G}|C \leq A \quad \mathcal{F}|A \leq E}{\mathcal{G}|\mathcal{F}|C \leq E}\ cut}{\vdots \rho_1, \rho_2}}{\mathcal{H}|C \leq E}}{\mathcal{H}|C < \circ E}$$

2. If the inferences were both $\circ_2$, so that the proof is

$$\dfrac{\dfrac{\begin{array}{c}\dfrac{\dfrac{\mathcal{G}|C \le A}{\mathcal{G}|\circ C \le \circ A}}{} \\ \vdots\rho_1\end{array}}{\mathcal{H}|\circ C \le \circ A} \qquad \dfrac{\begin{array}{c}\dfrac{\dfrac{\mathcal{F}|A \le E}{\mathcal{F}|\circ A \le \circ E}}{} \\ \vdots\rho_2\end{array}}{\mathcal{H}|\circ A \le \circ E}}{\mathcal{H}|\circ C \le \circ E}\ cut$$

then replace it with

$$\dfrac{\dfrac{\begin{array}{c}\dfrac{\dfrac{\mathcal{G}|C \le A \quad \mathcal{F}|A \le E}{\mathcal{G}|\mathcal{F}|C \le E}\ cut}{} \\ \vdots\rho_1,\rho_2\end{array}}{\mathcal{H}|C \le E}}{\mathcal{H}|\circ C \le \circ E}$$

3. If the inference on the left-hand side is $\circ_1$ and the inference on the right-hand side is an internal weakening, the proof will be of the form

$$\dfrac{\dfrac{\begin{array}{c}\dfrac{\dfrac{\mathcal{G}|B \le A}{\mathcal{G}|B < \circ A}}{} \\ \vdots\rho_1\end{array}}{\mathcal{H}|B < \circ A} \qquad \dfrac{\begin{array}{c}\dfrac{\dfrac{\mathcal{F}|B \le C}{\mathcal{F}|B \le \circ A|\circ A < C}}{} \\ \vdots\rho_2\end{array}}{\mathcal{H}|B \le \circ A|\circ A < C}}{\mathcal{H}|B \le A|B < C}\ cut$$

Replace it with

$$\dfrac{\dfrac{\begin{array}{c}\dfrac{\dfrac{\mathcal{G}|B \le A \quad \dfrac{\dfrac{\mathcal{F}|B \le C}{\mathcal{F}|B \le A|A < C}}{\mathcal{F}|B < \circ A|A < C}}{\mathcal{F}|\mathcal{G}|B < \circ A|B < C}\ cut}{} \\ \vdots\rho_1,\rho_2\end{array}}{\mathcal{H}|B < \circ A|B < C}}{\mathcal{H}|B \le \circ A|B < C}$$

4. If the inference on the left-hand side is $\circ_2$ and the inference on the right-hand side is an internal weakening, the proof will be of the form

$$\dfrac{\dfrac{\begin{array}{c}\dfrac{\dfrac{\mathcal{G}|B \le A}{\mathcal{G}|\circ B \le \circ A}}{} \\ \vdots\rho_1\end{array}}{\mathcal{H}|\circ B \le \circ A} \qquad \dfrac{\begin{array}{c}\dfrac{\dfrac{\mathcal{F}|B \le C}{\mathcal{F}|B \le \circ A|\circ A < C}}{} \\ \vdots\rho_2\end{array}}{\mathcal{H}|B \le \circ A|\circ A < C}}{\mathcal{H}|\circ B \le A|B < C}\ cut$$

Replace it with

$$\dfrac{\dfrac{\mathcal{G}|B \le A \qquad \dfrac{\mathcal{F}|B \le C}{\mathcal{F}|B \le C|A < C}}{\mathcal{F}|\mathcal{G}|B \le A|B < C}\ cut}{\mathcal{F}|\mathcal{G}|B < \circ A|B < C}$$

$$\dfrac{\dfrac{\vdots\rho_1, \rho_2}{\mathcal{H}|B < \circ A|B < C}}{\mathcal{H}|B \le \circ A|B < C}$$

5. If the inference on the right-hand side is $\circ_2$ and the inference on the left-hand side is an internal weakening, the proof will be of the form

$$\dfrac{\mathcal{F}|B \le D}{\mathcal{F}|B \le \circ A| \circ A < D} \qquad \dfrac{\mathcal{G}|A \le C}{\mathcal{G}| \circ A \le \circ C}$$

$$\dfrac{\dfrac{\vdots\rho_2}{\mathcal{H}|B \le \circ A| \circ A < D} \qquad \dfrac{\vdots\rho_1}{\mathcal{H}| \circ A \le \circ C}}{\mathcal{H}|B \le \circ C| \circ A < D}\ cut$$

Replace it with

$$\dfrac{\mathcal{F}|B \le D \qquad \dfrac{\mathcal{G}|A \le C}{\mathcal{G}| \circ A \le \circ C}}{\mathcal{F}|\mathcal{G}|B \le \circ C| \circ A < D}\ com$$

$$\dfrac{\vdots\rho_1, \rho_2}{\mathcal{H}|B \le \circ C| \circ A < D}$$

6. The final case is that both inferences are internal weakenings:

$$\dfrac{\mathcal{F}|B \le D}{\mathcal{F}|B \le \circ A| \circ A < D} \qquad \dfrac{\mathcal{G}|E \le F}{\mathcal{G}|E \le \circ A| \circ A < F}$$

$$\dfrac{\dfrac{\vdots\rho_1}{\mathcal{H}|B \le \circ A| \circ A < D} \qquad \dfrac{\vdots\rho_2}{\mathcal{H}|E \le \circ A| \circ A < F}}{\mathcal{H}|B < F| \circ A < D|E \le \circ A}\ cut$$

Replace it with

$$\dfrac{\mathcal{F}|B \le D \qquad \mathcal{G}|E \le F}{\mathcal{F}|\mathcal{G}|B \le F|E \le D}\ com$$

$$\dfrac{\dfrac{\vdots\rho_1, \rho_2}{\mathcal{H}|B < F|E < D} \qquad \dfrac{A \le A}{\circ A \le \circ A}}{\mathcal{H}|B < F| \circ A < D|E \le \circ A}\ com$$

$\square$

References

1. Baaz, M., Ciabattoni, A., Fermüller, C.G.: Cut-elimination in a sequents-of-relations calculus for Gödel logic. In: Proceedings of The International Symposium on Multiple-Valued Logic, pp. 181–186 (2001)
2. Baaz, M., Fasching, O.: Monotone operators on Gödel logic. Arch. Math. Logic **53**, 261–284 (2014)
3. Gabbay, D.M., Metcalfe, G., Olivetti, N.: Analytic sequent calculi for Abelian and Łukasiewicz logics. In: Egly, U., Fermüller, C. (eds.) TABLEAUX 2002. LNCS (LNAI), vol. 2381, pp. 191–205. Springer, Heidelberg (2002)
4. Gabbay, D., Metcalfe, G., Olivetti, N.: Proof Theory for Fuzzy Logics. Applied Logic, vol. 36. Springer, Netherlands (2008)
5. Takeuti, G.: Proof Theory. North-Holland, Amsterdam (1987)

A Classical Propositional Logic for Reasoning About Reversible Logic Circuits

Holger Bock Axelsen, Robert Glück, and Robin Kaarsgaard(✉)

DIKU, Department of Computer Science,
University of Copenhagen, Copenhagen, Denmark
{funkstar,glueck,robin}@di.ku.dk

Abstract. We propose a syntactic representation of reversible logic circuits in their entirety, based on Feynman's control interpretation of Toffoli's reversible gate set. A pair of interacting proof calculi for reasoning about these circuits is presented, based on classical propositional logic and monoidal structure, and a natural order-theoretic structure is developed, demonstrated equivalent to Boolean algebras, and extended categorically to form a sound and complete semantics for this system. We show that all strong equivalences of reversible logic circuits are provable in the system, derive an equivalent equational theory, and describe its main applications in the verification of both reversible circuits and template-based reversible circuit rewriting systems.

1 Introduction

Reversible computing–the study of computing models deterministic in both the forward and backward directions–is primarily motivated by a potential to reduce the power consumption of computing processes, but has also seen applications in topics such as static average-case program analysis [17], unified descriptions of parsers and pretty-printers [16], and quantum computing [6]. The potential energy reduction was first theorized by Rolf Landauer in the early 1960s [12], and has more recently seen experimental verification [2,14]. Reaping these potential benefits in energy consumption, however, requires the use of a specialized gate set guaranteeing reversibility, when applied at the level of logic circuits.

Boolean logic circuits correspond immediately to propositions in classical propositional logic (CPL): This is done by identifying input lines with propositional atoms, and logic gates with propositional connectives, reducing the problem of reasoning about circuits to that of reasoning about arbitrary propositions in a classical setting. However, although Toffoli's gate set for reversible circuit logic is equivalent to the Boolean one in terms of what can be computed [22], it falls short of this immediate and pleasant correspondence. This article seeks

The authors acknowledge support from the Danish Council for Independent Research | Natural Sciences under the *Foundations of Reversible Computing* project, and partial support from COST Action IC1405 *Reversible Computation*.
Colors in electronic version.

J. Väänänen et al. (Eds.): WoLLIC 2016, LNCS 9803, pp. 52–67, 2016.
DOI: 10.1007/978-3-662-52921-8_4

to establish such a correspondence by proposing a standardized way of syntactically representing and reasoning about reversible logic circuits. This is done by considering a reformulation, and slight extension, of the toolset of classical propositional logic. The main contributions of this article are the following:

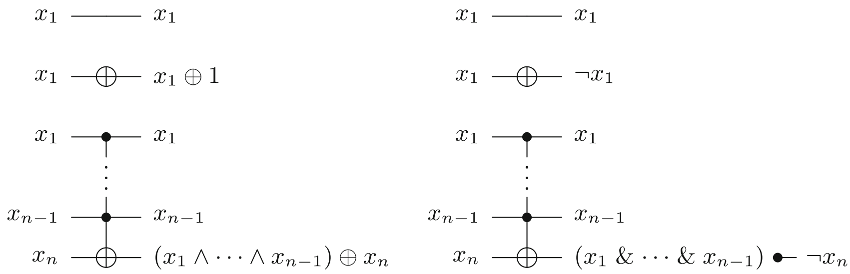
- A syntactic representation of entire reversible logic circuits as propositions, and a pair of proof calculi for reasoning about the semantics of thusly represented reversible logic circuits, sound and complete with respect to
- a categorical/algebraic semantics based on the free strict monoidal category over a *Toffoli lattice*, an order structure proven *equivalent to Boolean rings*,
- a proof that all strong equivalences of reversible logic circuits are provable, and
- an illustration of how the presented logic can be used to show strong equivalences of reversible circuits, and in particular to verify template-based reversible logic circuit rewriting systems.

The complexity of reversible circuits has been increasing while at the same time entirely new functional designs have been found (*e.g.* linear transforms [5], reversible microprocessors [21]). Established tools employing conventional Boolean logic are not geared towards the synthesis, transformation and verification of reversible circuits. Thus, it is important to find better ways of handling this new type of circuits, and some work has been approaching these problems from different angles (*e.g.* [4,23]). Our goal is to formally model the semantics of reversible circuits, and in particular to capture strong equivalence of such circuits as provable equivalence of propositions.

Overview: Sect. 2 introduces the syntax and intuitive interpretation of the connectives, and shows how reversible logic circuits can be represented as propositions by way of a simple annotation algorithm. Section 3 describes the proof calculi used to reason about circuits thus represented, and relates them to existing systems. Section 4 develops the concept of a *Toffoli lattice* as a semantics for the central proof calculus and extends it, via a categorical view on such a structure, into the final model category $\mathfrak{T}_{\otimes}$. Section 5 sketches the fundamental metatheorems of soundness, completeness and circuit completeness, Sect. 6 outlines the applications of the developed theory in reversible circuit rewriting, and Sect. 7 presents ideas for future work, and concludes on the results presented.

2 Circuits as Propositions

The correspondence between Boolean circuits and propositions, in all of its convenience to areas such as circuit design and computational complexity, did not happen by mistake: It is a well-known result that any Boolean function can be computed by a circuit composed of only NAND gates and constants, yet the Boolean gate set is still, in all of its redundancies, considered the *lingua franca* of logic circuit design, precisely due to this correspondence.



(a) Boolean ring semantics. (b) Propositional semantics.

Fig. 1. Toffoli's reversible gate set–consisting of, from top to bottom, the IDENTITY gate, the NOT gate, and the generalized TOFFOLI gate–annotated with their Boolean ring semantics, as well as our propositional semantics.

x_1	x_2	x_3		x_1	x_2	x_3
0	0	0		0	0	0
0	0	1		0	0	1
0	1	0		0	1	0
0	1	1	$\mapsto$	0	1	1
1	0	0		1	0	0
1	0	1		1	0	1
1	1	0		1	1	1
1	1	1		1	1	0

Reversible circuits are usually depicted as gate networks where computation flows from left to right. Here, we consider circuits composed of the gates in Toffoli's reversible gate set, shown in Fig. 1a. (This widely used gate set is known as the *Multiple-Control Toffoli* (MCT) library.) We provide a brief exposition, which the reader familiar with reversible circuit logic can safely skip.

The only gate that warrants particular explanation is the generalized TOFFOLI gate, since the remaining gates behave exactly as they do in Boolean circuit logic: This gate takes $n > 1$ input lines, of which $n - 1$ are *control lines* (marked with black dots), and the remaining one is the *target line* (marked with $\oplus$). If all control lines carry a value of 1, the value on the target line is negated – if not, the input of the target line simply passes through unchanged. As such, *the control lines control whether the* NOT *operation should be carried out on the target line*; in either case, the inputs to all control lines are carried through to the output unchanged (see also the truth table to the right for the generalized Toffoli gate where $n = 3$; x_1 and x_2 are control lines, x_3 is the target line). Circuits may be composed horizontally (*i.e.*, by ordinary function composition) and vertically (*i.e.*, by computation in parallel) so long as they remain finite in size and contain neither loops, fan-in, nor fan-out. Note also that even though the target line is placed at the bottom in Fig. 1a for purposes of illustration, it may be placed anywhere relative to the control lines.

Contrary to Toffoli's Boolean ring semantics for the gate set [22], our presentation embraces Feynman's control interpretation [6] not just in the intuitive explanation given above, but also directly in the formalism. Following Kaarsgaard [11], this is done by replacing exclusive disjunction (here, $\cdot \oplus \cdot$) with the connective $\cdot$ •— $\cdot$, read as *control*, and introducing the usual negation connective $\neg\cdot$ on the target. This results in the propositional semantics shown in Fig. 1b. In this case, the semantics of the target line for the generalized TOFFOLI gate

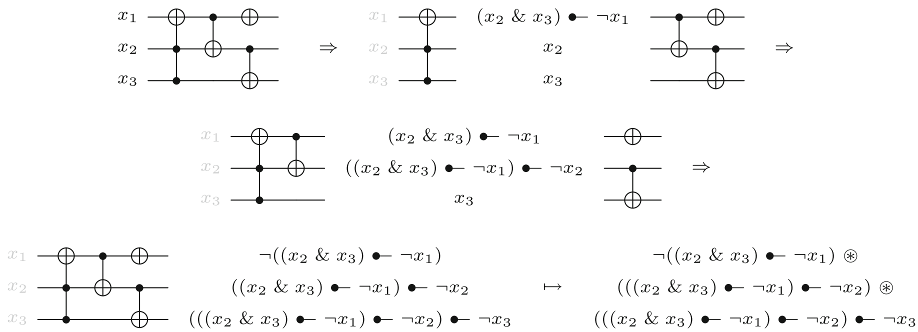


Fig. 2. An example of the annotation algorithm.

pleasingly reads as "x_1 and $\cdots$ and x_{n-1} control not x_n". While Soeken and Thomsen [20] have shown (with their box rules) that control is a general concept corresponding (roughly) to conditional execution of a subcircuit, it turns out that, at the level of individual circuit lines, control carries the same meaning as material bi-implication in CPL. We postpone the proof of equivalence of these two approaches to Sect. 5.

Although the target line of the generalized TOFFOLI gate is, in many ways, the heart of this gate's semantics, it only paints part of the picture. Since reversible circuits are, by definition, required to have the same number of output lines as input lines, parallelism plays a much larger role in reversible circuits than in Boolean ones: To capture the semantics of reversible logic circuits in their entirety, we need a way to capture this parallelism. We do this by introducing yet another connective, $\cdot \circledast \cdot$, read as *while*, with the meaning of $A \circledast B$ as the *multiplicative ordered conjunction* of propositions A and B, *i.e.* as string concatenation in a free monoid. Order is important: as stated earlier, we wish the provable equivalence relation to capture *strong equivalence* of reversible circuits (reversible circuits are strongly equivalent if they compute the same function up to function extensionality [8]), rather than equivalence up to arbitrary permutation of output lines.

Using these two new connectives, along with the usual connectives for conjunction (here, $\cdot \mathbin{\&} \cdot$) and negation, we can produce a straightforward annotation algorithm for extracting the semantics of reversible logic circuits as a proposition in this syntax. As also done for Boolean circuits, we identify each input line with a (fresh) propositional atom, and then propagate the semantics (as given in Fig. 1b) through until the entire circuit has been annotated, at which point we terminate and return the multiplicative ordered conjunction of these propositions, from top to bottom. An example of the annotation algorithm can be found in Fig. 2.

As also noted by Kaarsgaard [11], the syntax of propositions for forming reversible logic circuits using Toffoli's gate set in Fig. 1b is more restrictive than, *e.g.*, CPL; that is, (ordinary) conjunctions only appear as subpropositions of controls. Further, linear ordered conjunctions only appear as a way of "glueing" the propositions of individual circuit lines together (see, *e.g.*, the final step in Fig. 2).

This structure suggests a syntactic hierarchy, which we will illustrate by means of color: Blue propositions will be those that correspond to the semantics of a single circuit line (perhaps of many in a circuit), red propositions correspond to the semantics of entire circuits (or subcircuits), and yellow (or recolorable) propositions will be those that can be either of these two. Formally, we define such propositions to be those produced by the grammars

$$A_B, B_B, C_B := A_Y \mid \neg A_B \mid A_B \mathbin{\&} B_B \qquad \text{(Blue propositions)}$$
$$A_Y, B_Y, C_Y := a \mid \mathbf{0} \mid \mathbf{1} \mid \neg A_Y \mid A_B \multimapdotinv B_Y \mid A_Y \multimapdotinv B_B \qquad \text{(Yellow propositions)}$$
$$A_R, B_R, C_R := A_Y \mid A_R \circledast B_R \mid \mathbf{e} \qquad \text{(Red propositions)}$$

where a denotes any propositional atom; we will assume that there is a denumerable set P of these. For readability, we adopt the convention that $\neg\cdot$ binds tighter than $\cdot \mathbin{\&} \cdot$, which binds tighter than $\cdot \multimapdotinv \cdot$, which finally binds tighter than $\cdot \circledast \cdot$. Further, we will omit subscripts when the syntactic class is clear from the context.

Starting with blue and recolorable propositions, $\mathbf{0}$ and $\mathbf{1}$ represent the false respectively true proposition (corresponding to *ancillae*, lines of constant value, in circuit terms), $\neg A_Y$ the usual negation of a proposition, $A_B \multimapdotinv B_Y$ and $A_Y \multimapdotinv B_B$ as "A *control* B", and finally $A_B \mathbin{\&} B_B$ as the usual (additive) conjunction. Red propositions are interpreted as circuit structures, with $A_R \circledast B_R$ representing the ordered (parallel) structure made up of A_R and B_R, and $\mathbf{e}$ representing the empty structure (*i.e.*, the empty circuit). Further, we will denote the set of all such well-formed blue respectively red propositions by Φ_B respectively Φ_R.

In the same manner, well-formed blue and red contexts (a notion of a recolorable context is unnecessary) are those produced by the grammars

$$\Gamma_B, \Delta_B, \Pi_B := \cdot \mid \Gamma_B, A_B \qquad \text{(Blue contexts)}$$
$$\Gamma_R, \Delta_R, \Pi_R := \cdot \mid \Gamma_R, A_R \qquad \text{(Red contexts)}$$

The distinction between the empty blue context and the empty red one is important, since the two types of contexts will be interpreted in two different ways; blue contexts are interpreted as an additive (blue) conjunction with $\mathbf{1}$ as unit, while red contexts are interpreted as an ordered multiplicative (red) conjunction with $\mathbf{e}$ as unit. As we did for propositions, we will denote the set of all well-formed blue respectively red contexts by Φ_B^* respectively Φ_R^*.

3 Proof Calculi

As the syntax presented in the previous section perhaps already alludes to, we will use not one but two proof calculi to reason about propositions thus formed. Figures 3 and 4 show the two proof calculi–the blue and the red fragment, respectively–that make up the logic which we shall call $\mathrm{LRS}_{\circledast}$.

CORE RULES

$$\frac{}{\Gamma, A \vdash_B A}\ (\mathrm{BId}) \qquad \frac{\Gamma \vdash_B A \qquad \Gamma, A \vdash_B B}{\Gamma \vdash_B B}\ (\mathrm{BCut})$$

STRUCTURAL RULES

$$\frac{\Gamma \vdash_B B}{\Gamma, A \vdash_B B}\ (\mathrm{Wkn}) \qquad \frac{\Gamma, A, A \vdash_B B}{\Gamma, A \vdash_B B}\ (\mathrm{Cnt})$$

$$\frac{\Gamma, A, \Delta, B, \Pi \vdash_B C}{\Gamma, B, \Delta, A, \Pi \vdash_B C}\ (\mathrm{Exc})$$

UNITS

$$\frac{}{\Gamma \vdash_B \mathbf{1}}\ (\mathbf{1}\mathrm{I}) \qquad \text{(no elimination for } \mathbf{1}\text{)}$$

$$\text{(no introduction for } \mathbf{0}\text{)} \qquad \frac{\Gamma \vdash_B \mathbf{0}}{\Gamma \vdash_B A}\ (\mathbf{0}\mathrm{E})$$

CONJUNCTION

$$\frac{\Gamma \vdash_B A \qquad \Gamma \vdash_B B}{\Gamma \vdash_B A \,\&\, B}\ (\&\mathrm{I}) \qquad \frac{\Gamma \vdash_B A \,\&\, B}{\Gamma \vdash_B A}\ (\&\mathrm{E}_1)$$

$$\frac{\Gamma \vdash_B A \,\&\, B}{\Gamma \vdash_B B}\ (\&\mathrm{E}_2)$$

CONTROL

$$\frac{\Gamma, A \vdash_B B \qquad \Gamma, B \vdash_B A}{\Gamma \vdash_B A \bullet\!\!- B}\ (\bullet\!\!-\mathrm{I}) \qquad \frac{\Gamma \vdash_B A \bullet\!\!- B \qquad \Gamma \vdash_B B}{\Gamma \vdash_B A}\ (\bullet\!\!-\mathrm{E}_1)$$

$$\frac{\Gamma \vdash_B A \bullet\!\!- B \qquad \Gamma \vdash_B A}{\Gamma \vdash_B B}\ (\bullet\!\!-\mathrm{E}_2)$$

NEGATION

$$\frac{\Gamma, A \vdash_B \mathbf{0}}{\Gamma \vdash_B \neg A}\ (\neg\mathrm{I}) \qquad \frac{\Gamma \vdash_B A \qquad \Gamma \vdash_B \neg A}{\Gamma \vdash_B \mathbf{0}}\ (\neg\mathrm{E})$$

CLASSICAL RULES

$$\frac{\Gamma, A \vdash_B B \qquad \Gamma, \neg A \vdash_B B}{\Gamma \vdash_B B}\ (\mathrm{Lem}) \qquad \frac{\Gamma \vdash_B \neg\neg A}{\Gamma \vdash_B A}\ (\neg\neg\mathrm{E})$$

Fig. 3. The blue fragment of $\mathrm{LRS}_{\circledast}$. (Color figure online)

There are two judgment forms, $\Gamma_B \vdash_B \varphi_B$ and $\Gamma_R \vdash_R \varphi_R$, which differ not only by syntax, but also by the interpretation of the context: Blue contexts are understood to be an additive (ordinary) conjunction of its constituent propositions (as usual) with $\mathbf{1}$ as unit, while red contexts are understood as a multiplicative ordered conjunction of its constituent propositions with $\mathbf{e}$ as unit. This difference of interpretation is reflected directly in the core rules of the calculi; while the identity and cut rules for the red fragment use careful bookkeeping to ensure that order and linearity are not broken, the corresponding rules in the blue fragment display implicit use of the structural rules available in the blue fragment. More explicitly, the blue fragment contains the usual structural rules of CPL–weakening, contraction, and exchange–while the red fragment has *none* of these.

The blue fragment, largely similar to the sequent calculus of LRS [11], presents itself as a reformulation of CPL in which control (corresponding to material bi-implication) is taken as a fundamental connective, while implication and disjunction are omitted. In particular, the omission of disjunction as a connective presents a challenge for classical reasoning, as we can no longer express the law of the excluded middle axiomatically in a way which facilitates its easy use in derivations. To resolve this, we present the rule instead as an explicit case analysis, corresponding to a proof tree of the form

$$\text{Core rules} \quad \frac{}{A \vdash_R A}\,(\text{RId}) \qquad \frac{\Delta \vdash_R A \qquad \Gamma, A, \Pi \vdash_R B}{\Gamma, \Delta, \Pi \vdash_R B}\,(\text{RCut})$$

$$\text{Unit} \quad \frac{}{\cdot \vdash_R \mathbf{e}}\,(\mathbf{e}\text{I}) \qquad \frac{\Delta \vdash_R \mathbf{e} \qquad \Gamma, \Pi \vdash_R A}{\Gamma, \Delta, \Pi \vdash_R A}\,(\mathbf{e}\text{E})$$

$$\text{Ordered conjunction} \quad \frac{\Gamma \vdash_R A \qquad \Delta \vdash_R B}{\Gamma, \Delta \vdash_R A \circledast B}\,(\circledast\text{I}) \quad \frac{\Delta \vdash_R A \circledast B \qquad \Gamma, A, B, \Pi \vdash_R C}{\Gamma, \Delta, \Pi \vdash_R C}\,(\circledast\text{E})$$

$$\text{Recoloring} \quad (\dagger)\ \frac{A \vdash_B B}{A \vdash_R B}\,(\text{Rcl}) \qquad \dagger \ \text{Side condition: } A \text{ and } B \text{ are recolorable.}$$

Fig. 4. The red fragment of LRS$_\circledast$. (Color figure online)

$$\dfrac{\dfrac{}{\Gamma \vdash A \vee \neg A}\,(\text{Lem}) \qquad \begin{matrix}\vdots \\ \Gamma, A \vdash B\end{matrix} \qquad \begin{matrix}\vdots \\ \Gamma, \neg A \vdash B\end{matrix}}{\Gamma \vdash B}\,(\vee\text{E})$$

in CPL, which not only presents the common use case of the law of the excluded middle, but is also strong enough to derive the double negation elimination rule in the straightforward way. (Proof theoretically inclined buyers beware: Though this rule is sufficiently powerful, it threatens the subformula property [3] even in the face of cut-elimination.) Note that as we are not aiming for minimality, both rules are included in the blue fragment.

The red fragment offers little in terms of rules, since the only structure we are interested in is the parallel structure of circuit lines, captured by the rules for ordered multiplicative conjunction – essentially corresponding to concatenation of strings (though our formulation follows the conjunctive fragment of Polakow's presentation [15] of the Lambek calculus), with **e** corresponding to the empty string.

In our setting, by far the most interesting rule of the red fragment is the recoloring rule, which states that any logical deduction from a single recolorable proposition can be inserted into a structure of unit length, as long as the succedent is likewise recolorable. Recall that the recolorable propositions are precisely those that are well-formed as both blue and red propositions, so this (purely syntactic) side condition is entirely reasonable. Figure 5 gives a larger example of an LRS$_\circledast$ derivation, showing $\neg x_1 \circledast x_1 \bullet\!\!- \neg x_2 \vdash_R \neg x_1 \circledast \neg\neg x_1 \bullet\!\!- \neg x_2$.

Finally, it is worth noting that the syntax of red propositions is not strong enough to ensure that only reversible circuits can be represented. For example, the red proposition $x_1 \circledast x_1 \,\&\, x_2 \bullet\!\!- \neg x_3$ is perfectly well-formed, but does not correspond directly to a reversible circuit. On the other hand, every reversible circuit *can* adequately, and with minimal work, be represented as a red proposition, as we saw in Sect. 2. This turned out to result in an interesting tradeoff in the proof calculi: Naturally, it would be desirable if we could guarantee that every red proposition corresponded precisely to a reversible circuit–however, by not guaranteeing this property, we may consider the semantics of a single line or group of lines in isolation, without having to take the overall structure of the circuit into account at every step of a derivation, making for simpler overall logic.

4 Semantics

Given the obvious similarities between CPL and the blue fragment of LRS$_{\circledast}$, it would seem highly natural to adopt truth-functional semantics here as well. While this approach certainly works when considering the blue fragment in isolation, extending this approach to the red fragment runs into the problem of defining a single truth value–and for good reason, since truth should be interpreted relative to a circuit structure, taking order and resource use (*i.e.*, viewing circuit lines as ordered resources) into consideration.

For this reason, we will instead take the algebraic approach to semantics by considering what we call a *Toffoli lattice*, an order structure with obvious similarities to the blue fragment of LRS$_{\circledast}$. This approach has the immediate benefit that order structures can very easily be interpreted as categories, giving us a whole suite of tools to extend the semantics to the red fragment. We define Toffoli lattices, and their corresponding homomorphisms, as follows:

Definition 1. *A Toffoli lattice* $\mathfrak{A} = (A, \leq, \top, \bot, \wedge, \leftrightharpoons, \overline{\cdot})$ *consists of a partially ordered set* $(A, \leq)$ *furnished with the following operations and conditions:*

(i) *There is a greatest element* $\top$ *such that* $x \leq \top$ *for any element* x*.*
(ii) *There is a least element* $\bot$ *such that* $\bot \leq x$ *for any element* x*.*
(iii) *Given elements* a, b *there is an element* $a \wedge b$ *such that* $x \leq a \wedge b$ *iff* $x \leq a$ *and* $x \leq b$*.*
(iv) *Given elements* a, b *there is an element* $a \leftrightharpoons b$*, the* relative equivalence *of* a *and* b*, such that* $x \leq a \leftrightharpoons b$ *iff* $x \wedge a \leq b$ *and* $x \wedge b \leq a$*.*
(v) *Given an element* a*, there is an element* $\overline{a}$ *satisfying* $x \leq \overline{a}$ *iff* $x \wedge a \leq \bot$*,* $a \wedge \overline{a} \leq \bot$*, and if* $x \wedge a \leq b$ *and* $x \wedge \overline{a} \leq b$ *then* $x \leq b$*.*

As is often done, we will use $|\mathfrak{A}|$ *to denote the carrier set* A*.*

Definition 2. *Let* $\mathfrak{A}$ *and* $\mathfrak{B}$ *be Toffoli lattices. A Toffoli lattice homomorphism is a function* $h : |\mathfrak{A}| \to |\mathfrak{B}|$ *that preserves all lattice operations and constants, i.e.,* $h(\top) = \top$*,* $h(\bot) = \bot$*,* $h(a \wedge b) = h(a) \wedge h(b)$*,* $h(a \leftrightharpoons b) = h(a) \leftrightharpoons h(b)$*, and* $h(\overline{a}) = \overline{h(a)}$ *for all* $a, b \in |\mathfrak{A}|$*.*

From this definition, the truth-functional semantics appear by considering the set $\{0, 1\}$:

Example 1. The set $\{0, 1\}$ equipped with the usual partial order is a Toffoli lattice: Assigning the usual truth table semantics to $\top$, $\bot$, $\wedge$, and complement, and defining

$$0 \leftrightharpoons 0 = 1 \qquad 0 \leftrightharpoons 1 = 0 \qquad 1 \leftrightharpoons 0 = 0 \qquad 1 \leftrightharpoons 1 = 1$$

it is straightforward to verify that this yields a Toffoli lattice.

Though no explicit join operation is given, joins may be defined using meets and complements–*i.e.*, analogously to Boolean algebras, one can show that $\overline{\overline{x} \wedge \overline{y}}$ is the least upper bound of x and y.

Lemma 1. *Let $h : \mathfrak{A} \to \mathfrak{B}$ be a Toffoli lattice homomorphism. Then h is specifically monotonic, i.e. if $a \leq b$ then $h(a) \leq h(b)$.*

Like so many other structured sets, these definitions lead us, without much trouble, to show that Toffoli lattices with homomorphisms between them form a concrete category; a useful feature which we will use shortly to characterize the free Toffoli lattice.

Theorem 1. *The class of all Toffoli lattices with Toffoli lattice homomorphisms between them forms a category, **TL**.*

Careful inspection of the definition of a Toffoli lattice reveals a correspondence with the blue fragment of $\mathrm{LRS}_{\circledast}$ – of course, this is entirely by design, though this correspondence is missing one part, namely the propositional atoms (recall the assumption that these form a denumerable set P). To account for these, we observe that Toffoli lattices may be freely constructed, and apply this free construction to the set of propositional atoms P to form an order theoretic model of the blue fragment.

Theorem 2. *Toffoli lattices may be freely constructed,* i.e.*, the forgetful functor $U : \mathbf{TL} \to \mathbf{Sets}$ has a left adjoint $F : \mathbf{Sets} \to \mathbf{TL}$.*

Using this theorem, we take $\mathfrak{T} = FP$ (where P is the set of propositonal atoms) to be our model of the blue fragment. This allows us to define blue denotation and entailment:

Definition 3. *The denotation of a blue proposition $\varphi_B \in \Phi_B$, denoted $[\![\varphi_B]\!]$, is given by the function $[\![\cdot]\!] : \Phi \to |\mathfrak{T}|$ defined as follows:*

$$[\![\mathbf{1}]\!] = \top \qquad [\![a]\!] = a \qquad [\![A_B \,\&\, B_B]\!] = [\![A_B]\!] \wedge [\![B_B]\!]$$
$$[\![\mathbf{0}]\!] = \bot \qquad [\![\neg A_B]\!] = \overline{[\![A_B]\!]} \qquad [\![A_B \bullet\!\!- B_B]\!] = [\![A_B]\!] \leftrightharpoons [\![B_B]\!]$$

where a denotes any propositional atom in P. Further, the denotation of a blue context $\Gamma_B \in \Phi_B^*$ is given by the overloaded function $[\![\cdot]\!] : \Phi_B^* \to |\mathfrak{T}|$ defined by

$$[\![\cdot]\!] = \top \qquad [\![\Gamma'_B, A_B]\!] = [\![\Gamma'_B]\!] \wedge [\![A_B]\!]$$

Definition 4 (Blue entailment). *Let Γ be a well-formed blue context, and φ be a well-formed blue formula. Then we define the blue entailment relation by $\Gamma \vDash_B \varphi$ iff $[\![\Gamma]\!] \leq [\![\varphi]\!]$ in $\mathfrak{T}$.*

In the same manner as for any other partially ordered set, we can regard a single Toffoli lattice $\mathfrak{A}$ as a (skeletal preorder) category by considering each element of $|\mathfrak{A}|$ as an object of the category, and placing a morphism between objects X and Y iff $X \leq Y$ in $\mathfrak{A}$. This allows us to extend our model lattice $\mathfrak{T}$ by categorical means to form a model of the red fragment. A key insight in this regard is the role of monoidal categories in modelling linear logic [18]; in particular, a strict monoidal category is sufficient to model the red fragment. This leads to the following construction:

Definition 5. *Let* $\mathfrak{T}_\otimes$ *denote the* free strict monoidal category *over* $\mathfrak{T}$. *That is,* $\mathfrak{T}_\otimes$ *has as objects all strings* $X_1X_2\dots X_n$ *where all* X_i *are objects of* $\mathfrak{T}$*, and as morphisms all strings of morphisms* $f_1f_2\dots f_n : X_1X_2\dots X_n \to Y_1Y_2\dots Y_n$ *for morphisms* $f_i : X_i \to Y_i$ *of* $\mathfrak{T}$*. It has a monoidal tensor* $-\otimes- : \mathfrak{T}_\otimes \times \mathfrak{T}_\otimes \to \mathfrak{T}_\otimes$ *defined by concatenation; thus it is strictly associative and has a strict unit* i*, denoting the empty string.*

See, *e.g.*, Joyal and Street [9,10] for the construction of the free strict monoidal category (or, in their nomenclature, free strict tensor category) over a given category $\mathbf{C}$; it simply amounts to be the coproduct of all functor categories of the form $\mathbf{C}^n$, where n is the discrete category of n objects. This allows us to characterize $\mathfrak{T}_\otimes$ by means of a (Grothendieck) fibration (see, *e.g.*, Jacobs [7]) into the discrete category $\Delta_\mathbb{N}$ which has, as objects, all natural numbers[1]:

Theorem 3. *The functor* $\Psi : \mathfrak{T}_\otimes \to \Delta_\mathbb{N}$ *defined by mapping objects to their lengths as strings, and morphisms to the corresponding identities is a Grothendieck fibration and a monoidal functor. Specifically, each inverse image* $\Psi^{-1}(k)$ *for* k *in* $(\Delta_\mathbb{N})_0$ *is a full subcategory of* $\mathfrak{T}_\otimes$*.*

Proof. (Proof sketch). Since $\Delta_\mathbb{N}$ is discrete, for any object $X_1X_2\dots X_n$ of $\mathfrak{T}_\otimes$, the only possible morphism in $\Delta_\mathbb{N}$ of the form $u : K \to \Psi(X_1X_2\dots X_n)$ is the identity $1_{\Psi(X_1X_2\dots X_n)}$, which the identity morphism $1_{X_1X_2\dots X_n}$ is trivially cartesian over.

To see that Ψ is a strict monoidal functor, we note the obvious tensor product in $\Delta_\mathbb{N}$ given by addition, *i.e.*, by mapping objects $A\otimes B$ to their sum (as natural numbers) $A+B$, and likewise morphisms $1_A \otimes 1_B$ to 1_{A+B}. From this, it follows directly that $\Psi(A\otimes B) = \Psi(A)\otimes\Psi(B)$. □

This approach is closely related to the theory of PROs, PROPs, and operads (see, *e.g.*, Leinster [13])–indeed, $\mathfrak{T}_\otimes$ is a PRO–but we will avoid relying on this theory for the sake of a more coherent presentation.

In order to define denotation and entailment in the red propositions, we need one last lemma, stating the obvious isomorphism between $\Psi^{-1}(1)$ (the subcategory of strings of objects of $\mathfrak{T}$ of length precisely 1) and $\mathfrak{T}$:

Lemma 2. *There exist functors* $I : \mathfrak{T} \to \Psi^{-1}(1)$ *and* $J : \Psi^{-1}(1) \to \mathfrak{T}$ *witnessing* $\Psi^{-1}(1) \cong \mathfrak{T}$*.*

Definition 6. *The denotation of a red proposition* $\varphi_R \in \Phi_R$*, denoted* $[\![\varphi_R]\!]$*, is given by the function* $[\![\cdot]\!] : \Phi_R \to (\mathfrak{T}_\otimes)_0$ *defined as follows:*

$$[\![\mathbf{e}]\!] = i \quad [\![A_R \circledast B_R]\!] = [\![A_R]\!] \otimes [\![B_R]\!] \quad [\![A_R]\!] = I([\![A_B]\!]) \quad \textit{if } A_R \textit{is a recolorable.}$$

As we did for blue propositions, we overload the denotation function to apply to (in this case, red) contexts as well, by defining the function $[\![\cdot]\!] : \Phi_R^* \to (\mathfrak{T}_\otimes)_0$ *as*

$$[\![\cdot]\!] = i \qquad\qquad [\![\Gamma_R, A_R]\!] = [\![\Gamma_R]\!] \otimes [\![A_R]\!]$$

[1] We use the notation $\Delta_\mathbb{N}$ for the discrete category specifically to avoid confusion with the ordinal category ω, which some authors denote $\mathbb{N}$.

Definition 7 (Red entailment). *Let Γ be a well-formed red context, and φ be a well-formed red proposition. We define red entailment by $\Gamma \vDash_R \varphi$ iff $[\![\Gamma]\!] \leq [\![\varphi]\!]$, i.e. iff there exists a morphism between the objects $[\![\Gamma]\!]$ and $[\![\varphi]\!]$ in $\mathfrak{T}_\otimes$.*

5 Metatheorems

With a semantics for both the blue and red fragments, we are ready to take on the fundamental metatheorems of soundness and completeness. The hierarchical structure of the proof calculi (and their semantics) gives a natural separation of work, as the soundness and completeness of the red fragment depends directly, via the recoloring rule, on the corresponding theorems for the blue fragment.

Theorem 4 (Soundness). *If $\Gamma \vdash_B \varphi$ then $\Gamma \vDash_B \varphi$; and if $\Gamma \vdash_R \varphi$ then $\Gamma \vDash_R \varphi$.*

Both parts follow straightforwardly by induction; the only interesting case is recoloring, which follows by Lemma 2 and soundness of the blue fragment. The completeness theorems require a little more work; blue completeness relies on the Lindenbaum-Tarski method (*i.e.*, by taking the set of blue propositions quotiented by blue provable equivalence, $\Phi_B/\dashv\vdash_B$, and showing that this is isomorphic to $\mathfrak{T}$), while red completeness uses the characterization of objects of $\mathfrak{T}_\otimes$ given by Theorem 3 as an induction principle for objects of $\mathfrak{T}_\otimes$.

Theorem 5 (Completeness). *If $\Gamma \vDash_B \varphi$ then $\Gamma \vdash_B \varphi$; and if $\Gamma \vDash_R \varphi$ then $\Gamma \vdash_R \varphi$.*

We are finally ready to tackle our previous obligation to show our propositional semantics equivalent to Toffoli's Boolean ring semantics. The first step is to show that Boolean rings are equivalent to Toffoli lattices:

Theorem 6 (Universality). *$\mathfrak{A}$ is a Toffoli lattice iff it is a Boolean ring.*

Proof (Proof sketch). If $\mathfrak{A}$ is a Toffoli lattice, we define the constants and operations of a ring by

$$0 = \bot \qquad 1 = \top \qquad a \cdot b = a \wedge b \qquad a \oplus b = a \leftrightharpoons \bar{b}$$

for all elements a and b of $\mathfrak{A}$. From this, it is straightforwardly shown that $\mathfrak{A}$ forms an abelian group under addition (with each a as its own additive inverse, and 0 as unit), and a monoid under multiplication (with 1 as unit) which further distributes over addition; thus it is a ring, and that it is Boolean follows directly by the idempotence of meets.

In the other direction, suppose $\mathfrak{A}$ is a Boolean ring; then it is also a Boolean algebra [19], so it suffices to show that a Boolean algebra is also a Toffoli lattice. But then we can construct relative equivalences by $a \leftrightharpoons b = (\bar{a} \vee b) \vee (\bar{b} \vee a)$ for all elements $a, b \in |\mathfrak{A}|$; that $\mathfrak{A}$ is then a Toffoli lattice follows by algebraic manipulation. □

We now extend this result to the full generality of entire reversible circuits. Let the order of a reversible circuit denote its number of input (equivalently output) lines; having the same order is thus a trivial requirement for two reversible circuits to be strongly equivalent, as the functions they compute (denote this function $f_{\mathcal{C}}$ for a circuit $\mathcal{C}$) will otherwise differ fundamentally by domain and codomain. Further, we will use $\mathfrak{B} = (\{0,1\}, 0, 1, \oplus, \cdot)$ to denote the Boolean ring on the set $\{0,1\}$ with exclusive disjunction as addition, and conjunction as multiplication, and $\mathfrak{B}^n$ to be the direct product of $\mathfrak{B}$ with itself n times. Using Toffoli's Boolean ring semantics (as presented in Sect. 2, Fig. 1a), we will develop a semantic preorder on reversible circuits – but to do this, we need a way to handle ancillae (lines of constant value) in a clean way. This is done by the ancilla restriction on a circuit, defined as follows:

Definition 8. *Let $\mathcal{C}$ be a reversible circuit of order n, and $\mathbf{x} \in |\mathfrak{B}^n|$. We define the* ancilla restriction *on $\mathbf{x}$ with respect to $\mathcal{C}$ to be $\mathbf{x}_{|\mathcal{C}} = (c_1, c_2, \dots, c_n)$ where each c_i is given by*

$$c_i = \begin{cases} k & \text{if the } i^{th} \text{ input of } \mathcal{C} \text{ is an ancilla of value } k \\ \pi_i(\mathbf{x}) & \text{otherwise} \end{cases}$$

This allows the following preorder on the set of reversible logic circuits, and in turn, the category induced by this preorder:

Lemma 3. *The relation on reversible circuits defined by $\mathcal{C} \leq_R \mathcal{D}$ iff $f_{\mathcal{C}}(\mathbf{x}_{|\mathcal{C}}) \leq f_{\mathcal{D}}(\mathbf{x}_{|\mathcal{D}})$ for all $\mathbf{x} \in |\mathfrak{B}^n|$ and circuits $\mathcal{C}, \mathcal{D}$ of equal order n, where the order relation $\cdot \leq \cdot$ denotes the usual (component-wise) ordering on Boolean vectors of length n, is a preorder.*

Definition 9. *Let* **RC** *denote the category which has reversible circuits as objects, and a single morphism between circuits $\mathcal{C}$ and $\mathcal{D}$ iff $\mathcal{C} \leq_R \mathcal{D}$.*

Note particularly from this definition that objects $\mathcal{C}$ and $\mathcal{D}$ of **RC** are isomorphic (i.e., $\mathcal{C} \leq_R \mathcal{D}$ and $\mathcal{D} \leq_R \mathcal{C}$) precisely when they are strongly equivalent. This allows us to show that all strong equivalences of reversible logic circuits are contained in $\mathfrak{T}_\otimes$:

Theorem 7 (Embedding of RC). *There exists a functor $H : \mathbf{RC} \to \mathfrak{T}_\otimes$ which constitutes an embedding of* **RC** *in $\mathfrak{T}_\otimes$, i.e., it is fully faithful; in particular $H(\mathcal{C}) \cong H(\mathcal{D})$ iff $\mathcal{C} \cong \mathcal{D}$.*

Proof. We define $H : \mathbf{RC} \to \mathfrak{T}_\otimes$ on objects by taking circuits to their annotation, as given by the annotation algorithm (see Sect. 2 and the example in Fig. 2), and on morphisms by taking $\mathcal{C} \leq \mathcal{D}$ to the morphism $H(\mathcal{C}) \leq H(\mathcal{D})$: That this morphism exists in $\mathfrak{T}$ follows by induction on the order of $\mathcal{C}$ (equivalently $\mathcal{D}$) by Theorem 6, since the order on the outputs is an order on Boolean ring terms, which are equivalent to Toffoli lattice terms via

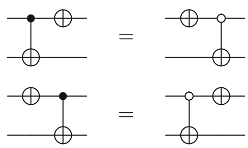
$$a \cdot b = a \wedge b \qquad a \oplus b = a \leftrightharpoons \overline{b} \qquad a \oplus 1 = a \leftrightharpoons \overline{\top} = a \leftrightharpoons \bot = \overline{a}$$

which shows, by soundness, completeness and the denotation of the propositional semantics, the exact correspondence between Toffoli's Boolean ring semantics and our propositional semantics (see Sect. 2, Figs. 1a and 1b). That $H(\mathcal{C}) \cong H(\mathcal{D})$ iff $\mathcal{C} \cong \mathcal{D}$ (equivalently, that H is fully faithful) follows likewise by induction on the order of $\mathcal{C}$ (equivalently $\mathcal{D}$) using Theorem 6. $\square$

6 Applications

Above, we have shown that the logic of $\mathrm{LRS}_{\circledast}$ is sound and complete with respect to a semantics that includes all strong equivalences of reversible logic circuits. This property suggests, as an obvious first application, a general method for proving such strong equivalences: Use the annotation algorithm of Sect. 2 to extract propositional representations of each circuit, and then use $\mathrm{LRS}_{\circledast}$ to show that their propositional representations are provably equivalent.

This approach can be applied directly in the optimization of reversible circuits. When used on very large circuits, the annotation algorithm may produce propositional representations that are infeasibly large to work with, however. Where the approach really shines is in the development and verification of template-based reversible circuit rewriting systems (see, *e.g.*, [1,20]). Template-based rewriting works by identifying certain forms of sub-circuits, allowing these to be substituted with equivalent ones.



Since such templates are typically quite modest in size, one can often extract corresponding propositions from templates with only a few steps of the annotation algorithm. A concrete example of such a template-based rewriting rule is Soeken and Thomsen's rule R2, shown on the right. Annotating these two circuits with our algorithm, the rule states precisely the equivalences

$$\neg x_1 \circledast x_1 \bullet\!- \neg x_2 \dashv\vdash_R \neg x_1 \circledast \neg\neg x_1 \bullet\!- \neg x_2 \tag{1}$$

and

$$\neg x_1 \circledast \neg x_1 \bullet\!- \neg x_2 \dashv\vdash_R \neg x_1 \circledast \neg x_1 \bullet\!- \neg x_2 \ . \tag{2}$$

which are both, indeed, provable. Note that (2) follows directly by red identity, as the annotation the two circuits resulted in syntactically identical propositions. One of the two derivations proving the (1) is shown in Fig. 5.

Using diagrammatic notation for such rewriting systems is both convenient and intuitive to use for humans. Although this has provided real insights into the rewriting behavior of reversible circuits, showing completeness (with respect to reversible circuits) for such rewriting systems has proven difficult.

Because $\mathrm{LRS}_{\circledast}$ provides sound and complete proof calculi for reasoning about reversible circuits, we can go the other way around and extract an equational theory from this that is sound and complete with respect to reversible circuits. Further, since the blue fragment of $\mathrm{LRS}_{\circledast}$ is sound and complete with respect to

$$\mathcal{D}_1 = \dfrac{\dfrac{\dfrac{}{x_1 \bullet\!\!- \neg x_2 \vdash_B x_1 \bullet\!\!- \neg x_2}\,(\text{BId})}{x_1 \bullet\!\!- \neg x_2, \neg\neg x_1 \vdash_B x_1 \bullet\!\!- \neg x_2}\,(\text{Wkn}) \qquad \dfrac{\dfrac{}{x_1 \bullet\!\!- \neg x_2, \neg\neg x_1 \vdash_B \neg\neg x_1}\,(\text{BId})}{x_1 \bullet\!\!- \neg x_2, \neg\neg x_1 \vdash_B x_1}\,(\neg\neg\text{E})}{x_1 \bullet\!\!- \neg x_2, \neg\neg x_1 \vdash_B \neg x_2}\,(\bullet\!\!-\text{E}_2)$$

$$\mathcal{D}_2 = \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{x_1 \bullet\!\!- \neg x_2 \vdash_B x_1 \bullet\!\!- \neg x_2}\,(\text{BId})}{x_1 \bullet\!\!- \neg x_2, \neg x_2 \vdash_B x_1 \bullet\!\!- \neg x_2}\,(\text{Wkn}) \qquad \dfrac{}{x_1 \bullet\!\!- \neg x_2, \neg x_2 \vdash_B \neg x_2}\,(\text{BId})}{x_1 \bullet\!\!- \neg x_2, \neg x_2 \vdash_B x_1}\,(\bullet\!\!-\text{E}_1)}{x_1 \bullet\!\!- \neg x_2, \neg x_2, \neg x_1 \vdash_B x_1}\,(\text{Wkn}) \qquad \dfrac{}{x_1 \bullet\!\!- \neg x_2, \neg x_2, \neg x_1 \vdash_B \neg x_1}\,(\text{BId})}{x_1 \bullet\!\!- \neg x_2, \neg x_2, \neg x_1 \vdash_B \mathbf{0}}\,(\neg\text{E})}{x_1 \bullet\!\!- \neg x_2, \neg x_2 \vdash_B \neg\neg x_1}\,(\neg\text{I})$$

$$\mathcal{D}_0 = \dfrac{\dfrac{}{\neg x_1 \vdash_R \neg x_1}\,(\text{RId}) \qquad \dfrac{\dfrac{\dfrac{\mathcal{D}_1}{x_1 \bullet\!\!- \neg x_2, \neg\neg x_1 \vdash_B \neg x_2} \qquad \dfrac{\mathcal{D}_2}{x_1 \bullet\!\!- \neg x_2, \neg x_2 \vdash_B \neg\neg x_1}}{x_1 \bullet\!\!- \neg x_2 \vdash_B \neg\neg x_1 \bullet\!\!- \neg x_2}\,(\bullet\!\!-\text{I})}{x_1 \bullet\!\!- \neg x_2 \vdash_R \neg\neg x_1 \bullet\!\!- \neg x_2}\,(\text{RCl})}{\neg x_1, x_1 \bullet\!\!- \neg x_2 \vdash_R \neg x_1 \circledast \neg\neg x_1 \bullet\!\!- \neg x_2}\,(\circledast\text{I})$$

$$\dfrac{\dfrac{}{\neg x_1 \circledast x_1 \bullet\!\!- \neg x_2 \vdash_R \neg x_1 \circledast x_1 \bullet\!\!- \neg x_2}\,(\text{RId}) \qquad \dfrac{\mathcal{D}_0}{\neg x_1, x_1 \bullet\!\!- \neg x_2 \vdash_R \neg x_1 \circledast \neg\neg x_1 \bullet\!\!- \neg x_2}}{\neg x_1 \circledast x_1 \bullet\!\!- \neg x_2 \vdash_R \neg x_1 \circledast \neg\neg x_1 \bullet\!\!- \neg x_2}\,(\circledast\text{E})$$

Fig. 5. Derivation in $\text{LRS}_\circledast$ for verifying the first direction of Soeken and Thomsen's rule R2.

Toffoli lattices, we can instead extract an equational theory for the blue fragment from the definition of a Toffoli lattice, using the following lemma:

Lemma 4. *In any Toffoli lattice, $a \leq b$ iff $a \wedge \overline{b} = \bot$.*

This lemma allows us to straightforwardly recast the definition of a Toffoli lattice in purely equational terms (although the result is not exactly elegant). What this *does* give us, is a set of equations that must hold for all Toffoli lattices, and which any other complete equational theory must therefore be equivalent to, and the means to show such an equivalence by converting equalities to statements about the underlying order structure and vice versa.

Figure 6 shows a more pleasing equational theory for the blue fragment, presented in the syntax of $\text{LRS}_\circledast$ (the intrinsic properties of equality, *i.e.*, reflexivity, symmetry, transitivity, and congruences are not shown,) proven equivalent (and therefore sound and complete) exactly in the way outlined above (by the power of boring algebra). Deriving an equational theory for the red fragment is simpler, as it is sound and complete with respect to the free monoidal part of $\mathfrak{T}_\otimes$, which is already expressed in equational terms. As such, the equational theory for the red fragment shown in Fig. 6 is sound and complete by definition, though congruences applied in the red fragment are syntactically restricted by recolorability; that is, we can only replace recolorable propositions by recolorable propositions.

The usefulness of such an equational theory is evident in that we can, *e.g.*, now prove the soundness of the R2 rules directly by applying equation (B9) in Fig. 6. Such equational theories can themselves also be used to develop new rewriting systems for reversible circuits, in particular to suggest new templates.

$$\varphi \circledast (\psi \circledast \chi) \overset{(R1)}{=} (\varphi \circledast \psi) \circledast \chi \qquad (\varphi \mathbin{\&} \psi) \mathbin{\&} \chi \overset{(B4)}{=} \varphi \mathbin{\&} (\psi \mathbin{\&} \chi)$$

$$\varphi \circledast \mathbf{e} \overset{(R2)}{=} \varphi \qquad \varphi \bullet\!\!- \psi \overset{(B5)}{=} \psi \bullet\!\!- \varphi$$

$$\mathbf{e} \circledast \varphi \overset{(R3)}{=} \varphi \qquad (\varphi \bullet\!\!- \psi) \bullet\!\!- \chi \overset{(B6)}{=} \varphi \bullet\!\!- (\psi \bullet\!\!- \chi)$$

$$\varphi \mathbin{\&} \neg(\psi \mathbin{\&} \chi) \overset{(B7)}{=} \neg(\neg(\varphi \mathbin{\&} \neg\psi) \mathbin{\&} \neg(\varphi \mathbin{\&} \neg\chi))$$

$$\varphi \mathbin{\&} \mathbf{1} \overset{(B1)}{=} \varphi \qquad \varphi \bullet\!\!- \psi \overset{(B8)}{=} \neg(\varphi \mathbin{\&} \neg\psi) \mathbin{\&} \neg(\psi \mathbin{\&} \neg\varphi)$$

$$\varphi \mathbin{\&} \mathbf{0} \overset{(B2)}{=} \mathbf{0} \qquad \neg\neg\varphi \overset{(B9)}{=} \varphi$$

$$\varphi \mathbin{\&} \psi \overset{(B3)}{=} \psi \mathbin{\&} \varphi \qquad \varphi \mathbin{\&} \neg\varphi \overset{(B10)}{=} \mathbf{0}$$

Fig. 6. Sound and complete equational theories for the two calculi. (Color figure online)

7 Conclusion and Future Work

In this article, we have presented a syntactic representation of reversible logic circuits centered around the control interpretation of Toffoli's reversible gate set, and shown, via two proof calculi of natural deduction, that a variant of classical propositional logic extended with ordered multiplicative conjunction is sufficient to reason about these. We have developed an algebraic and categorical semantics, shown that the proof calculi are sound and complete with respect to these, and that this model subsumes the established notion of strong equivalence of reversible logic circuits. Finally, we have shown how our work can be used to prove strong equivalences of reversible logic circuits, to verify existing systems of reversible logic circuit rewriting, and to develop new such rewriting systems.

The approach has been successful in enabling reasoning about reversible logic circuits, but it is not quite on even footing with the template-based approaches to reversible circuit rewriting, as these use a graphical circuit notation which, by definition, asserts circuit reversibility on every rewriting step. Although our approach faithfully models circuit semantics, it is not currently clear when looking at an arbitrary proposition whether it corresponds to a reversible circuit or not. On the other hand, by decoupling the propositions from the graphical representations, the current logic may allow for much shorter rewritings than if each step must yield representations which directly translate to circuits in this way.

References

1. Arabzadeh, M., Saeedi, M., Zamani, M.S.: Rule-based optimization of reversible circuits. In: Proceedings of the ASP-DAC 2010, pp. 849–854. IEEE (2010)
2. Bérut, A., Arakelyan, A., Petrosyan, A., Ciliberto, S., Dillenschneider, R., Lutz, E.: Experimental verification of Landauer's principle linking information and thermodynamics. Nature **483**(7388), 187–189 (2012)
3. Buss, S.R.: Handbook of Proof Theory. Elsevier, Amsterdam (1998)
4. De Vos, A.: Reversible Computing. Wiley-VCH, Weinheim (2010)

5. De Vos, A., Burignat, S., Glück, R., Mogensen, T.Æ., Axelsen, H.B., Thomsen, M.K., Rotenberg, E., Yokoyama, T.: Designing garbage-free reversible implementations of the integer cosine transform. ACM J. Emerg. Tech. Com. **11**(2), 11:1–11:15 (2014)
6. Feynman, R.P.: Quantum mechanical computers. Found. Phys. **16**(6), 507–531 (1986)
7. Jacobs, B.: Categorical Logic and Type Theory. Elsevier, Amsterdam (1999)
8. Jordan, S.P.: Strong equivalence of reversible circuits is coNP-complete. Quantum Inf. Comput. **14**(15–16), 1302–1307 (2014)
9. Joyal, A., Street, R.: The geometry of tensor calculus I. Adv. Math. **88**(1), 55–112 (1991)
10. Joyal, A., Street, R.: Braided tensor categories. Adv. Math. **102**(1), 20–78 (1993)
11. Kaarsgaard, R.: Towards a propositional logic for reversible logic circuits. In: Proceedings of the ESSLLI 2014 Student Session, pp. 33–41 (2014). http://www.kr.tuwien.ac.at/drm/dehaan/stus2014/proceedings.pdf
12. Landauer, R.: Irreversibility and heat generation in the computing process. IBM J. Res. Dev. **5**(3), 261–269 (1961)
13. Leinster, T.: Higher Operads, Higher Categories. London Mathematical Society Lecture Note Series, vol. 298. Cambridge University Press, Cambridge (2004)
14. Orlov, A.O., Lent, C.S., Thorpe, C.C., Boechler, G.P., Snider, G.L.: Experimental test of Landauer's principle at the sub-$k_{\mathrm{b}}t$ level. Japan. J. Appl. Phys. **51**, 06FE10 (2012)
15. Polakow, J.: Ordered linear logic and applications. Ph.D. thesis, CMU (2001)
16. Rendel, T., Ostermann, K.: Invertible syntax descriptions: unifying parsing and pretty printing. ACM SIGPLAN Notices, vol. 45, No. 11, pp. 1–12 (2010)
17. Schellekens, M.P.: MOQA: unlocking the potential of compositional static average-case analysis. J. Log. Algebr. Program. **79**(1), 61–83 (2010)
18. Seely, R.A.G.: Linear logic, *-autonomous categories and cofree coalgebras. Contemp. Math. **92**, 371–382 (1989)
19. Sikorski, R.: Boolean Algebras. Springer, Heidelberg (1969)
20. Soeken, M., Thomsen, M.K.: White dots *do* matter: rewriting reversible logic circuits. In: Dueck, G.W., Miller, D.M. (eds.) RC 2013. LNCS, vol. 7948, pp. 196–208. Springer, Heidelberg (2013)
21. Thomsen, M.K., Glück, R., Axelsen, H.B.: Reversible arithmetic logic unit for quantum arithmetic. J. Phys. A Math. Theor. **43**(38), 382002 (2010)
22. Toffoli, T.: Reversible computing. In: de Bakker, J., van Leeuwen, J. (eds.) Automata, Languages and Programming. LNCS, vol. 85, pp. 632–644. Springer, Heidelberg (1980)
23. Wille, R., Drechsler, R.: Towards a Design Flow for Reversible Logic. Springer, Heidelberg (2010)

Foundations of Mathematics: Reliability and Clarity: The Explanatory Role of Mathematical Induction

John T. Baldwin(✉)

University of Illinois at Chicago, Chicago, USA
jbaldwin@uic.edu

Abstract. While studies in the philosophy of mathematics often emphasize reliability over clarity, much study of the explanatory power of proof errs in the other direction. We argue that Hanna's distinction between 'formal' and 'acceptable' proof misunderstands the role of proof in Hilbert's program. That program explicitly seeks the existence of a justification; the notion of proof is not intended to represent the notion of a 'good' proof. In particular, the studies reviewed here of mathematical induction miss the explanatory heart of such a proof; how to proceed from suggestive example to universal rule. We discuss the role of algebra in attaining the goal of generalizability and abstractness often taken as keys to being explanatory. In examining several proofs of the closed form for the sum of the first n natural numbers, we expose the hidden inductive definitions in the 'immediate arguments' such as Gauss's proof. This connection with inductive definition leads to applications far beyond verifying numerical identities. We discuss some objections, which we find more basic than those in the literature, to Lange's general argument that proofs by mathematical induction are not explanatory. We conclude by arguing that whether a proof is explanatory depends on a context of clear hypothesis and understanding what is supposedly explained to who.

Lange [Lan09] describes a striking disagreement among philosophers concerning the explanatory power of mathematical induction, 'Some philosophers are quite confident that these arguments are generally explanatory, even in the face of other philosophers who appear equally confident of their intuition to the contrary. Very little in the way of argument is offered for either view.' We argue that the failure to see the explanatory nature of the usual inductive proofs is in fact a misunderstanding of what it is that is explained. We isolate the explanatory feature of specific inductive proofs and indicate why apparently simpler 'proofs' are incomplete.

The contrast raised by many philosophers, e.g., [Han90, Lan09, Man08, RK87] between explanatory and non-explanatory proofs can be phrased as 'A non-explanatory proof merely shows the result is 'true' while an explanatory proof provokes understanding of why it is 'true'.' I put scare quotes around 'true' as in fact proofs do not show truth. They show that a result is a consequence of the hypotheses. A major difficulty with this literature concerning induction is

J. Väänänen et al. (Eds.): WoLLIC 2016, LNCS 9803, pp. 68–82, 2016.
DOI: 10.1007/978-3-662-52921-8_5

that much of the analysis ignores a crucial criterion that Hanna [Han90] makes explicit (and then ignores), 'the proof must proceed from specific and accepted hypotheses'. In particular cases, making the implicit hypotheses explicit illuminates what the proof is supposed to explain.

In Sect. 1 we observe that the traditional emphasis on *the foundation* of mathematics leads to a misunderstanding of proof as simply a matter of 'reliable inference' and misses explanatory motivations in mathematics. In Sect. 2, we show how this distortion of the role of formal proof is reflected in a misapprehension of the goals of even basic proofs involving mathematical induction. Section 3 underlines the symbiosis between inductive definition and inductive proof that is implicit in many if not most proofs by induction. Complementing this discussion of how inductive proofs are explanatory, in Sect. 4 we expound some fundamental objections to Lange's assertion that inductive proofs are inherently non-explanatory. We return in Sect. 5 to our underlying theme: analyzing proofs solely as a way to verify truth obscures their explanatory nature. This tendency is amplified by a misunderstanding of Hilbert's proof theory; it aims to analyze not *proof*, but *provability*.

We critique several attempts to give a general characterization of mathematical explanation by examining how they fare in studying mathematical practice. We don't attempt to produce a positive theory but obey the injunction that concludes the Hafner and Mancosu article [HM05], 'It is our hope that this kind of testing of theories of mathematical explanation against mathematical practice will pave the way for future studies in the same vein. This seems to us the most promising approach for making progress in this treacherous area.'

I thank the referees for a number of helpful and incisive comments.

1 Reliability vrs Clarity

Much study of the foundations of mathematics focuses on the issue of finding a ground for all of mathematics and on issues of ontology and reliability. But, in fact, many of the 19th century foundational studies were concerned with the clarification of concepts and the relations between them. In [Bal14, Bal15] I expound the role of formal theories in clarifying notions and making mathematical progress. Here we focus on the relation between clarity and reliability in the context of mathematical induction and (generalized) inductive definition. As Coffa describes below, many histories of foundational studies have emphasized reliability over clarity. Perhaps in reaction, many discussions of mathematical induction have sacrificed the reliability aspect in favor of an explanation of one phenomena: why is this particular closed form of some expression hypothesized? But this analysis ignores the phenomena that inductive proof is designed to explain: the passage from example to universal. Regarding foundations, Coffa reported the history,

> [We consider] the sense and purpose of foundationalist or reductionist projects such as the reduction of mathematics to arithmetic or arithmetic

> to logic. It is widely thought that the principles inspiring such reconstructive efforts were basically a search for certainty. This is a serious error. It is true, of course, that most of those engaging in these projects believed in the possibility of achieving something in the neighborhood of Cartesian certainty for principles of logic or arithmetic on which a priori knowledge was to based. But it would be a gross misunderstanding to see in this belief the basic aim of the enterprise. A no less important purpose was the clarification of what was being said ...
> The search for rigor might be, and often was, a search for certainty, for an unshakable "Grund". But it was also a search for a clear account of the basic notions of a discipline.[1]

We argue below that in the study of the explanatory power of mathematical induction there are several issues to be considered. And some of them might be dismissed as 'mere reliability'. But we insist that explanation is a fundamental goal of mathematics. We focus on mathematical induction as there is a substantial literature on its purported non-explanatory value.

Resnik and Kushner remark, 'Mathematicians rarely describe themselves as explaining' (page 151 of [RK87]). As a mathematician, I can only explain such a remark as a lack of exposure to mathematicians[2]. Perhaps the difficulty is that the notions being explained are abstruse. I give from a popular source (wikipedia) the first example that popped into my head.

> In mathematics, monstrous moonshine, or moonshine theory, is a term devised by John Conway and Simon P. Norton in 1979, used to describe the unexpected connection between the monster group M and modular functions, in particular, the j function. It is now known that lying behind monstrous moonshine is a certain conformal field theory having the monster group as symmetries. The conjectures made by Conway and Norton were proved by Richard Borcherds in 1992 using the no-ghost theorem from string theory and the theory of vertex operator algebras and generalized KacMoody algebras [Ano16].

Many why-questions in mathematics arise from exploring unexpected connections across widely divergent areas. In this example, Conway and Norton observed certain strange sequences of numbers (1,196884, 21493760, ...) that arose in finite group theory also arose in complex function theory. Physics and functional analysis were involved in explaining this non-coincidence. One part of the solution involved collaboration between two finite group theorists (Paul Fong, Steve Smith) at University of Illinois in Chicago and our colleague A.O.L. Atkin. As a pioneer in using the computer for number theoretic calculations[3] and an expert in modular forms, Atkin instantly recognized these coefficients.

[1] Page 26 of [Cof91].

[2] Section 3 of [HM05] give several specific examples of mathematicians using 'explain' in various senses.

[3] He and another U.I.C. mathematician, Neil Rickert, once held the record for the largest pair of twin primes.

The Langlands program is an even bigger example of an explanatory project that crosses many fields to explain certain analogies.

Of course, such major projects as described in the last paragraph can not be analyzed in a short paper or with my (lack of) expertise. In a series of vignettes, 'How to explain number theory at a dinner party', Harris ([Har15], page 51) presents a more concrete example. His foil says 'a number theorist sits at a desk and answers questions about numbers all day'. He replies,

> Actually, number theorists are not especially interested in answering questions about numbers. We really get excited when we notice that answers seem to be coming out in a certain way, and then we try to explain why that is. For example, the equation $x^4 14x^2 + 121 = 0$ – the question, what number solves that equation? – has not one but four answers: $\sqrt{2} + 3i$, $\sqrt{2} - 3i$, $-\sqrt{2} + 3i$, $-\sqrt{2} - 3i$. There's a pattern: you can permute $\sqrt{2}$ with $-\sqrt{2}$ and $3i$ with $-3i$. What does it mean? What does it tell us about solutions to other equation?
> When our ideas about possible explanations are sufficiently clear, we set ourselves the goal of finding the correct explanation and then justifying it.

2 The Explanatory Function of Mathematical Induction

Hanna [Han90], distinguishes between formal and acceptable proof. Although, we discuss difficulties with her version of each notion in Sect. 5, this divide is useful for understanding the situation.

1. Formal proof: proof as a theoretical concept in formal logic (or metalogic), which may be thought of as the ideal which actual mathematical practice only approximates.
2. Acceptable proof: proof as a normative concept that defines what is acceptable to qualified mathematicians.

Although this distinction is irrelevant to much of her analysis, our fifth proof below uses the notion of formal proof. Her argument rather depends on a dichotomy between a *proof that proves* (which is an alias for acceptable) versus *a proof that explains*. She writes (page 10 of [Han90]), 'I prefer to use the term *explain* only when the proof reveals and uses the mathematical ideas that motivate it.' A difficulty in her analysis is that while she defines 'acceptable proof' in terms of 'qualified mathematician', she omits discussion of the 'qualified' mathematician's understanding of the hypotheses of the theorem – this understanding crucially impacts the explanatory nature of the proof.

I take formal/acceptable to be the same distinction I make in [Bal13] between *Hilbert-Gödel-Tarski* and *Euclid-Hilbert* proof. The first requires the definition of a formal syntax and rules of inference and Tarski's name is adjoined to consider semantics. The second takes place in natural language. While there are specified definitions and axioms, the rules of inference may be implicit. With this background we consider some examples.

Hanna's examples are from [Ste78]; I address Steiner's more sophisticated approach to explanation later. Hanna writes[4]:

> The following example illustrates the difference between a proof that proves (acceptable) and proof that explains (explanatory):
> Prove that the sum of the first n positive integers, $S(n)$, is equal to $\frac{n(n+1)}{2}$ (page 10 of [Han90]).

There are two issues that have to be explained here.

1. Why do we choose the formula $\frac{n(n+1)}{2}$?
2. Why does our observation that this formula works on some examples extend to all natural numbers?

Modern attempts to answer the second question date from Maurolycus in 1575 [Bus17], followed quickly by Pascal. But Maurolycus simply argues for a few cases that the property extends from n to $n+1$. Cajori [Caj18] attributes the explicit step of proving for each k, $P(k) \rightarrow P(k+1)$ to Jakob Bernoulli in 1713. Dedekind announces in his introduction (page 32 of [Ded63]) that a major result of his essay on number is 'a complete proof that the form of argument known as complete induction (or the inference from n to $n+1$) is really conclusive.' The modern formal version appears in Peano. Dedekind and Peano thus address an even deeper why question, why does this 'rule of complete induction', gradually clarified until the late 19th century, actually justify the passage from example to universal.

We consider five alternative 'proofs' and see how they answer each of these questions. We try to extract from the argument what the writer of the proof is actually taking as the hypotheses.

Gauss 1. Hanna rates the standard proof by induction as acceptable but not explanatory and the Gauss argument as both explanatory and acceptable. The Gauss argument is the following.

$$\begin{array}{ccccc} 1 & + & 2 & + \dots & \text{n} \\ \text{n} & + & \text{(n-1)} & + \dots & 1 \\ \hline \text{n+1} & + & \text{n+1} & + \dots & \text{n+1} \end{array}$$

Now there are n addition problems that each add to $n+1$ and each number up to n occurs twice as a summand so the sum of the numbers up to n is $\frac{n(n+1)}{2}$.

I argue below that by Hanna's criteria the standard inductive proof is explanatory of (2) but not (1) and is acceptable in terms of verifying truth. While the hypotheses are not stated, when they are made clear, the argument can clearly be carried out in arithmetic with induction.

And I say that 'Gauss's' proof as given is explanatory of (1) but not (2). It is not acceptable because the expression $1 + 2 \dots n$ is not defined (and is not easy

[4] I added the parenthetical descriptions. And, I modified the question because as originally phrased, the first question is not asked. But the explanations proffered by Hanna all deal with it.

to define). Remember, addition is a binary function. But $1 + 2 \ldots + n$ is what is variously called *anadic*, *variadic* addition or an example of plural quantification. It is not part of basic mathematical notation because no arity is prescribed. The function must be introduced by some form of inductive definition[5]. Such functions are implemented in many (e.g. list processing) programming languages.

To make Gauss' argument into a proof we have to clarify our estimate of what the 'qualified mathematician' is assuming. Here is one approach[6]. Work in informal set theory and for fixed n and $k < n$, define $f(k) = (n + 1) - k$. Then for each k with $0 \leq k \leq n$, $k + f(k) = n + 1$. So the sum of the first n numbers is

$$\frac{(1 + f(1)) + (2 + f(2)) + \ldots (n + f(n))}{2}$$

since the numerator of this expression contains each number up to n twice. Now there are certainly n such k since the domain of f is n and by the (generalized) distributive law the sum is $\frac{n(n+1)}{2}$.

Note that the function $f(n)$ is usually taken as a definition of $\Sigma_{k=1}^{n} k$. So we see a way to fill the gap and produce an acceptable proof is to assume the generalized algebraic laws (i.e. distribution, associativity and commutativity over arbitrary finite sums) AND that $\Sigma_{k=1}^{n} k$ is well-defined. This is a hidden induction. But as Sally's proof (below) shows, these additional assumptions represent a 'qualified mathematician's hypotheses'. Of course the generalized algebraic laws work as well for any ring (a proof by induction on the number of addends); but the definition of $\Sigma_{k=1}^{n} k$ does *not*. The ring might not be ordered, let alone well-ordered.

Gauss 2. We now rephrase Gauss's proof in an even more concrete form[7]. A standard problem for introducing the problem of finding a closed form for $\Sigma_{i=1}^{n} i$ is the 'handshake problem'. There are $n + 1$ people at a party and each shakes hands with each other person. How many handshakes are there. One analysis is, 'Each of the $n + 1$ people shakes hands with n others so there are $n(n + 1)$ handshakes. Whoops! I counted each handshake twice. So there are $\frac{n(n+1)}{2}$ handshakes'. But another way to count says the first person shakes hands with n others, avoiding repetitions the second shakes hand with $n - 1$, the third with $n - 2$. So the sum is

$$\Sigma_{i=1}^{n}(n - i) = \Sigma_{i=1}^{n} i.$$

Since both calculations give us the number of handshakes, the values are equal.

Here the two steps are actually quite separate. The calculation of the number of handshakes does not depend on induction. But then another method of

---

[5] There are various logics for studying this topic [Lin14]; but they are not considered in the papers under discussion.

[6] In the paper Hanna draws on, Steiner [Ste78] refers to Quine's set theory book [Qui69].

[7] While this statement is appealing to students, a more formal version with the same proof is: How many edges are there in the complete symmetric graph on n vertices?.

calculation is introduced. The first calculation has no trace of induction[8], although it certainly relies on the connections between the natural numbers and actual counting. The second certainly uses the inductive definition of the Σ notation. There is an apparent use of generalized associativity and commutativity, but not distributivity.

The Standard Proof. We notice that $1+2=\frac{2(3)}{2}$ and check this for a few more small numbers. Now the mantra says, show $(\forall n)P(n) \rightarrow P(n+1)$. So one asks what is $\frac{n(n+1)}{2}+(n+1)$? With a common denominator of 2, we have[9]

$$\frac{n(n+1)+2n+2}{2}=\frac{n^2+3n+2}{2}=\frac{(n+2)(n+1)}{2}.$$

But why is this explanatory of the formula holding for *all* n? *This* proof translates the basic intuition; I can transform my calculation for 1 into a calculation for 2 into a calculation for 3 ... into a single step by the power of algebra. That is, the calculation with the variable n which can interpreted as any number. So stepping through the numbers will establish the result for all natural numbers. The procedure is not meaningless. The proof scheme is a distilled hint of how to understand why the proposition is true for all n.

The choice of the particular formula $\frac{n(n+1)}{2}$ may remain a mystery; but adding any of the many geometric pictures for motivating this step can complete the explanation of the result.

The general aim of inductive proofs is to move from observing $P(n)$ for a few n to the statement $\forall n P(n)$. The key contribution of the axiom/rule of mathematical induction is to reduce the intuitive iterative calculation to a finite statement $\forall k[P(k) \rightarrow P(k+1)$. This step is amenable to 'algebraic proof'.

Sally's Proof: Systematic Generalization. Paul Sally's sequence of proofs[10], which follow using both the generalized algebraic laws and the Σ notation, show the generalizing power of algebraic methods. Consider the equation:

$$(n+1)^2-1=\Sigma_{k=1}^{n}((k+1)^2-k^2). \tag{1}$$

Notice that the right hand side telescopes. (The subtracted term in one summand is the positive term in the previous summand.) So the right hand side

[8] This assertion of course depends on where I begin my arithmetic. There is no trace of arithmetic, if my assumption is that $(N,+,\times)$ is a semi-ring (ring without additive inverse). But there is if I go back one step further and define multiplication inductively from addition.

[9] There is also a geometrical picture to understand the algebra in the numerator of this calculation. Represent $n(n+1)$ by an n high by $n+1$ wide rectangle. Then to add $2(n+1)$, place two 1 by n strips on top of the rectangle.

[10] Paul Sally presented this argument at a University of Chicago class for high school teachers on Aug. 3/4, 2012. Doubtless, the approach is old; the use of telescoping series dates at least to the Bernoulli's, Euler and Goldbach [BVP06]. Sally was not only a distinguished researcher in p-adic analysis and representation theory but a national leader in Mathematics Education.

simplifies to the left and the equality is true. We can easily give a geometric motivation for this formula. Write a square that is $n+1$ units on a side as a union 1×1 square in the lower left hand corner, then a 2×2 with the same lower left corner, etc. Then note the difference between the successive small squares gives a disjoint cover of the large square.

Note that the *kth* summand on the right side of Eq. 1 simplifies to $2k+1$ so the right hand side *en toto* simplifies to $n + 2\Sigma_{k=1}^{n} k$. So we have

$$(n+1)^2 - 1 = n + 2\Sigma_{k=1}^{n} k$$

and a little algebra gives

$$\Sigma_{k=1}^{n} k = \frac{n(n+1)}{2}.$$

But this approach yields more. The same telescoping argument shows

$$(n+1)^3 - 1 = \Sigma_{k=1}^{n}((k+1)^3 - k^3).$$

Again we analyze the right hand side. Each summand is $k^3 + 3k^2 + 3k + 1 - k^3$ which equals $3k^2 + 3k + 1$. So we have

$$(n+1)^3 - 1 = 3(\Sigma_{k=1}^{n} k^2 + \Sigma_{k=1}^{n} k) + n.$$

Using the formula for the sum of the first n positive integers and moving all but the first term on the right to the left, $3(\Sigma_{k=1}^{n} k^2)$ equals $n^3 + 3n^2 + 3n + 1 - 1 - n - 3(\frac{n(n+1)}{2})$. Now again a little, very basic, algebra gives

$$\Sigma_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}.$$

Let us emphasize what is explanatory about this proof. If we think of the formulas for the first n squares, cubes, 4th powers etc. as a sequence of distinct problems that require a new inspiration for the formulas for each larger power, they are a great mystery. But this proof provides a unified strategy for attacking all the problems. There isn't (here) a geometric picture for the formulas after the first couple of dimensions but we have (implicitly) a procedure for generating each formula. The telescoping procedure can be visualized intuitively in exactly the same way as the Gauss argument.

What are the hypotheses of this proof? The algebra of polynomials is assumed (including generalized associativity and generalized distributivity which must be established by a separate induction) and the whole argument is an induction on the power of k in the sequence being summed.

Assertions of the non-explanatory nature of the standard inductive proof seem to be based on the idea that the algebraic manipulations in going from $P(n)$ to $P(n+1)$ are inherently non-explanatory. The glory of algebra is that one does not need (and possibly cannot) keep track of the meaning of each term in a derivation; nevertheless, the variables have the same interpretation at the end of the derivation as the beginning. The thought seems to be that losing track of the explicit reference of each term means the argument is non-explanatory but mere calculation. We have just seen the fallacy of this assertion in Sally's methods of generalization.

Steiner [Ste78] quotes Kreisel and Feferman as suggesting that the more explanatory proof is the more general and/or abstract. He gives three more specific versions of this assertion; I quote only the third.

> (c) Of two proofs of the same theorem the more explanatory is the more abstract (or general).
> Kreisel explicitly adopts (c) – in a private communication –writing that "familiar axiomatic analysis in terms of the greater generality of (the theorems occurring in) one proof than (in) the other" is 'sufficient' to distinguish between proofs' explanatory value (page 136 of [Ste78]).

Steiner rejects this view arguing (page 144 of [Ste78]) 'It is not, then, the general proof which explains; it is the generalizable proof.' This description fits Sally's proof well. Steiner refines this notions further and writes 'an explanatory proof depends on a characterizing property of something mentioned in the theorem: if we 'deform' the proof, substituting the characterizing property of a related entity, we get a related theorem.' Unfortunately this notion of characterizing property is elusive as demonstrated by the examples in [HM05]. But in this particular case, it seems we can identify the characterizing property as the representation of $(n+1)^2 - 1$ as a telescoping series. The tool of telescoping series is exploited.

The Generality of a Formal Proof. Recall the hypotheses (Gauss 1) of my first explanation of writing a correct proof in naive set theory. Equally well it could be thought of as a proof in second order arithmetic. The definition of n-adic addition is in fact made by a recursive definition (in the technical sense). Both that definition and the generalized algebraic laws can formalized in first order Peano arithmetic. So, writing PA^1 for first order Peano arithmetic, we have a theorem:

$$PA^1 \vdash \Sigma_{k=1}^{n} k = \frac{n(n+1)}{2}.$$

Now from the standpoint of Hanna this seems like a hair-splitting analysis of the hypotheses. What it actually shows is that this formula holds not just for the Natural Numbers but for any model of first order Peano arithmetic. It is a vast generalization. Even more the same analysis applies to Sally's argument.

We have presented five 'proofs' of the formula for the sum of the first n numbers. We see that in order to assess the explanatory value of each we must carefully stipulate the hypotheses. We see that one can avoid an actual induction, if one assumes the generalized algebraic rules and the definition of the Σ symbol. But this just hides the explanatory core of the solution.

3 Inductive Definition and Inductive Proof

A mathematical induction almost always is a response to an inductive definition. The motivating definition is often hidden from view as in the case of the examples from arithmetic analyzed above, where the generalized algebraic sum and anadic addition are often taken as just part of 'general mathematical knowledge'. But it is much more evident in proofs that involve algebraic constructions.

Generalized inductive definition[11] is required to construct such objects as the closure of a set to a subgroup or in a logic, the set of formulas in logic or theorems of a theory.

Hafner and Mancosu [HM05] criticize the Resnik and Kushner [RK87] assertion that Henkin's proof of the completeness theorem is explanatory, asking 'what the explanatory features of this proof are supposed to consist of'. Here is an answer. The relevant form of the completeness theorem asserts, 'every syntactically consistent first order theory has a model.' We sketch the Henkin proof[12]: T has the *witness property* if for every formula $\phi(x)$, there is a *witness constant* c_ϕ such that

$$T \vdash (\exists x)\phi(x) \to \phi(c_\phi).$$

Now the proof has two steps: (1) every syntactically consistent theory can be extended to a complete theory with the witness property (2) every complete syntactically consistent theory with the witness property has a model.

Both steps are explanatory. The extension of an arbitrary consistent T to one satisfying the witnessing property depends precisely on the axioms and rules of inference of the logic. Extending to a complete theory is often done by Zorn's lemma. If one finds this method unexplanatory, it can be done by an inductive construction adding ϕ_α or $\neg\phi_\alpha$ at stage α; the goal is to ensure that each sentence is decided. To construct the model, consider the set of witnesses, M and show that after modding out by the equivalence relation cEd if and only $T \vdash c = d$, the structure $M' = M/E$ satisfies T. More precisely, show by induction on formulas that for any formula $\psi(\mathbf{c})$,

$$T \vdash \phi(\mathbf{c}) \text{ if and only if } M' \models \phi(\mathbf{c}).$$

Note that here the formulas and theorems of T arise by a generalized inductive definition; so we must induct on the structure of the formulas to complete the proof.

This induction shows exactly how the new structure arises from the syntactic data; in contrast, Gödel reduces the first order case to propositional logic. Further, Steiner's deformation is realized in the many variants of this argument. In particular, the first step of the proof will have minor variants depending on which deductive system is chosen. But the second stage will be the same. Moreover, completeness for many other logics (type theory, infinitary, modal, 2nd order, etc.), the omitting types theorem, and a long list of more technical

[11] A set X is defined by generalized inductive definition if there is rule assigning to each finite subset X_0 of X some larger set X_0' (the closure of X_0) and for each $X_0 \subset X$, $X_0' \subset X$. This notion is given a more inductive format if one starts with a set Y and successively closes it to obtain X [Sho67].

[12] We take the version form [Mar02] but similar accounts can be found in any modern logic text.

results in model theory all derive from this method of model construction. A later adaptation of the idea of constructing an actual algebraic object from the syntactic description is applied in a proof of the Hilbert Nullstellensatz (page 89 of [Mar02]). Thus, here we can exhibit Steiner's characterizing property as this uniformly defined transfer from a collection of sentences in a formal language to a mathematical structure.

4 Objections to Lange's Argument that Proofs by Mathematical Induction are not Explanatory

Lange [Lan09] presents an argument that proofs by induction are generally not explanatory. I have already described how these are arguments are explanatory. Here I try to identify some of the flaws in Lange's analysis. I generally agree with specific objections[13] to his argument raised by various authors [Lan, Bak10, Car16, Wys]. I now advance several further objections that I regard as more basic.

The most basic is that while his argument is against the explanatory ability of a *proof by induction*, Lange reduces the discussion to whether the premises of the argument explain the conclusion. Since it is a serious issue (e.g. [Han90, Ste78, HM05]) whether one proof is more explanatory than another, this reduction clearly misses a major problem.

I now sketch Lange's argument. Noncontroversially, he notes that a proof by mathematical induction proceeds by the following rule of inference[14]:

If $P(1)$ and for each natural number k, $P(k) \to P(k+1)$ then for all k, $P(k)$.

He then asserts, 'The *explanans* would be (for some particular property P) the fact that $P(1)$ and that (for any natural number k) if $P(k)$ then $P(k+1)$.'

He then proposes an alternative rule:

If $P(5)$ and both (a) for each natural number k, $P(k) \to P(k+1)$ and (b) for each natural number k, $P(k) \to P(k-1)$ then for all k, $P(k)$.

Lange argues that not both of these rules can be explanatory. He writes 'Relations of explanatory priority are asymmetric. Otherwise mathematical explanation would be nothing like scientific explanation.' What are the relations between? We have argued that the explanatory object is a proof and such is the title of his paper. But he concludes (where P is some property that may or may not hold of a natural number), 'It cannot be that $P(1)$ helps to explain why $P(5)$ holds *and* that $P(5)$ helps to explain why $P(1)$ holds, on pain of mathematical explanations running in a circle.' This leap exacerbates the reduction of

[13] Particularly relevant is the recognition by Wysocki [Wys] and Cariani [Car16] that the basic thrust of up versus down induction breaks down for argument based on generalized induction definition (such as induction of formulas or closing a subset of a group G to a subgroup of G).

[14] This ignores, of course, the many uses of mathematical induction to prove $P(n)$ and for each $k \geq n$, $P(k) \to P(k+1)$ then for all $k \geq n$, $P(k)$. His argument could be complicated to handle this case as well as it does the one it explicitly addresses, but he doesn't even consider such situations.

the argument to considering only 'premises and consequence' by conflating the entire explanation with any component of it[15].

Lange's Objection is not About Induction. Lange's argument applies more generally to show that for any domain A and $a \in A$, $P(a)$ cannot be a partial explanation for the assertion $A \models (\forall x)P(x)$. We will show several examples of arguments of this form that contradict this assertion.

A standard mathematical technique is to show that all elements of some collection[16] Q satisfy $P(x)$ by first showing an *underlying fact* that restricted to Q, $(\exists x)P(x) \leftrightarrow (\forall x)P(x)$. (In Lange's case, Q is the set of natural numbers.) Now to show that all elements satisfy P, we need only find a convenient a such that $P(a)$ holds. There could be another a' almost as convenient and we could conclude the result for still another a'' that would be very hard to check. This theme shows up in such proof paradigms as proving a function is well-defined, showing that an equivalence relation is a congruence, and in many other situations that are more complicated than appropriate for discussion here. Perhaps Lange would argue that such proofs are not explanatory. But they are direct answers to, for example, 'Why is this function well-defined?'

Here are two more specific examples. Let G be a group with a subgroup K. Here Q is the set of cosets of K. Question: Does aK have an element of even order? Underlying fact: If one element of a coset aK has an element of even order then every element has even order. The key (easy) fact is that if a homomorphism f of groups maps x to y, and $y \neq 1$ then the order of x is divisible by the order of y. So if the order of a is even so is the order of $f(a)$ and any member of $f^{-1}(a)$. Now to determine if aK has an element of even order, we can check the order of any element of the coset (preferably one with low order).

Normal form arguments illustrate the same point. To take a high school example, all quadratic equations can be expressed in *each* of three normal forms: (by degree: ax^2+bx+c, factored form: $a(x-r_1)(x-r_2)$, vertex form: $a(x-h)^2+k$). Here Q is a collection of quadratic polynomials that are all equivalent by the usual algebraic operations. Any of the infinitely many of the forms of the polynomial have the same vertex and the same roots. Factored normal form makes the roots evident; vertex normal form makes the vertex evident. Algebraic transformations are used to put the polynomial in a convenient form for the problem at hand. But one must compute the vertex or roots and the normal form makes this easier. So the choice of one (indeed any) particular equivalent to the original polynomial is partially explanatory of finding the roots or vertex of the polynomial. Normal form arguments are, in fact, a clear example of explanatory argument. The ability to reduce to a normal form is the key point of the explanation.

Tappenden (page 171 of [Tap05]) gives a slightly different example that illustrates the same point. The value of an integral over a plane area does not depend

[15] Baker [Bak10] notes this objection but does not develop it.

[16] If one were to develop this argument in first order logic, Q would be a formula. However, in the spirit of the general discussion of induction we describe here informal mathematical arguments.

on the choice of coordinates[17]; but the ease of evaluation does. So one might find the evaluation by a particular choice of coordinates more explanatory than by another. Thus, one specific case can be a partial explanation of another.

We noted that Lange's argument did not really address inductive proof but any argument for a univesral proposition. These examples demonstrate the failure of Lange's contention that an argument for a universal statement $(\forall x)P(x)$ cannot be explanatory if the arguments appeals to any instance of P. And thus his claim that no proof by mathematical induction can be explanatory also fails.

5 Proof versus Provability

We began with Hanna's distinction between formal and acceptable proof. Neither of these is an appropriate analysis of proof. The second, as she interprets it, is ambiguous about what assumptions are intended. The first was never intended to be such an analysis. Hilbert's goal was to study the *existence* of a proof by providing certain minimal characteristics. He deliberately ignored in this focus on reliability such aspects of a 'good' proof as motivation, irredundancy, organization. While we hailed generalizability as a hallmark of an explanatory proof, one must also note that a proof can be too general. In saying this, we bring out still another aspect of 'explanatory proof'; the quality of an explanation depends on the intended audience[18] The importance of audience is emphasized by this description by Fields Medalist William Thurston of the reaction to his proof of the 'geometric Haken conjecture, a revolutionary result in low-dimensional topology.

> It became dramatically clear how much proofs depend on the audience. We prove things in a social context and address them to a certain audience. Parts of this proof I could communicate in two minutes to the topologists, but the analysts would need an hour lecture before they would begin to understand it. Similarly, there were some things that could be said in two minutes to the analysts that would take an hour before the topologists would begin to get it [Thu94].

None of these characteristics of 'good proof' are captured by 'a proof is a sequence of statements each of which is an axiom or deduced from prior statements by one of clearly stated list of rules of inference'. As Burgess [Bur10] puts it, 'For formal provability to be a good model of informal provability it is not necessary that formal proof should be a good model of informal proof.'

On the other hand, the proofs that contain only motivation for the inductive step, miss the real difficulty. How can a finite process of proof justify a statement about infinitely many objects? Thus, in constructing more explanatory proofs

[17] Here, Q is the set of possible coordinatizations.

[18] In fact, Hanna's article in an educational journal reflects the common use of Gauss' proof for future American teachers of middle school mathematics. The goal of the activity is not understanding the step from example to universal but just some notion of justification.

above, we have included the both inductive definition and proof that is essential for explaining this step but also a motivation (often geometric) for the induction step. Such proofs both verify and explain.

After much of this paper was written, we found our view summarised in the earlier work of Resnik and Kushner who, employing Van Fraassen's notion of a why-question, wrote,

> nothing is an explanation *simpliciter* but only relative to the context-dependent why-question(s) that it answers. ... Whether or not a given proof counts as an explanation depends on the why-question with which it is approached (page 153 of [RK87]).

We return to our original theme of the interaction of reliability and clarity. As Tappenden explores with rich examples and from several perspectives in [Tap05], the notion of mathematical explanation needs to be treated in the general context of the development of an area of mathematics. Our examples, even while focusing on the proof of specific propositions, have demonstrated several aspects of context dependence: the exact choice of hypotheses, what precisely is to be explained, and to who.

The notion of good proof restores a proper balance between 'reliability' and 'clarity' that is lost by mistakenly identifying 'provable' with 'a proof'.

References

[Ano16] Anon. Montrous moonshine. https://en.wikipedia.org/wiki/Monstrous_moonshine. Accessed Apr 2016

[Bak10] Baker, A.: Mathematical induction and explanation. Analysis **70**, 681–689 (2010)

[Bal13] Baldwin, J.T.: Formalization, primitive concepts, and purity. Rev. Symb. Log. **6**, 87–128 (2013). http://homepages.math.uic.edu/ jbaldwin/pub/purityandvocab10.pdf

[Bal14] Baldwin, J.T.: Completeness and categoricity (in power): formalization without foundationalism. Bull. Symb. Log. **20**, 39–79 (2014). http://homepages.math.uic.edu/jbaldwin/pub/catcomnovbib2013.pdf

[Bal15] Baldwin, J.T.: Formalization Without Foundationalism; Model Theory and the Philosophy of Mathematics Practice. Book manuscript available on request (2015)

[Bur10] Burgess, J.P.: Putting structuralism in its place. Preprint (2010)

[Bus17] Bussey, W.H.: The origin of mathematical induction. Am. Math. Mon. **24**, 199–207 (1917)

[BVP06] Bibiloni, L., Viader, P., Paradís, J.: On a series of Goldbach and Euler. Bull. AMS **113**, 206–221 (2006)

[Caj18] Cajori, F.: Origin of the name "mathematical induction". Am. Math. Mon. **25**, 197–201 (1918). https://archive.org/stream/jstor-2972638/2972638_djvu.txt

[Car16] Cariani, F.: Mathematical induction and explanatory value in mathematics. Preprint (2016)

[Cof91] Coffa, A.: The Semantic Traditin from Kant to Carnap: To the Vienna Station. Cambridge University Press, Cambridge (1991)
[Ded63] Dedekind, R.: Essays on the Theory of Numbers. Dover, New York (1963). As first published by Open Court Publications 1901: first German 1888th
[Han90] Hanna, G.: Some pedagogical aspects of proof. Interchange **21**, 6–13 (1990)
[Har15] Harris, M.: Mathematics Without Apologies: Portrait of a Problematic Vocation. Princeton University Press, Princeton (2015)
[HM05] Hafner, J., Mancosu, P.: The varieties of mathematical explanation. In: Mancosu, P., Jørgensen, K.F., Pedersen, S. (eds.) Visualization, Explanation, and Reasoning Styles in Mathematics. Synthese Library, vol. 327, pp. 215–250. Springer, Netherlands (2005)
[Lan] Lange, A.M.: Explanation by induction. Preprint
[Lan09] Lange, M.: Why proofs by mathematical induction are generally not explanatory. Analysis **69**, 203–211 (2009)
[Lin14] Linnebo, Ø.: Plural quantification. In: The Stanford Encyclopedia of Philosophy (Fall 2014th edn.) (2014). http://plato.stanford.edu/archives/fall2014/entries/plural-quant/
[Man08] Mancosu, P.: Mathematical explanation: why it matters. In: Mancosu, P. (ed.) The Philosophy of Mathematical Practice, pp. 134–150. Oxford University Press, Oxford (2008)
[Mar02] Marker, D.: Model Theory: An Introduction. Graduate Texts in Mathematics, vol. 217. Springer, New York (2002)
[Qui69] Quine, W.V.O.: Set Theory and Its Logic. Harvard, Cambridge (1969)
[RK87] Resnik, M., Kushner, D.: Explanation, independence, and realism in mathematics. Ann. Math. Log. **38**, 141–158 (1987)
[Sho67] Shoenfield, J.: Mathematical Logic. Addison-Wesley, Reading (1967)
[Ste78] Steiner, M.: Mathematical explanation. Philos. Stud. **34**, 135–151 (1978)
[Tap05] Tappenden, J.: Proof style and understanding in mathematics I: visualization, unification and axiom choice. In: Mancosu, P., Jørgensen, K.F., Pedersen, S. (eds.) Visualization, Explanation, and Reasoning Styles in Mathematics. Synthese Library, vol. 327, pp. 147–214. Springer, Netherlands (2005)
[Thu94] Thurston, W.P.: On proof and progress in mathematics. Bull. Am. Math. Soc. **30**, 161–177 (1994)
[Wys] Wysocki, T.: Mathematical induction, grounding, and causal explanation. Presentation at APA meeting Chicago, March 2016

Justified Belief and the Topology of Evidence

Alexandru Baltag[1], Nick Bezhanishvili[1], Aybüke Özgün[1,2(✉)], and Sonja Smets[1]

[1] University of Amsterdam, Amsterdam, The Netherlands
thealexandrubaltag@gmail.com, ozgunaybuke@gmail.com
{n.bezhanishvili,s.j.l.smets}@uva.nl
[2] LORIA, CNRS - Université de Lorraine, Nancy, France

Abstract. We introduce a new topological semantics for evidence, evidence-based justifications, belief and knowledge. This setting builds on the evidence model framework of van Benthem and Pacuit, as well as our own previous work on (a topological semantics for) Stalnaker's doxastic-epistemic axioms. We prove completeness, decidability and finite model property for the associated logics, and we apply this setting to analyze key issues in Epistemology: "no false lemma" Gettier examples, misleading defeaters, and undefeated justification versus undefeated belief.

1 Introduction

In this paper we propose a topological semantics for *evidence-based belief*, as well as for a notion of "soft" (defeasible) *knowledge*, and explore their connections with various notions of *evidence possession*. This work is largely based on looking from a new perspective at the models for evidence and belief proposed by van Benthem and Pacuit [21], and developed further in [20].

The *basic pieces of evidence* possessed by an agent are modeled as non-empty sets of possible worlds. A *combined evidence* (or just "evidence", for short) is any non-empty intersection of finitely many pieces of evidence. This notion of evidence is not necessarily factive[1], since the pieces of evidence are possibly false (and possibly inconsistent with each other). The family of (combined) evidence sets forms a topological basis, that generates what we call *the evidential topology*. This is the smallest topology in which all the basic pieces of evidence are open, and it will play an important role in our setting. We study the operator of "having (a piece of) evidence for a proposition P" proposed by van Benthem and Pacuit, but we also investigate other interesting variants of this concept: "having (combined) evidence for P", "having a (piece of) *factive* evidence for P" and "having (combined) factive evidence for P". We show that the last notion coincides with the *interior operator* in the evidential topology, thus matching

[1] Factive evidence is true in the actual world. In Epistemology it is common to reserve the term "evidence" for factive evidence. But we follow here the more liberal usage of this term in [20], which agrees with the common usage in day to day life, e.g. when talking about "uncertain evidence", "fake evidence", "misleading evidence" etc.

J. Väänänen et al. (Eds.): WoLLIC 2016, LNCS 9803, pp. 83–103, 2016.
DOI: 10.1007/978-3-662-52921-8_6

McKinsey and Tarski's original topological semantics for modal logic [15]. We also show that the two factive variants of evidence-possession operators are more expressive than the original (non-factive) one, being able (when interacting with the global modality) to define the non-factive variants, as well as many other doxastic/epistemic operators.

We propose a *'coherentist' semantics for justification and justified belief*, that is obtained by extending, generalizing and (to an extent) "streamlining" the evidence-model framework for beliefs introduced in [21]. An *argument for* P consists of one or more (combined) evidence sets supporting the same proposition P (thus providing multiple evidential paths towards a common conclusion). A *justification for* P is an argument for P that is consistent with every other evidence. Our proposed definition of belief is equivalent to requiring that: P *is believed iff there is some (evidence-based) justification for* P. According to this setting, in order to believe P one needs to have an "undefeated" argument for P: one that is not refuted by any available evidence. We show that our notion of belief coincides with the one of van Benthem and Pacuit [21] for *finite* models, but involves a different generalization of their notion in the infinite case. But, in contrast to the later one, our semantics *always ensures consistency of belief*, even when the available pieces of evidence are mutually inconsistent.[2] Our proposal is also very natural from a topological perspective: essentially, P *is believed iff* P *is true in "almost all" epistemically-possible worlds* (where 'almost all' is interpreted topologically: all except for a nowhere-dense set).

Moving on to 'knowledge', there are a number of different notions one may consider. First, there is "absolutely certain" or "infallible" knowledge, akin to Aumann's concept of 'partitional knowledge' or van Benthem's concept of 'hard information'. In our single-agent setting, this can be simply defined as the global modality (quantifying universally over all epistemically-possible worlds). There are propositions that are 'known' in this infallible way (-e.g. the ones known by introspection or by logical proof), but very few: most facts in science or real-life are unknown in this sense. Hence, it is more interesting to look at notions of knowledge that are less-than-absolutely-certain: so-called 'defeasible knowledge'. The famous Gettier counterexamples [7] show that simply adding "factivity" to belief will not do: true (justified) belief is extremely fragile (i.e. it can be too easily lost), and it is consistent with having only wrong justifications for an (accidentally) true conclusion.

Clark's [5] influential "no false lemma" proposal is to require a *correct justification*: one that doesn't use any falsehood. We formalize this notion by saying that P *is known if there is a factive (true) justification for* P. Note though that our proposal imposes a stronger requirement than Clark's, since our concept of justification requires consistence with all the available (combined) evidence. In our terminology, Clark only requires a factive *argument* for P. So Clark's approach is 'local', assessing a knowledge claim based only on the truth of the

[2] Another, purely technical advantage of our setting is that the resulting doxastic logic has finite model property, in contrast to the one in [21].

evidence pieces (and the correctness of the inferences) that are used to justify it. Our proposal is coherentist, and thus 'holistic', assessing knowledge claims by their coherence with all of the agent's acceptance system: justifications need to be checked against all the other arguments that can be constructed from the agent's current evidence.

Another approach to knowledge (also stronger than the no-false-lemma requirement) was championed by Lehrer, Klein and others [11–14,17], under the name of "Defeasibility Theory of Knowledge". According to this view, P is known (in the in-defeasible sense) only if there is a factive justification for P that cannot be defeated by any further true evidence. This means that the justification is consistent, not only with the currently available evidence, but also with *any potential (new) factive evidence* that the agent might learn in the future. This version of the theory has been criticized as being too strong: some new evidence might be 'misleading' or 'deceiving' despite being true. A weaker version of Defeasibility Theory requires that knowledge is undefeated only by "non-misleading" evidence. In our setting, a proposition P is said to be a *potentially misleading evidence* if it can indirectly generate false evidence (i.e. if by adding P to the family of currently available pieces of evidence we obtain at least one false combined evidence). Misleading propositions include all the false ones, but they may also include some true ones. We show that our notion of knowledge matches this weakened version of Defeasibility Theory (though not the strong version).

Yet another path leading to our setting in this paper goes via our previous work [1,2] on a topological semantics for the doxastic-epistemic axioms proposed by Stalnaker [18]. These axioms were meant to capture a notion of fallible knowledge, in close interaction with a notion of "strong belief" (defined as "subjective certainty" or the "feeling of knowledge"). The main principle specific to this system was that "believing implies believing that you know" ($Bp \to BKp$), which goes in direct contradiction to Negative Introspection for Knowledge.[3] The topological semantics that we proposed for these concepts in [1,2] was overly restrictive (being limited to the rather exotic class of "extremally disconnected" topologies). In this paper, we show that these notions can be interpreted on arbitrary topological spaces, without changing their logic. Indeed, our definitions of belief and knowledge above can be seen as the natural generalizations to arbitrary topologies of the notions in [1,2].

We apply our models to various Gettier-type examples, and completely axiomatize the resulting logics, proving their decidability and finite model property. Our hardest result refers to our richest logic (that can define all the modal operators mentioned above). We end with a discussion of possible research lines for future work.

[3] Indeed, the logic of Stalnaker's knowledge is not $S5$, but the modal logic $S4.2$.

2 Evidence, Belief and Knowledge in Topological Spaces

2.1 Topological Models for Evidence

Definition 1 (Evidence Models). *(van Benthem and Pacuit)*[4] Given a countable set of propositional letters Prop, an *evidence model* for Prop is a tuple $\mathcal{M} = (X, E_0, V)$, where: X is a non-empty set of "states"; $E_0 \subseteq \mathcal{P}(X) \setminus \{\emptyset\}$ is a family of non-empty sets called *basic evidence sets* (or *pieces of evidence*), with $X \in E_0$; and $V : \text{Prop} \to \mathcal{P}(X)$ is a valuation function.

Given an evidence model $\mathcal{M} = (X, E_0, V)$, a family $F \subseteq E_0$ of pieces of evidence is *consistent* if $\bigcap F \neq \emptyset$, and *inconsistent* otherwise. A *body of evidence* is a family $F \subseteq E_0$ s.t. every non-empty finite subfamily is consistent. We denote by $\mathcal{F}$ the family of all bodies of evidence, and by $\mathcal{F}^{finite}$ the family of all finite ones. A body of evidence F *supports a proposition* P iff P is true in all worlds satisfying the evidence in F (i.e. $\bigcap F \subseteq P$).

The *strength order* between bodies of evidence is given by inclusion: $F \subseteq F'$ means that F' is at least as strong as F. Note that stronger bodies of evidence support more propositions: if $F \subseteq F'$ then every proposition supported by F is also supported by F'. A body of evidence is *maximal* ("strongest") if it's not included in any other such body. We denote by $Max_{\subseteq}\mathcal{F} = \{F \in \mathcal{F} : \forall F' \in \mathcal{F}(F \subseteq F' \Rightarrow F = F')\}$ the family of all maximal bodies of evidence. By Zorn's Lemma, *every body of evidence can be strengthened to a maximal body of evidence*: $\forall F \in \mathcal{F} \exists F' \in Max_{\subseteq}\mathcal{F}(F \subseteq F')$.

A *combined evidence* (or just "*evidence*", for short) is any non-empty intersection of finitely many pieces of evidence. We denote by $E := \{\bigcap F : F \in \mathcal{F}^{finite} \text{ s.t. } \bigcap F \neq \emptyset\}$ the family of all (combined) evidence.[5] A (combined) evidence $e \in E$ *supports* a proposition $P \subseteq X$ if $e \subseteq P$. (In this case, we also say that e is *evidence for* P.) Note that the natural strength order between combined evidence sets goes the other way around (reverse inclusion): $e \supseteq e'$ means that e' is at least as strong as e.[6]

The intuition is that $e \in E_0$ represent the basic pieces of "direct" evidence (obtained say by observation or via testimony) that are possessed by the agent, while the combined evidence $e \in E$ represents indirect evidence that is obtained by combining finitely many pieces of direct evidence. Not all of this evidence is necessarily true though.

We say that some (basic or combined) evidence $e \in E$ is *factive evidence* at world $x \in X$ whenever it is true at x (i.e. $x \in e$). A body of evidence F is factive if all the pieces of evidence in F are factive (i.e. $x \in \bigcap F$).

---

[4] The notion of evidence model in [21] is more general, covering cases in which evidence depends on the actual world, but we stick with what they call 'uniform' models, since this corresponds to restricting to agents who are "evidence-introspective".

[5] This is a difference in notation with the setting in [20, 21], where E is used to denote the family of basic evidence sets (denoted here by E_0).

[6] This is both to fit with the strength order on bodies of evidence (since $F \subseteq F'$ implies $\bigcap F \supseteq \bigcap F'$), and to ensure that stronger evidence supports more propositions: since, if $e \supseteq e'$, then every proposition supported by e is supported by e'.

The *plausibility (pre)order* $\sqsubseteq_E$ associated to an evidence model is given by:

$$x \sqsubseteq_E y \text{ iff } \forall e \in E_0 \, (x \in e \Rightarrow y \in e) \text{ iff } \forall e \in E \, (x \in e \Rightarrow y \in e).$$

Definition 2 (Topological Space). A *topological space* is a pair $\mathcal{X} = (X, \tau)$, where X is a non-empty set and τ is a *topology* on X, i.e. a family $\tau \subseteq \mathcal{P}(X)$ containing X and $\emptyset$, and closed under finite intersections and arbitrary unions. Given a family $E \subseteq \mathcal{P}(X)$ of subsets of X, the *topology generated by* E is the smallest topology τ_E on X such that $E \subseteq \tau_E$. A set $A \subseteq X$ is *closed* iff it is the complement of an open set, i.e. it is of the form $X \setminus U$ with $U \in \tau$. Let $\tau^c = \{X \setminus U \,|\, U \in \tau\}$ denote the family of all closed sets of $\mathcal{X} = (X, \tau)$. In any topological space $\mathcal{X} = (X, \tau)$, one can define two important operators, namely *interior* $Int : \mathcal{P}(X) \to \mathcal{P}(X)$ and *closure* $Cl : \mathcal{P}(X) \to \mathcal{P}(X)$, given by $IntP := \bigcup\{U \in \tau \,|\, U \subseteq P\}$, $ClP := \bigcap\{C \in \tau^c \,|\, P \subseteq C\}$. A set $A \subseteq X$ is called *dense* in $\mathcal{X}$ if $ClA = X$ and it is called *nowhere dense* if $IntClA = \emptyset$. For a topological space $\mathcal{X} = (X, \tau)$, the *specialization preorder* $\sqsubseteq_\tau$ is given by: $x \sqsubseteq_\tau y$ iff $\forall U \in \tau \, (x \in U \Rightarrow y \in U)$.

Special Case: Relational Spaces. A topological space is called *Alexandroff* iff the topology is closed under arbitrary intersections. An Alexandroff topology is fully captured by its specialization preorder: *in this case, the interior operator coincides with the Kripke modality for the specialization relation* (i.e. $IntP = \{x \in X \,|\, \forall y \, (x \sqsubseteq_\tau y \Rightarrow y \in P)\}$). There is a canonical bijection between Alexandroff topologies $\mathcal{X} = (X, \tau)$ and *preordered spaces*[7] $(X, \leq)$, mapping (X, τ) to $(X, \sqsubseteq_\tau)$; the inverse map takes $(X, \leq)$ into $(X, Up(X))$, where $Up(X)$ is the family of upward-closed sets[8].

An Even More Special Case: (Grove/Lewis) Sphere Spaces. These are topological spaces in which the opens are "nested", i.e. for every $U, U' \in \tau$, we have either $U \subseteq U'$ or $U' \subseteq U$. Sphere spaces are Alexandroff, and moreover they correspond exactly to *totally preordered spaces* (i.e. sets X endowed with a total preorder $\leq$).

Definition 3 (Topological Evidence Models). A *topological evidence model* ("*topo-e-model*", for short) is a structure $\mathcal{M} = (X, E_0, \tau, V)$, where (X, E_0, V) is an evidence model and $\tau = \tau_E$ is the topology generated by the family of combined evidence E (or equivalently, by the family of basic evidence sets E_0)[9], which will be called *the evidential topology*. It is easy to see that *the plausibility order* $\sqsubseteq_E$ *of* $\mathcal{M}$ *coincides with the specialization order* of the associated topology: $\sqsubseteq_E = \sqsubseteq_\tau$.

Since any family $E_0 \subseteq \mathcal{P}(X)$ generates a topology, topo-e-models are just another presentation of (uniform) evidence models. We use this special terminology to stress our focus on the topology, and to avoid ambiguities (since our

---

[7] A *preorder* on X is a reflexive-transitive relation on X.

[8] A subset $A \subseteq X$ is said to be *upward-closed* wrt $\leq$ if $\forall x, y \in X \, (x \in A \wedge x \leq y \Rightarrow y \in A)$.

[9] These families generate the same topology. We denote it by τ_E only because the family E of combined evidence forms a *basis* of this topology.

definition of belief in topo-e-models will be different from the definition of belief in evidence models in [21]).

A topo-e-model is said to be *Alexandroff* iff the underlying topology is Alexandroff. So they can be understood as *relational (plausibility) models*, in terms of a preorder $\leq$ ("plausibility relation"). A special case is the one of *Grove-Lewis (topological) evidence models*: this is the case when the basic pieces of evidence are nested (i.e. for all $e, e' \in E_0$ we have either $e \subseteq e'$ or $e' \subseteq e$). It is easy to see that in this case all the opens of the generated topology are also nested, so the topology is that of a sphere space.

Proposition 1. *Given a topo-e-model $\mathcal{M} = (X, E_0, \tau, V)$, the following are equivalent:*

1. *$\mathcal{M}$ is Alexandroff;*
2. *The family E of (combined) evidence is closed under arbitrary non-empty intersections (i.e. if $F \subseteq E$ and $\bigcap F \neq \emptyset$, then $\bigcap F \in E$);*
3. *Every consistent body of evidence is equivalent to a finite body of evidence (i.e. $\forall F \in \mathcal{F}(\bigcap F \neq \emptyset \Rightarrow \exists F' \in \mathcal{F}^{finite}$ s. t. $\bigcap F = \bigcap F'$)).*

Arguments and Justifications. We can use this setting to formalize a "coherentist" view on justification. An *argument for P* is a disjunction $U = \bigcup_{i \in I} e_i$ of (some non-empty family of) (combined) evidences $e_i \in E$ that all support P (i.e. $e_i \subseteq P$ for all $i \in I$). Thus, an argument may provide *multiple evidential paths e_i* to support a common conclusion P. Topologically, an argument for P is the same as a *non-empty open subset of P* ($U \in \tau_E$ s.t. $U \subseteq P$). Also, the interior $IntP$ is the *weakest (most general) argument for P*.

A *justification for P* is an argument U for P that is consistent with every (combined) evidence (i.e. $U \cap e \neq \emptyset$ for all $e \in E$, which in fact implies that $U \cap U' \neq \emptyset$ for all $U' \in \tau_E \setminus \{\emptyset\}$). So justifications are arguments that are not defeated by any available evidence. Topologically, we can see that a justification for P is just an *(everywhere) dense open subset* of P (i.e. $U \in \tau_E$ s.t. $U \subseteq P$ and $Cl_{\tau_E}(U) = X$). As for evidence, an argument or a justification for P is said to be *factive* (or "correct") if it is true in the actual world. The fact that arguments are open in the generated topology encodes the principle that *any argument should be evidence-based*: whenever an argument is correct, then it is supported by some factive evidence. To anticipate further: in our setting, justifications will form the basis of *belief*, while correct justifications will form the basis of *(defeasible) knowledge*. But for now we'll introduce a stronger form of "knowledge": the absolutely-certain and irrevocable kind.

Infallible Knowledge: Possessing Hard Information. We use $\forall$ for the so-called *global* modality, which associates to every proposition $P \subseteq X$, some other proposition $\forall P$, given by putting: $(\forall P) := X$ iff $P = X$, and $(\forall P) := \emptyset$ otherwise. In other words: $(\forall P)$ holds (at any state) iff P holds at all states. In this setting, $\forall$ is interpreted as "absolutely certain, *infallible knowledge*", defined as truth in all the worlds that are consistent with the agent's information.[10] This is not a realistic concept of knowledge, but just a limit notion, encompassing all epistemic possibilities.

Having *Basic* Evidence for a Proposition. van Benthem and Pacuit define, for every proposition $P \subseteq X$, another proposition[11] E_0P given by putting: $E_0P := X$ if $\exists e \in E_0\,(e \subseteq P)$, and $E_0P := \emptyset$ otherwise. Essentially, E_0P means that "the agent *has basic evidence for* P", i.e. P is supported by some available piece of evidence. One can also introduce a *factive* version of this proposition: $\Box_0 P$, read as "the agent has *factive basic evidence for* P", is given by putting

$$\Box_0 P := \{x \in X : \exists e \in E_0\,(x \in e \subseteq P)\}.$$

Having (Combined) Evidence for a Proposition. If in the above definitions of E_0P and $\Box_0P$ we replace basic pieces of evidence by combined evidence, we obtain two other operators EP, meaning that "the agent *has (combined) evidence for* P", and $\Box P$, meaning that "the agent *has factive (combined) evidence for* P". More precisely:

$$EP := X \text{ if } \exists e \in E\,(e \subseteq P), \quad \text{and } EP := \emptyset \text{ otherwise;}$$

$$\Box P := \{x \in X : \exists e \in E\,(x \in e \subseteq P)\}.$$

Observation 1. Note that *the agent has evidence for a proposition P iff she has an argument for P*. So EP can also be interpreted as "having an argument for P". Similarly, $\Box P$ can be interpreted as "having a *correct (i.e. factive) argument* for P".

Observation 2. Note that the agent has factive evidence for P at x iff x is in the interior of P. So our modality $\Box$ *coincides with the interior operator*: $\Box P = Int P$.

2.2 Belief

Belief à la van Benthem-Pacuit [21]. The notion of belief proposed by van Benthem and Pacuit, which we will denote by Bel, is that P *is believed iff every maximal body of evidence supports* P: $BelP$ holds (at any state of X) iff we have $\bigcap F \subseteq P$ for every $F \in Max_{\subseteq}\mathcal{F}$. As already noticed in [21], this is equivalent to treating evidence models as special cases of plausibility models [3,4,19], with the plausibility relation given by $\sqsubseteq_E$ (or equivalently, as Grove-Lewis "sphere models" [9] where the spheres are the sets that are upward closed wrt $\sqsubseteq_E$), and

[10] In a multi-agent model, some worlds might be consistent with one agent's information, while being ruled out by another agent's information. So, in a multi-agent setting, $\forall_i$ will only quantify over all the states in agent i's current information cell (according to a partition Π_i of the state space reflecting agent i's hard information).

[11] They denote this by EP, but we use E_0P for this notion, since we reserve the notation EP for having *combined* evidence for P.

applying the standard definition (due to Grove) of belief as "truth in all the most plausible worlds".[12] Grove's definition works well when the plausibility relation is well-founded (and also in the somewhat more general case given by the Grove-Lewis Limit Assumption), but it yields inconsistent beliefs in the case that there are *no* most plausible worlds. But note that in evidence models $\sqsubseteq_E$ may be non-wellfounded. Indeed, *belief à la van Benthem-Pacuit can be inconsistent*:

Example 1. *Consider the evidence model $\mathcal{M} = (\mathbb{N}, E_0, V)$, where the state space is the set $\mathbb{N}$ of natural numbers, $V(p) = \emptyset$, and the basic evidence family $E_0 = \{e \subseteq \mathbb{N} : \mathbb{N} \setminus e \text{ finite}\}$ consists of all co-finite sets. The only maximal body of evidence in E_0 is E_0 itself. However, $\bigcap E_0 = \emptyset$. So $Bel\bot$ holds in $\mathcal{M}$.*

This phenomenon only happens in (some cases of) *infinite* models, so it is *not* due to the inherent mutual inconsistency of the available evidence. The "good" examples in [21] are the ones in which (possibly inconsistent) evidence is processed to yield consistent beliefs. So it seems to us that the intended goal (only partially fulfilled) in [21] was to ensure that the agents are able to form *consistent beliefs* based on the available evidence. We think this to be a natural requirement for idealized "rational" agents, and so we consider doxastic inconsistency to be "a bug, not a feature", of the van Benthem-Pacuit framework. Hence, we now propose a notion that agrees with the one in [21] in all the "good" cases, but also produces in a natural way only consistent beliefs.

Our Notion of Belief. The intuition is that P is believed iff *it is entailed by all the "sufficiently strong" (combined) evidence*. Formally, BP holds iff *every finite body of evidence can be strengthened to some finite body of evidence which supports P*:

$$BP \text{ holds (at any state)} \quad \text{iff} \quad \forall F \in \mathcal{F}^{finite} \exists F' \in \mathcal{F}^{finite} (F \subseteq F' \wedge \bigcap F' \subseteq P).$$

Our notion of belief B coincides with Bel in the *finite* case, or, more generally, in *evidence models in which every maximal body of evidence is consistent*. But, unlike Bel, our notion of belief B is *always consistent* (i.e. $B\bot = B\emptyset = \emptyset$), and moreover it satisfies the axioms of the standard doxastic logic $KD45$. Another nice feature is that our belief B is a *purely topological notion*, as can be seen from the following:

Proposition 2. *In every evidence model (X, E_0, V), the following are equivalent, for any proposition $P \subseteq X$:*

1. *BP holds (at any state);*
2. *every (combined) evidence can be strengthened to some evidence supporting P ($\forall e \in E \exists e' \in E$ s.t $e' \subseteq e \cap P$);*

[12] Note that all the notions of belief we consider are global: they do not depend on the state of the world, i.e. we have either $BelP = X$ or $BelP = \emptyset$ (similar to the sets $\forall P, E_0P, EP$). This expresses the assumption that belief is a purely internal notion, thus transparent and hence absolutely introspective. This is standard in logic and accepted by most philosophers.

3. *every argument (for anything) can be strengthened to an argument for* P *(*$\forall U \in \tau_E \setminus \{\emptyset\} \exists U' \in \tau_E \setminus \{\emptyset\}$ *s.t.* $U' \subseteq U \cap P$*);*
4. there is a justification for P*: i.e. some argument for* P *which is consistent with any available evidence (*$\exists U \in \tau_E$ *s.t.* $U \subseteq P$ *and* $U \cap e \neq \emptyset$ *for all* $e \in E$*);*
5. P *includes some dense open set;*
6. $IntP$ *is* dense *in* τ_E *(i.e.* $Cl(IntP) = X$*), or equivalently* $X \setminus P$ is nowhere dense*;*
7. $\forall\Diamond\Box P$ *holds (at any state: i.e.* $\forall\Diamond\Box P \neq \emptyset$*, or equivalently* $\forall\Diamond\Box P = X$*), where* $\Diamond P := \neg\Box\neg P$ *is the dual of the* $\Box$ *operator.*

Proposition 2 part (4) can be interpreted as saying that *our notion of belief* B *is the same as "justified belief"*: a proposition P is believed iff the agent has a justification for P. In this case, there exists a *weakest (most general) justification* for P, namely $IntP$. Moreover, part (6) shows that our proposal is very natural from a topological perspective: it is equivalent to saying that *P is believed iff the complement of P is nowhere dense*. Since nowhere dense sets are one of the topological concepts of "small" or "negligible sets", this amounts to *believing propositions if they are true in "almost all" epistemically-possible worlds* (where 'almost all' is interpreted topologically). Finally, part (7) tells us that *belief is definable in terms of the operators* $\forall$ and $\Box$.

Our notion of belief can be viewed as a formalization of a "coherentist" epistemology of belief. The requirement that a belief's justification must be *open* in the evidential topology simply means that the justification is ultimately based on the available evidence; while the requirement that the justification is *dense* (in the same topology) means that all the agent's beliefs must be coherent with all her evidence.[13]

Conditional Belief. For sets $Q, Q' \subseteq X$, we say that Q' is *Q-consistent* iff $Q \cap Q' \neq \emptyset$. A body of evidence F is *Q-consistent* iff $\bigcap F \cap Q \neq \emptyset$. We say that P *is believed given* Q, and write $B^Q P$, iff *every finite Q-consistent body of evidence can be strengthened to some finite Q-consistent body of evidence supporting* $Q \to P$ (i.e. $\neg Q \cup P$). Similarly to Proposition 2, $B^Q P$ is equivalent to any of the following: every Q-consistent evidence can be strengthened to some Q-consistent evidence supporting $Q \to P$; every Q-consistent argument can be strengthened to a Q-consistent argument for $Q \to P$; there is a Q-consistent argument for $Q \to P$ which is consistent with any Q-consistent evidence; $Q \to P$ includes some Q-consistent open set which is dense in Q; $\forall(Q \to \Diamond(Q \wedge \Box(Q \to P))) = X$; etc.

2.3 Knowledge

We now define a "softer" notion of knowledge, that is closer to the common usage of the word than "infallible" knowledge. Formally, we put $KP := \{x \in$

[13] Lehrer uses the metaphor of a Subjective Justification Game [13]: rational beliefs are based on justifications that survive a game between the Believer and an inner Critic, who tries to defeat them using the Believer's own "acceptance system".

$X : \exists U \in \tau\,(x \in U \subseteq P \wedge Cl(U) = X)\}$. So KP holds at x iff P *includes a dense open neighborhood of* x; equivalently, iff $x \in IntP$ and $IntP$ is dense. Essentially, this says that *knowledge is "correctly justified belief"*: KP holds at world x iff there exists some justification $U \in \tau$ for P such that $x \in U$. In other words, P is known iff there *exists some correct (i.e. factive) argument for P that is consistent with all the available evidence.*

Note that K satisfies Stalnaker's Strong Belief Principle $BP = BKP$: *from a subjective point of view, belief is indistinguishable from knowledge* [18].[14]

Example 2. *Consider the model $\mathcal{X} = ([0,1], E_0, V)$, where $E_0 = \{(a,b) \cap [0,1] : a, b \in \mathbb{R},\ a < b\}$ and $V(p) = \emptyset$. The generated topology τ_E is the standard topology on $[0,1]$. Let $P = [0,1] \setminus \{\frac{1}{n} : n \in \mathbb{N}\}$ be the proposition stating that the actual state is* not *of the form $\frac{1}{n}$, for any $n \in \mathbb{N}$. Since the complement $\neg P = \{\frac{1}{n} : n \in \mathbb{N}\}$ is nowhere dense, the agent* believes P, *and e.g. $U = \bigcup_{n \geq 1}(\frac{1}{n+1}, \frac{1}{n})$ is a (dense, open) justification for P. This belief is* true at world $0 \in P$. *But this* true belief is not knowledge *at* 0*: no justification for P is true at* 0*, since P doesn't include any open neighborhood of* 0*, so $0 \notin IntP$ and hence $0 \notin KP$. (However, P is known at all the* other *worlds $x \in P \setminus \{0\}$, since $\forall x \in P \setminus \{0\} \exists \epsilon > 0$ s.t. $x \in (x - \epsilon, x + \epsilon) \subseteq P$, hence $x \in IntP$.)*

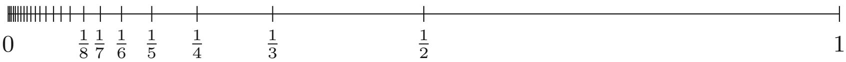


Fig. 1. $([0,1], \tau_E)$

This 'soft' type of knowledge is *defeasible*. In contrast, the usual assumption in Logic is that *knowledge acquisition is monotonic*. As a result, logicians typically assume that knowledge is "irrevocable": once acquired, it cannot be defeated by any further evidence. In our setting, the only irrevocable knowledge is the infallible one, captured by the operator $\forall$. Clearly, K is *not* irrevocable.

Epistemologists have made various other proposals on how a realistic concept of knowledge should be defined. A conception that is very close to (though subtly different from) our notion is the one held by the proponents of the so-called Defeasibility Theory of Knowledge, e.g. Lehrer and Paxson [14], Lehrer [13], Klein [11,12]: "in-defeasible knowledge" cannot be defeated by any *factive* evidence that might be gathered later (though it may be defeated by false "evidence"). In its simplest version, this says that "an agent knows that P if and only if P is true, she believes that P, and she continues to believe P if any *true* information is received" (Stalnaker [18]). In our formalism, this would require

[14] As we'll see, K and B satisfy all the Stalnaker axioms for knowledge and belief [1,2,16] and further generalizes our previous work on a topological interpretation of Stalnaker's doxastic-epistemic axioms, which was based on extremally disconnected spaces.

P to be believed conditional on every true "new evidence": i.e. P is known in world x iff $B^Q P$ holds for every $Q \subseteq X$ with $x \in Q$. This simple version is what Rott calls "the Stability Theory of Knowledge" [17]. In contrast, the full-fledged version of the Defeasibility Theory, as held by Lehrer and others, insists that, in order to know P, not only the belief in P has to stay undefeated, but also its justification (i.e. what we call here "an argument for P"). In other words, there must exist an argument for P that is believed conditional on every true evidence. Clearly, this implies that the belief in P is stable; but the converse is not at all obvious. Indeed, Lehrer claims that the converse is false. The problem is that, when confronted with various new pieces of evidence, the agent might keep switching between different justifications (for believing P); thus, she may keep believing in P conditional on any such new true evidence, without actually having any "good" justification (i.e. one that remains itself undefeated by all true evidence). To have 'knowledge', we thus need a *stable justification*.[15]

However, many authors attacked the above interpretation (of both the stability and the defeasibility theory) as being *too strong*: if we allow as potential defeaters *all* factive propositions (i.e. all sets of worlds P containing the actual world), then there are intuitive examples showing that knowledge KP *can* be defeated. Here is such an example, discussed by a leading proponent of the defeasibility theory (Klein [12]). Loretta filled in her federal taxes, following very carefully all the required procedures on the forms, doing all the calculations and double checking everything. Based on this evidence, she correctly believes that she owes \$500, and she seems perfectly justified to believe this. So it seems obvious that she *knows* this. But suppose now that, being aware of her own fallibility, she asks her accountant to check her return. The accountant finds no errors, and so he sends her his reply reading "Your return contains *no* errors"; but he inadvertently leaves out the word "no". If Loretta would learn the true fact that the accountant's letter actually reads "Your return contains errors", she would lose her belief that she owed \$500! So it seems that there exist defeaters that are true but "misleading".

We can formalize this counterexample as follows.

Example 3. *Consider the model $\mathcal{M} = (X, E_0, V)$, where $X = \{x_1, x_2, x_3, x_4, x_5\}$, $V(p) = \emptyset$, $E_0 = \{X, O_1, O_2\}$, $O_1 = \{x_1, x_2, x_3\}$, $O_2 = \{x_3, x_4, x_5\}$. The resulting set of combined evidence is $E = \{X, O_1, O_2, \{x_3\}\}$. Assume the actual world is x_1. Then O_1 is* known*, since $x_1 \in Int(O_1) = O_1$ and $Cl(O_1) = X$. Now consider the model $\mathcal{M}^{+O_3} = (X, E_0^{+O_3}, V)$ obtained by adding the new evidence $O_3 = \{x_1, x_5\}$. We have $E_0^{+O_3} = \{X, O_1, O_2, O_3\}$, so $E^{+O_3} = \{X, O_1, O_2, O_3, \{x_1\}, \{x_3\}, \{x_5\}\}$. Note that the new evidence is* true *($x_1 \in O_3$). But O_1 is* not even believed *in $\mathcal{M}^{+O_3}$ anymore (since $O_1 \cap \{x_5\} = \emptyset$, so O_1 is no longer dense in $\tau_{E^{+O_3}}$), thus O_1 is* no longer known *after the true evidence O_3 was added!*

[15] Lehrer uses the metaphor of an 'Ultra-Justification Game' [13], according to which 'knowledge' is based on arguments that survive a game between the Believer and an omniscient truth-telling Critic, who tries to defeat the argument by using both the Believer's current "justification system" *and any new true evidence.*

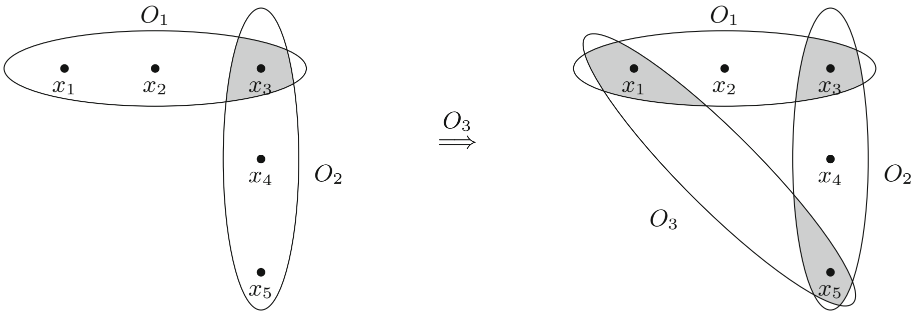


Fig. 2. From $\mathcal{M}$ to $\mathcal{M}^{+O_3}$

Klein's story corresponds to taking O_1 to represent Loretta's direct evidence (based on careful calculations) that she owes \$500, O_2 to represent her prior evidence (based on past experience) that the accountant doesn't make mistakes in his replies to her, and O_3 the potential new evidence provided by the letter. In conclusion, our notion of knowledge is incompatible with the above-mentioned strong interpretations of both stability and defeasibility theory, thus confirming the objections raised against them.

Klein's solution is that one should exclude such 'misleading' defeaters, which may "unfairly" defeat a good justification. But how can we distinguish them from genuine defeaters? Klein's diagnosis, in Foley's more succinct formulation [6], is that "a defeater is misleading if it justifies a falsehood in the process of defeating the justification for the target belief". In the example, the falsehood is that the accountant had discovered errors in Loretta's tax return. It seems that the new evidence O_3 (the existence of the letter as actually written) supports this falsehood, but how? According to us, it is the combination $O_2 \cap O_3$ of the new (true) evidence O_3 with the old (false) evidence O_2 that supports the new falsehood: the true fact (about the letter saying what it says) entails a falsehood *only* if it is taken in conjunction with Loretta's prior evidence (or blind trust) that the accountant cannot make mistakes. So intuitively, *misleading defeaters are the ones which may lead to new false conclusions when combined with some of the old evidence.*

We proceed now to formalize this distinction. Given a topo-e-model $\mathcal{M}$, a proposition $Q \subseteq X$ is *misleading at $x \in X$ wrt E* if evidence-addition with Q produces some false new evidence; i.e. if there is some $e' \in E^{+Q} \setminus E$ s.t. $x \notin e'$; equivalently, there is some $e \in E$ s.t. $x \notin (e \cap Q) \notin E \cup \{\emptyset\}$. It is easy to see that: *old evidence in E is by definition non-misleading wrt E* (i.e. if $e \in E$ then e is non-misleading wrt E), and *new non-misleading evidence must be true* (i.e. if $Q \notin E$ is non-misleading at x then $x \in Q$).

We are now in the position to formulate precisely the "weakened" versions of both stability and defeasibility theory that we are looking for. The Weak Stability Theory will stipulate that P is known if it is undefeated by every non-misleading proposition: i.e. $B^Q P$ holds for every non-misleading $Q \subseteq X$. The Weak Defeasibility Theory will require that there exists some justification (argument) for P

that is undefeated by every non-misleading proposition. Finally, there is a third formulation, which one might call Epistemic Coherence theory, saying that P is known iff there exists some justification (argument) for P which is consistent with every non-misleading proposition.

The following counterexample shows that weak stability is (only a necessary, but) not a sufficient condition for knowledge:

Example 4. *Consider the model $\mathcal{M} = (X, E_0, V)$, where $X = \{x_0, x_1, x_2\}$, $V(p) = \emptyset$, $E_0 = \{X, O_1, O_2\}$, $O_1 = \{x_1\}$, $O_2 = \{x_1, x_2\}$. The resulting set of combined evidence is $E = E_0$. Assume the actual world is x_0, and let $P = \{x_0, x_1\}$. Then P is* believed *(since its interior $IntP = \{x_1\}$ is dense) but it is* not known *(since $x_0 \notin IntP = \{x_1\}$). However, we can show that P* is believed conditional on any non-misleading proposition*. For this, note that the family of non-misleading propositions (at x_0) is $E \cup \{P, \{x_0\}\} = \{X, O_1, O_2, P, \{x_0\}\}$. It is easy to see that for each set Q in this family, we have $B^Q P$.*

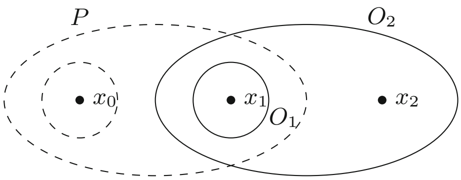


Fig. 3. $\mathcal{M} = (X, E_0, V)$: The continuous ellipses represent the currently available pieces of evidence, while the dashed ones represent the other non-misleading propositions.

One should stress that our counterexample agrees with the position taken by most proponents of Defeasibility Theory: stability of (justified) belief is not enough for knowledge. Intuitively, what happens in the above example is that, although the agent continues to believe P given any non-misleading evidence, her justification keeps changing: there is *no* uniform justification for P that works for every non-misleading evidence Q.

The next result shows that *our notion of knowledge exactly matches the weakened version of Defeasibility Theory*, as well as the Epistemic Coherence formulation:

Proposition 3. *Let $\mathcal{M}$ be a topo-e-model, and assume $x \in X$ is the actual world. The following are equivalent for all $P \subseteq X$:*

1. *P is known ($x \in KP$).*
2. *there is an argument for P that cannot be defeated by any non-misleading proposition; i.e. $\exists U \in \tau_E \setminus \{\emptyset\}$ s.t. $U \subseteq P$ and $B^Q U$ for all non-misleading $Q \subseteq X$.*
3. *there is an argument for P that is consistent with every non-misleading proposition; i.e. $\exists U \in \tau_E \setminus \{\emptyset\}$ s.t. $U \subseteq P$ and $U \cap Q \neq \emptyset$ for all non-misleading $Q \subseteq X$.*

3 Logics for Evidence, Belief and Knowledge

In this section, we present formal languages for evidence, belief and knowledge, and provide sound, complete and decidable proof systems for the resulting logics.

The *topological language* $\mathcal{L}$ is given by the following grammar

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid B\varphi \mid K\varphi \mid \forall\varphi \mid B^{\varphi}\varphi \mid \Box\varphi \mid E\varphi$$

where $p \in \text{Prop}$. We employ the usual abbreviations for propositional connectives $\top, \bot, \vee, \rightarrow, \leftrightarrow$, and for the dual modalities $\langle B\rangle$, $\langle K\rangle$, $\langle E\rangle$ etc., except that some of them have special abbreviations: $\exists\varphi := \langle\forall\rangle\varphi$ and $\Diamond\varphi := \langle\Box\rangle\varphi$.

Several fragments of $\mathcal{L}$ have special importance: $\mathcal{L}_B$ is the fragment having the belief B as the only modality; $\mathcal{L}_K$ has only the knowledge operator K; $\mathcal{L}_{KB}$ has only operators K and B; $\mathcal{L}_{\forall K}$ has only operators $\forall$ and K; $\mathcal{L}_{\forall\Box}$ has only operators $\forall$ and $\Box$.

We also consider an *extension* $\mathcal{L}_{E_0\Box_0}$ of $\mathcal{L}$, called the *evidence language*: this is obtained by extending $\mathcal{L}$ with *two new operators* E_0 *and* $\Box_0$. The expressivity of $\mathcal{L}_{E_0\Box_0}$ goes beyond purely topological properties: the meaning of E_0 and $\Box_0$ does not depend only on the topology, but also on the basic evidence family E_0. Finally, we will consider one very important fragment of $\mathcal{L}_{E_0\Box_0}$, namely the language $\mathcal{L}_{\forall\Box\Box_0}$ having only the operators $\forall$, $\Box$ and $\Box_0$. Its importance comes from that $\mathcal{L}_{\forall\Box\Box_0}$ is *co-expressive* with $\mathcal{L}_{E_0\Box_0}$.

The *semantics* for these languages is obvious: given a topo-e-model $\mathcal{M} = (X, E_0, \tau, V)$, we recursively extend the valuation map V to an interpretation map $||\varphi||$ for all formulas φ, by interpreting the Boolean connectives and the modalities using the corresponding semantic operators: e.g. $||\forall\varphi|| = \forall||\varphi||$, $||\Box\varphi|| = \Box||\varphi||$ etc.

Proposition 4. *The following equivalences are valid in all topo-e-models:*

1. $B\varphi \leftrightarrow \langle K\rangle K\varphi \leftrightarrow \exists K\varphi \leftrightarrow \forall\Diamond\Box\varphi$
2. $E\varphi \leftrightarrow \exists\Box\varphi$
3. $E_0\varphi \leftrightarrow \exists\Box_0\varphi$
4. $K\varphi \leftrightarrow \Box\varphi \wedge B\varphi \leftrightarrow \Box\varphi \wedge \forall\Diamond\Box\varphi$
5. $B^{\theta}\varphi \leftrightarrow \forall(\theta \rightarrow \Diamond(\theta \wedge \Box(\theta \rightarrow \varphi)))$
6. $\forall\varphi \leftrightarrow B^{\neg\varphi}\bot$

So, all the other modalities of $\mathcal{L}_{E_0\Box_0}$ can be defined in $\mathcal{L}_{\forall\Box\Box_0}$.

Theorem 1. *The system* $KD45$ *(for the* B *operator) is sound and complete for* $\mathcal{L}_B$*.*

Theorem 2. *The system* $S4.2$ *(for the* K *operator) is sound and complete for* $\mathcal{L}_K$*.*

Theorem 3. *A sound and complete axiomatization for* $\mathcal{L}_{KB}$ *is given by Stalnaker's system*[16] KB *in [18], consisting of the following:*

1. *the* $S4$ *axioms and rules for Knowledge* K

[16] This shows that the semantics in this paper correctly generalizes the one in [1,2,16] for the system KB.

2. *Consistency of Belief:* $B\phi \to \neg B\neg\phi$;
3. *Knowledge implies Belief:* $K\phi \to B\phi$;
4. *Strong Positive and Negative Introspection for Belief:* $B\phi \to KB\phi$; $\neg B\phi \to K\neg B\phi$;
5. *the "Strong Belief" axiom:* $B\phi \to BK\phi$.

Theorem 4 [8]. *The following system is sound and complete for* $\mathcal{L}_{\forall\Box}$:

1. *the S5 axioms and rules for* $\forall$
2. *the S4 axioms and rules for* $\Box$
3. $\forall\varphi \to \Box\varphi$

By Proposition 4, $\mathcal{L}_{\forall\Box}$ can define *all* the other operators of $\mathcal{L}$. So a complete system for $\mathcal{L}$ is obtained by adding the relevant axiom-definitions to the above system.

Theorem 5. *The following system is sound and complete for* $\mathcal{L}_{\forall K}$:

1. *the* $S5$ *axioms and rules for* $\forall$	3. $\forall\varphi \to K\varphi$
2. *the* $S4$ *axioms and rules for* K	4. $\exists K\varphi \to \forall\langle K\rangle\varphi$

Since belief is definable in $\mathcal{L}_{\forall K}$, a complete system for the language with this additional belief operator is obtained by adding the axiom-definition $B\varphi \leftrightarrow \exists K\varphi$ to the above system for $\mathcal{L}_{\forall K}$.

Theorem 6 (Soundness, Completeness, Finite Model Property and Decidability). *The logic* $\mathcal{L}_{\forall\Box\Box_0}$ *is* completely axiomatizable *and has the* finite model property, *and hence it is decidable. A complete axiomatization is given by the following system* $L_{\forall\Box\Box_0}$:

1. *the S5 axioms and rules for* $\forall$
2. *the S4 axioms and rules for* $\Box$
3. $\Box_0\varphi \to \Box_0\Box_0\varphi$
4. *the Monotonicity Rule for* $\Box_0$*: from* $\varphi \to \psi$*, infer* $\Box_0\varphi \to \Box_0\psi$
5. $\forall\varphi \to \Box_0\varphi$
6. $\Box_0\varphi \to \Box\varphi$
7. *the Pullout Axiom*[17]: $(\Box_0\varphi \wedge \forall\psi) \to \Box_0(\varphi \wedge \forall\psi)$

The proof of Theorem 6 is the most difficult result of the paper, and we present it in full in the Appendix. The key difficulty of the proof consists in guaranteeing that the natural topology for which $\Box$ acts as interior operator is exactly the topology generated by the neighborhood family associated to $\Box_0$. Though the main steps of the proof involve known methods (a canonical quasi-model construction, a filtration argument, and then making multiple copies of the worlds), addressing the above-mentioned difficulty requires an innovative use of these methods, and a careful treatment of each of the steps. The proofs of the other results are standard, and so are left for the extended version of this paper, available at http://www.illc.uva.nl/Research/Publications/Reports/.

[17] This axiom originates from [20], where it is stated as an equivalence rather than an implication. But the converse is provable in our system.

4 Further Developments and Future Work

The above-mentioned extended version contains an investigation of several types of evidential *dynamics* (building on the work in [21]), as well as complete axiomatizations of the corresponding dynamic-epistemic logics.

One line of further inquiry involves adding to the semantic structure a larger set $E_0^\Diamond \supseteq E_0$ of *potential evidence*, meant to encompass all the evidence that might be learnt in the future. This would connect well with the topological program in Inductive Epistemology [10], based on a learning-theoretic investigation of convergence of beliefs to the truth in the limit, when the agent observes a stream of incoming evidence.

We also plan to extend our framework to notions of *group knowledge* for a group G. There are at least two different natural options for *common knowledge*: the Aumann concept (the infinite conjunction of "everybody knows that everybody knows etc"), and Lewis' concept, based on *shared evidence* (the intersection $\bigcap_{a \in G} E_0^a$ of the evidence families E_0^a of all agents $a \in G$). Similarly, there are now two different models for a *group's epistemic potential*: the standard concept of distributed knowledge, versus the one obtained by *sharing the evidence* (i.e. taking the union $E_0^G = \bigcup_{a \in G} E_0^a$ of all the evidence families E_0^a).

Acknowledgments. We thank Johan van Benthem, Kevin Kelly, Thomas Icard and Eric Pacuit for feedback and discussions on the topic of this paper. We also thank to the anonymous referees of WoLLIC 2016 for their useful comments. A. Özgün's work was funded by the European Research Council grant EPS 313360. Sonja Smets' research was funded by the European Research Council under the European Community's Seventh Framework Programme (FP7/2007-2013)/ERC Grant agreement no. 283963.

A Appendix: Proof of Theorem 6

A *quasi-model* is a tuple $\mathcal{M} = (X, E_0, \leq, V)$, where: $E_0 \subseteq \mathcal{P}(X)$ satisfies the same constraints as a topo-e-model, V is a valuation, $\leq$ is a preorder s.t. every $e \in E_0$ is upward-closed wrt $\leq$. The *semantics* is the same as on topo-e-models, except that $\Box$ gets a Kripke semantics: $\|\Box\phi\| := \{x \in X \mid \forall y \in X(x \leq y \Rightarrow y \in \|\phi\|)\}$.

A quasi-model $\mathcal{M} = (X, E_0, \leq, V)$ is called *Alexandroff* if the topology τ_E is Alexandroff and $\leq = \sqsubseteq_E$ is the specialization preorder. There is a natural *bijection B between Alexandroff quasi-models and Alexandroff topo-e-models*, given by putting, for any Alexandroff quasi-model $\mathcal{M} = (X, E_0, \leq, V)$, $B(\mathcal{M}) := (X, E_0, \tau_E, V)$. Moreover, *$\mathcal{M}$ and $B(\mathcal{M})$ satisfy the same formulas of $\mathcal{L}_{\forall\Box\Box_0}$ at the same points*. So Alexandroff quasi-models are just another presentation of Alexandroff models.

Proposition 5. *Let $\mathcal{M} = (X, E_0, \leq, V)$ be a quasi-model. The following are equivalent:*

1. *$\mathcal{M}$ is Alexandroff (hence, equivalent to an Alexandroff topo-e-model);*

2. τ_E *coincides with the family of all upward-closed sets (with respect to* $\leq$*);*
3. for every $x \in X$*,* $\uparrow x$ *is in* τ_E*.*

Proof. (1 $\Rightarrow$ 3) Suppose $\mathcal{M}$ is Alexandroff, i.e., τ_E is Alexandroff and $\leq = \sqsubseteq_E$. Let $x \in X$. Then we have: $\uparrow x = \{y \mid x \leq y\} = \{y \mid x \sqsubseteq_E y\} = \{y \mid \forall U \in \tau_E (x \in U \Rightarrow y \in U)\} = \bigcap\{U \in \tau_E \mid x \in U\}$. Since τ_E is an Alexandroff space, we have $\bigcap\{U \in \tau_E \mid x \in U\} \in \tau_E$, and hence $\uparrow x = \bigcap\{U \in \tau_E \mid x \in U\} \in \tau_E$.
(3 $\Rightarrow$ 2) Let $Up(X)$ be the set of all upward-closed subsets of X. It is easy to see that $\tau_E \subseteq Up(X)$ (since τ_E is generated by E_0 and every element of E_0 is upward-closed). Now let $A \in Up(X)$. Since A is upward-closed, we have $A = \bigcup\{\uparrow x \mid x \in A\}$. Then, by (3) (and τ_E being closed under arbitrary unions), we obtain $A \in \tau_E$.
(2 $\Rightarrow$ 1) Suppose (2) and let $\mathcal{A} \subseteq \tau_E$. By (2), every $U \in \mathcal{A}$ is upward-closed; hence, $\bigcap \mathcal{A}$ is upward-closed, so by (2) $\bigcap \mathcal{A} \in \tau_E$. This proves that τ_E is Alexandroff. (2) also implies that $\uparrow x$ is the least open neighbourhood of x in τ_E, i.e., $\uparrow x \subseteq U$, for all U such that $x \in U \in \tau_E$. Therefore, $\leq \subseteq \sqsubseteq_E$. For the other direction, suppose $x \sqsubseteq_E y$. This implies, in particular, $y \in \uparrow x$ (since $x \in \uparrow x \in \tau_E$), i.e., $x \leq y$.

The proof of Theorem 6 goes through *three steps*: (1) strong completeness for quasi-models; (2) finite quasi-model property; (3) every finite quasi-model is modally equivalent to a finite Alexandroff quasi-model (hence, to a topo-e-model).

Proposition 6 *(STEP 1).* $L_{\forall\Box\Box_0}$ *is sound and strongly complete for quasi-models.*

Proof. Soundness is easy. Completeness goes via a canonical quasi-model:

Lemma 1 *(*Lindenbaum Lemma*). Every consistent set of sentences in* $\mathcal{L}_{\forall\Box\Box_0}$ *can be extended to a maximally consistent one.*

Proof. Standard.

Let us now fix a consistent set of sentence Φ_0. Our goal is to construct a quasi-model for Φ_0. By Lemma 1, there exists a maximally consistent theory T_0 s. t, $\Phi_0 \subseteq T_0$. For any two maximally consistent theories T and S, we put: $T \sim S$ iff for all $\phi \in \mathcal{L}_{\forall\Box\Box_0} : ((\forall\phi) \in T \Rightarrow \phi \in S)$; and $T \leq S$ iff for all $\phi \in \mathcal{L}_{\forall\Box\Box_0}$: $((\Box\phi) \in T \Rightarrow \phi \in S)$.

Canonical Quasi-Model for T_0. This is a structure $\mathcal{M} = (X, E_0, \leq, V)$, where: $X := \{T : T$ maximally consistent theory with $T \sim T_0\}$; $E_0 := \{\widehat{\Box_0\phi} : \phi \in \mathcal{L}_{\forall\Box\Box_0}$ with $(\exists\Box_0\phi) \in T_0\}$, where we used notation $\hat{\theta} := \{T \in X : \theta \in T\}$; $\leq$ is the restriction of the above preorder $\leq$ to X; and $V(p) := \hat{p}$. In the following, variables $T, S, \ldots$ range over X.

Lemma 2. $\mathcal{M}$ *is a quasi-model.*

Proof. Easy verification.

Lemma 3 *(Existence Lemma for $\forall$). $\widehat{\exists\varphi} \neq \emptyset$ iff $\hat{\varphi} \neq \emptyset$.*

Proof. Easy (along standard lines of the so-called Diamond Lemma for $\exists$).

Lemma 4 *(Existence Lemma for $\Box$). $T \in \widehat{\Diamond\varphi}$ iff $(\exists)\, S \in \hat{\varphi}$ s. t. $T \leq S$.*

Proof. Standard again.

Lemma 5 *(Existence Lemma for $\Box_0$). $T \in \widehat{\Box_0\varphi}$ iff $(\exists)\, e \in E_0$ s. t. $T \in e \subseteq \hat{\varphi}$.*

Proof. Left-to-right: Assume $T \in \widehat{\Box_0\varphi}$, i.e. $(\Box_0\varphi) \in T$. From $T \in X$ and $T \sim T_0$ we get $(\exists\Box_0\varphi) \in T_0$. Taking $e := \widehat{\Box_0\varphi}$, we get $e \in E_0$ and $T \in e$. To show that $e \subseteq \hat{\varphi}$, we use the theorem $\Box_0\varphi \to \varphi$, which implies that $\widehat{\Box_0\varphi} \subseteq \hat{\varphi}$, i.e. $e \subseteq \hat{\varphi}$.

Right-to-Left: Let $T \in X$ and $e \in E_0$, s.t. $T \in e \subseteq \hat{\varphi}$. Then $e = \widehat{\Box_0\theta}$ for some θ s.t. $(\exists\Box_0\theta) \in T_0$. So $T \in e = \widehat{\Box_0\theta} \subseteq \hat{\varphi}$. We now prove the following:

Claim: *The set $\Gamma := \{\Box_0\theta\} \cup \{\forall\psi : \forall\psi \in T\} \cup \{\neg\varphi\}$ is inconsistent.*

Proof of Claim: Suppose that $\Gamma \nvdash \perp$. By Lemma 1, there exists some $S \in X$ s. t. $\Gamma \subseteq S$. From $(\neg\varphi) \in S$ we get $S \notin \hat{\varphi}$ (by the consistency of S), and from $(\Box_0\theta) \in S$ we get $S \in \widehat{\Box_0\theta}$. So $S \in \widehat{\Box_0\theta} \setminus \hat{\varphi}$, contradicting $\widehat{\Box_0\theta} \subseteq \hat{\varphi}$.

Given the Claim, there exists a *finite* $\Gamma_0 \subseteq \Gamma$ with $\Gamma_0 \vdash \perp$. By the theorem $(\forall\psi_1 \wedge \ldots \forall\psi_n) \leftrightarrow \forall(\psi_1 \wedge \ldots \psi_n)$, we can assume that $\Gamma_0 = \{\Box_0\theta, \forall\psi, \neg\varphi\}$, for some ψ s. t. $(\forall\psi) \in T$. From $\Gamma_0 \vdash \perp$ we get the theorem $(\Box_0\theta \wedge \forall\psi) \to \varphi$. Using the Monotonicity Rule for $\Box_0$, the formula $\Box_0(\Box_0\theta \wedge \forall\psi) \to \Box_0\varphi$ is also a theorem. From the axiom $\Box_0\theta \to \Box_0\Box_0\theta$ and the Pullout Axiom, we get the theorem $(\Box_0\theta \wedge \forall\psi) \to \Box_0\varphi$. Since $(\Box_0\theta) \in T$ and $(\forall\psi) \in T$, it follows that $(\Box_0\varphi) \in T$, i.e. $T \in \widehat{\Box_0\varphi}$, as desired.

Lemma 6 (Truth Lemma). *For every formula $\phi \in \mathcal{L}_{\forall\Box\Box_0}$, we have: $\|\phi\|_{\mathcal{M}} = \hat{\phi}$.*

Proof. Standard proof by induction on the complexity of ϕ.

Consequence: $T_0 \models_{\mathcal{M}} \Phi_0$. This proves Step 1 (Proposition 6).

Theorem 7 *(STEP 2). The logic $\mathcal{L}_{\forall\Box\Box_0}$ has Strong Finite Quasi-Model Property.*

Proof of Theorem 7: Let ϕ_0 be a consistent formula. By Step 1, take T_0 a maximal consistent theory s.t. $\phi_0 \in T_0$, and let $\mathcal{M} = (X, E_0, \leq, V)$ be the canonical quasi-model for T_0. We will use two facts about this model:

1. $\|\varphi\|_{\mathcal{M}} = \hat{\varphi}$, for all $\varphi \in \mathcal{L}_{\forall\Box\Box_0}$,
2. $E_0 = \{\widehat{\Box_0\varphi} : (\exists\Box_0\varphi) \in T_0\} = \{\|\Box_0\varphi\|_{\mathcal{M}} : (\exists\Box_0\varphi) \in T_0\}$.

Let Σ be a *finite* set such that: (1) $\phi_0 \in \Sigma$; (2) Σ is closed under subformulas; (3) if $(\Box_0\varphi) \in \Sigma$ then $(\Box\Box_0\varphi) \in \Sigma$; (4) Σ is closed under single negations; (5) $(\Box_0\top) \in \Sigma$. For $x, y \in X$, put: $x \equiv_\Sigma y$ iff $\forall\psi \in \Sigma(x \in \|\psi\|_{\mathcal{M}} \iff y \in \|\psi\|_{\mathcal{M}})$, and denote by $|x| := \{y \in X : x \equiv_\Sigma y\}$ the equivalence class of x

modulo $\equiv_\Sigma$. Also, put $X^f := \{|x| : x \in X\}$, and more generally put $e^f := \{|x| : x \in e\}$ for every $e \in E_0$.

We now define a "*filtrated model*" $\mathcal{M}^f = (X^f, E_0^f, \leq^f, V^f)$, by taking: as set of worlds the set X^f (of equivalence classes) defined above; as for the rest, we put: $|x| \leq^f |y|$ iff for all $(\Box\psi) \in \Sigma : (x \in \|\Box\psi\|_\mathcal{M} \Rightarrow y \in \|\Box\psi\|_\mathcal{M})$; $E_0^f := \{e^f : e = \widehat{\Box_0\psi} = \|\Box_0\psi\|_\mathcal{M} \in E_0$ for some ψ s. t $(\Box_0\psi) \in \Sigma\}$; $V^f(p) := \{|x| : x \in V(p)\}$.

Lemma 7. *$\mathcal{M}^f$ is a finite quasi-model (of size bounded by a computable function of ϕ_0).*

Proof. X^f is finite, since Σ is finite so there are only finitely many equivalence classes modulo $\equiv_\Sigma$. In fact, the size is at most $2^{|\Sigma|}$. It's obvious that $\leq^f$ is a preorder, that $X^f \in E_0^f$ (since $X = \|\Box_0\top\|_\mathcal{M}$ and $(\Box_0\top) \in \Sigma$, so $X^f \in E_0^f$) and that every $e^f \in E_0^f$ is non-empty (since it comes from some non-empty $e \in E_0$). So we only have to prove that the evidence sets are upward–closed: for this, let $e^f \in E_0^f$, with $e = \widehat{\Box_0\psi} \in E_0$, $(\Box_0\psi) \in \Sigma$ and let $|x| \in e^f$ and $|y| \in X^f$ s.t. $|x| \leq^f |y|$. We need to show that $|y| \in e^f$.

Since $|x| \in e^f$, there exists some $x' \equiv_\Sigma x$ s.t. $x' \in \widehat{\Box_0\psi} = \|\Box_0\psi\|_\mathcal{M}$. From $(\Box_0\psi) \in \Sigma$ and $x' \equiv_\Sigma x$, we get $x \in \|\Box_0\psi\|_\mathcal{M}$. By the theorem $\Box_0\psi \to \Box\Box_0\psi$, we have $x \in \|\Box\Box_0\psi\|_\mathcal{M}$. But $(\Box\Box_0\psi) \in \Sigma$ (by the closure assumptions on Σ), so $|x| \leq^f |y|$ gives us $y \in \|\Box\Box_0\psi\|_\mathcal{M}$. By the T-axiom $\Box\phi \to \phi$, we get $y \in \|\Box_0\psi\|_\mathcal{M} = \widehat{\Box_0\psi} = e$, hence $|y| \in e^f$.

Lemma 8 (Filtration Lemma). *For every formula $\phi \in \Sigma$: $\|\phi\|_{\mathcal{M}^f} = \{|x| : x \in \|\phi\|_\mathcal{M}\}$.*

Proof. Proof by induction on $\phi \in \Sigma$. The atomic case, inductive cases for propositional connectives and modalities $\forall\phi$ and $\Box\phi$ are treated as usual (-in the last case using the filtration property of $\leq^f$). We only prove here the inductive case for the modality $\Box_0\phi$:

Left-to-right inclusion: Let $|x| \in \|\Box_0\phi\|_{\mathcal{M}^f}$. This means that there exists some $e^f \in E_0^f$ s.t. $|x| \in e^f \subseteq \|\phi\|_{\mathcal{M}^f}$. By the definition of E_0^f, there exists some ψ s.t.: $(\Box_0\psi) \in \Sigma$ and $e = \widehat{\Box_0\psi} = \|\Box_0\psi\|_\mathcal{M} \in E_0$. From $|x| \in e^f$, it follows that there is some $x' \equiv_\Sigma x$ s.t. $x' \in e = \|\Box_0\psi\|_\mathcal{M}$, and since $(\Box_0\psi) \in \Sigma$, we have $x \in \|\Box_0\psi\|_\mathcal{M} = e$. It is easy to see that we also have $e \subseteq \|\phi\|_\mathcal{M}$. (Indeed, let $y \in e$ be *any* element of e; then $|y| \in e^f \subseteq \|\phi\|_{\mathcal{M}^f}$, so $|y| \in \|\phi\|_{\mathcal{M}^f}$, and by the induction hypothesis $y \in \|\phi\|_\mathcal{M}$.) So we have found an evidence set $e \in E_0$ s.t. $x \in e \subseteq \|\phi\|_\mathcal{M}$, i.e., shown that $x \in \|\Box_0\phi\|_\mathcal{M}$.

Right-to-left inclusion: Let $x \in \|\Box_0\phi\|_\mathcal{M}$, with $(\Box_0\phi) \in \Sigma$. It is easy to see that $(\exists\Box_0\phi) \in x$ (by the theorem $\Box_0\phi \to \exists\Box_0\phi$) and so also $(\exists\Box_0\phi) \in T_0$ (since $x \in X$ so $x \sim T_0$). This means that the set $e := \widehat{\Box_0\phi} = \|\Box_0\phi\|_\mathcal{M} \in E_0$ is an evidence set in the canonical model, and since $(\Box_0\phi) \in \Sigma$, we conclude that $e^f \in E_0^f$ is an evidence set in the filtrated model. We obviously have $x \in e$, and so $|x| \in e^f$. By the (T) axiom, $e = \|\Box_0\phi\|_\mathcal{M} \subseteq \|\phi\|_\mathcal{M}$, and hence $e^f \subseteq \{|y| : y \in \|\phi\|_\mathcal{M}\} = \|\phi\|_{\mathcal{M}^f}$ (by the induction hypothesis). Thus, we have found $e^f \in E_0^f$ s.t. $|x| \in e^f \subseteq \|\phi\|_{\mathcal{M}^f}$, i.e., shown that $|x| \in \|\Box_0\phi\|_{\mathcal{M}^f}$.

Theorem 8 *(STEP 3). Every finite quasi-model is modally equivalent to a finite Alexandroff quasi-model (and so to a topo-e-model).*

Proof of Theorem 8: Let $\mathcal{M} = (X, E_0, \leq, V)$ be a finite quasi-model. We form a new structure $\tilde{\mathcal{M}} = (\tilde{X}, \tilde{E_0}, \tilde{\leq}, \tilde{V})$, by putting: $\tilde{X} := X \times \{0,1\}$; $\tilde{V}(p) := V(p) \times \{0,1\}$; $(x,i)\tilde{\leq}(y,j)$ iff: $x \leq y$ and $i = j$; $\tilde{E_0} := \{e_i : e \in E_0, i \in \{0,1\}\} \cup \{e_i^y : y \in e \in E_0, i \in \{0,1\}\} \cup \{\tilde{X}\}$, where we used notations $e_i := e \times \{i\} = \{(x,i) : x \in e\}$ and $e_i^y := \uparrow y \times \{i\} \cup e \times \{1-i\} = \{(x,i) : y \leq x\} \cup e_{1-i}$.

Lemma 9. *$\tilde{\mathcal{M}}$ is a (finite) quasi-model.*

Proof. Easy verification.

Notation: For any set $\tilde{Y} \subseteq \tilde{X}$, put $\tilde{Y}_X := \{y \in X : (y,i) \in \tilde{Y}$ for some $i \in \{0,1\}\}$ for the set consisting of first components of all members of $\tilde{Y}$. It is easy to see that we have: $(\tilde{Y} \cup \tilde{Z})_X = \tilde{Y}_X \cup \tilde{Z}_X$, and $\tilde{X}_X = X$.

Lemma 10. *If $y \in e \in E_0$, $i \in \{0,1\}$ and $\tilde{e} \in \{e_i, e_i^y\}$, then we have:*

1. $\tilde{e}_X = e$;
2. $e_i^y \cap e_i = \uparrow(y,i)$, *where* $\uparrow(y,i) = \{\tilde{x} \in \tilde{X} : (y,i)\tilde{\leq}\tilde{x}\} = \{(x,i) : y \leq x\}$.

Proof

1. If $\tilde{e} = e_i$, then $\tilde{e}_X = (e \times \{i\})_X = e$. If $\tilde{e} = e_i^y$, then $\tilde{e}_X = (\uparrow y \times \{i\})_X \cup (e \times \{1-i\})_X = \uparrow y \cup e = e$ (since e is upward-closed and $y \in e$, so $\uparrow y \subseteq e$).
2. $e_i^y \cap e_i = (\uparrow y \times \{i\} \cup e \times \{1-i\}) \cap (e \times \{i\}) = (\uparrow y \cap e) \times \{i\} = \uparrow y \times \{i\} = \uparrow(y,i)$ (since $\uparrow y \subseteq e$).

Lemma 11. *$\tilde{\mathcal{M}}$ is an Alexandroff quasi-model (and thus also a topo-e-model).*

Proof. By Proposition 5, it is enough to show that, for every $(y,i) \in \tilde{X}$, the upward-closed set $\uparrow(y,i)$ is open in the topology τ_E generated by E_0. But this follows directly from part 2 of Lemma 10.

Lemma 12 (Modal-Equivalence Lemma). *For all $\varphi \in \mathcal{L}_{\forall\Box\Box_0}$: $\|\varphi\|_{\tilde{\mathcal{M}}} = \|\varphi\|_{\mathcal{M}} \times \{0,1\}$.*

Proof. Induction on φ. The base case, and the inductive steps for Boolean connectives and operators $\forall$ and $\Box$, are straightforward. We only prove the inductive step for $\Box_0$:

Left-to-Right Inclusion: Suppose that $(x,i) \in \|\Box_0\varphi\|_{\tilde{\mathcal{M}}}$. Then there exists some $\tilde{e} \in \tilde{E}$ such that $(x,i) \in \tilde{e} \subseteq \|\varphi\|_{\tilde{\mathcal{M}}} = \|\varphi\|_{\mathcal{M}} \times \{0,1\}$ (where we used the induction hypothesis for φ at the last step). From this, we obtain that $x \in \tilde{e}_X \subseteq (\|\varphi\|_{\mathcal{M}} \times \{0,1\})_X = \|\varphi\|_{\mathcal{M}}$. But by the construction of $\tilde{E}$, $\tilde{e} \in \tilde{E}$ means that either $\tilde{e} = \tilde{X}$ or there exist $e \in E_0$, $y \in e$ and $j \in \{0,1\}$ such that $\tilde{e} \in \{e_j, e_j^y\}$. If the former is the case, we have $x \in \tilde{e}_X = X \subseteq \|\varphi\|_{\mathcal{M}}$. Since $X \in E_0$, by the semantics of $\Box_0$, we obtain $x \in \|\Box_0\varphi\|_{\mathcal{M}}$. If the latter is the case, by part

1 of Lemma 10, we have $\tilde{e}_X = e$, so we conclude that $x \in \tilde{e}_X = e \subseteq \|\varphi\|_{\mathcal{M}}$. Therefore, again by the semantics of $\Box_0$, we have $x \in \|\Box_0\varphi\|_{\mathcal{M}}$.

Right-to-Left Inclusion: Suppose that $x \in \|\Box_0\varphi\|_{\mathcal{M}}$. Then there exists some $e \in E_0$ such that $x \in e \subseteq \|\varphi\|_{\mathcal{M}}$. Take now the set $e_i = e \times \{i\} \in \tilde{E}$. Clearly, we have $(x, i) \in e_i \subseteq \|\varphi\|_{\mathcal{M}} \times \{i\} \subseteq \|\varphi\|_{\mathcal{M}} \times \{0, 1\} = \|\varphi\|_{\tilde{\mathcal{M}}}$ (where we used the induction hypothesis for φ at the last step), i.e. we have $(x, i) \in \|\Box_0\varphi\|_{\tilde{\mathcal{M}}}$.

Theorem 8 follows immediately from the above Lemma: the same formulas are satisfied at x in $\mathcal{M}$ as at (x, i) in $\tilde{\mathcal{M}}$. Theorem 6 is an immediate corollary of Theorem 8.

References

1. Baltag, A., Bezhanishvili, N., Özgün, A., Smets, S.: The Topology of belief, belief revision and defeasible knowledge. In: Grossi, D., Roy, O., Huang, H. (eds.) LORI. LNCS, vol. 8196, pp. 27–40. Springer, Heidelberg (2013)
2. Baltag, A., Bezhanishvili, N.: Özgün, A., Smets, S.: The topological theory of belief (2015). http://www.illc.uva.nl/Research/Publications/Reports/PP-2015-18.text.pdf (Submitted)
3. Baltag, A., Smets, S.: Conditional doxastic models: a qualitative approach to dynamic belief revision. In: Proceedings of WOLLIC, vol. 165, pp. 5–21 (2006)
4. Baltag, A., Smets, S.: A qualitative theory of dynamic interactive belief revision. Texts Logic Games **3**, 9–58 (2008)
5. Clark, M.: Knowledge and grounds: a comment on Mr. Gettier's paper. Analysis **24**, 46–48 (1963)
6. Foley, R.: When is True Belief Knowledge?. Princeton University Press, Princeton (2012)
7. Gettier, E.: Is justified true belief knowledge? Analysis **23**, 121–123 (1963)
8. Goranko, V., Passy, S.: Using the universal modality: gains and questions. J. Log. Comput. **2**, 5–30 (1992)
9. Grove, A.: Two modellings for theory change. J. Phil. Logic **17**, 157–170 (1988)
10. Kelly, K.: The Logic of Reliable Inquiry. Oxford University Press, Oxford (1996)
11. Klein, P.: A proposed definition of propositional knowledge. J. Philos. **68**, 471–482 (1971)
12. Klein, P.: Certainty, a Refutation of Scepticism. University of Minneapolis Press, Minneapolis (1981)
13. Lehrer, K.: Theory of Knowledge. Routledge, London (1990)
14. Lehrer, K., Paxson, T.J.: Knowledge: undefeated justified true belief. J. Philos. **66**, 225–237 (1969)
15. McKinsey, J.C.C., Tarski, A.: The algebra of topology. Ann. of Math. **2**(45), 141–191 (1944)
16. Özgün, A.: Topological Models for Belief and Belief Revision. Master's thesis, University of Amsterdam, Amsterdam, The Netherlands (2013)
17. Rott, H.: Stability, strength and sensitivity: converting belief into knowledge. Erkenntnis **61**, 469–493 (2004)
18. Stalnaker, R.: On logics of knowledge and belief. Phil. Studies **128**, 169–199 (2006)
19. van Benthem, J.: Dynamic logic for belief revision. JANCL **17**, 129–155 (2007)
20. van Benthem, J., Fernández-Duque, D., Pacuit, E.: Evidence and plausibility in neighborhood structures. Ann. Pure Appl. Logic **165**, 106–133 (2014)
21. van Benthem, J., Pacuit, E.: Dynamic logics of evidence-based beliefs. Studia Logica **99**(1), 61–92 (2011)

Semantic Acyclicity for Conjunctive Queries: Approximations and Constraints

Pablo Barceló(✉)

DCC, Center for Semantic Web Research, University of Chile, Santiago, Chile
pbarcelo@dcc.uchile.cl

Abstract. Evaluation of conjunctive queries (CQs) is NP-complete, but becomes tractable for syntactically defined fragments. One of the oldest and most studied such fragments is the class of acyclic CQs. Here we look at the problem of semantic acyclicity, i.e., given a CQ q, is there an acyclic CQ q' that is equivalent to it? This notion is important in CQ evaluation, as semantically acyclic CQs can be evaluated in polynomial time. The notion of semantic acyclicity itself is decidable, with the same complexity as the usual static analysis tasks for CQs, i.e., NP-complete.

Unfortunately, semantic acyclic is not general enough for practical purposes, as only CQs whose core is acyclic belong to this class. In this tutorial we present two approaches that have been developed to make the notion more flexible and take better advantage of the ideas that underlie it. These are computing approximations and making use of semantic information in the form of constraints. For approximations, we look at the case when q is not semantically acyclic and explain how to find and evaluate those acyclic CQs q' that are as "close" as possible to q in terms of containment. As for constraints, they enrich semantic acyclicity since they can be applied on a CQ q to produce an acyclic reformulation of it. We present results that establish the boundary of decidability for semantic acyclicity under usual database constraints such as tuple and equality-generating dependencies, and show their applicability in query evaluation.

1 Extended Abstract

Query optimization is a fundamental database task that amounts to transforming a query into one that is arguably more efficient to evaluate. The database theory community has developed several principled methods for optimization of conjunctive queries (CQs), many of which are based on *static-analysis* tasks such as containment [1]. In a nutshell, such methods compute a *minimal* equivalent version of a CQ, where minimality refers to number of atoms. As argued by Abiteboul, Hull, and Vianu [1], this provides a theoretical notion of "true optimality" for the reformulation of a CQ, as opposed to practical considerations based on heuristics. For each CQ q the minimal equivalent CQ is its *core* q' [15]. Although the static analysis tasks that support CQ minimization are NP-complete [9], this is not a major problem for real-life applications, as the input (the CQ) is small.

J. Väänänen et al. (Eds.): WoLLIC 2016, LNCS 9803, pp. 104–108, 2016.
DOI: 10.1007/978-3-662-52921-8_7

An important shortcoming of the previous approach, however, is that there is no theoretical guarantee that the minimized version of a CQ is in fact easier to evaluate (recall that, in general, CQ evaluation is NP-complete [9]). We know, on the other hand, quite a bit about classes of CQs which can be evaluated efficiently. It is thus a natural problem to ask whether a CQ can be reformulated as one in such tractable classes, and if so, what is the cost of computing such reformulation. Following Abiteboul et al., this would provide us with a theoretical guarantee of "true efficiency" for those reformulations. Here we concentrate on one of the oldest and most studied tractability conditions for CQs; namely, *acyclicity*. It is known that acyclic CQs can be evaluated in linear time [19].

More formally, we study the following problem (we write $q \equiv q'$ to denote that q and q' are *equivalent*, i.e., they have the same output over every database):

PROBLEM :	SEMANTIC ACYCLICITY
INPUT :	A CQ q
QUESTION :	Is there an acyclic CQ q' s.t. $q \equiv q'$?

Basic properties of semantic acyclicity, such as the complexity of (1) checking whether a CQ is semantically acyclic, and (2) evaluating semantically acyclic CQs, are by now well-understood. In particular:

1. It is known that a CQ q is semantically acyclic iff its core q' is acyclic (recall that such q' is the minimal equivalent CQ to q). It follows that checking semantic acyclicity of CQs is NP-complete (see, e.g., [5,11]).
2. Regarding evaluation, semantically acyclic CQs can be evaluated efficiently [10,11,14].

Item (1) tells us that if q is semantically acyclic, then the only reason why q is not acyclic in the first hand is because it has not been minimized. Therefore, semantic acyclicity is not really different from usual minimization, which severely restricts its applicability in practical scenarios. Two approaches have been developed in the literature to enrich this notion and take better advantage of its underlying ideas. These are computing approximations and make use of semantic information in the form of constraints, as we explain next.

1.1 Approximations

When a CQ q is not semantically acyclic, it might be convenient to compute an acyclic *approximation* of it. This corresponds to an acyclic CQ q' such that (1) q' never returns false answers with respect to q, and (2) q' is as "close" as possible to q among all acyclic CQs that satisfy q. In order to satisfy (1), we need q' to be contained in q (denoted $q' \subseteq q$), which means that the result of q' is contained in that of q over every database. To formalize (2), the following definition is often used in the literature [4,5]:

$$\text{there is no acyclic CQ}\, q'' \,\text{such that}\, q' \subset q'' \subseteq q.$$

That is, there is no acyclic CQ q'' that is closer to q than q' with respect to the partial order defined by the containment relation $\subseteq$.

We will present the following important properties of acyclic approximations based on [4]:

– *Approximations always exist:* Every CQ q has an acyclic overapproximation. Moreover, the set of all acyclic approximations of a CQ q can be computed in exponential time.
– *Evaluating approximations is fixed-parameter tractable:* In particular, computing the results of all approximations of q over a database D can be done in time $|D| \cdot 2^{O(|q|)}$. When D is large, this constitutes an important improvement over the general cost of CQ evaluation, which is $|D|^{O(|q|)}$.

We also present an exponential lower bound on the number of acyclic approximations of CQs and establish DP-completeness of the problem of checking whether q' is an acyclic approximation of q.

1.2 Taking Advantage of Constraints

It is known that semantic information about the data, in the form of integrity constraints, alleviates query optimization by reducing the space of possible reformulations. Here we concentrate on the two most important classes of database constraints; namely, *tuple-generating dependencies* (tgds) and *equality-generating dependencies* (egds).

Earlier, we defined CQ equivalence over *all* databases. Adding constraints yields a refined notion of CQ equivalence, which holds over those databases that satisfy a given set of constraints only. But finding a minimal equivalent CQ in this context is notoriously more difficult than before. This is because basic static analysis tasks such as containment become undecidable when considered in full generality. This motivated a long research program for finding larger "islands of decidability" of such containment problem, based on syntactical restrictions on constraints [2,6–8,16,17].

It is an easy observation that the presence of constraints enriches the notion of semantic acyclicity. This is because constraints can be applied on CQs to produce acyclic reformulations of them. We present basic properties of semantic acyclicity in the presence of constraints based on recent results [3,13]. More in particular, we study the following problems:

– Decidability: For which classes of tgds and egds is the problem of semantic acyclicity decidable? In such cases, what is the computational complexity of the problem?
– Evaluation: What is the computational cost of evaluating semantically acyclic CQs under constraints?

Semantic Acyclicity Under tgds. We notice that having a decidable CQ containment problem is a necessary condition for semantic acyclicity to be decidable under tgds. Surprisingly enough, it is not a sufficient condition. This means that, contrary to what one might expect, there are natural classes of tgds for which CQ containment but not semantic acyclicity is decidable. In particular, this is the case for the well-known class of *full* tgds (i.e., Datalog programs). In conclusion, we cannot directly export techniques from CQ containment to deal with semantic acyclicity.

In view of the previous result, we concentrate on classes of tgds that (a) have a decidable CQ containment problem, and (b) do not contain the class of full tgds. These restrictions are satisfied by several expressive languages considered in the literature. Such languages can be classified into three main families depending on the techniques used for studying their containment problem: (i) *guarded* tgds [6], which contain inclusion and linear dependencies, (ii) *non-recursive* [12], and (iii) *sticky* sets of tgds [7]. We show that for all of them semantic acyclicity is decidable; more in particular, it is (a) 2EXPTIME-complete for guarded tgds (and NP-complete for a fixed schema), and (b) in NEXPTIME for both non-recursive and sticky sets of tgds (and again NP-complete if the schema is fixed).

Semantic Acyclicity Under egds. We show that semantic acyclicity under the important class of egds defined by *unary functional dependencies* is decidable (NP-complete). The latter has been independently established, and generalized in a nontrivial way, in a recent paper by Figueira [13]. Decidability for general egds remains open.

Evaluation. It is possible to show that for tgds for which semantic acyclicity is decidable (guarded, non-recursive, sticky) there is a fixed-parameter tractable algorithm for evaluating q on a database. No such algorithm is believed to exist for CQ evaluation in general [18]; thus, semantically acyclic CQs under these constraints behave better than the general case in terms of evaluation.

Recall, on the other hand, that in the absence of constraints one can do better: Evaluating semantically acyclic CQs in such context is in polynomial time. It is natural to ask if this also holds in the presence of constraints. We show this to be the case for guarded tgds and functional dependencies. For the other classes the problem remains to be investigated.

References

1. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison-Wesley, Boston (1995)
2. Baget, J.-F., Mugnier, M.-L., Rudolph, S., Thomazo, M.: Walking the complexity lines for generalized guarded existential rules. In: IJCAI, pp. 712–717 (2011)
3. Barceló, P., Gottlob, G., Pieris, A.: Semantic acyclicity under constraints. In: PODS (2016)
4. Barceló, P., Libkin, L., Romero, M.: Efficient approximations of conjunctive queries. SIAM J. Comput. **43**(3), 1085–1130 (2014)
5. Barceló, P., Romero, M., Vardi, M.Y.: Semantic acyclicity on graph databases. In: PODS, pp. 237–248 (2013)
6. Calì, A., Gottlob, G., Kifer, M.: Taming the infinite chase: query answering under expressive relational constraints. J. Artif. Intell. Res. **48**, 115–174 (2013)

7. Calì, A., Gottlob, G., Pieris, A.: Towards more expressive ontology languages: the query answering problem. Artif. Intell. **193**, 87–128 (2012)
8. Calvanese, D., De Giacomo, G., Lenzerini, M.: Conjunctive query containment and answering under description logic constraints. ACM Trans. Comput. Log. **9**(3), 22.1–22.31 (2008)
9. Chandra, A.K., Merlin, P.M.: newblock Optimal implementation of conjunctive queries in relational data bases. In: STOC, pp. 77–90 (1977)
10. Chen, H., Dalmau, V.: Beyond hypertree width: Decomposition methods without decompositions. In: CP, pp. 167–181 (2005)
11. Dalmau, V., Kolaitis, P.G., Vardi, M.Y.: Constraint satisfaction, bounded treewidth, and finite-variable logics. In: CP, pp. 310–326 (2002)
12. Fagin, R., Kolaitis, P.G., Miller, R.J., Popa, L.: Data exchange: semantics and query answering. Theor. Comput. Sci. **336**(1), 89–124 (2005)
13. Figueira, D.: Semantically acyclic conjunctive queries under functional dependencies. In: LICS (2016)
14. Gottlob, G., Greco, G., Marnette, B.: HyperConsistency width for constraint satisfaction: algorithms and complexity results. In: Lipshteyn, M., Levit, V.E., McConnell, R.M. (eds.) Graph Theory, Computational Intelligence and Thought. LNCS, vol. 5420, pp. 87–99. Springer, Heidelberg (2009)
15. Hell, P., Nešetřil, J.: Graphs and Homomorphisms. Oxford University Press, Oxford (2004)
16. Johnson, D.S., Klug, A.C.: Testing containment of conjunctive queries under functional and inclusion dependencies. J. Comput. Syst. Sci. **28**(1), 167–189 (1984)
17. Krötzsch, M., Rudolph, S.: Extending decidable existential rules by joining acyclicity and guardedness. In: IJCAI, pp. 963–968 (2011)
18. Papadimitriou, C.H., Yannakakis, M.: On the complexity of database queries. J. Comput. Syst. Sci. **58**(3), 407–427 (1999)
19. Yannakakis, M.: Algorithms for acyclic database schemes. In: VLDB, pp. 82–94 (1981)

Expressivity of Many-Valued Modal Logics, Coalgebraically

Marta Bílková and Matěj Dostál(✉)

Institute of Computer Science, The Czech Academy of Sciences,
Prague, Czech Republic
dostamat@math.feld.cvut.cz

Abstract. We apply methods developed to study coalgebraic logic to investigate expressivity of many-valued modal logics which we consider as coalgebraic languages interpreted over set-coalgebras with many-valued valuations. The languages are based on many-valued predicate liftings. We provide a characterization theorem for a language generated by a set of such modalities to be expressive for bisimilarity: in addition to the usual condition on the set of predicate liftings being separating, we indicate a sufficient and sometimes also necessary condition on the algebra of truth values which guarantees expressivity. Thus, adapting results of Schröder [16] concerning expressivity of boolean coalgebraic logics to many-valued setting, we generalize results of Metcalfe and Martí [13], concerning Hennessy-Milner property for many-valued modal logics based on $\Box$ and $\Diamond$.

1 Introduction

The abstract theory of coalgebras has recently become one of the most important bridges connecting modal logic and computer science: from a logician's point of view it provides techniques and a new level of generality for studying various modal logics, while from a computer-scientist's point of view it provides a general framework for designing expressive modal languages describing behavior of abstract transition systems modeled as coalgebras. It is then natural to ask what benefits a coalgebraic approach brings to study of many-valued modal logics: from a logician's point of view we can generalize logics of many-valued Kripke-style relational semantics [2,13], where valuations and the accessibility relation take values in a given algebra $\mathscr{V}$, to the coalgebraic level, while from a computer-scientist's point of view we generalize coalgebraic logics to the many-valued setting, allowing for a many-valued observable phenomena to be captured by modal languages with genuinely many-valued semantics. The notion

M. Bílková—The work of the first author has been supported by the joint project of Austrian Science Fund (FWF) I1897-N25 and Czech Science Foundation (GACR) 15-34650L.

M. Dostál—The work of the second author has been supported by the project No. GA13-14654S of the Czech Science Foundation.

J. Väänänen et al. (Eds.): WoLLIC 2016, LNCS 9803, pp. 109–124, 2016.
DOI: 10.1007/978-3-662-52921-8_8

of behavioral equivalence is central in studying coalgebras, and it can often be captured by bisimilarity, a central notion in model theory of modal logics. In particular, for coalgebras with a finitary type of behaviour, we are interested in finitary modal languages being expressive for bisimilarity, i.e., logics satisfying the Hennessy-Milner property.

We adopt the approach based on understanding modalities as predicate liftings and apply it in a many-valued setting. Such languages for classical coalgebraic logics were developed and their expressivity investigated by Pattinson in [14,15] and further by Schröder in [16]. In particular, a sufficient condition on a set of predicate liftings, namely being separating, is given to ensure that the resulting modal logic is expressive for behavioral equivalence, respectively for bisimilarity, depending on the setting. We address the limitative results of Metcalfe and Martí [13] providing a sufficient and necessary condition on the algebra of truth values $\mathscr{V}$ ensuring the Hennessy-Milner property for the $\mathscr{V}$-valued modal language with box and diamond over image-finite Kripke frames with two-valued accessibility relation, where $\mathscr{V}$ is an MTL-chain. The condition they provide says that we can distinguish truth values in $\mathscr{V}$ with propositional formulas, so, in contrast to the boolean case, also expressivity of the purely propositional part of the language matters. If we want to avoid including constants for all truth values of $\mathscr{V}$ in the language, the condition rules out many interesting fuzzy modal logics: for $\mathscr{V}$ being a complete BL-chain with finite universe or $[0, 1]$, this yields expressivity if and *only if* $\mathscr{V}$ is a MV-chain or the ordinal sum of two (hoop reducts of) MV-chains, leaving out most Gödel modal logics.

We shall apply the approach of [16] to generalize the results of [13]. In particular, we also address logics of Kripke frames with many-valued accessibility relation, probabilistic Kripke frames and extend the negative results on Gödel logics. On a positive side, we provide a countable expressive language for Kripke frames and any MTL chain as the algebra $\mathscr{V}$ of truth values.

2 Set Coalgebras as Models of Many-Valued Modal Logics

We fix an endofunctor $T : \mathsf{Set} \to \mathsf{Set}$. We say that T is *standard* if it preserves inclusions and the equaliser $0 \to 1 \rightrightarrows 2$. For such a standard T we define $T_\omega : \mathsf{Set} \to \mathsf{Set}$ on objects as $T_\omega X = \bigcup\{TY \mid Y \subseteq X, Y \text{ finite}\}$. We say that T is *finitary* if $T = T_\omega$ holds. For a standard finitary T and an element $t \in TX$ we define *base* of t as $\mathrm{b}(t) \subseteq X$ to be the smallest subset of X such that $t \in T(\mathrm{b}(t))$ holds. We refer the reader to [10] for more details about bases of various functors and properties of bases.

Definition 1. *Given an endofunctor* $T : \mathsf{Set} \to \mathsf{Set}$*, a* T-coalgebra *is a morphism* $c : X \to TX$ *in* Set*. Given two* T*-coalgebras* $c : X \to TX$ *and* $d : Y \to TY$*, the morphism* $h : X \to Y$ *is a homomorphism of* T*-coalgebras if the diagram*

$$\begin{tikzcd} X \arrow[r, "h"] \arrow[d, "c"'] & Y \arrow[d, "d"] \\ TX \arrow[r, "Th"'] & TY \end{tikzcd}$$

commutes.

Example 1 (Functors and their respective coalgebras)

1. The covariant powerset functor $P : \mathsf{Set} \to \mathsf{Set}$ assigns to each set X the set of its subsets PX; a mapping $f : X \to Y$ is sent to the direct image mapping $Pf : PX \to PY$, sending $Z \subseteq X$ to $Pf(Z) = f[Z] = \{f(z) \in Y \mid z \in Z\}$. The coalgebras for P are mappings $c : X \to PX$, modeling Kripke frames by assigning to each $x \in X$ its set $c(x) \subseteq X$ of successors. The finitary powerset functor P_ω assigns to each set X the set of its *finite* subsets $P_\omega X$ and acts on morphisms as described above. The P_ω coalgebras are *image finite* Kripke frames.
2. Let $\mathscr{V}$ be any residuated lattice. The (covariant) functor $P^{\mathscr{V}} : \mathsf{Set} \to \mathsf{Set}$ assigns to a set X the set $P^{\mathscr{V}}(X) = [X, \mathscr{V}]$ of all mappings from X to $\mathscr{V}$. A mapping $f : X \to Y$ is sent to a mapping $[X, \mathscr{V}] \to [Y, \mathscr{V}]$ which assigns to $s : X \to \mathscr{V}$ a map $t : Y \to \mathscr{V}$ defined as $t(y) = \bigvee_{f(x)=y} s(x)$. The coalgebras for $P^{\mathscr{V}}$ are maps $c : X \to [X, \mathscr{V}]$ which model many-valued Kripke frames, i.e., frames for which the accessibility relation takes values from $\mathscr{V}$. Whenever $\mathscr{V}$ is distributive, which is the case in all our examples, the functor $P^{\mathscr{V}}$ preserves weak pullbacks. See Lemma 2.4.12 of [3]. The finitary functor $P^{\mathscr{V}}_\omega$ assigns to a set X the set $P^{\mathscr{V}}(X) = [X, \mathscr{V}]$ of all mappings from X to $\mathscr{V}$ with *finite support*, i.e. only finitely many non-zero values.
3. Given a set X, denote by DX the set of all probabilistic distributions on X: that is, DX is the set of mappings $d : X \to [0, 1]$ such that $\Sigma_{x \in X} d(x) = 1$. Similarly to the functor above, the mapping $Df : DX \to DY$ assigns to a distribution $d : X \to [0, 1]$ the distribution $e : Y \to [0, 1]$ for which $e(y) = \Sigma_{f(x)=y} d(x)$ holds. The coalgebras for the functor D correspond precisely to probabilistic Kripke frames. Again, finitary version of the functor D_ω is defined to assign to a set X the set of probabilistic distributions on X with finite support.

Propositional Part of the Logic: For simplicity of presentation and since all our examples come mostly from fuzzy logics, we restrict ourselves to propositional logics extending commutative Full Lambek substructural logic with weakening FL_{ew}, whose semantics are commutative integral residuated lattices [4]. We also fix a commutative integral residuated lattice $\mathscr{V}$, and a countable set of propositional variables At. The propositional language is then defined as follows[1]:

$$a := p \mid \top \mid \bot \mid a \wedge b \mid a \vee b \mid a \to b \mid a \,\&\, b,$$

with additional defined connectives $\neg a = a \to \bot$ and $a \leftrightarrow b = (a \to b) \wedge (b \to a)$. Given a coalgebra $c : X \to TX$ and a valuation of atoms

$$\|.\|_c : At \to [X, \mathscr{V}]$$

the semantics $\|*(a_1, \dots, a_n)\|_c$ is computed inductively for each n-ary connective $*$, as

$$X \xrightarrow{\overrightarrow{\|a\|_c}} \mathscr{V}^n \xrightarrow{*_{\mathscr{V}}} \mathscr{V}.$$

The semantics of the language can be seen as a local $\mathscr{V}$-valued relation between states and formulas given by

$$x \Vdash_c a \;=\; \|a\|_c(x).$$

Definition 2. *A relation $B \subseteq X \times Y$ is a T-*bisimulation *between $c : X \to TX$ and $d : Y \to TY$ iff there is a coalgebra structure b on B which makes the projections into coalgebra morphisms:*

$$\begin{tikzcd}
X \arrow[d, "c"'] & B \arrow[l, "p_0"'] \arrow[r, "p_1"] \arrow[d, "b"] & Y \arrow[d, "d"] \\
TX & TB \arrow[l, "Tp_0"] \arrow[r, "Tp_1"'] & TY
\end{tikzcd}$$

Equivalently (if T preserves weak pullbacks) B is a T-bisimulation if, using the *relation lifting* of B by the functor T, i.e. the relation $\overline{T}(B)$ as follows:

$$B(x, y) \;\text{ implies }\; \overline{T}(B)(c(x), d(y)).$$

To unravel this definition (we refer to [10,11] for a general definition and properties of the lifting),

$$\overline{T}(B)(c(x), d(y)) \;\text{ iff }\; \exists z \in TB(c(x) = (Tp_0)(z) \;\&\; (Tp_1)(z) = d(y)).$$

In this paper we use coalgebras as models of many-valued modal logics and therefore also $\mathscr{V}$-valued valuations will play a role in defining T-bisimilarity: Two states $x \in X$ and $y \in Y$ in coalgebras c and d are T-*bisimilar* if there exists a T-bisimulation $B \subseteq X \times Y$ such that $B(x, y)$ holds, and moreover the atomic harmony,

$$x' \Vdash_c p = y' \Vdash_d p,$$

holds for all atoms $p \in At$, and all $x'By'$[2].

[1] We would like to stress that we do not include constants for elements of $\mathscr{V}$ in the language (cf. Examples 3 and 7).

[2] This in fact says that B is a $T \times \mathscr{V}^{At}$-bisimulation, where the second part of the functor encodes the valuations.

Remark 1. Using the notion of *many-valued* relation lifting, it is possible to define a *many-valued* variant of the notion of bisimulation. However, it has been shown in [1,3] that the two-valued notion of bisimilarity arising from many-valued bisimulations coincides with the usual notion of bisimilarity.

Example 2 (T-bisimulations). A P-*bisimulation* between many-valued crisp Kripke models [2,13] (P coalgebras with many-valued valuations) is a relation $B \subseteq X \times Y$ satisfying: xBy implies

1. $x \Vdash_c p = y \Vdash_d p$ for all atoms,
2. $\forall x' \in c(x)\ \exists y'(y' \in d(y)\ \&\ x'By')$ and $\forall y' \in d(y)\ \exists x'(x' \in c(x)\ \&\ x'By')$.

A $P^{\mathscr{V}}$-*bisimulation* between many-valued Kripke models [2,13] ($P^{\mathscr{V}}$ coalgebras with many-valued valuations) is a relation $B \subseteq X \times Y$ satisfying: xBy implies

1. $x \Vdash_c p = y \Vdash_d p$ for all atoms,
2. $c(x)(x') \leq \bigvee_{y':x'By'} d(y)(y')$ and $d(y)(y') \leq \bigvee_{x':x'By'} c(x)(x')$.

A D-*bisimulation* between probabilistic Kripke frames is defined similarly, with the sum in place of join.

3 Many Valued Predicate Liftings

We want to apply the existing theory of modal logics defined using predicate liftings, and classifications of sets of predicate liftings, by [15,16] to study expressivity of many-valued modal logics.

In the text and in the examples of Sect. 3 we assume T is finitary and that T preserves weak pullbacks. The $\mathscr{V}$-valued modal language for T-coalgebras will be given via extending the propositional language defined in the previous section by a set of modal operators, modalities, describing behaviour of T-coalgebras. Modalities can arise semantically as an abstract way of lifting predicates on X (maps in $[X, \mathscr{V}]$) to predicates on TX (maps in $[TX, \mathscr{V}]$). Following ideas of [15,16] concerning two-valued predicates and their liftings, we define $\mathscr{V}$-valued n-ary *predicate liftings* to be maps:

$$\hat{\heartsuit}_X : [X, \mathscr{V}^n] \to [TX, \mathscr{V}],$$

natural in X. Let's start with an easy observation that n-ary predicate liftings are essentially the same things as the maps

$$\heartsuit : T\mathscr{V}^n \to \mathscr{V},$$

which we will call the n-ary modalities. The two concepts are in the following one-to-one correspondence:

$$\hat{\heartsuit}_X(Q) = \heartsuit(TQ) : TX \to \mathscr{V} \text{ and } \heartsuit = \hat{\heartsuit}_{\mathscr{V}^n}(\mathrm{id}_{\mathscr{V}^n}).$$

We can now extend the propositional language with a set of such modalities (possibly all), stating that whenever $a_1, \ldots, a_n$ are formulas and $\heartsuit$ is an n-ary modality, $\heartsuit(a_1, \ldots, a_n)$ is a formula. For a set Λ of modalities, we denote by $\mathscr{L}(\Lambda)$ the resulting modal language.

On a coalgebra $c : X \to TX$ with valuations $\|a_i\|_c : X \to \mathscr{V}$ the formula $\heartsuit(a_1, \ldots, a_n)$ is interpreted as follows (cf. [5]):

$$X \xrightarrow{c} TX \xrightarrow{T\overrightarrow{\|a\|}} T(\mathscr{V}^n) \xrightarrow{\heartsuit} \mathscr{V}$$

Example 3 (Boxes)

1. The boolean semantics of $\Box : P2 \to 2$ is the map assigning 1 to $\emptyset$ and $\{1\}$, i.e. the meet $\bigwedge$ on the two-element boolean algebra. On a coalgebra $c : X \to PX$, $\Box a$ is interpreted as follows:

$$X \xrightarrow{c} PX \xrightarrow{P\|a\|} P(2) \xrightarrow{\bigwedge} 2$$

for any x the result is 1 iff $c(x) = \emptyset$ or $\forall y \in c(x)\ y \Vdash a$.
2. Many-valued semantics of $\Box : P\mathscr{V} \to \mathscr{V}$ is also the meet, now computed in $\mathscr{V}$. On a coalgebra $c : X \to PX$, $\Box a$ is interpreted as follows:

$$X \xrightarrow{c} PX \xrightarrow{P\|a\|} P(\mathscr{V}) \xrightarrow{\bigwedge} \mathscr{V}$$

for any x the result is $c(x) \Vdash \Box a = \bigwedge_{y \in c(x)} y \Vdash a$ (cf. [2,13]).
3. Many-valued semantics of $\Box : [\mathscr{V}, \mathscr{V}] \to \mathscr{V}$ is given by a mapping:

$$\sigma \ \mapsto \ \bigwedge_{v \in \mathscr{V}} (\sigma(v) \to v).$$

On a coalgebra $c : X \to P^{\mathscr{V}}$, $\Box a$ is interpreted as follows:

$$X \xrightarrow{c} [X, \mathscr{V}] \xrightarrow{P^{\mathscr{V}}\|a\|} [\mathscr{V}, \mathscr{V}] \xrightarrow{\Box} \mathscr{V}$$

for any x the result is $c(x) \Vdash \Box a = \bigwedge_y (c(x)(y) \to y \Vdash a)$ (cf. [2]).

Lemma 1 (Adequacy). *The modal language $\mathscr{L}(\Lambda)$ for Λ a set of predicate liftings (modalities) for T is invariant under bisimilarity.*

Proof. Routine induction on the complexity of a modal formula, using the semantics defined as above.

3.1 Separating Sets of Predicate Liftings

To prove expressivity of a modal language resulting from a set of modalities ar predicate liftings, we have in particular to ensure we have enough modalities to distinguish, or to separate, different behaviours of coalgebras in question. Such sets of modalities are called *separating*. Although we will see later that having a separating set of modalities is in general not enough to prove expressivity, as also the propositional language has to be expressive enough to handle the modalities[3], we separate this two issues and concentrate first on the modalities only. What follows in this and the next subsection are mostly direct analogues of theorems in Schröder's [16], generalizing from the case when $\mathscr{V} = 2$. First notice that predicate liftings have their transpose:

$$\hat{\heartsuit}_X : [X, \mathscr{V}^n] \to [TX, \mathscr{V}] \quad \mapsto \quad \hat{\heartsuit}^\flat_X : TX \to [[X, \mathscr{V}^n], \mathscr{V}]$$

Definition 3. *A set Λ of predicate liftings for a finitary functor T is called* separating *iff*

$$(\hat{\heartsuit}^\flat_X : TX \to [[X, \mathscr{V}^n], \mathscr{V}])_{n<\omega, \heartsuit\in\Lambda}$$

is jointly injective *for all X. This means that for each $t \neq t'$ in TX, there are some n, $\sigma : X \to \mathscr{V}^n$ and n-ary $\heartsuit \in \Lambda$, such that (in $\mathscr{V}$)*

$$\heartsuit(T\sigma)(t) \neq \heartsuit(T\sigma)(t').$$

Theorem 1. *A finitary functor T admits a separating set of predicate liftings iff the source*

$$(Tf : TX \to T(\mathscr{V}^n))_{n<\omega, f:X\to\mathscr{V}^n}$$

is jointly injective for each X.

Proof. Assume a separating set of predicate liftings Λ is given, and $t \neq t'$ in TX. Then by the definition there are some n, $\sigma : X \to \mathscr{V}^n$ and $\heartsuit \in \Lambda$, such that (in $\mathscr{V}$) $\heartsuit(T\sigma)(t) \neq \heartsuit(T\sigma)(t')$, but then clearly in $T(\mathscr{V}^n)$ $(T\sigma)(t) \neq (T\sigma)(t')$. For the other direction, the condition implies that the set of *all* n-ary predicate liftings is separating.

Notice that the theorem in particular holds if we rephrase it for sets of *unary* predicate liftings, and restrict n in the condition to be always $n = 1$ (cf. Corollary 18 of [16]). The (unrestricted) Theorem 1 has the following corollary [16]:

Corollary 1. *Every finitary functor admits a separating set of predicate liftings. Namely, the set of* all *n-ary liftings is separating.*

Proof. By [16] [Corollary 38], this is true for $\mathscr{V} = 2$. But we can define an injective $g : 2 \to \mathscr{V}$ by $g(0) = 0$ and $g(1) = 1$ and use it to obtain mappings f required by Theorem 1, using the fact that T is standard and therefore preserves injective maps.

---

[3] In case that $\mathscr{V} = 2$ separability is in fact sufficient for expressivity. The reason is that the classical propositional logic is functionally complete and each boolean function $\sigma : 2^n \to 2$ is definable by a formula with n variables (cf. Definition 4).

This abstract theorem tells us that in principle we can always find a separating set of modalities, but the set of all modalities with unrestricted arity is certainly not what we want to consider in specific examples. Not only is the set of all predicate liftings too big to handle, but it in particular contains some modalities which can be seen as modal constants, naming elements in $\mathscr{V}$. For example, if $T = P_\omega$, we can define $\langle v \rangle : P_\omega \mathscr{V} \to \mathscr{V}$ as the map detecting v: $\langle v \rangle(Y) = 1$ iff $v \in Y$ and 0 otherwise. We would like in practice to restrict ourselves to some specific subsets of predicate liftings and therefore we need to come up with a sufficient condition for a subset of all predicate liftings being separating. We do one such case-study in the next section.

3.2 Monadic Predicate Liftings

A natural restriction, sufficient for most of examples in this paper, is to consider only *unary* predicate liftings. The following example illustrates that already when $\mathscr{V} = 2$ this is a real restriction, as not all finitary functors admit a separating set of unary predicate liftings:

Example 4. $\mathscr{V} = 2$, $T = P_\omega P_\omega$ (the double covariant powerset functor[4]) does not admit a separating set of *unary* predicate liftings. By Theorem 1 restricted to $n = 1$, given a finite set X and any $f : X \to 2$,

$$(Tf)\{A \subseteq X \mid |A| \leq 2\} = (Tf)P_\omega X.$$

However, we might restrict our attention to unary modalities in the present paper, as for all our essential examples the functors admit a separating set of unary predicate liftings:

Example 5. By Theorem 1 and a similar reasoning as in the proof of Corollary 1, if a finitary T admits a set of unary predicate liftings for $\mathscr{V} = 2$, it does so for arbitrary $\mathscr{V}$ considered in this paper. In particular, $P_\omega, P_\omega^{\mathscr{V}}, D_\omega$ admit a separating set of unary predicate liftings for any $\mathscr{V}$ considered in this paper, including the case D_ω for $\mathscr{V} = [0,1]_{\text{Ł}}$.

The example above shows that the set of *all* unary predicate liftings for the functors mentioned thereof is separating. To be able to recognize the separating property for subsets of unary predicate liftings, we prove the following theorem (cf. Theorem 20 of [16]):

Theorem 2. *Assume a finitary T admits a separating set of unary predicate liftings Λ_T. Then for all $\Lambda \subseteq \Lambda_T$ TFAE:*

1. *Λ is separating, i.e. for each X and each $t \neq t'$ in TX, there is an $\sigma : X \to \mathscr{V}$ and $\heartsuit \in \Lambda$, such that*
$$\heartsuit(T\sigma)(t) \neq \heartsuit(T\sigma)(t').$$

[4] Not to be confused with the double contravariant powerset functor whose coalgebras are neighbourhood frames.

2. for each n, $t \neq t'$ in $T\mathscr{V}^n$ implies $\exists f : \mathscr{V}^n \to \mathscr{V}, \heartsuit \in \Lambda$ such that

$$\heartsuit(Tf)(t) \neq \heartsuit(Tf)(t').$$

Proof. For 1. $\to$ 2. simply consider $X = \mathscr{V}^n$. For the other direction assume $t \neq t'$ in TX. By assumption, T admits a separating set of unary predicate liftings, and we can use Theorem 1 for $n = 1$. Therefore there is an $f : X \to \mathscr{V}$ such that $Tf(t) \neq Tf(t')$ in $T\mathscr{V}$. Now, by (ii), there is an $f' : \mathscr{V} \to \mathscr{V}$ and $\heartsuit \in \Lambda$ such that $\heartsuit(Tf')(Tf)(t) \neq \heartsuit(Tf')(Tf)(t')$. Thus $f'f : X \to \mathscr{V}$ witnesses (i).

Example 6. 1. for $T = P_\omega$ and $\mathscr{V} = 2$, $\{\Box\}$ is a separating set, and $\{\Diamond\}$ is also a separating set.
2. for $T = P_\omega$ and any $\mathscr{V}$, $\{\Diamond\}$ is a separating set[5]: consider $t \neq s$ in $P_\omega\mathscr{V}^n$, w.l.o.g. assume there is some $\overrightarrow{v} \in t$ different from each $\overrightarrow{w} \in s$. Define $f(\overrightarrow{v}) = 1$ and define $f(\overrightarrow{w}) = 0$ for all $\overrightarrow{w} \neq \overrightarrow{v}$ (it would be enough if $f(\overrightarrow{w}) < 1$). Then
$$\bigvee f[t] > \bigvee f[s].$$
Also $\{\Box\}$ is separating, which can be proved using a similar argument.
3. for $T = D_\omega$ and $\mathscr{V} = 2$, the set $\{\langle\pi\rangle \mid \pi \in [0,1]\}$ with semantics
$$x \Vdash \langle\pi\rangle a \text{ iff } \sum_{y \Vdash a} c(x)(y) \geq \pi,$$
is separating. The functor however does not admit a separating set of *continuous* predicate liftings, i.e. an expressive normal modal logic (cf. Example 32 of [16]).

3.3 An Algebraic Condition

In the case of $\mathscr{V} = 2$, i.e. the classical coalgebraic logic, the fact that a set of predicate liftings Λ is separating is enough to prove that the resulting modal language is expressive for bisimilarity. This is no longer the case for a general $\mathscr{V}$: also the propositional language given by $\mathscr{V}$ matters. As was shown by Metcalfe and Martí [13], already for $T = P_\omega$ and $\Box, \Diamond$ it is not always the case. Generalizing their approach, we isolate an abstract condition on the algebra $\mathscr{V}$ to guarantee expressivity. Recall that a set Λ of unary modalities is separating iff

$$(\hat{\heartsuit}^\flat_X : TX \to [[X, \mathscr{V}], \mathscr{V}])_{\heartsuit \in \Lambda}$$

is jointly injective for all X, meaning that for each $t \neq t'$ in TX, there is $\sigma : X \to \mathscr{V}$ and $\heartsuit \in \Lambda$, such that $\heartsuit(T\sigma)(t) \neq \heartsuit(T\sigma)(t')$. To be able to prove expressivity, we need such witnessing σ to be expressible as a predicate $\|a\|_X$ given by an actual formula in the propositional language of $\mathscr{V}$. Now, in the condition above, read X as $\mathscr{V}^n$ (cf. how separating sets of PL are characterized in Theorem 2). This in a way motivates the following definition:

[5] cf. Examples 7 and 10. This does not entail expressivity.

Definition 4 (The condition on $\mathscr{V}$). *We call a function $f : \mathscr{V}^n \to \mathscr{V}$* expressible*, if there is a term σ in n variables in the language of $\mathscr{V}$, such that*

$$\sigma[x_1, \ldots, x_n/v_1, \ldots, v_n] = f(v_1, \ldots, v_n).$$

Λ is then called $\mathscr{V}$-separating*, if the collection of expressible functions separates values in $T\mathscr{V}^n$, i.e., the following condition holds: $t \neq t'$ in $T\mathscr{V}^n$ implies there exists $f : \mathscr{V}^n \to \mathscr{V}$ expressible, and $\heartsuit \in \Lambda$ such that*

$$\heartsuit(Tf)(t) \neq \heartsuit(Tf)(t').$$

Example 7. 1. for $\mathscr{V} = 2$, each $f : 2^n \to 2$ is expressible in the boolean language (simply because propositional logic is functionally complete). Thus the condition is vacuous if we work over classical logic because any separating Λ is 2-separating.

2. for $T = P_\omega$, $\Lambda = \{\Box, \Diamond\}$ and $\mathscr{V}$ a complete MTL chain, the condition of Λ being $\mathscr{V}$-separating is equivalent to the one given by Metcalfe and Martí in [13] [Theorem 3.5] in terms of distinguishing formula property: assume $t \neq s$ in $P_\omega\mathscr{V}^n$, and assume w.l.o.g. that there is some $\overrightarrow{v} \in t$ different from each $\overrightarrow{w} \in s$. Then, if by [13] there is a distinguishing formula σ with $\sigma(\overrightarrow{v}) < \overrightarrow{w}$ for each $\overrightarrow{w} \in s$ (or the same with $>$), the modality separating t and s is $\Box$ ($\Diamond$ resp.). If on the other hand we know that Λ is $\mathscr{V}$-separating, and $\overrightarrow{v}$ different from each $\overrightarrow{w} \in s$, we know there is a formula σ with $\bigwedge \sigma(\overrightarrow{v}) = \sigma(\overrightarrow{v}) \neq \bigwedge \sigma[s]$ (or the same with $\bigvee$). $\mathscr{V}$ is a chain, thus either $\sigma(\overrightarrow{v}) < \bigwedge \sigma[s]$ or $\sigma(\overrightarrow{v}) > \bigwedge \sigma[s]$, in both cases σ is a distinguishing formula.
   In particular, for $T = P_\omega$, $\Lambda = \{\Box, \Diamond\}$ and $\mathscr{V}$ an MV chain, including finite MV chains and the standard algebra $[0, 1]_{Ł}$, or Gödel three-element chain G_3, box and diamond are $\mathscr{V}$-separating by [13]. On the other hand, for Gödel chain $\mathscr{V} = G_n$ with $n \geq 4$, $\Lambda = \{\Box, \Diamond\}$ fails to be $\mathscr{V}$-separating as shown in [13].
3. If constants for all elements of $\mathscr{V}$ were included in the propositional language, then we could express a similar function f to that used in Example 6 (ii): observe that we can express the projections $\pi_i : \mathscr{V}^n \to \mathscr{V}$ by formulas, and we can detect values using the $\leftrightarrow$ connective, as $u \leftrightarrow v = 1$ iff $u = v$. Thus if $T = P_\omega$, then $\{\Box, \Diamond\}$ is always $\mathscr{V}$-separating , provided the constants are in the language.

3.4 Expressivity

We can finally show that the modal language for T-coalgebras, generated by a $\mathscr{V}$-separating set of predicate liftings Λ, is expressive for bisimilarity:

Theorem 3 (Expressivity). *Let T be finitary, w.p.p., and Λ a $\mathscr{V}$-separating set of predicate liftings. Then $\mathscr{L}(\Lambda)$ is expressive for bisimilarity.*

Proof. Fix coalgebras $c : X \to TX$ and $d : Y \to TY$ and respective valuations of formulas. Let the modal equivalence $\equiv_{\mathscr{L}(\Lambda)}$ between states x and y of coalgebras c and d be defined as

$$x \equiv y \quad \text{iff} \quad \forall a \in \mathscr{L}(\Lambda)(x \Vdash_c a = y \Vdash_d a).$$

We prove that $\equiv_{\mathscr{L}(\Lambda)}$ is a T-bisimulation, in particular that

$$x \equiv y \text{ implies } c(x)\ \overline{T}{\equiv}\ d(y).$$

Assume $\neg(c(x)\ \overline{T}{\equiv}\ d(y))$, we find a distinguishing formula witnessing $x \not\equiv y$.

Consider bases of $c(x)$ and $d(y)$ to be finite sets $\mathrm{b}(c(x)) = \{x_1 \dots x_k\}$ and $\mathrm{b}(d(y)) = \{y_1 \dots y_l\}$ resp., and for each $x_i \not\equiv y_j$ fix a distinguishing formula $a_{i,j}$ with $x_i \Vdash_c a_{i,j} \neq y_j \Vdash_d a_{i,j}$, thus we have up to kl distinguishing formulas, including the possibility there are none. Put n be the number of the formulas we obtained, we distinguish two cases:

$n = 0$: Consider the following two maps:

$$f = \|\top\|_c : \mathrm{b}(c(x)) \to \mathscr{V} \quad \text{and} \quad g = \|\top\|_d : \mathrm{b}(d(y)) \to \mathscr{V}.$$

Then f, g are constant maps with the value 1, and because the two bases were pairwise indistinguishable, $\equiv$ (more precisely, the corresponding restriction of $\equiv$ to bases of $c(x)$ and $d(y)$) is the pullback of the two maps (it is in fact the product of the two bases, but we do not need this fact here). T weakly preserves pullbacks and therefore $T{\equiv}$ is a weak pullback of Tf and Tg. Because $\neg(c(x)\ \overline{T}{\equiv}\ d(y))$ (and relation lifting commutes with restrictions, see e.g. [10]), we know, using the weak pullback, that $(Tf)c(x) \neq (Tg)(d(y))$ in $T\mathscr{V}$. By the separation property, there is $\heartsuit \in \Lambda$ and $\sigma : \mathscr{V} \to \mathscr{V}$ *expressible* such that

$$\heartsuit(T\sigma)(Tf)(c(x)) \neq \heartsuit(T\sigma)(Tg)(d(y)).$$

Therefore the formula $\heartsuit(\sigma(\top))$ distinguishes x and y, which is not hard to see.

$n > 0$: Denote by $\overrightarrow{\|a\|_c}$ and $\overrightarrow{\|a\|_d}$ the corresponding vectors of valuations of the distinguishing formulas in c and d. Consider the following two maps:

$$f = \overrightarrow{\|a\|_c} : \mathrm{b}(c(x)) \to \mathscr{V}^n \quad \text{and} \quad g = \overrightarrow{\|a\|_d} : \mathrm{b}(d(y)) \to \mathscr{V}^n.$$

Then it is not hard to see that $\equiv$ (again, more precisely, the corresponding restriction of $\equiv$ to bases of $c(x)$ and $d(y)$) is the pullback of the two maps. T preserves pullbacks weakly and therefore $T{\equiv}$ is a weak pullback of Tf and Tg. Because $\neg(c(x)\ \overline{T}{\equiv}\ d(y))$, we know that $(Tf)c(x) \neq (Tg)(d(y))$ in $T\mathscr{V}^n$. By the separation property, there is $\heartsuit \in \Lambda$ and $\sigma : \mathscr{V}^n \to \mathscr{V}$ *expressible* such that

$$\heartsuit(T\sigma)(Tf)(c(x)) \neq \heartsuit(T\sigma)(Tg)(d(y)).$$

Therefore the formula $\heartsuit(\sigma(\overrightarrow{a}))$ distinguishes x and y.

Corollary 2. *The modal logic of P_ω coalgebras based on a $\mathscr{V}$-separating Λ is expressive for bisimilarity. In particular, modal logic of $\{\Box, \Diamond\}$ for $\mathscr{V}$ being an MV algebra, $\mathscr{V} = 2$, or $\mathscr{V} = G_3$ is expressive. The condition on $\mathscr{V}$ in this case is not only sufficient, but also necessary.*

Proof. We show that for $T = P_\omega$, if Λ is not $\mathscr{V}$-separating, then there are indistinguishable but not bisimilar states of two P_ω-coalgebras: assume for some n there are $t \neq s$ in $P_\omega \mathscr{V}^n$, $|t| = k, |s| = l$ such that for each formula in n variables $\sigma : \mathscr{V}^n \to \mathscr{V}$ and each $\heartsuit \in \Lambda$ we have

$$\heartsuit(T\sigma)(t) = \heartsuit(T\sigma)(s).$$

Define $X = \{x_0, \ldots, x_k\}$ and $Y = \{y_0, \ldots, y_k\}$ with coalgebras c on X given by $c(x_0) = \{x_1, \ldots, x_k\}$ (and $\emptyset$ otherwise) and d on Y given by $d(y_0) = \{y_1, \ldots, y_k\}$ (and $\emptyset$ otherwise). Fix atoms $p_1, \ldots, p_n$ and define valuations $\overrightarrow{\|p\|} : X \to \mathscr{V}^n$ given by $\overrightarrow{\|p\|}(x_0) = \overrightarrow{0}$ and $\overrightarrow{\|p\|}(x_i) = t_i$. Similarly, valuations $\overrightarrow{\|p\|} : Y \to \mathscr{V}^n$ given by $\overrightarrow{\|p\|}(y_0) = \overrightarrow{0}$ and $\overrightarrow{\|p\|}(y_i) = s_i$. Now x_0 and y_0 are not bisimilar by $s \neq t$, but are modally equivalent.

We apply the theory above to a few particular examples, namely we show (i) there is an expressive language for P_ω coalgebras and $\mathscr{V}$ being a Gödel chain bigger then 3, it cannot however be based on unary modalities, (ii) the results of [13] about expressivity of Łukasziewicz's logics with box and diamond extend to the $\mathscr{V}$-valued finitary powerset functor, and (iii) we present an expressive language for D_ω coalgebras based on a finite set of modalities and the standard Łukasziewicz algebra.

Example 8 (Łukasziewicz logics). Let $\mathscr{V}$ be the standard Łukasziewicz algebra, i.e. the real interval $[0,1]_{\text{Ł}}$, with $a \,\&\, b = max\{0, a + b - 1\}$ and $a \to b = min\{1, 1 - a + b\}$ (and therefore $\neg a = (a \to 0) = 1 - a$). Let $T = P_\omega^{\mathscr{V}}$. Notice in particular that $a \,\&\, 0 = 0$ for each a, and that $\&$ distributes over $\vee$. Consider the unary modality $\Diamond : [\mathscr{V}, \mathscr{V}] \to \mathscr{V}$ given, for a $t : \mathscr{V} \to \mathscr{V}$ with finite support, by

$$\Diamond t = \bigvee_{u \in \mathscr{V}} (t(u) \,\&\, u).$$

We show that $\Lambda = \{\Diamond\}$ is $\mathscr{V}$-separating: Assume $t \neq s$ in $[\mathscr{V}^n, \mathscr{V}]$ and assume that their finite supports are subsets of $\{\overrightarrow{u_1}, \ldots, \overrightarrow{u_k}\}$. Since s and t have finite support, they can also differ only on finitely many distinct points from $\{\overrightarrow{u_1}, \ldots, \overrightarrow{u_k}\}$. Assume w.l.o.g. that for some i, $t(\overrightarrow{u_i}) > s(\overrightarrow{u_i})$. Then we can "separate" $\overrightarrow{u_i}$ from each of the other $\overrightarrow{u_j}$ with $j \neq i$ by a rational point $\overrightarrow{v} \in \mathscr{V}^n \cap Q^n$, and define a piece-wise linear function $f : \mathscr{V}^n \to \mathscr{V}$ with rational coefficients so that $f(\overrightarrow{u_i}) = 1$ while for $j \neq i$ $f(\overrightarrow{u_j}) = 0$. By McNaughton theorem [12] there is a propositional formula $\sigma(\overrightarrow{p})$ with $\sigma[\overrightarrow{p}/\overrightarrow{u}] = f(\overrightarrow{u})$ for all $\overrightarrow{u} \in \mathscr{V}^n$. In particular, $\sigma(\overrightarrow{u_i}) = 1$ while for all $j \neq i$ we have $\sigma(\overrightarrow{u_j}) = 0$. Recall that on any $\overrightarrow{u} \notin \{u_1, \ldots, u_k\}$ both $s(\overrightarrow{u}) = t(\overrightarrow{u}) = 0$ by the finite-support assumption. It is not hard to see that then we have:

$$\bigvee_{\overrightarrow{u}\in\mathscr{V}^n}(t(\overrightarrow{u})\ \&\ \sigma(\overrightarrow{u})) > \bigvee_{\overrightarrow{u}\in\mathscr{V}^n}(s(\overrightarrow{u})\ \&\ \sigma(\overrightarrow{u})),$$

but by

$$\Diamond(T\sigma)t = \bigvee_{w\in\mathscr{V}}((\bigvee_{\overrightarrow{u}:\sigma(\overrightarrow{u})=w} t(\overrightarrow{u}))\ \&\ w) = \bigvee_{\overrightarrow{u}\in\mathscr{V}^n}(t(\overrightarrow{u})\ \&\ \sigma(\overrightarrow{u})),$$

$$\Diamond(T\sigma)s = \bigvee_{w\in\mathscr{V}}((\bigvee_{\overrightarrow{u}:\sigma(\overrightarrow{u})=w} s(\overrightarrow{u}))\ \&\ w) = \bigvee_{\overrightarrow{u}\in\mathscr{V}^n}(s(\overrightarrow{u})\ \&\ \sigma(\overrightarrow{u}))$$

we obtain $\Diamond(T\sigma)t \neq \Diamond(T\sigma)s$ as required. A similar argument (and in fact a simpler one based on distinguishing formulas) can be find also for finite Łukasziewicz chains. Therefore the modal logics of image finite many-valued Kripke frames ($P_\omega^{\mathscr{V}}$ coalgebras) based on $\Diamond$ (or $\Box$ or both) and $\mathscr{V}$ the standard or finite Łukasziewicz chain are expressive for bisimilarity.

Example 9 (Probabilistic Łukasziewicz logics). We show that we can find a finite expressive language for probabilistic Kripke frames, i.e. D_ω coalgebras, based on $\mathscr{V} = [0,1]_{\text{Ł}}$ using a very similar idea as in the example above. Note that the 2-valued logic for the same coalgebras needs to contain infinitely many modalities (cf. Example 6).

In $[0,1]_{\text{Ł}}$, the following truncated sum is definable: $a \oplus b = \neg(\neg a\ \&\ \neg b) = max\{1, a+b\}$. Consider a unary modality $\Diamond : [\mathscr{V}, [0,1]] \to \mathscr{V}$ given, for a $t : \mathscr{V} \to [0,1]$ with finite support, by[6]

$$\Diamond(t) = \bigoplus_{u\in\mathscr{V}}(t(u)\cdot u).$$

We show that $\Lambda = \{\Diamond\}$ is $\mathscr{V}$-separating: Assume $t \neq s$ in $[\mathscr{V}^n, [0,1]]$ and assume that their finite supports are subsets of $\{\overrightarrow{u_1}, \ldots, \overrightarrow{u_k}\}$. By very similar reasoning as in the above example we can construct a McNaughton function definable by a formula $\sigma(\overrightarrow{p})$ satisfying $\sigma(\overrightarrow{u_i}) = 1$ while for all $j \neq i$ we have $\sigma(\overrightarrow{u_j}) = 0$. Then we obtain:

$$\sum_{\overrightarrow{u}\in\mathscr{V}^n}(t(\overrightarrow{u})\cdot\sigma(\overrightarrow{u})) > \sum_{\overrightarrow{u}\in\mathscr{V}^n}(s(\overrightarrow{u})\cdot\sigma(\overrightarrow{u})),$$

but by

$$\Diamond(T\sigma)t = \bigoplus_{w\in\mathscr{V}}((\sum_{\overrightarrow{u}:\sigma(\overrightarrow{u})=w} t(\overrightarrow{u}))\cdot w) = \sum_{\overrightarrow{u}\in\mathscr{V}^n}(t(\overrightarrow{u})\cdot\sigma(\overrightarrow{u})),$$

$$\Diamond(T\sigma)s = \bigoplus_{w\in\mathscr{V}}((\sum_{\overrightarrow{u}:\sigma(\overrightarrow{u})=w} s(\overrightarrow{u}))\cdot w) = \sum_{\overrightarrow{u}\in\mathscr{V}^n}(s(\overrightarrow{u})\cdot\sigma(\overrightarrow{u}))$$

[6] Defined like this, using the multiplication of reals, the semantics of $\Diamond$ is not expressed by a first-order formula of Łukasziewicz logic.

we obtain $\diamondsuit(T\sigma)t \neq \diamondsuit(T\sigma)s$ as required. Therefore the modal logic of image finite probabilistic Kripke frames (D_ω coalgebras) based on a single modality $\diamondsuit$ and the standard Łukasziewicz algebra is expressive for bisimilarity. The modality $\diamondsuit$ is rather many-valued then probabilistic: a good way how to understand a formula $\diamondsuit a$ is *not* the probability of a, but rather the truth-value of an expression "probably a". (cf. with a two-layer modal logic proposed by Hájek [6].)

Example 10 (Monadic modal Gödel logics). Let us take P_ω for the coalgebra functor and $\mathscr{V}$ being a Gödel chain G_4 with 4 elements, say $0 < u < v < 1$. In this case, $\& = \wedge$ and $a \to b = 1$ if $a \leq b$ and $a \to b = b$ else, $\perp = 0$ and $\top = 1$. There is no $\mathscr{V}$-separating set of unary modalities. Therefore in such a case by Corollary 2, there is no *monadic* Gödel modal logic expressive for bisimilarity. It suffices to find two different elements $t, t' \in P_\omega \mathscr{V}^2$ such that for every binary term σ in the language of $\mathscr{V}$, $\sigma[t] = \sigma[t']$. Indeed, let $t = \{(u, v), (u, 1), (1, v)\}$ and $t' = \{(u, 1), (1, v)\}$. It is tedious but not hard to check that there is no binary term σ such that $P\sigma$ would distinguish t and t'.

Moreover, this approach easily generalizes to the cases of Gödel chains with more than 4 elements, and shows that for no such chain $\mathscr{V}$ admits a $\mathscr{V}$-separating set Λ of unary modalities. Let $\mathscr{V}$ be a Gödel chain with more than 4 elements. This means that $\mathscr{V}$ contains a subchain $G_4 : 0 < u < v < 1$, and by the definition of the basic operations of $\mathscr{V}$, the subchain G_4 is a (residuated) sublattice of $\mathscr{V}$. Because of this we can take for t and t' the same elements of $P\mathscr{V}^2$ as above, and no binary term σ distinguishes them.

In the following example we show how to exploit the abstract theory and obtain an expressive modal logic for P_ω coalgebras and any MTL chain, using a countable set of modalities of unbounded arities[7]:

Example 11 (An expressive polyadic modal logic for MTL chains). If we allow for modalities to take unbounded arities, we can define an expressive logic for P_ω coalgebras and a general MTL chain $\mathscr{V}$. We will find a (countable) separating set Λ of modalities which is $\mathscr{V}$-separating and does not rely on using constants representing the truth values of $\mathscr{V}$.

For an arbitrary natural number n, we need to distinguish any pair t, t' of elements of $P_\omega \mathscr{V}^n$ by some n-ary modality $\heartsuit \in \Lambda$. The idea is to use the lexicographical linear order on $\mathscr{V}^n$, and use modalities that pick from a finite subset of $\mathscr{V}^n$ the j-th projection of the i-th n-tuple in the set. More precisely, we consider the lexicographic order on $\mathscr{V}^n$ and define an auxiliary mapping $r_n : P_\omega \mathscr{V}^n \to P_\omega \mathscr{V}^n$ by setting $r_n(\emptyset) = \emptyset$, and sending a finite nonempty chain (w.r.t. the lexicographic order) $X \subseteq_\omega \mathscr{V}$ to the chain $X' \subseteq X$ that has the least element of X removed. By induction, we define $\langle i \mid - \rangle_n : P_\omega \mathscr{V}^n \to \mathscr{V}^n$ for each i as follows. As a base case

$$\langle 1 \mid - \rangle_n : P_\omega \mathscr{V}^n \to \mathscr{V}^n \text{ is given by } X \mapsto \bigwedge_{\text{lex}} X,$$

[7] It is straightforward to generalize Theorem 3 to the polyadic setting, and in this particular example we will not need any expressible propositional formulas.

where the meet is computed with respect to the lexicographic ordering (and the empty set being mapped to to $(1, \ldots, 1)$). The mapping $\langle m+1 \mid - \rangle_n : P_\omega \mathscr{V}^n \to \mathscr{V}^n$ is defined as the composite $\langle m \mid - \rangle_n \cdot r_n : P_\omega \mathscr{V}^n \to \mathscr{V}^n$ for each natural $m > 1$. We can then define a modality $\langle i \mid j \rangle_n : P_\omega \mathscr{V}^n \to \mathscr{V}$ as the composite $\pi_j \cdot \langle i \mid - \rangle_n : P_\omega \mathscr{V}^n \to \mathscr{V}$, with π_j being the j-th projection mapping. The set $\Lambda = \{\langle i \mid j \rangle_n \mid n, i < \omega, 1 \leq j \leq n\}$ of modalities is countable and $\mathscr{V}$-separating: Consider two elements $t \neq t'$ in $P_\omega \mathscr{V}^n$ and assume (without loss of generality) there is an n-tuple $\overrightarrow{v} = (v_1, \ldots, v_n)$ that belongs to t but not to t'. Then $\overrightarrow{v}$ is the i-th least element in t for some i, according to the lexicographic order. Since $\overrightarrow{v} \notin t'$, the i-th least element in t' (say $\overrightarrow{v'}$) differs from $\overrightarrow{v}$ on some index j, i.e., $v_j \neq v'_j$. Then the modality $\langle i \mid j \rangle_n$ distinguishes t and t', as $v_j = \langle i \mid j \rangle_n(t) \neq \langle i \mid j \rangle_n(t') = v'_j$. Here, we did not need to use any expressible propositional formulas.

Concluding Remarks

We have shown how to adapt and use existing colgebraic methods to study many-valued modal logics, namely their expressivity w.r.t. bisimilarity. The case-study in the previous section outlines some generalisations of results obtained in [13]. One may pursuit the topic further by considering algebras other than those arising from fuzzy logics, and cover for example paraconsistent coalgebraic logics. We have also restricted ourselves to study expressivity w.r.t. bisimilarity, and not behavioral equivalence, which seems to be preferred in the context of coalgebraic logic (e.g. for probabilistic coalgebras). Assuming the functor T preserves weak pullbacks, the two notions of equivalence coincide. The assumption (which we have used proving expressivity) however leaves out some natural examples of functors: in particular some instances of $P^{\mathscr{V}}$ where $\mathscr{V}$ is not distributive, or double contravariant powerset functor or its many-valued variant, whose coalgebras are neighbourhood frames or fuzzy neighbourhood frames. One way of extending our approach further would be to study expressivity using the dual adjunction approach, going back at least to [8,9], to define the modal logics in the spirit of Proposition 3.6 of [11]. Expressivity then corresponds to certain *injectivity* property, and following approach of [7] we may obtain an alternative, and in a way more general, view at the phenomenon in the many-valued setting.

References

1. Bílková, M., Dostál, M.: Many-valued relation lifting and Moss' coalgebraic logic. In: Heckel, R., Milius, S. (eds.) CALCO 2013. LNCS, vol. 8089, pp. 66–79. Springer, Heidelberg (2013)
2. Bou, F., Esteva, F., Godo, L., Rodríguez, R.: On the minimum many-valued modal logic over a finite residuated lattice. J. Log. Comput. **21**(5), 739–790 (2011)
3. Dostál, M.: Many valued coalgebraic logic. Master thesis, Czech Technical University (2013)
4. Galatos, N., Jipsen, P., Kowalski, T., Ono, H.: Residuated Lattices: An Algebraic Glimpse at Substructural Logics. Elsevier, Amsterdam (2007)

5. Gumm, H.P., Zarrad, M.: Coalgebraic simulations and congruences. In: Bonsangue, M.M. (ed.) CMCS 2014. LNCS, vol. 8446, pp. 118–134. Springer, Heidelberg (2014)
6. Hájek, P.: Metamathematics of Fuzzy Logic. Kluwer Academic Publishers, Dordrecht (1998)
7. Jacobs, B., Sokolova, A.: Exemplaric expressivity of modal logics. J. Log. Comput. **20**, 1041–1068 (2010)
8. Kupke, C., Kurz, A., Pattinson, D.: Algebraic semantics for coalgebraic logics. In: Coalgebraic Methods in Computer Science, ENTCS, vol. 106, pp. 219–241. Elsevier (2004)
9. Kupke, C., Kurz, A., Venema, Y.: Stone coalgebras. Theoret. Comput. Sci. **327**, 109–134 (2004)
10. Kupke, C., Kurz, A., Venema, Y.: Completeness for the coalgebraic cover modality. Log. Methods Comput. Sci. **8**(3), 1–76 (2012)
11. Kurz, A., Leal, R.: Modalities in the Stone age: a comparison of coalgebraic logics. Theoret. Comput. Sci. **430**, 88–116 (2012)
12. McNaughton, R.: A theorem about infinite-valued sentential logic. J. Symbolic Log. **16**, 1–13 (1951)
13. Metcalfe, G., Martí, M.: A Hennessy-Milner property for many-valued modal logics. Adv. Modal Log. **10**, 407–420 (2014)
14. Pattinson, D.: Expressivity results in the modal logic of coalgebras (2001)
15. Pattinson, D.: Expressive logics for coalgebras via terminal sequence induction. Notre Dame J. Formal Log. **45**, 19–33 (2004)
16. Schröder, L.: Expressivity of coalgebraic modal logic: the limits and beyond. Theoret. Comput. Sci. **390**, 230–247 (2008)

Second-Order False-Belief Tasks: Analysis and Formalization

Torben Braüner(✉), Patrick Blackburn, and Irina Polyanskaya

Roskilde University, Roskilde, Denmark
{torben,patrickb,irinap}@ruc.dk

Abstract. We first give a coarse-grained modal-logical analysis of the four best known second-order false-belief tasks. This preliminary analysis shows that the four tasks share a common logical structure in which a crucial role is played by a "principle of inertia" which says that an agent's belief is preserved over time unless the agent gets information to the contrary. It also reveals informational symmetries (all four possibilities inherent in the two dimensions of deception versus no-deception and change-in-world versus change-in-belief-only are realized) and reveals a rather puzzling feature common to all four tasks. We then take a closer look at how the principle of inertia is used, which leads to a fine-grained analysis in terms of perspective shifting. We formalize this analysis using a natural deduction system for hybrid logic, and show that the proof modelling the solution to the first-order Sally-Anne task is nested inside the proof modelling the second-order solution.

1 Introduction

In this paper we use modal and hybrid logic to analyse four second-order false-belief tasks. We begin with our running example, the second-order Sally-Anne task (which was introduced in [3]):

> *A child is shown a scene with two doll protagonists, Sally and Anne, with a basket and a box respectively. Sally first places a marble into her basket. Then Sally leaves the scene, and in her absence, Anne moves the marble and puts it in her box.* ***However, although Anne does not realise this, Sally is peeking through the keyhole and sees what Anne is doing.*** *Then Sally returns, and the child is asked: "Where* **does Anne think that** *[Sally will] look for her marble?"*

Experiments have shown that typically developing children above the age of six usually handle second-order tasks correctly; see [12,13]. They answer that Anne thinks that Sally will look in the basket, which is where Anne (falsely) believes that Sally believes the marble to be. Younger children usually answer that Anne thinks that Sally will look in the box: this is indeed where Sally knows the marble to be, but Anne does not know that Sally knows this, and hence the response is incorrect. In short, to pass the test, the experimental subject must ascribe a *false* belief to Anne, thus ensuring that the answer can't be explained

J. Väänänen et al. (Eds.): WoLLIC 2016, LNCS 9803, pp. 125–144, 2016.
DOI: 10.1007/978-3-662-52921-8_9

as the subject simply reporting what is true — it really is Anne's *belief* that is being reported. For children with *Autism Spectrum Disorder (ASD)*, the shift to correct responses tends to occur at a later age, if it happens at all.

If the bold font material is deleted, and [Sally will] is switched to 'will Sally', our statement of the second-order Sally-Anne task becomes a statement of the well known first-order Sally-Anne task (it was introduced in [2]):

> *A child is shown a scene with two doll protagonists, Sally and Anne, with a basket and a box respectively. Sally first places a marble into her basket. Then Sally leaves the scene, and in her absence, Anne moves the marble and puts it in her box. Then Sally returns, and the child is asked: "Where will Sally look for her marble?"*

Extensive experimental work with first-order false-belief tasks has shown the existence of a transition age, and it is lower than in the second-order case: children above the age of *four* will usually say that Sally will look in the basket, which is where Sally (falsely) believes the marble to be. But younger children will usually say that Sally will look in the box: this is where the marble is, but Sally does not know this, and hence the answer is incorrect. Once again, to pass the test, the experimental subject must ascribe a *false* belief, this time to Sally. This ensures that the subject's answer can't be explained as the subject simply reporting what is true. For children with ASD, the shift to correct responses usually occurs at a later age.

Handling first-order false-belief tasks correctly is viewed as a milestone in the acquisition of *Theory of Mind (ToM)*, the ability to ascribe mental states such as beliefs to oneself and others, and some researchers account for ASD using what is called the *ToM deficit hypothesis* (see [2]). A wide range of first-order false-belief tasks have been devised, and over the past 30 years both correlational and training studies (involving both typically developing and children with ASD) have yielded robust results across various countries and various task manipulations; see, for example, the meta-analysis [21].

Second-order false-belief mastery, the topic of this paper, is also regarded as a key step in the acquisition of ToM, but much less is known about it, and many conclusions are tentative [12,13]. There are far fewer second-order tests (the four we shall discuss pretty much cover the entire range) and they are less varied in design than their first-order cousins.[1] Moreover, there is no consensus

[1] The three other task we consider are the bake-sale task, the ice-cream task, and the puppy task; see Tables 4, 5 and 6 in the Appendix. The ice-cream task was the very first second-order false-belief task to be used; it was introduced in 1985 by Wimmer and Perner in [14]. The bake-sale task is a variant of the ice-cream task, and, as is explained in [11], pages 323–324: "The stories were modeled after Wimmer and Perner's (1985) "ice cream truck story". In contrast to their stories, we made sure that the beliefs of the two main protagonists in the story did not overlap, both at first-order and second-order level: each protagonist had his or her own distinct belief which was different from that of the other protagonist, as well as from the belief of the participants." The puppy task was introduced in 1994 in [19], again as a simplification of Wimmer and Perner's ice-cream task.

on the status of the shift from first-order to second-order competency. Some researchers, starting with [19], have viewed it as a straightforward extension of first-order mastery: acquisition of second-order mastery occurs when the child has sufficiently strengthened his or her information processing capacities; following Miller [12,13] we call this the *complexity only* position. Other researchers, starting with [14], have argued that the transition marks a more fundamental cognitive shift; following Miller, we call this the *conceptual change* position. In this paper we argue for a version of the conceptual change position. Our argument is grounded in ideas from modal and hybrid logic, but the backdrop to our discussion is our on-going training study on Danish speaking children with ASD in which we investigate whether training in linguistic recursion can lead to improvement in second-order false-belief competency.

We proceed as follows. In Sect. 2 we note earlier work on logical analysis of false-belief tasks. In Sect. 3 we give a coarse-grained modal-logical analysis of the four tasks and show that they share a common logical structure. A crucial role is played by a "principle of inertia", which says that an agent's belief is preserved over time, unless the agent gets information to the contrary. The coarse-grained analysis also reveals informational symmetries: all four possibilities inherent in the two binary dimensions of deception versus no-deception and change-in-world versus change-in-belief-only are realized. Moreover, the analysis reveals a somewhat puzzling feature concerning first-order information shared by all four tasks. In Sect. 4 we develop the coarse analysis into a fine-grained analysis by asking: how exactly is the principle of inertia used? Whereas the coarse-grained analysis simply uses the fact that Anne's belief is preserved, the fine-grained analysis builds on the observation that *Anne thinks that* Sally's belief is preserved to explain why; this is our stepping stone to the nested perspectival analyses that we formalize in hybrid-logic. In Sect. 5 we present the relevant fragment of hybrid logic, and formalize the first-order Sally-Anne task. In Sect. 6 we extend this to a formalization of the second-order task; as we shall see, the proof modelling the first-order solution is nested inside the proof modelling the second-order solution. In Sect. 7 we conclude.

2 Logic and False-Belief Tasks

Frege and Husserl both tried to divorce logic and psychology, but post-1945 work in cognitive science and artificial intelligence put logic-based models of cognitive abilities back on the agenda, and the 2008 publication of Stenning and Van Lambalgen [18] brought logic and psychology even closer. This pioneering work considers a wide range of psychological tasks, including the first-order Sally-Anne tasks, which it analyses using non-monotonic closed-world reasoning. Stenning and Van Lambalgen make use of the principle of inertia, and draw a useful distinction between *belief formation* and *belief manipulation*, which we will adopt in our discussion below. The first-order Sally-Anne task has also been formalized using an interactive theorem prover for a many-sorted first-order modal logic, an approach which also makes use of the principle of inertia; see [1]. But we know of few examples of logical modelling of second-order false-belief tasks: the clearest is the Dynamic Epistemic Logic based analysis given in [4], though the use of game theory in [20] to investigate performance in higher-order social reasoning, for instance, is also relevant.

This paper builds on recent hybrid-logical work on false-beliefs [6–8]. The distinguishing feature of the hybrid-logical approach is that *perspective shift* is taken as fundamental. That is, it formalizes the local shifts of perspective required by the experimental subject when reasoning about the agents in the scenario (in our running example, Sally and Anne). The intuition is this: correctly handling the first-order Sally-Anne task seems to involve taking the perspective of Sally, and reasoning about what she believes. So to speak, you have to put yourself in Sally's shoes. As we shall argue below, correctly handling the second-order Sally-Anne task seems to involve taking the perspective of another agent, namely Anne, and reasoning about her perspective on Sally's belief: you have to put yourself in Anne's shoes while she is putting herself in Sally's shoes. In this paper we turn these shoes into nested natural deduction proofs.

3 A Coarse-Grained Analysis

We now give a course-grained analysis of the four second-order tasks: we isolate the belief-states involved, and informally describe the reasoning leading from one belief-state to another. Three distinct times (t_0, t_1, and t_2) are significant in each story,[2] and in Table 2 in the Appendix we have described the belief-states at each of these times; the logical symbolism should be self-explanatory.

The reasoning pattern underlying all four tasks is clear.[3] First, note that in all four examples we make use of $B\neg\psi \rightarrow \neg B\psi$, the (contraposed form) of a modal principle called D: if we believe ψ to be false then we don't believe ψ. But in all four cases the crucial ingredient is the application of a "principle of inertia" saying that an agent's belief is preserved over time unless the agent has information to the contrary. For example, in the second-order Sally-Anne task, it is initially the case that Anne believes that Sally thinks that the marble is in the basket, formalized as $\boldsymbol{B_{anne}B_{sally}basket(t_0)}$. Initially it is also the case that Sally thinks that the marble is in the basket, $\boldsymbol{B_{sally}basket(t_0)}$, but Sally's belief changes at the intermediate stage t_1 since she sees through the keyhole the marble being moved, so $\boldsymbol{\neg B_{sally}basket(t_2)}$. Anne, however, does not know that Sally saw this, so Anne continues to believe that Sally thinks that the marble is in the basket, hence $\boldsymbol{B_{anne}B_{sally}basket(t_2)}$, the correct answer to the task.

---

[2] Some stories use more times than this: the bake-sale story, for example, makes use of (at least) four. But the sequence t_0, t_1, and t_2 constitutes the narrated time of the story, and here it is pointless to distinguish the times when Sam and Maria learn that there are no chocolate cookies for sale.

[3] In this section we adopt the following convention: belief-states that are part of this common pattern are typeset in bold (and displayed in blue in the online version), other belief-states are typeset in normal font.

This pattern underlies all four tasks: the correct answer is always a formula of the form $\boldsymbol{B_x B_y \phi}$ whose truth is preserved from stage t_0 to stage t_2, and subformula $\boldsymbol{B_y \phi}$ always becomes false at stage t_1 — unbeknownst to agent x, who ends up in t_2 with a false belief about the belief of agent y. So to derive the correct answer $\boldsymbol{B_x B_y \phi(t_2)}$, the experimental subject must work out that agent x does not know that something led to a changed belief for agent y.

Zero-Order, First-Order and Second-Order Information. Let's dig a little deeper for commonalities and differences. Table 3 in the Appendix summarizes the potentially relevant information available in the tasks, not just the information used in the coarse-grained analysis.

We start with the Sally-Anne task, where the formula $\boldsymbol{B_x B_y \phi(t_2)}$ is instantiated to $\boldsymbol{B_{anne} B_{sally}\, basket(t_2)}$. Note that in Table 3 we have focused solely on the predicate occurring in the correct answer, namely *basket*, and ignored the predicate *box*. That is, we assume that what matters is whether or not Sally believes the marble has been moved from the basket, not where it has been moved to. With this restriction, rows 1–5 in Table 3 summarize the potentially relevant zero-order, first-order and second-order information in the Sally-Anne task.

Similarly, rows 6–10 in Table 3 summarize the information in the bake-sale task, where $\boldsymbol{B_x B_y \phi(t_2)}$ is instantiated to $\boldsymbol{B_{maria} B_{sam}\, chocolate(t_2)}$. We have again focussed on the predicate occurring in the correct answer, which in this case is *chocolate*, so we are assuming that what matters is whether or not chocolate cookies are for sale, not what else is. We have also restricted our attention to Maria and Sam, the agents involved in the correct answer, and ignored Mom and the mailman, as their perspectives seem irrelevant.

In a similar fashion, rows 11–15 and 16–20 summarize the information available in the ice-cream and the puppy tasks. Here we also restrict attention to the predicate ϕ and the agents x and y involved in the correct answer $\boldsymbol{B_x B_y \phi(t_2)}$. These restrictions enable us to compare the information in the various tasks in a uniform way, which we will now do.[4]

Let's start by comparing second-order information. First, note that in the Sally-Anne case, there is an asymmetry in the agents' second-order information (see rows 4 and 5): from time t_1 on, Sally believes that Anne believes that the marble has been moved away from the basket, since Sally can see Anne moving the marble. But Anne is not aware of this (Anne is deceived). On the other hand, in the bake-sale case, the second-order information (rows 9 and 10) is symmetric: at all three times Maria believes that Sam believes they sell chocolate cookies, and Sam also believes that Maria believes they sell chocolate cookies

[4] Note that rows 4,9,14,19 in Table 3 have the same form $\boldsymbol{B_x B_y \phi(t_0)}$, $\boldsymbol{B_x B_y \neg\phi(t_1)}$, $\boldsymbol{B_x B_y \neg\phi(t_2)}$ and are part of the common reasoning pattern leading to the correct answer, hence they are typeset in bold (and blue in the online version). Similarly, rows 2,7,12,17 have the same form $\boldsymbol{B_y \phi(t_0)}$, $\boldsymbol{B_y \neg\phi(t_1)}$, $\boldsymbol{B_y \neg\phi(t_2)}$ and are part of the common reasoning pattern, so they are also bold (and blue). That is, the information in these rows is part of the experimental design, and is intended to ensure that agent x ends up having a false belief about the belief of agent y.

Table 1. Two dimensions of information variation

Task	Zero-order information	Second-order information
Ice-cream	Change-in-world	Symmetry
Bake-sale	Change-in-belief-only	Symmetry
Sally-Anne	Change-in-world	Asymmetry (deception)
Puppy	Change-in-belief-only	Asymmetry (deception)

(so there is no deception).[5] So second-order information in the bake-sale case is *symmetric* whereas in the Sally-Anne case it is not. Similarly, the ice-cream task is symmetric (rows 14 and 15), but the puppy task is not (rows 19 and 20) (Table 3).

Next, let's consider the zero-order information. In the Sally-Anne case we have $basket(t_0)$, $\neg basket(t_1)$, and $\neg basket(t_2)$ (see row 1 of Table 3). So the formula $\boldsymbol{B_y \phi}$ becomes false at t_1 since both the world and the belief agent y has about the world change. On the other hand, in bake-sale we have $\neg chocolate(t_0)$, $\neg chocolate(t_1)$, and $\neg chocolate(t_2)$ (see row 6). In this case, the falsification of $\boldsymbol{B_y \phi}$ at t_1 is not caused by a change in the world, but only by a change in the belief agent y has about the world. We shall say that there is a *change-in-the-world* in the Sally-Anne case, and a *change-in-belief-only* in the bake-sale case. Similarly, there is a change-in-the-world in the ice-cream task (row 11), but a change-in-belief-only in the puppy task (row 16).

Table 3 sums up the zero-order and the second-order informational differences between the tasks. It shows that the bake-sale and (second-order) Sally-Anne tasks are maximally different — they differ both at zero-order and second-order levels — as are the ice-cream and the puppy stories.

Analyzing the first-order information reveals something curious. First, observe that in all four tasks we have $B_x \neg\phi(t_2)$. So at the last stage t_2 of each story, agent x believes — indeed *knows* — that ϕ is false. For example, in the Sally-Anne case, Anne knows that the marble is not in the basket (as she has moved it), and in the bake-sale case, Maria knows that no chocolate cookies are for sale. But in all four tasks we also have $B_x B_y \phi(t_2)$.

That is: in all four tasks there are false beliefs in *two* layers: there is the *outer* layer where the experimental subject has to ascribe a false belief to agent x, but there is also an *inner* layer where agent x ascribes a belief in a proposition to agent y, but agent x knows that this proposition is false. To put it another way: what we might call *inner first-order deception* is built into all four tasks. Note that this is different from the overt second-order deception present in the Sally-Anne and Puppy tasks: second-order deception plays a clear role in their experimental designs. But this inner first-order deception does *not* seem to be

[5] The distinction between tasks that do and do not involve deception is considered important for first-order false beliefs, as deception in a story may signal the relevance of detecting falsehood. But it has been little discussed for second-order tasks; see [13], especially pages 48–49, for discussion and pointers to the literature.

a part of the experimental design of the four second-order false belief tasks: $B_x \neg\phi(t_2)$ is *not* used to derive the correct answers.[6] Nonetheless, it seems hard to devise second-order scenarios which don't have inner first-order deception built into them without the experimental design falling apart. But as far as we are aware, the general presence of this kind of 'deception' is not something that has been noted or discussed in the literature on second-order false-beliefs.

4 A Fine-Grained Analysis

We now make the coarse-grained analysis fine-grained by examining the role of the principle of inertia in more detail. Consider how it is used in the first-order Sally-Anne task. There the child (who we will call Peter) is asked: *Where will Sally look for her marble?* The inertia principle is clearly involved in Peter's reasoning: he takes it for granted that it can be applied to Sally's understanding. Indeed, learning to take it for granted in such circumstances is part of what is meant by acquiring first-order false-belief competence.

In the second-order case, Peter is asked: *Where does Anne think Sally will look for her marble?* Now, this is a question about Anne, thus it might seem that the key reasoning step for Peter is (once again) to take for granted that inertia applies, this time to *Anne's* understanding. After all, Anne never leaves the room, so she is right in front of that marble all the time, so inertia seems relevant. And as Peter observes, Anne does *not* "receive information to the contrary" (because she does not see Sally peek) and so the inertia principle applies and Anne's belief about Sally's belief will be preserved from t_0 to t_2.

But this analysis does not go deep enough. How does Peter "observe" that Anne does not "receive information to the contrary"? He certainly observes that Anne does not see Sally peek—*but what links this observation with Anne's beliefs?* There is a gap here. Peter cannot simply apply the principle of inertia to Anne's understanding; rather, he must understand that Anne is applying inertia to Sally's understanding. Anne reasons that Sally will preserve her belief in the marble being in the basket, *because Anne believes that Sally does not see the marble being moved.* This belief fills the missing gap—it builds a logical "bridge" to Peter's observation.

Summing up, in the fine-grained analysis the principle of inertia is applied *by Anne to Sally's belief* (and not by Peter to Anne's belief). And this has an interesting consequence. It means that Anne is playing the same role in the second-order Sally-Anne task (namely, reasoning about Sally's belief) that Peter played in the first-order task. And this suggests a road to formalization: take a proof that formalizes the first-order task (Peter's reasoning about Sally) and view it instead as formalizing Anne's reasoning about Sally. Nest this proof (at the appropriate place) inside a formalization of Peter's reasoning about the second-order task; this will fill in the missing details about Anne's use of the inertia

[6] Which is why this information is not typeset in bold (and why in the online version, it is not in blue), and also why in Table 2 (the coarse-grained reasoning analysis) it has been put in parentheses.

principle. This is the goal of the following two sections, where we will use natural deduction in hybrid logic to formalize the perspectival reasoning involved.

5 Formalizing the First-Order Sally-Anne Task

First we define the syntax and semantics of the fragment of hybrid logic we use for the formalization of the first-order Sally-Anne task, namely a version of Seligman's [17] *Logic of Correct Description (LCD)*. We assume we are given a set of propositional symbols (to be thought of as placeholders for information that is seen, believed, deduced ..., and so on) and a set of nominals (to be thought of as names of the agents in the scenarios: Sally and Anne in our running example). We assume these sets are disjoint. We use p, q, r, ..., for ordinary propositional symbols and s, a, b, c, ..., for nominals.

Definition 1. *Formulas of LCD are defined by the following grammar:*

$$S ::= \ p \mid a \mid S \wedge S \mid S \rightarrow S \mid \bot \mid @_a S$$

Negation is defined by the convention that $\neg\phi$ is an abbreviation for $\phi \rightarrow \bot$.

Definition 2. *A* model *for LCD is a tuple* $(W, \{V_w\}_{w \in W})$ *where:*

1. *W is a non-empty set; think of these as the agents in the scenario of interest.*
2. *For each w, V_w is a function that to each ordinary propositional symbol assigns an element of* $\{0, 1\}$.

Given a model $\mathfrak{M} = (W, \{V_w\}_{w \in W})$, an *assignment* is a function g that to each nominal assigns an element of W. The relation $\mathfrak{M}, g, w \models \phi$, where g is an assignment, w is an element of W, and ϕ is a formula, is defined as follows:

$$\begin{aligned}
\mathfrak{M}, g, w &\models p \text{ iff } V_w(p) = 1\\
\mathfrak{M}, g, w &\models a \text{ iff } w = g(a)\\
\mathfrak{M}, g, w &\models \phi \wedge \psi \text{ iff } \mathfrak{M}, g, w \models \phi \text{ and } \mathfrak{M}, g, w \models \psi\\
\mathfrak{M}, g, w &\models \phi \rightarrow \psi \text{ iff } \mathfrak{M}, g, w \models \phi \text{ implies } \mathfrak{M}, g, w \models \psi\\
\mathfrak{M}, g, w &\models \bot \text{ iff falsum}\\
\mathfrak{M}, g, w &\models @_a\phi \text{ iff } \mathfrak{M}, g, g(a) \models \phi
\end{aligned}$$

Two remarks. First, nominals should be thought of as naming the unique agent they are true at. For example, we shall use s as a nominal true at Sally; in effect it is a 'name' or 'constant' that picks her out.[7] But nominals are also used to make modalities: if ϕ is an arbitrary formula and s is the nominal that names

[7] There are some interesting possibilities here: we could make our formalization more fine-grained by taking some nominals to stand for times, or go two-dimensional by taking nominals to stand for person-time pairs. But here we stick with the simpler setup just defined, as it has the same granularity as Stenning and Van Lambalgen's work on first-order false-belief tasks, cf. [18], pages 251–259.

$$\frac{c \qquad \phi}{@_c\phi}\,(@I) \qquad\qquad \frac{c \qquad @_c\phi}{\phi}\,(@E)$$

$$\frac{\phi_1 \quad \ldots \quad \phi_n \qquad \begin{array}{c}[\phi_1]\ldots[\phi_n][c]\\ \vdots\\ \psi\end{array}}{\psi}\,(\mathit{Term})^* \qquad\qquad \frac{\begin{array}{c}[c]\\ \vdots\\ \psi\end{array}}{\psi}\,(\mathit{Name})^\dagger$$

$*$ $\phi_1, \ldots, \phi_n$ and ψ are satisfaction statements, and there are no undischarged assumptions in the derivation of ψ besides the specified occurrences of $\phi_1, \ldots, \phi_n$ and c.
$\dagger$ The nominal c does not occur in ψ or in any undischarged assumptions other than the specified occurrences of c.

Fig. 1. Natural deduction rules for LCD

Sally, then a new formula $@_s\phi$ can be built. The $@_s$ prefix is called a *satisfaction operator* and the formula $@_s\phi$ is called a *satisfaction statement*. Satisfaction statements let us switch perspectives: if we evaluate the satisfaction statement $@_s\phi$ at *any* agent in a model, it will be true iff ϕ is true at Sally.

Second, note that we have not introduced any modalities apart from the satisfaction operators. But this is not an oversight. In what follows the reader will encounter expressions of the form $@_sS\phi$ (that is, Sally sees ϕ) and $@_sB\phi$ (that is, Sally believes ϕ). But as far as the analysis of *first-order* false-belief tasks is concerned, expressions containing the modalities S and B are not used in genuinely modal reasoning. Indeed, expressions of the form $S\phi$ and $B\phi$ are essentially complicated-looking propositional symbols: they are only used in simple propositional reasoning and then fed (once) into a perspective-shifting natural-deduction rule called *Term*. This will change (at least for the B operator) in the following section when we formalize the second-order task.

This brings us to natural deduction system we shall use to analyse the first-order Sally-Anne task.[8] We use the system for *LCD* obtained by extending the standard natural deduction system for classical propositional logic with the rules in Fig. 1; the symbol c is an arbitrary nominal (that is, the name of an arbitrary agent). This is a modified version of Seligman's original natural deduction system for *LCD* [17]; these rules here are from Chap. 4 of [5]. We omit the rules for the boolean connectives: they are standard, and we prefer the more perspicuous proof trees obtained by 'compiling down' the simple propositional reasoning involved into additional rules (see the examples in the Appendix). In [5], this natural deduction system is proved to be sound and complete:

[8] Natural deduction was originally developed to model mathematical argumentation, but there is now some experimental backing for the claim that it is a mechanism underlying human deductive reasoning more generally; see [16]. One of the reasons we chose hybrid logic for our analysis (rather than, say, a multi-agent doxastic logic) was because of its well-behaved natural deduction systems; see [5].

Theorem 1. *Let ψ be a formula and Γ a set of LCD wffs. The first statement below implies the second statement (soundness) and vice versa (completeness).*

1. *The formula ψ is derivable from Γ in Seligman's natural deduction system.*
2. *For any model $\mathfrak{M}$, any world w, and any assignment g, if, for any formula $\theta \in \Gamma$, it is the case that $\mathfrak{M}, g, w \models \theta$, then $\mathfrak{M}, g, w \models \psi$.*

Let's take a closer look. The rules $@I$ and $@E$ in Fig. 1 are the introduction and elimination rules for satisfaction operators. The $@I$ rule says that if we have the information c (so we are reasoning about the agent called c) and we also have the information ϕ, then we can introduce the satisfaction operator $@_c$ and conclude $@_c\phi$, which says that ϕ holds from c's perspective. The $@E$ rule says: suppose that when reasoning about the agent named c, we also have the information that $@_c\phi$. Then we can eliminate $@_c$ and conclude ϕ.

But it is the *Term* rule that is central to the formalization. This rule lets us switch to another agent's perspective using hypothetical reasoning: the bracketed expressions $[\phi_1] \dots [\phi_n][c]$ in the statement of the rule are (discharged) assumptions. The key assumption is c, which can be glossed as: let's switch perspective and temporarily adopt c's point of view.[9] The remaining (discharged) assumptions $[\phi_1] \dots [\phi_n]$ in the rule's statement are additional assumptions we may wish to make about the information available from c's perspective.[10]

The rule works as follows. Suppose that on the basis of assumptions $\phi_1 \dots \phi_n, c$ we deduce ψ from c's perspective. Then the *Term* rule tells us that if $\phi_1 \dots \phi_n$ are available in the original perspective,[11] then we can discharge the assumption (which we do by bracketing them, thus obtaining $[\phi_1] \dots [\phi_n][c]$) and conclude ψ unconditionally in the original perspective.

The *Term* rule is a subtle and powerful rule.[12] Indeed, as was first shown in [6], the hybrid logical analysis of the first-order Sally-Anne task boils down to a *single* application of *Term*. Recall that Peter is the child performing the task. To answer the question (*Where will Sally look for her marble?*) Peter reasons as follows. At the time t_0, Sally believed the marble to be in the basket. She saw no action to move it, so she still believed this at t_1. When she returned at t_2, she still believed the marble to be in the basket (after all, she was out of the room when Anne moved it at time t_1). Peter concludes that Sally believes that the marble is still in the basket.

[9] Incidentally, when using the *Term* rule we make at least one assumption c, but we can make several, and this is often necessary to drive the proof through.

[10] The *Name* rule tells us that if we can prove the information ϕ by adopting some *arbitrary* perspective c, then ϕ also holds from the original perspective. As we won't use this rule in our analysis, we refer to [5] for further discussion.

[11] Indicated by the premisses $\phi_1 \dots \phi_n$ listed just above the horizontal line in the statement of *Term* given in Fig. 1.

[12] A subtlety worth emphasising is that (as is stated in Fig. 1) the assumptions $[\phi_1] \dots [\phi_n]$ must all be satisfaction statements, otherwise the rule is not sound. We refer the reader to [17] and Chap. 4 of [5] for further discussion.

To formalize this we use the nominal s to name Sally, and the modal operators S (*sees that*) and B (*believes that*). The predicate $l(i, t)$ means that *the marble is at location i at time t*. Predicate $m(t)$ means that *the marble is moved at time t*. We take time to be discrete, and use $t + 1$ as the successor of t. Using this vocabulary we can express the four belief formation principles we need:[13]

(D) $B\neg\phi \rightarrow \neg B\phi$
($P1$) $S\phi \rightarrow B\phi$
($P2$) $Bl(i, t) \wedge \neg Bm(t) \rightarrow Bl(i, t + 1)$
($P3$) $Bm(t) \rightarrow Sm(t)$

With the help of these principles, the perspectival reasoning involved in the Sally-Anne task can be formalized as the derivation in Fig. 3 (in the Appendix). We have already given Peter's informal perspectival reasoning; the formal proof mirrors it in full detail using a single application of *Term* in which the assumptions of s model the shift to Sally's perspective. The first two premises $@_s Sl(basket, t_0)$ and $@_s S\neg m(t_0)$ taken together say that Sally at the earlier time t_0 saw that the marble was in the basket and that no action was taken to move it. The third premise, $@_s\neg Sm(t_1)$, says that Sally did not see the marble being moved at the time t_1 (since she was absent). Note that when applying the belief formation principles, we simply use them as rules.[14]

The bulk of the reasoning on the right-hand-side of the proof tree in Fig. 3 simply consists of a sequence of applications of belief formation principles until the crucial formula $@_s Bl(basket, t_2)$ — *Sally believes the ball is in the basket* — is deduced. What turns this into a formalisation of correct reasoning in the Sally-Anne task is the way the sequencing of belief formation principles is perspectivized. The right-hand-side sequencing occurs between the initial assumptions of s (which perspectivizes it as *Sally's* reasoning) and the final application of *Term* which lets us conclude that the crucial formula is also true from *Peter's* point of view. In short, the analysis consists of *Belief Formation + Perspectival Reasoning* correctly combined.

---

[13] As we mentioned earlier, "belief formation" (and "belief manipulation") is terminology we have borrowed from [18], and we discuss them in more detail shortly. As for the belief formations principles themselves, we have already met Principle (D) which says if we believe that something is false, then we don't believe it. Principle ($P1$) states that a belief in ϕ may be formed as a result of seeing ϕ; this is principle (9.2) in [18], page 251. Principle ($P2$) is (pretty clearly) a principle of inertia: a belief that the predicate l is true is preserved from a time t to its successor $t + 1$, unless it is believed that the marble moved at t. This is essentially Principle (9.11) from [18], page 253, and axiom $[A_5]$ in [1], page 20. Principle ($P3$) encodes the information that *seeing* the marble being moved is the only way a belief that the marble is being moved can be acquired. Obviously this is not a general truth, but the point of the formalization is simply to capture Peter's reasoning in the Sally-Anne scenario.

[14] As we remarked earlier, we do this to 'compile down' the simple propositional reasoning involved. Strictly speaking, deducing $B\phi$ from $S\phi$ requires us to apply the propositional rule of *modus ponens* to $S\phi \rightarrow B\phi$. Using the belief formation principles as additional natural deduction rules enables us to omit such steps and reduce the size of the proof tree.

Analagous remarks are made by Stenning and Van Lambalgen about their own analysis of first-order false-belief tasks; see [18], page 257. They note that the bulk of the reasoning involves belief formation principles and their analysis succeeds because it is carrying out using closed world reasoning; we might summarise their approach as *Belief Formation + Closed World Reasoning* correctly combined. However they then go on to remark that what they call *Belief Manipulation* rules (which codify how to reason from one belief state to another) are unnecessary. Now, as far as first-order false-belief reasoning is concerned, we agree completely. Indeed, until now we have provided no proof rules for manipulating the belief operator B beyond the belief formation principles. And that is because, for the first-order Sally-Anne task, we had no need of anything else. But a belief manipulation rule will be needed if we are to extend our perspectival analysis to the second-order Sally-Anne task.[15] We turn to this task now.

6 Formalizing the Second-Order Sally-Anne Task

As we remarked at the end of Sect. 4, Anne plays the same role in the second-order Sally-Anne task (namely, reasoning about Sally's belief) that Peter played in the first-order task. This suggests that we should take the proof we have just given (formalizing Peter's reasoning about Sally), view it as formalizing Anne's reasoning about Sally, and nest it (at the appropriate place) inside a formalization of Peter's reasoning about the second-order task. That is, we should add another level of nesting to the perspectival analysis. To make this work we have to introduce a recursive belief manipulation rule for B. We have chosen the rule given in Fig. 2. We call it BM. It is a version of a rule from [9] that fits naturally our tree-style natural deduction proofs.[16]

$$\dfrac{B\phi_1 \quad \dots \quad B\phi_n \qquad \begin{array}{c}[\phi_1] \dots [\phi_n] \\ \vdots \\ \psi\end{array}}{B\psi}\ (\mathsf{BM})^*$$

$*$ There are no undischarged assumptions in the derivation of ψ except the specified occurrences of $\phi_1, \dots, \phi_n$.

Fig. 2. Belief manipulation rule for the B operator

[15] Stenning and Van Lambalgen do not analyse second-order false-belief tasks.

[16] So we are adding natural deduction machinery for the minimal modal logic K and thus treating B as a full-fledged modal operator. In this paper we won't discuss the model-theoretic changes required — but we *do* believe that the fact that a semantic enrichment is called for at this point adds weight to our argument that the transition from first- to second-order reasoning involves conceptual change.

And now to complete the formalization. We shall use the nominal a as a name for Anne, read $D\phi$ as ϕ is deducible, and make use of a natural deduction formulation of the following belief formation principle:

$(P0) \qquad D\phi \rightarrow B\phi$

This says that if we can deduce the information ϕ then we believe ϕ (this is principle (9.4) in [18], page 251). With this machinery in place, the reasoning in the second-order Sally-Anne task can be formalized by the proof tree in Fig. 4 in the Appendix. Note that the first-order proof in nested inside: the dots in the upper-right corner of Fig. 4 indicate where.

The proof's conclusion, $@_a B @_s Bl(basket, t_2)$, says that Anne believes that Sally believes that the marble is in the basket at the time t_2, and this is indeed the correct response to the second-order task. And Peter can prove this as follows.

The first two premises used in the application of *Term* with which the proof concludes, $@_a S @_s Sl(basket, t_0)$ and $@_a S @_s S \neg m(t_0)$, say that at time t_0, Anne saw that Sally saw that the marble was in the basket and that no action was taken to move it. The third premise used in the concluding application of the *Term* rule, $@_a D @_s \neg Sm(t_1)$, says that Anne deduced that Sally did not see the marble being moved at the time t_1, which is true.

But the essential step is the way the belief manipulation rule BM glues together the two levels of perspectival reasoning. The embedded proof (which reasons from Sally's perspective) yields the conclusion $@_s Bl(basket, t_2)$, the correct response to the first-order task. But Peter can't use this information directly: he needs to know that *Anne believes this*. But the application of BM prefixes the belief operator to form $B @_s Bl(basket, t_2)$, and the very next step of the proof shows that this belief holds from Anne's point of view. Thus the reasoning on the right has now been incorporated back into Anne's perspective, and so can be fed into *Term*, and Peter has his answer.

7 Concluding Discussion

Second-order reasoning is more complex than first-order — the previous section with its embedded proof and use of the BM rule showed this clearly.[17] Nonetheless, our analysis also suggests that the transition to second-order competence marks a more significant development than is suggested by the complexity only position: the full *reification* of beliefs. Attainment of first-order false-belief competence marks the stage at which the child becomes aware of the fact that beliefs held by other agents can be false; second-order competence, on the other hand, marks the stage where beliefs become objects in their own right that can be manipulated. This shift is mirrored in our analysis: we jumped from a logic that

[17] Indeed, our analysis allows us to tentatively indicate the shift in complexity. The *LCD* fragment is NP-complete. By adding BM we have moved to a PSPACE-hard modal logic. So our analysis of the first-order Sally-Anne task is carried out in computationally simpler logic than the second-order case (assuming P $\neq$ NP).

permitted only *Belief Formation + Perspectival Reasoning* to one that allowed unrestricted *Belief Manipulation* as well.

This is a significant advance. Beliefs are special objects: they are abstract, invisible, and though ‘about’ the world, they may very well be false. Typically developing children learn this first lesson around the age of four, but there is a further lesson they must learn: that beliefs can be embedded one inside another and freely manipulated. Something like the BM rule seems to be required to capture this step. It is tempting to speculate that at this developmental stage some sort of “recursion module” is adapted to handle these strange new objects, but be that as it may, in typically developing children the reasoning architecture is certainly enriched in an important way at around the age of six.[18]

Recursively stacked beliefs lie at the heart of this transition, which brings us to our empirical work [15]. Our logical investigations were carried out as part of an ongoing training study involving Danish speaking children with ASD. Our empirical work is driven by the hypothesis that, in case of children with ASD, improving linguistic recursion competency predicts belief manipulation mastery required by second-order false-belief tasks. We are investigating whether children with ASD use language as a “scaffolding” to support developing understanding of other minds, an explanation advanced in the first-order case by [10].

Acknowledgements. We thank the referees for their valuable comments and questions. The authors acknowledge the funding received from the VELUX FOUNDATION for the project *Hybrid-Logical Proofs at Work in Cognitive Psychology* (VELUX 33305).

Appendix

The Appendix contains the coarse-grained reasoning for the four tasks (Table 2), the table listing their information content (Table 3), the texts of the bake-sale, ice-cream and puppy tasks (Tables 4, 5 and 6 respectively) and the formalization of the first-order and second-order Sally-Anne tasks (Figs. 3 and 4).

[18] Our formalization does suggest a hypothesis which may be empirically testable. Although we have talked of acquiring *second-order* competency, to acquire (something like) the BM rule is to acquire a fully recursive competency. That is, once the child has acquired BM, there should be nothing more to learn, for the rule covers the third, fourth, fifth, . . . , and all higher-order levels. That is, we suspect that false-belief competency comes in two stages for typically developing children: first-order competency (at around the age of four) and all the rest (at around the age of six). But designing an experiment to test this is likely to be difficult. Apart from anything else, higher levels of reasoning impose heavy cognitive loads very fast, and it is unclear how such performance effects could be disentangled experimentally.

Table 2. A coarse-grained analysis of second-order false-belief tasks in terms of belief-states

	Time t_0	*Time t_1*	*Time t_2*
Second-order Sally-Anne task	*Sally leaves after having put the marble in the basket* Anne now believes that Sally thinks that the marble is in the basket $B_{anne}B_{sally}basket(t_0)$	*Anne moves the marble from the basket to the box* Sally sees through the keyhole that the marble is moved $B_{sally}box(t_1)$ So $B_{sally}\neg basket(t_1)$ and hence $\neg B_{sally}basket(t_1)$ (Anne knows that the marble is moved so $B_{anne}\neg basket(t_1)$)	*Sally has returned* Correct answer: "Anne believes that Sally thinks that the marble is in the basket" $B_{anne}B_{sally}basket(t_2)$ Derivable by inertia from t_0 as Anne does not know that Sally's belief changed at t_1
Bake-sale task	*Maria tells Sam that she will go to buy chocholate cookies* Maria now believes that Sam thinks that they sell chocolate cookies $B_{maria}B_{sam}chocolate(t_0)$	*Mom comes home and Maria arrives at the bake sale* Sam is told that they sell pumpkin pie $B_{sam}pumpkin(t_1)$ So $B_{sam}\neg chocolate(t_1)$ and hence $\neg B_{sam}chocolate(t_1)$ (Maria realizes that they sell brownies so $B_{maria}\neg chocolate(t_1)$)	*The mailman talks to Maria* Correct answer to the mailman: "I [Maria] believe that Sam thinks that they sell chocolate cookies" $B_{maria}B_{sam}chocolate(t_2)$ Derivable by inertia from t_0 as Maria does not know that Sam's belief changed at t_1
Ice-cream task	*Mary leaves the park after having seen the van* John believes that Mary thinks that the van is in the park $B_{john}B_{mary}park(t_0)$	*John and Mary independently talk to the ice-cream man* Mary is told that the van drives to the church $B_{mary}church(t_1)$ So $B_{mary}\neg park(t_1)$ and hence $\neg B_{mary}park(t_1)$ (John is told that the van drives to the church so $B_{john}\neg park(t_1)$)	*The van has arrived* Correct answer: "John believes that Mary thinks that the van is in the park" $B_{john}B_{mary}park(t_2)$ Derivable by inertia from t_0 as John does not know that Mary's belief changed at t_1
Puppy task	*Mom tells Peter that she has got him a toy* Mom now believes that Peter thinks that he will get a toy $B_{mom}B_{peter}toy(t_0)$ (Mom knows that Peter will get a puppy so $B_{mom}\neg toy(t_0)$ is true)	*Peter finds a puppy in the basement* Peter realizes that he will get a puppy $B_{peter}puppy(t_1)$ So $B_{peter}\neg toy(t_1)$ and hence $\neg B_{peter}toy(t_1)$	*Grandmother talks to Mom* Correct answer to Grandmother: "I [Mom] believe that Peter thinks that he will get a toy" $B_{mom}B_{peter}toy(t_2)$ Derivably by inertia from t_0 as Mom does not know that Peter's belief changed at t_1

Table 3. Zero-order, first-order and second-order information in the tasks

Sally-Anne	*Time* t_0	*Time* t_1	*Time* t_2	
Zero-order	$basket(t_0)$	$\neg basket(t_1)$	$\neg basket(t_2)$	1
First-order	$\boldsymbol{B_{sally}\, basket(t_0)}$	$\boldsymbol{B_{sally} \neg basket(t_1)}$	$\boldsymbol{B_{sally} \neg basket(t_2)}$	2
	$B_{anne}\, basket(t_0)$	$B_{anne} \neg basket(t_1)$	$B_{anne} \neg basket(t_2)$	3
Second-order	$\boldsymbol{B_{anne} B_{sally}\, basket(t_0)}$	$\boldsymbol{B_{anne} B_{sally}\, basket(t_1)}$	$\boldsymbol{B_{anne} B_{sally}\, basket(t_2)}$	4
	$B_{sally} B_{anne}\, basket(t_0)$	$B_{sally} B_{anne} \neg basket(t_1)$	$B_{sally} B_{anne} \neg basket(t_2)$	5
Bake-sale				
Zero-order	$\neg chocolate(t_0)$	$\neg chocolate(t_1)$	$\neg chocolate(t_2)$	6
First-order	$\boldsymbol{B_{sam}\, chocolate(t_0)}$	$\boldsymbol{B_{sam} \neg chocolate(t_1)}$	$\boldsymbol{B_{sam} \neg chocolate(t_2)}$	7
	$B_{maria}\, chocolate(t_0)$	$B_{maria} \neg chocolate(t_1)$	$B_{maria} \neg chocolate(t_2)$	8
Second-order	$\boldsymbol{B_{maria} B_{sam}\, chocolate(t_0)}$	$\boldsymbol{B_{maria} B_{sam}\, chocolate(t_1)}$	$\boldsymbol{B_{maria} B_{sam}\, chocolate(t_2)}$	9
	$B_{sam} B_{maria}\, chocolate(t_0)$	$B_{sam} B_{maria}\, chocolate(t_1)$	$B_{sam} B_{maria}\, chocolate(t_2)$	10
Ice-cream				
Zero-order	$park(t_0)$	$\neg park(t_1)$	$\neg park(t_2)$	11
First-order	$\boldsymbol{B_{mary}\, park(t_0)}$	$\boldsymbol{B_{mary} \neg park(t_1)}$	$\boldsymbol{B_{mary} \neg park(t_2)}$	12
	$B_{john}\, park(t_0)$	$B_{john} \neg park(t_1)$	$B_{john} \neg park(t_2)$	13
Second-order	$\boldsymbol{B_{john} B_{mary}\, park(t_0)}$	$\boldsymbol{B_{john} B_{mary}\, park(t_1)}$	$\boldsymbol{B_{john} B_{mary}\, park(t_2)}$	14
	$B_{mary} B_{john}\, park(t_0)$	$B_{mary} B_{john}\, park(t_1)$	$B_{mary} B_{john}\, park(t_2)$	15
Puppy				
Zero-order	$\neg toy(t_0)$	$\neg toy(t_1)$	$\neg toy(t_2)$	16
First-order	$\boldsymbol{B_{peter}\, toy(t_0)}$	$\boldsymbol{B_{peter} \neg toy(t_1)}$	$\boldsymbol{B_{peter} \neg toy(t_2)}$	17
	$B_{mom} \neg toy(t_0)$	$B_{mom} \neg toy(t_1)$	$B_{mom} \neg toy(t_2)$	18
Second-order	$\boldsymbol{B_{mom} B_{peter}\, toy(t_0)}$	$\boldsymbol{B_{mom} B_{peter}\, toy(t_1)}$	$\boldsymbol{B_{mom} B_{peter}\, toy(t_2)}$	19
	$B_{peter} B_{mom}\, toy(t_0)$	$B_{peter} B_{mom} \neg toy(t_1)$	$B_{peter} B_{mom} \neg toy(t_2)$	20

Table 4. The bake-sale task (quoted from [11], pictures and some questions omitted)

Sam and Maria are playing together. They look outside and see that the church is having a bake sale. Maria tells Sam: “I am going to buy chocolate chip cookies for us there,” and she walks away.

Mom comes home and she tells Sam that she just drove past the bake sale. “Are they selling chocolate chip cookies?” Sam asks. No, mum says, “they are only selling pumpkin pie.” “Maria will now probably get pumpkin pie at the bake sale,” Sam says.

Maria has arrived at the bake sale. “I would like to buy chocolate chip cookies,” she says. “All we have left are brownies,” says the lady behind the stall. Since Maria also likes brownies, she decides to get some brownies.

On her way back, Maria meets the mailman. She tells the mailman: “I have just bought some brownies. I am going to share them with my brother Sam. It is a surprise”. “That is nice of you,” says the mailman. Then he asks Maria: “Does Sam know what you bought him?”

Ignorance: What does Maria tell the mailman?

Then the mailman asks: “What does Sam think they are selling at the bake sale?”

Second-order false-belief question: What does Maria tell the mailman?

Table 5. The ice-cream task (introduced in [14], quoted from [12], a question omitted)

This is a story about John and Mary who live in this village. This morning John and Mary are together in the park. In the park there is also an ice-cream man in his van.
Mary would like to buy an ice cream but she has left her money at home. So she is very sad. “Don’t be sad,” says the ice-cream man, “you can fetch your money and buy some ice cream later. I’ll be here in the park all afternoon.” “Oh good,” says Mary, “I’ll be back in the afternoon to buy some ice cream. I’ll make sure I won’t forget my money then.”
So Mary goes home. . . . She lives in this house. She goes inside the house. Now John is on his own in the park. To his surprise he sees the ice-cream man leaving the park in his van. “Where are you going?” asks John. The ice-cream man says, “I’m going to drive my van to the church. There is no one in the park to buy ice cream; so perhaps I can sell some outside the church.”
The ice-cream man drives over to the church. On his way he passes Mary’s house. Mary is looking out of the window and spots the van. “Where are you going?” she asks. “I’m going to the church. I’ll be able to sell more ice cream there,” answers the man. “It’s a good thing I saw you,” says Mary. Now John doesn’t know that Mary talked to the ice-cream man. He doesn’t know that!
Now John has to go home. After lunch he is doing his homework. He can’t do one of the tasks. So he goes over to Mary’s house to ask for help. Mary’s mother answers the door. “Is Mary in?” asks John. “Oh,” says Mary’s mother. “She’s just left. She said she was going to get an ice cream.”
Test question: So John runs to look for Mary. Where does he think she has gone?

Table 6. The puppy task (introduced in [19], quoted from [12], some questions omitted)

Tonight it’s Peter’s birthday and Mom is surprising him with a puppy. She has hidden the puppy in the basement. Peter says, “Mom, I really hope you get me a puppy for my birthday.” Remember, Mom wants to surprise Peter with a puppy. So, instead of telling Peter she got him a puppy, Mom says, “Sorry Peter, I did not get you a puppy for your birthday. I got you a really great toy instead.”
Now, Peter says to Mom, “I’m going outside to play.” On his way outside, Peter goes down to the basement to fetch his roller skates. In the basement, Peter finds the birthday puppy! Peter says to himself, “Wow, Mom didn’t get me a toy, she really got me a puppy for my birthday.” Mom does not see Peter go down to the basement and find the birthday puppy.
Now, the telephone rings, ding-a-ling! Peter’s grandmother calls to find out what time the birthday party is. Grandma asks Mom on the phone, “Does Peter know what you really got him for his birthday?”
Now remember, Mom does not know that Peter saw what she got him for his birthday. Then, Grandma says to Mom, “What does Peter think you got him for his birthday?”
Second-order false-belief question: What does Mom say to Grandma?

$$
\dfrac{@_s Sl(\mathit{basket}, t_0)\ \ @_s S\neg m(t_0)\ \ @_s \neg Sm(t_1) \qquad \dfrac{[s] \qquad \dfrac{\dfrac{\dfrac{\dfrac{[s]\ [@_s Sl(\mathit{basket}, t_0)]}{Sl(\mathit{basket}, t_0)}\,(@E)}{Bl(\mathit{basket}, t_0)}\,(P1) \quad \dfrac{\dfrac{\dfrac{[s]\ [@_s S\neg m(t_0)]}{S\neg m(t_0)}\,(@E)}{B\neg m(t_0)}\,(P1)}{\neg Bm(t_0)}\,(\mathsf{D})}{Bl(\mathit{basket}, t_1)}\,(P2) \quad \dfrac{\dfrac{[s]\ [@_s \neg Sm(t_1)]}{\neg Sm(t_1)}\,(@E)}{\neg Bm(t_1)}\,(P3)}{Bl(\mathit{basket}, t_2)}\,(P2)}{@_s Bl(\mathit{basket}, t_2)}\,(@I)}{@_s Bl(\mathit{basket}, t_2)}\,(\mathit{Term})
$$

Fig. 3. Formalization of the child's correct response in the first-order Sally-Anne task

$$
\dfrac{@_a S @_s Sl(\mathit{basket}, t_0)\ @_a S @_s S\neg m(t_0)\ @_a D @_s \neg Sm(t_1) \qquad \dfrac{\dfrac{\dfrac{\dfrac{[a][@_a S @_s Sl(\mathit{basket}, t_0)]}{S @_s Sl(\mathit{basket}, t_0)}}{B @_s Sl(\mathit{basket}, t_0)}\,(P1) \quad \dfrac{\dfrac{[a][@_a S @_s S\neg m(t_0)]}{S @_s S\neg m(t_0)}}{B @_s S\neg m(t_0)}\,(P1) \quad \dfrac{\dfrac{[a][@_a D @_s \neg Sm(t_1)]}{D @_s \neg Sm(t_1)}}{B @_s \neg Sm(t_1)}\,(P0) \quad \begin{matrix}[@_s Sl(\mathit{basket}, t_0)][@_s S\neg m(t_0)][@_s \neg Sm(t_1)] \\ \vdots \\ @_s Bl(\mathit{basket}, t_2)\end{matrix}}{B @_s Bl(\mathit{basket}, t_2)}\,(\mathsf{BM}) \quad [a]}{@_a B @_s Bl(\mathit{basket}, t_2)}}{@_a B @_s Bl(\mathit{basket}, t_2)}\,(\mathit{Term})
$$

The vertical dots in the upper-right corner represent the derivation in Figure 3. So this proof contains two applications of *Term*: the concluding application, which is shown, and the one inside the earlier proof, which is not. To save space, we have omitted names of the introduction and elimination rules for the @ operator.

Fig. 4. Formalization of the child's correct response in the second-order Sally-Anne task

References

1. Arkoudas, K., Bringsjord, S.: Toward formalizing common-sense psychology: an analysis of the false-belief task. In: Ho, T.-B., Zhou, Z.-H. (eds.) PRICAI 2008. LNCS (LNAI), vol. 5351, pp. 17–29. Springer, Heidelberg (2008)
2. Baron-Cohen, S., Leslie, A.M., Frith, U.: Does the autistic child have a 'theory of mind'? Cognition **21**(1), 37–46 (1985)
3. Baron-Cohen, S., O'Riordan, M., Stone, V., Jones, R., Plaisted, K.: Recognition of faux pas by normally developing children and children with Asperger syndrome or high-functioning autism. J. Autism Dev. Disord. **29**(5), 407–418 (1999)
4. Bolander, T.: Seeing is believing: formalising false-belief tasks in dynamic epistemic logic. In: Herzig, A., Lorini, E. (eds.) Proceedings of the European Conference on Social Intelligence (ECSI-2014), pp. 87–107. Toulouse University, France, IRIT-CNRS (2014)
5. Braüner, T.: Hybrid Logic and its Proof-Theory. Applied Logic Series, vol. 37. Springer, Heidelberg (2011)
6. Braüner, T.: Hybrid-logical reasoning in the Smarties and Sally-Anne tasks. J. Logic Lang. Inf. **23**, 415–439 (2014)
7. Braüner, T.: Hybrid-logical reasoning in the Smarties and Sally-Anne tasks: what goes wrong when incorrect responses are given? In: Proceedings of the 37th Annual Meeting of the Cognitive Science Society, pp. 273–278. Cognitive Science Society, Pasadena, California (2015)
8. Braüner, T., Blackburn, P., Polyanskaya, I.: Recursive belief manipulation and second-order false-beliefs. In: Proceedings of the 38th Annual Meeting of the Cognitive Science Society. Cognitive Science Society, Philadelphia, Pennsylvania, USA (2016, to appear)
9. Fitting, M.: Modal proof theory. In: Blackburn, P., van Benthem, J., Wolter, F. (eds.) Handbook of Modal Logic, pp. 85–138. Elsevier, New York (2007)
10. Hale, C., Tager-Flusberg, H.: The influence of language on theory of mind: a training study. Dev. Sci. **6**, 346–359 (2003)
11. Hollebrandse, B., van Hout, A., Hendriks, P.: Children's first and second-order false-belief reasoning in a verbal and a low-verbal task. Synthese **191**, 321–333 (2014)
12. Miller, S.: Children's understanding of second-order mental states. Psychol. Bull. **135**, 749–773 (2009)
13. Miller, S.: Theory of Mind: Beyond the Preschool Years. Psychology Press, New York (2012)
14. Perner, J., Wimmer, H.: "John thinks that Mary thinks that..": attribution of second-order beliefs by 5-to 10-year-old children. J. Exp. Child Psychol. **39**, 437–471 (1985)
15. Polyanskaya, I., Braüner, T., Blackburn, P.: Linguistic recursion and Autism Spectrum Disorder. Manuscript (2016)
16. Rips, L.: Logical approaches to human deductive reasoning. In: Adler, J., Rips, L. (eds.) Reasoning: Studies of Human Inference and Its Foundations, pp. 187–205. Cambridge University Press, Cambridge (2008)
17. Seligman, J.: The logic of correct description. In: de Rijke, M. (ed.) Advances in Intensional Logic. Applied Logic Series, vol. 7, pp. 107–135. Kluwer, Dordrecht (1997)
18. Stenning, K., van Lambalgen, M.: Human Reasoning and Cognitive Science. MIT Press, Cambridge (2008)

19. Sullivan, K., Zaitchik, D., Tager-Flusberg, H.: Preschoolers can attribute second-order beliefs. Dev. Psychol. **30**, 395–402 (1994)
20. Szymanik, J., Meijering, B., Verbrugge, R.: Using intrinsic complexity of turn-taking games to predict participants' reaction times. In: Knauff, M., Pauen, M., Sebanz, N., Wachsmuth, I. (eds.) Proceedings of the 35th Annual Conference of the Cognitive Science Society, pp. 1426–1432. Cognitive Science Society, Austin (2013)
21. Wellman, H., Cross, D., Watson, J.: Meta-analysis of theory-of-mind development: the truth about false-belief. Child Dev. **72**, 655–684 (2001)

Categories: How I Learned to Stop Worrying and Love Two Sorts

Willem Conradie[1], Sabine Frittella[2(✉)], Alessandra Palmigiano[1,2], Michele Piazzai[2], Apostolos Tzimoulis[2], and Nachoem M. Wijnberg[3]

[1] Department of Pure and Applied Mathematics, University of Johannesburg, Johannesburg, South Africa
wconradie@uj.ac.za

[2] Faculty of Technology, Policy and Management, Delft University of Technology, Delft, The Netherlands
{S.S.A.Frittella,A.Palmigiano,M.Piazzai,A.Tzimoulis-1}@tudelft.nl

[3] Amsterdam Business School, University of Amsterdam, Amsterdam, The Netherlands
n.m.wijnberg@uva.nl

Abstract. RS-frames were introduced by Gehrke as relational semantics for substructural logics. They are two-sorted structures, based on RS-polarities with additional relations used to interpret modalities. We propose an intuitive, epistemic interpretation of RS-frames for modal logic, in terms of categorization systems and agents' subjective interpretations of these systems. Categorization systems are a key to any decision-making process and are widely studied in the social and management sciences.

A set of objects together with a set of properties and an incidence relation connecting objects with their properties forms a polarity which can be 'pruned' into an RS-polarity. Potential categories emerge as the Galois-stable sets of this polarity, just like the concepts of Formal Concept Analysis. An agent's beliefs about objects and their properties (which might be partial) is modelled by a relation which gives rise to a normal modal operator expressing the agent's beliefs about category membership. Fixed-points of the iterations of the belief modalities of all agents are used to model categories constructed through social interaction.

Keywords: Lattice-based modal logic · RS-frames · Categorization theory · Epistemic logic · Formal concept analysis

1 Introduction

Relational semantic frameworks for logics algebraically captured by varieties of normal lattice expansions[1] have been intensely investigated for more than three decades

[1] A *normal lattice expansion* is a bounded lattice endowed with operations of finite arity, each coordinate of which is either positive (i.e. order-preserving) or negative (i.e. order-reversing). Moreover, these operations are either finitely join-preserving (resp. meet-reversing) in their positive (resp. negative) coordinates, or are finitely meet-preserving (resp. join-reversing) in their positive (resp. negative) coordinates.

J. Väänänen et al. (Eds.): WoLLIC 2016, LNCS 9803, pp. 145–164, 2016.
DOI: 10.1007/978-3-662-52921-8_10

[3,15,17,19,22,25,27,30,31,33,34,39,40]. However, none of these frameworks has gained the same pre-eminence and success as Kripke semantics. Indeed, the extant proposals are regarded as significantly less intuitive than Kripke structures, especially w.r.t. their possibility to support the various established interpretations of modal operators (e.g. epistemic, temporal, dynamic), and hence doubts have been raised as to the suitability of these logics for applications. Various directions have been explored to try and cope with these difficulties, such as: (a) attempts to provide a conceptual justification to some of the distinctive features of these semantics (for instance, in [25], a conceptual motivation has been given for the 'two-sortedness' of the relational semantics for substructural logics introduced in the same paper in terms of a duality between states and information quanta); (b) recapturing the usual definition of the interpretation clause of modal operators in a generalized context [27,28]; (c) improving the modularity of mathematical theories such as correspondence theory, to facilitate the transfer of results across different semantic settings. The latter direction has been implemented specifically for lattice-based logics in [6,9,10], and pursued more in general in [4,5,7,8,11–14,21,26,37,38].

The contribution of the present paper pertains to direction (a): we propose categorization theory in management science as a concrete frame of reference for understanding the *RS-semantics* of lattice-based modal logic, and we argue that, when understood in this light, a natural epistemic interpretation can be given to the modal operators, which captures e.g. the factivity and positive introspection of knowledge.

Our starting point is the connection, mentioned also in [25], between RS-semantics and Formal Concept Analysis (FCA) [24]. Namely, *RS-frames* for normal lattice-based modal logics are based on *polarities*, that is, tuples $(A, X, \perp)$ such that A and X are sets, and $\perp \subseteq A \times X$. In FCA, polarities can be understood as *formal contexts*, consisting of objects (the elements of A) and properties (the elements of X) with the relation $\perp$ indicating which object satisfies which property. It is well known that any polarity induces a Galois connection between the powersets of A and X, the stable sets of which form a complete lattice, and in fact, any complete lattice is isomorphic to one arising from some polarity. This representation theory for general lattices, due to Birkhoff, provides the polarity-to-lattice direction of the duality developed in [25], and is also at the heart of FCA. Indeed, the Galois-stable sets arising from formal contexts can be interpreted as *formal concepts*. One of the most felicitous insights of FCA is that concepts are endowed by construction with a double interpretation: an *extensional* one, specified by the objects which are instances of the formal concept, and an *intensional* one, specified by the properties shared by any object belonging to the concept.

The second key step is the arguably natural idea that categories and classification systems, as studied in social sciences and management science, are a very concrete setting of application of the insights of FCA.

Indeed, in social science and management science, *categories* are understood as types of collective identities for broad classes e.g. of market products, organizations or individuals. Categorization theory recognizes categories as a key aspect of any decision-making process, in that they structure the space of options by defining the boundaries of meaningful comparisons between the available alternatives [29,32,42]. Also, categories function as cognitive sieves, filtering out those features which are redundant or less essential to the decision-making, thus contributing to minimize the agents' cognitive efforts. Examples of categories are musical genres, which are widely applied as tools to compress and convey relevant information about a musical product to its potential audience. Structuring information and decision-making along the faultlines of genres is so established a practice in the creative industries that genres have become the main way to structure competition as well as to create consumer group identity.

An aspect of categories which is very much highlighted in the categorization theory literature is that they never occur in isolation; rather, they arise in the context of *categorization systems* (e.g. taxonomies), which are typically organized in hierarchies of super- (i.e. less specified) and sub- (i.e. more specified) categories. This observation agrees with the FCA treatment, according to which concepts arise embedded in their concept lattice.

One of the open challenges in the extant literature is how to reconcile the view on categories which defines them in terms of the objects (e.g. products) belonging to that category with another view which defines categories in terms of the features enjoyed by its members. The intensional and extensional perspectives on concepts brought about by FCA provide an elegant reconciliation of the two views on categories, which gives a second clue that the FCA perspective on categorization theory can be fruitful.

In recent years, a substantial research stream in social and management science explores the dynamic aspects of categorization [29,35]. For instance, *category emergence* investigates how new categories are created, either ex nihilo or through the recombination of existing ones, and how the interaction of relevant groups of agents, such as the media or the reviewers, plays a role in this process. The aspect of social interaction is essential to understand how categories arise and are put to use: although they can be seen to arise from factual pieces of information about the world (e.g., the products available in a given market and their features), a critical component of their nature cannot be reduced to factual information. In other words, categories are *social artifacts*, and reasoning about them requires a peculiar combination of factual truth, individual perception and social interaction.

The main point of interest and the conceptual contribution of the present proposal concerns precisely the formalization of the subjective and social aspects of this emergence. Namely, we observe that the agents' subjective perspective on products and features can be naturally modelled by associating each agent with a binary relation $R \subseteq A \times X$ on the database $(A, X, \perp)$, which represents the subjective filters superimposed by each agent on the information of the database. That is, for every product $a \in A$ and every feature $x \in X$, we read aRx as 'product a has feature x according to the agent'. By general order-theoretic facts, these relations[2] induce normal modal operators on the categorization system associated with the database. These modal operators enrich the basic propositional logic of the categorization systems. In this enriched logical language, it is easy to distinguish between 'objective' information (stored in the database), encoded in the formulas of the modal-free fragment of the language, and the agents' subjective interpretation of the 'objective' information, encoded in formulas in which modal operators occur. This language is expressive enough to encode agents' beliefs/perceptions regarding other agents' beliefs/perceptions, and so on. Again, this

[2] Actually, those which are RS-compatible, cf. Definition 4.

makes it possible to define *fixed points* of these regressions, similarly to the way in which common knowledge is defined in classical epistemic logic [20]. Intuitively, these fixed points represent the stabilization of a process of social interaction; for instance, the consensus reached by a group of agents regarding a given category. Clearly, market dynamics are bound to create further destabilization, necessitating a new round of interaction in order to establish a new equilibrium. Further directions will be to generalize the framework of dynamic epistemic logic [2] to the setting outlined in the present paper, and further develop the theory of lattice-based mu-calculus initiated in [6].

Structure of the Paper. In Sect. 2, we collect the necessary definitions and basic facts about RS-semantics. In Sect. 3, we discuss how the mathematical environment introduced in the previous section can be understood using categories and categorization systems as the framework of reference. In particular, we show how normal modal operators on lattices can support an epistemic interpretation. In Sect. 4, we build on the epistemic interpretation of the modal operators, and introduce a common knowledge-type construction to account for a view of categories as the outcome of social interaction. In Sect. 5 we collect our conclusions. More technical background is relegated to Appendix A, while the proofs of some technical lemmas can be found in Appendix B.

2 Preliminaries

In this section we recall some preliminaries on perfect lattices, RS-polarities, generalized Kripke frames and formal concept analysis. We will assume familiarity with the basics of lattice theory (see e.g. [16]).

2.1 Perfect Lattices

A bounded lattice $\mathbb{L} = (L, \wedge, \vee, 0, 1)$ is *complete* if all subsets $S \subseteq L$ have both a supremum $\bigvee S$ and an infimum $\bigwedge S$. An element a in $\mathbb{L}$ is *completely join-irreducible* if, for any $S \subseteq \mathbb{L}$, $a = \bigvee S$ implies $a \in S$. *Complete meet-irreducibility* is defined order-dually. The sets of completely join- and meet-irreducible elements of $\mathbb{L}$ are denoted by $J^\infty(\mathbb{L})$ and $M^\infty(\mathbb{L})$, respectively.

A complete lattice $\mathbb{L}$ is called *perfect* if it is join-generated by its completely join-irreducibles, and meet-generated by its completely meet-irreducibles. That is, $\mathbb{L}$ is perfect if for any $u \in \mathbb{L}$, we have $\bigvee\{j \in J^\infty(\mathbb{L}) \mid j \leq u\} = u = \bigwedge\{m \in M^\infty(\mathbb{L}) \mid u \leq m\}$.

2.2 Polarities and Birkhoff's Representation Theorem

Definition 1. *A* polarity *is a triple $\mathbb{P} = (A, X, \perp)$ where A and X are sets, and $\perp \subseteq A \times X$ is a relation. For every polarity $\mathbb{P}$, we define the functions $(\cdot)^\uparrow$ (upper) and $(\cdot)^\downarrow$ (lower)*[3] *between the posets $(\mathcal{P}(A), \subseteq)$ and $(\mathcal{P}(X), \subseteq)$, as follows:*

$$\begin{aligned} for\ U \in \mathcal{P}(A), let U^\uparrow &:= \{x \in X \mid \forall a(a \in U \to a \perp x)\},\\ for\ V \in \mathcal{P}(X), let V^\downarrow &:= \{a \in A \mid \forall x(x \in V \to a \perp x)\}. \end{aligned}$$

[3] In what follows, we abuse notation and write $a^\uparrow$ for $\{a\}^\uparrow$ and $x^\downarrow$ for $\{x\}^\downarrow$ for every $a \in A$ and $x \in X$.

The maps $(\cdot)^{\uparrow}$ and $(\cdot)^{\downarrow}$ form a **Galois connection** between $(\mathcal{P}(A), \subseteq)$ and $(\mathcal{P}(X), \subseteq)$, i.e. $V \subseteq U^{\uparrow}$ iff $U \subseteq V^{\downarrow}$ for all $U \in \mathcal{P}(A)$ and $V \in \mathcal{P}(X)$. Well-known consequences of this fact are: the composition maps $(\cdot)^{\uparrow\downarrow} := (\cdot)^{\downarrow} \circ (\cdot)^{\uparrow}$ and $(\cdot)^{\downarrow\uparrow} := (\cdot)^{\uparrow} \circ (\cdot)^{\downarrow}$ are closure operators on $(\mathcal{P}(A), \subseteq)$ and $(\mathcal{P}(X), \subseteq)$, respectively;[4] The set of all *Galois-stable* subsets of A (i.e. those $U \in \mathcal{P}(A)$ such that $U^{\uparrow\downarrow} = U$) forms a complete sub-semilattice of $(\mathcal{P}(A), \bigcap)$, which we denote by $\mathbb{P}^+$;[5] since it is complete, the semilattice $\mathbb{P}^+$ is in fact a lattice, where meet is set-theoretic intersection and join is the closure of the set-theoretic union. If fact, Birkhoff showed that every complete lattice is isomorphic to $\mathbb{P}^+$ for some polarity $\mathbb{P}$. This lattice can be identified with the lattice of *concepts* arising from $\mathbb{P}$ (this terminology comes from Formal Concept Analysis), i.e. tuples (C, D) s.t. $C \subseteq A$, $D \subseteq X$ and $D^{\downarrow} = C$ and $C^{\uparrow} = D$.[6] Concepts (resp. Galois stable subsets of X and of A) can be characterized as (members of) tuples $(U^{\uparrow\downarrow}, U^{\uparrow})$ and $(V^{\downarrow}, V^{\downarrow\uparrow})$ for any $U \subseteq A$ and $V \subseteq X$.

Let us conclude the present subsection by introducing some notation and showing some useful facts. Polarities $(A, X, \perp)$ induce 'specialization pre-orders' on A and X defined as follows: $x \leq y$ iff $\forall a(a \perp x \to a \perp y)$ for all $x, y \in X$, and $a \leq b$ iff $\forall x(b \perp x \to a \perp x)$ for all $a, b \in A$. Clearly, $\leq \circ \perp \circ \leq \subseteq \perp$. For every $b \in A$ and $z \in X$, let $z{\uparrow} := \{x \mid z \leq x\}$, and $b{\downarrow} := \{a \mid a \leq b\}$. The proofs of the following lemma and corollary can be found in Appendix B.

Lemma 1. *$z{\uparrow}$ and $b{\downarrow}$ are Galois-stable for all $b \in A$ and $z \in X$.*

Corollary 1. *$z^{\downarrow\uparrow} = z{\uparrow}$ and $b^{\uparrow\downarrow} = b{\downarrow}$ for all $b \in A$ and $z \in X$.*

Summing up, the concepts generated by each $a \in A$ and $x \in X$ are $(a{\downarrow}, a^{\uparrow})$ and $(x^{\downarrow}, x{\uparrow})$ respectively.

2.3 RS-polarities and Dual Correspondence for Perfect Lattices

As mentioned early on, every complete lattice is isomorphic to $\mathbb{P}^+$ for some polarity $\mathbb{P}$. When specializing to distributive lattices and Boolean algebras, the well-known dualities obtain between set-theoretic structures and *perfect* algebras. In particular, perfect distributive lattices are dual to posets, and perfect (i.e. complete and atomic) Boolean algebras are dual to sets. The question then arises: which polarities are dual to perfect lattices? The answer was given by Gehrke in [25], where the so-called reduced and separated polarities, or *RS-polarities*, have been characterized as duals to perfect lattices, by rephrasing in a model-theoretic way the duality for perfect lattices given in [18]. In what follows, we will recall what it means for a polarity to be reduced and separated, and briefly explain how these two properties guarantee the perfection of the dual lattice. First, the route from perfect lattices to polarities is given by the following definition:

[4] Recall that a closure operator on a poset $(S, \leq)$ is a map $f : S \to S$ which is extensive $(\forall a \in S[a \leq f(a)])$, monotone $(\forall a, b \in S[a \leq b \Rightarrow f(a) \leq f(b)])$ and idempotent $(\forall a \in S[f(a) = f(f(a))])$.

[5] Likewise, The set of all *Galois-stable* subsets of X (i.e. those $V \in \mathcal{P}(X)$ such that $V^{\downarrow\uparrow} = V$) forms a complete sub-semilattice of $(\mathcal{P}(X), \bigcap)$.

[6] Sometimes C and D are referred to as the *extension* and the *intension* of a concept, respectively.

Definition 2. *For every perfect lattice* $\mathbb{L}$*, the* polarity associated with $\mathbb{L}$ *is the triple* $\mathbb{L}_+ := (J^\infty(\mathbb{L}), M^\infty(\mathbb{L}), \perp_+)$ *where* $\perp_+$ *is the lattice order* $\leq_\mathbb{L}$ *restricted to* $J^\infty(\mathbb{L}) \times M^\infty(\mathbb{L})$.

Definition 3 *(cf.* [25, Definitions 2.3 and 2.12]*). A polarity* $\mathbb{P} = (A, X, \perp)$ *is:*

1. separating *if the following conditions are satisfied:*
   (s1) for all $a, b \in A$*, if* $a \neq b$ *then* $a^\uparrow \neq b^\uparrow$*, and*
   (s2) for all $x, y \in Y$*, if* $x \neq y$ *then* $x^\downarrow \neq y^\downarrow$.
2. reduced *if the following conditions are satisfied:*
   (r1) for every $a \in A$*, some* $x \in X$ *exists s.t.* a *is* $\leq$*-minimal in* $\{b \in A \mid b \not\perp x\}$.
   (r2) for every $x \in X$*, some* $a \in A$ *exists s.t.* x *is* $\leq$*-maximal in* $\{y \in X \mid x \not\perp a\}$.
3. *an* RS-polarity[7] *if it is separating and reduced.*

If $\mathbb{P}$ is separating, then, denoting $S := \{b \mid b \in A \text{ and } b < a\} = a{\downarrow}\backslash\{a\}$ for each $a \in A$, notice that $a{\downarrow}$ is completely join-irreducible in $\mathbb{P}^+$ iff $\bigvee_{b \in S} b{\downarrow} \subsetneq a{\downarrow}$ iff $a^\uparrow \subsetneq \bigcap_{b \in S} b^\uparrow$, i.e. some $x \in X$ exists such that $b \perp x$ for all $b \in S$ and $a \not\perp x$, which is condition (r1). Similarly, (r2) dually characterizes the condition that, for every $x \in X$, the subset $x{\uparrow}$ is completely meet-irreducible in $\mathbb{P}^+$, represented as a sub meet-semilattice of $\mathcal{P}(X)$.

Proposition 1 *(cf.* [25, Remark 2.13] and [18, Proposition 4.7, Corollary 4.9]*). For every perfect lattice* $\mathbb{L}$ *and RS-polarity* $\mathbb{P}$*,*

1. $\mathbb{L}_+$ *is an RS-polarity and* $(\mathbb{L}_+)^+ \cong \mathbb{L}$.
2. $\mathbb{P}^+$ *is a perfect lattice and* $(\mathbb{P}^+)_+ \cong \mathbb{P}$.

2.4 RS-frames and Models

In the present section, we report on the definition of a relational semantics, based on RS-polarities, for an expansion $\mathcal{L}$ of the basic lattice language with a unary normal box-type connective. We also give semantics for a further expansion of $\mathcal{L}$ with a unary normal diamond-type connective $\blacklozenge$, and with two special sorts of variables $\mathbf{i}, \mathbf{j}$ called *nominals*, and $\mathbf{m}, \mathbf{n}$ called *co-nominals*. This semantics is the outcome of a dual characterization which is discussed in detail and in full generality in [9, Sect. 2], and is reported on in the appendix for the part directly relevant to this paper. The most peculiar feature of this semantics is that formulas are *satisfied* at $a \in A$ and *co-satisfied* (*refuted*) at $x \in X$.

Definition 4. *An RS-frame for* $\mathcal{L}$ *is a structure* $\mathbb{F} = (\mathbb{P}, R)$ *where* $\mathbb{P} = (A, X, \perp)$ *is an RS-polarity, and* $R \subseteq A \times X$ *such that the images and pre-images of singletons under* R *are Galois-closed, i.e. for every* $x \in X$ *and* $a \in A$*,*

$$R^{-1}[x]^{\uparrow\downarrow} \subseteq R^{-1}[x] \text{ and } R[a]^{\downarrow\uparrow} \subseteq R[a].$$

Relations R *which satisfy this condition are called* RS-compatible.

[7] In [25], RS-polarities are referred to as RS-frames. Here we reserve the term RS-frame for RS-polarities endowed with extra relations used to interpret the operations of the lattice expansion.

The additional conditions on R are compatibility conditions guaranteeing that the following assignments respectively define the operations $\Box$ and $\blacklozenge$ associated with R on the lattice $\mathbb{P}^+$: for every $U \in \mathbb{P}^+$,

$$\Box U := \bigcap\{R^{-1}[x] \mid U \subseteq x^{\downarrow}\} \text{ and } \blacklozenge U := \bigvee\{R[a] \mid a^{\uparrow\downarrow} \subseteq U\}.$$

Definition 5. *For every RS-frame $\mathbb{F} = (\mathbb{P}, R)$, its complex algebra is the lattice expansion $\mathbb{F}^+ := (\mathbb{P}^+, \Box)$ where $\Box$ is defined as above.*

Lemma 2. $\leq \circ R \circ \leq \subseteq R$ *for every RS-frame $\mathbb{F} = (\mathbb{P}, R)$.*

An *RS-model* for $\mathcal{L}$ on $\mathbb{F}$ is a structure $\mathbb{M} = (\mathbb{F}, v)$ such that $\mathbb{F}$ is an RS-frame for $\mathcal{L}$ and v is a variable assignment mapping each $p \in \mathsf{PROP}$ to a pair $(V_1(p), V_2(p))$ of Galois-stable sets in $\mathcal{P}(A)$ and $\mathcal{P}(X)$ respectively. In a model for the expanded language with $\blacklozenge$, nominals and conominals, variable assignments also map nominals $\mathbf{j}$ to $(j^{\uparrow\downarrow}, j^{\uparrow})$ for some j in A and co-nominals $\mathbf{m}$ to $(m^{\downarrow}, m^{\downarrow\uparrow})$ for some m in X.

The following table reports the recursive definition of the satisfaction and co-satisfaction relations on $\mathbb{M}$:

$$
\begin{array}{llll}
\mathbb{M}, a \Vdash 0 & \text{never} & \mathbb{M}, x \succ 0 & \text{always} \\
\mathbb{M}, a \Vdash 1 & \text{always} & \mathbb{M}, x \succ 1 & \text{never} \\
\mathbb{M}, a \Vdash p & \text{iff } a \in V_1(p) & \mathbb{M}, x \succ p & \text{iff } x \in V_2(p) \\
\mathbb{M}, a \Vdash \mathbf{i} & \text{iff } a \in V_1(\mathbf{i}) & \mathbb{M}, x \succ \mathbf{i} & \text{iff } x \in V_2(\mathbf{i}) \\
\mathbb{M}, a \Vdash \mathbf{m} & \text{iff } a \in V_1(\mathbf{m}) & \mathbb{M}, x \succ \mathbf{m} & \text{iff } x \in V_2(\mathbf{m})
\end{array}
$$

$$
\begin{array}{ll}
\mathbb{M}, a \Vdash \phi \wedge \psi & \text{iff } \mathbb{M}, a \Vdash \phi \text{ and } \mathbb{M}, a \Vdash \psi \\
\mathbb{M}, x \succ \phi \wedge \psi & \text{iff for all } a \in A, \text{ if } \mathbb{M}, a \Vdash \phi \wedge \psi, \text{ then } a \perp x \\
\mathbb{M}, a \Vdash \phi \vee \psi & \text{iff for all } x \in X, \text{ if } \mathbb{M}, x \succ \phi \vee \psi, \text{ then } a \perp x \\
\mathbb{M}, x \succ \phi \vee \psi & \text{iff } \mathbb{M}, x \succ \phi \text{ and } \mathbb{M}, x \succ \psi \\
\mathbb{M}, a \Vdash \Box\phi & \text{iff for all } x \in X, \text{ if } \mathbb{M}, x \succ \phi, \text{ then } aRx \\
\mathbb{M}, x \succ \Box\phi & \text{iff for all } a \in A, \text{ if } \mathbb{M}, a \Vdash \Box\phi, \text{ then } a \perp x \\
\mathbb{M}, a \Vdash \blacklozenge\phi & \text{iff for all } x \in X, \text{ if } \mathbb{M}, x \succ \blacklozenge\phi, \text{ then } a \perp x \\
\mathbb{M}, x \succ \blacklozenge\phi & \text{iff for all } a \in A, \text{ if } \mathbb{M}, a \Vdash \phi, \text{ then } aRx.
\end{array}
$$

The following lemma is proven easily by simultaneous induction on ϕ and ψ using the truth definitions above. The base cases for 0 and 1 use conditions (r1) and (r2) and those for proposition letters, nominals and co-nominals follow from the way valuations are defined.

Lemma 3. *For all formulas ϕ and ψ it holds that*

1. $\mathbb{M}, a \Vdash \phi$ *iff for all $x \in X$, if $\mathbb{M}, x \succ \phi$ then $a \perp x$, and*
2. $\mathbb{M}, x \succ \psi$ *iff for all $a \in A$, if $\mathbb{M}, a \Vdash \psi$ then $a \perp x$.*

An inequality $\phi \leq \psi$ is **true** in $\mathbb{M}$, denoted $\mathbb{M} \Vdash \phi \leq \psi$, if for all $a \in A$ and all $x \in X$, if $\mathbb{M}, a \Vdash \phi$ and $\mathbb{M}, x \succ \psi$ then $a \perp x$.

Remark 1. It follows from Lemma 3 that $\mathbb{M} \Vdash \phi \leq \psi$ iff for all $a \in A$, if $\mathbb{M}, a \Vdash \phi$ then $\mathbb{M}, a \Vdash \psi$. It also follows that $\mathbb{M} \Vdash \phi \leq \psi$ iff for all $x \in X$, if $\mathbb{M}, x \succ \psi$ then $\mathbb{M}, x \succ \phi$. We will find these equivalent characterizations of truth in a model useful when treating examples.

2.5 Standard Translation on RS-frames

As in the Boolean case, each RS-model $\mathbb{M}$ for $\mathcal{L}$ can be seen as a first-order structure, albeit two-sorted. Accordingly, we define correspondence languages as follows.

Let L_1 be the two-sorted first-order language with equality built over the denumerable and disjoint sets of individual variables A and X, with binary relation symbol $\perp$, R, and two unary predicate symbols P_1, P_2 for each $p \in \mathsf{PROP}$.[8]

We will further assume that L_1 contains denumerably many individual variables $i, j, \ldots$ corresponding to the nominals $\mathbf{i}, \mathbf{j}, \ldots \in \mathsf{NOM}$ and $n, m, \ldots$ corresponding to the co-nominals $\mathbf{n}, \mathbf{m} \in \mathsf{CO\text{-}NOM}$. Let L_0 be the sub-language which does not contain the unary predicate symbols corresponding to the propositional variables. Let us now define the **standard translation** of $\mathcal{L}^+$ into L_1 recursively:[9]

$$
\begin{array}{ll}
\mathrm{ST}_a(0) := a \neq a & \mathrm{ST}_x(0) := x = x \\
\mathrm{ST}_a(1) := a = a & \mathrm{ST}_x(1) := x \neq x \\
\mathrm{ST}_a(p) := P_1(a) & \mathrm{ST}_x(p) := P_2(x) \\
\mathrm{ST}_a(\mathbf{j}) := a \leq j & \mathrm{ST}_x(\mathbf{j}) := j \perp x \\
\mathrm{ST}_a(\mathbf{m}) := a \perp m & \mathrm{ST}_x(\mathbf{m}) := m \leq x \\
\mathrm{ST}_a(\phi \vee \psi) := \forall x[\mathrm{ST}_x(\phi \vee \psi) \to a \perp x] & \mathrm{ST}_x(\phi \vee \psi) := \mathrm{ST}_x(\phi) \wedge \mathrm{ST}_x(\psi) \\
\mathrm{ST}_a(\phi \wedge \psi) := \mathrm{ST}_a(\phi) \wedge \mathrm{ST}_a(\psi) & \mathrm{ST}_x(\phi \wedge \psi) := \forall a[\mathrm{ST}_a(\phi \wedge \psi) \to a \perp x] \\
\mathrm{ST}_a(\Box\phi) := \forall x[\mathrm{ST}_x(\phi) \to aRx] & \mathrm{ST}_x(\Box\phi) := \forall a[\mathrm{ST}_a(\Box\phi) \to a \perp x] \\
\mathrm{ST}_a(\blacklozenge\phi) := \forall x[\mathrm{ST}_x(\blacklozenge\phi) \to a \perp x] & \mathrm{ST}_x(\blacklozenge\phi) := \forall a[\mathrm{ST}_a(\phi) \to aRx]
\end{array}
$$

The following is a variant of [9, Lemma 2.5].

Lemma 4. *For any $\mathcal{L}$-model $\mathbb{M}$ and any $\mathcal{L}^+$-inequality $\phi \leq \psi$, it holds that $\mathbb{M} \Vdash \phi \leq \psi$ iff $\mathbb{M} \models \forall a \forall x[\mathrm{ST}_a(\phi) \wedge \mathrm{ST}_x(\psi) \to a \perp x]$ iff $\mathbb{M} \models \forall a[\mathrm{ST}_a(\phi) \to \mathrm{ST}_a(\psi)]$ iff $\mathbb{M} \models \forall x[\mathrm{ST}_x(\psi) \to \mathrm{ST}_x(\phi)]$.*

2.6 Examples

So far we have seen that the environment of RS-frames provides a mathematically motivated generalization of the correspondence theory which was key to the success of classical normal modal logic as a formal framework in multiple settings. The focus of this paper is to try and understand whether and how this generalized environment can retain some of the intuition which made Kripke semantics and modal logic so appealing. Let us start with the inequality $\Box 0 \leq 0$, which corresponds on Kripke frames to the condition that every state has a successor.

$$
\begin{aligned}
\Box 0 \leq 0 \text{ iff } & \forall a[\mathrm{ST}_a(\Box 0) \to \forall x(\mathrm{ST}_x(0) \to a \perp x)] \\
\text{iff } & \forall a[\forall y(y = y \to aRy) \to \forall x(x = x \to a \perp x)] \\
\text{iff } & \forall a[\forall y(aRy) \to \forall x(a \perp x)] \\
\text{iff } & \forall a \exists y(\neg(aRy))
\end{aligned}
$$

---

[8] The intended interpretation links P_1 and P_2 in the way suggested by the definition of $\mathcal{L}$-valuations. Indeed, every $p \in \mathsf{PROP}$ is mapped to a pair $(V_1(p), V_2(p))$ of Galois-stable sets as indicated in Subsect. 2.4. Accordingly, the interpretation of pairs (P_1, P_2) of predicate symbols is restricted to such pairs of Galois-stable sets, and hence the interpretation of universal second-order quantification is also restricted to range over such sets.

[9] Recall that $a \leq j$ abbreviates $\forall x(j \perp x \to a \perp x)$ and $m \leq x$ abbreviates $\forall a(a \perp m \to a \perp x)$.

To justify the last equivalence, notice that by definition, in RS-polarities no object a verifies $\forall x(a \perp x)$. Hence the condition in the penultimate line is true precisely when the premise of the implication is false. This condition says that every state is not R-related to some co-state; the condition on Kripke frames is recognizable modulo suitable insertion of negations. Next, let us consider the inequality $\Box p \leq p$, which corresponds on Kripke frames to the condition that R is reflexive.

$$\begin{aligned}\forall p(\Box p \leq p) &\text{ iff } \forall \mathbf{m}(\Box\mathbf{m} \leq \mathbf{m})\\ &\text{ iff } \forall a \forall \mathbf{m}[\mathrm{ST}_a(\Box\mathbf{m}) \to \mathrm{ST}_a(\mathbf{m})]\\ &\text{ iff } \forall a \forall m(aRm \to a \perp m),\end{aligned}$$

since by definition, $\mathrm{ST}_a(\mathbf{m}) = a \perp m$, and $\mathrm{ST}_a(\Box\mathbf{m}) = \forall y(m \leq y \to aRy)$ can be rewritten as $m\!\uparrow\ \subseteq R[a]$, which is equivalent to aRm, since $R \circ \leq \subseteq R$ (cf. Lemma 2). To recognize the connection with the usual reflexivity condition, observe that $\forall a \forall m(aRm \to a \perp m)$ is equivalent to $R \subseteq \perp$, and the reflexivity of a relation $R \subseteq A \times A$ can be written as $\mathrm{Id} \subseteq R$, which is equivalent to $R^c \subseteq \mathrm{Id}^c$.

Clearly, $\Box p \leq p$ implies $\Box\Box p \leq \Box p$. Let us consider the converse inequality, which in the classical setting corresponds to transitivity:

$$\begin{aligned}\forall p(\Box p \leq \Box\Box p) &\text{ iff } \forall \mathbf{m}(\Box\mathbf{m} \leq \Box\Box\mathbf{m})\\ &\text{ iff } \forall a \forall \mathbf{m}(\mathrm{ST}_a(\Box\mathbf{m}) \to \mathrm{ST}_a(\Box\Box\mathbf{m}))\\ &\text{ iff } \forall a \forall m(aRm \to R^{-1}[m]^{\uparrow} \subseteq R[a]),\end{aligned}$$

where

$$\begin{aligned}\mathrm{ST}_a(\Box\Box\mathbf{m}) &= \forall y[\mathrm{ST}_y(\Box\mathbf{m}) \to aRy]\\ &= \forall y[\forall b(\mathrm{ST}_b(\Box\mathbf{m}) \to b \perp y) \to aRy]\\ &= \forall y[\forall b(bRm \to b \perp y) \to aRy] \qquad (**)\\ &= R^{-1}[m]^{\uparrow} \subseteq R[a].\end{aligned}$$

While, again, with a bit of work it is possible to retrieve the transitivity condition in this new interpretation, already with a relatively simple inequality such as $\Box p \leq \Box\Box p$ this game is not really useful for the purpose of gaining a better intuitive understanding of this semantics, since it requires jumping through too many hoops (the accessibility relation on states is here encoded into a 'non unaccessibility' relation between states and co-states), and quickly becomes awkward and unintuitive. In the next section, we will argue that better results can be achieved by taking it as primitive, rather than as the generalization of some other semantics.

3 Conceptualizing RS-semantics via Categorization Theory

In the present section, we propose a conceptualization of the notions introduced in the previous section based on ideas from categorization theory in management science. The starting point of this conceptualization is the very well known idea, core to Formal Concept Analysis, that polarities $(A, X, \perp)$ are abstract representations of databases, in which A and X are sets of *objects* and *properties* respectively, and $\perp$ encodes information about whether a given object satisfies a given property. More specifically, we propose to think of a given polarity $\mathbb{P} = (A, X, \perp)$ as a database such that A is the set of all *products* in a given market at a certain moment (e.g. all models of cars, or models of togas on sale in the Netherlands in a given year), and X all the relevant observable *features* of these products. The specialization pre-order $a \leq b$ on objects (a has at least all the features that b has) can then be read as 'product a is at least as specified (i.e. rich in features) as product b' and the one on features $x \leq y$ (any product having x has also y) as 'feature y is more generic than feature x'. The RS-conditions on the database can then be understood as follows:

(s1): Any two distinct products can be told apart by some feature;
(s2): For any two distinct features there is a product having one but not the other;
(r1): For any product a, if there are strictly more specified products than a in the market, then they all share some feature x which a does not have;
(r2): For any feature x, if there are strictly more generic features than x, then some product a exists which has all of them but not x.

The separation conditions (s1) and (s2) seem rather intuitive and do not require much explanation; (r1) can be enforced by suitably adding 'artificial' features to the database, and (r2) can be enforced by removing features from the database which are the exact intersection of two or more generic features.[10] Clearly, removing such features can always be done without loss of descriptive power. We can always enforce the separation and reduction conditions, since the finite polarities we consider are a subclass of the so-called doubly founded polarities, for which this is always possible, see [23].

Arguably, the reformulation of the RS-conditions in terms of products and features makes them easier to grasp.

Further, we propose to understand the lattice $\mathbb{P}^+$ as the collection of 'candidate categories'. That is, each element of $\mathbb{P}^+$ is a set of products which is completely identified by the set of features common to its elements. That is, any product with all these features is a member of the 'candidate category'. We refer to these categories as 'candidate' since they are purely implicit in the database, and not necessarily the target of any social construction. In particular, only a restricted subset of candidate categories will support the interpretation of socially meaningful categories (which have *labels* such as western, opera, bossa-nova, SUV, smart phone etc.). Labels of socially meaningful categories can be assigned to 'candidate categories' in the usual way, namely, by means of an assignment v which associates each *atomic category label* $p \in \mathsf{PROP}$ to a category viewed both extensionally as $V_1(p) \subseteq X$ and intensionally as $V_2(p) \subseteq A$.[11] Notice the perfect match between the encoding of the *meaning* of atomic propositions on Kripke models and of atomic category labels on RS-models: the meaning of atomic proposition p is given as the set of states at which p holds true; the meaning of atomic category label p is given as the set of products which are the *members* of p, and the set of features which *describe* p. In what follows, we will refer to the intension of a category (cf. Footnote 6) as its *description*, and we say that a feature *describes* a category if it belongs to its description.

[10] For instance, consider the following features of a soft drink: x: = 'with vitamin A', y: = 'with vitamin C', z: = 'with vitamin A and C'. Clearly, a database with these features would violate (r2). This can be remedied by removing z from the set X of the database.

[11] Recall that for such an assignment, $V_1(p) = V_2(p)^{\downarrow}$ and $V_2(p) = V_1(p)^{\uparrow}$.

Given such an assignment,[12] the database is endowed with a structure of an $\mathcal{L}$-model $\mathbb{M}$, in such a way that, for every formula (category label) $\phi \in \mathcal{L}$, any $a \in A$ and $x \in X$, the symbols $\mathbb{M}, a \Vdash \phi$ and $\mathbb{M}, x \succ \phi$ can be understood as 'object a is a member of category ϕ', and 'feature x describes category ϕ'. One immediately apparent advantage of this conceptualization is that it provides an intuitive way to understand $\succ$ from first principles rather than as the negative counterpart of $\Vdash$.

The other advantage concerns the understanding of the connectives $\wedge$ and $\vee$ in the general lattice environment. The issue is that their standard interpretation as conjunction and disjunction does not seem completely right, since distributivity seems hardwired in the way we understand 'and' and 'or' in natural language. The satisfaction clauses for $\wedge$ and $\vee$ formulas read:

$$\begin{array}{l}
\mathbb{M}, a \Vdash \phi \wedge \psi \text{ iff } \mathbb{M}, a \Vdash \phi \text{ and } \mathbb{M}, a \Vdash \psi \\
\mathbb{M}, x \succ \phi \wedge \psi \text{ iff for all } a \in A, \text{ if } \mathbb{M}, a \Vdash \phi \wedge \psi, \text{ then } a \perp x \\
\mathbb{M}, a \Vdash \phi \vee \psi \text{ iff for all } x \in X, \text{ if } \mathbb{M}, x \succ \phi \vee \psi, \text{ then } a \perp x \\
\mathbb{M}, x \succ \phi \vee \psi \text{ iff } \mathbb{M}, x \succ \phi \text{ and } \mathbb{M}, x \succ \psi
\end{array}$$

These clauses say that the category $\phi \wedge \psi$ is the one whose members are members of both categories ϕ and ψ; hence, these products will satisfy at least both the description of ϕ and of ψ, and hence the description of $\phi \wedge \psi$ contains at least the union of these descriptions. The category $\phi \vee \psi$ is described by the intersection of the descriptions of ϕ and of ψ. Hence, membership in $\phi \vee \psi$ only requires products to satisfy this smaller set of features, and typically includes much more than the union of the members of the two categories. So for instance, $\mathsf{bird} \vee \mathsf{cat}$ would exclude reptiles, insects and fish, but include vertebrate homeothermic species such as the platypus. This interpretation of $\wedge$ and $\vee$ makes it possible to understand intuitively why distributivity fails. Indeed, a member of $(\mathsf{phone} \vee \mathsf{smartphone}) \wedge (\mathsf{kettle} \vee \mathsf{smartphone})$ is guaranteed to have all the features in the description of phone (and in fact, $\mathsf{kettle} \vee \mathsf{smartphone}$ is so general that can be assumed to not add any feature that phone does not have already). However, this might be not enough for it to be a member of $(\mathsf{phone} \wedge \mathsf{kettle}) \vee \mathsf{smartphone}$, given that the category $\mathsf{phone} \wedge \mathsf{kettle}$ has no members (hence its description consists of *all* features), and so the members of $(\mathsf{phone} \wedge \mathsf{kettle}) \vee \mathsf{smartphone}$ must have at least all the features in the description of $\mathsf{smartphone}$.

Now that we have a working understanding of $\Vdash$ and $\succ$, we can recognize the normal box-type operator on $\mathbb{P}^+$ as the perspective of a single agent on categories. Accordingly, $\mathbb{M}, a \Vdash \Box\phi$ and $\mathbb{M}, x \succ \Box\phi$ can be understood as 'object a is a member of category ϕ according to the agent', and 'feature x describes category ϕ according to the agent'. The normality conditions $\Box\top = \top$ and $\Box(\phi \wedge \psi) = \Box\phi \wedge \Box\psi$ can be understood as rationality requirements: that is, the agent correctly recognizes the 'uninformative' category $\top$ as such, and her understanding/perception of the greatest common subcategory of any two categories ϕ and ψ is the greatest common subcategory of the categories she understands as ϕ and ψ.

On the side of the database, the agent is modelled as a relation $R \subseteq A \times X$. Hence, aRx intuitively reads 'object a has feature x according to the agent'. Unsurprisingly, the additional properties of R (cf. Lemma 2) can be also understood as rationality requirements: if aRx then aRy for every $y \geq x$ says that if the agent attributes feature x to

[12] Empirically, there are many ways to generate such an assignment [36].

product a, then the agent will attribute to a also all the features which are 'implied' by x. Likewise, if aRx then bRx for every $b \leq a$ says that if the agent attributes feature x to product a, then the agent will attribute x also to all the products which are 'more specified' than a.

Like in the classical case, two modal operators, $\Box$ and $\blacklozenge$, are associated with the same relation R. However, these operations are not dual to each other, in the sense of e.g. $\Diamond := \neg\Box\neg$, but are rather adjoints to each other, that is, for all $u, v \in \mathbb{P}^+$,

$$\blacklozenge u \leq v \text{ iff } u \leq \Box v.$$

In fact, rather than encoding the dual perspective on the subjectivity of the agent that $\Box$ encodes, the operation $\blacklozenge$ encodes the same perspective that $\Box$ encodes, only geared towards objects while $\Box$ is geared towards features. Indeed, for every object j and every feature m, denoting by $\mathbf{j}$ and $\mathbf{m}$ the categories respectively generated by j and m,

$$\blacklozenge\mathbf{j} \leq \mathbf{m} \text{ iff } jRm \text{ iff } \mathbf{j} \leq \Box\mathbf{m}.$$

Thus, the information jRm ('the agent attributes feature m to object j') is encoded on the side of the categories both by saying that m describes the category $\blacklozenge\mathbf{j}$ (the one the agent understands as the category generated by j), and by saying that j is a member of the category $\Box\mathbf{m}$ (the one the agent understands as the category generated by m). As to the defining clauses of the recursive definition of $\Vdash$ and $\succ$, by definition, $\mathbb{M}, a \Vdash \Box\phi$ is the case iff for all features x, if $\mathbb{M}, x \succ \phi$, then aRx. That is, product a is recognized by the agent as member of category ϕ iff the agent attributes to a all the features that belong to the description of ϕ.

Moreover, by definition, $\mathbb{M}, x \succ \Box\phi$ iff for all $a \in A$, if $\mathbb{M}, a \Vdash \Box\phi$, then $a \perp x$. That is, feature x pertains to the description of category ϕ according to the agent iff x is verified by each object a that the agent recognizes as a member of ϕ.

Two modal axioms commonly considered in epistemic logic are 'reflexivity' $\Box p \leq p$ and 'transitivity' ($\Box p \leq \Box\Box p$). The axiom $\Box p \leq p$ is interpreted epistemically as the *factivity* of knowledge ('if the agent knows that p then p is true'). The first-order correspondent of the factivity axiom on RS-frames is $\forall a \forall x(aRx \rightarrow a \perp x)$, which indeed expresses a form of factivity, in that it requires that whenever the agent attributes any feature x to any product a, then it is indeed the case that x is a feature of a. The axiom $\Box p \leq \Box\Box p$ is interpreted epistemically as the *positive introspection* of knowledge ('if the agent knows that p, then the agent knows that she knows that p'). The first-order correspondent of the positive introspection axiom on RS-frames is $\forall a \forall m(aRm \rightarrow R^{-1}[m]^{\uparrow} \subseteq R[a])$, expressing the condition that if an agent attributes feature m to product a, then she will attribute to a all the features which are shared by the products to which she attributes m. To understand the link between this condition and positive introspection, consider the category $\Box\mathbf{m}$, i.e. the category which the agent understands as the one generated by a given feature m.[13] This category can be identified with the tuple $(R^{-1}[m], R^{-1}[m]^{\uparrow})$. That is, the members of $\Box\mathbf{m}$ are the products to which the agent attributes m (recall that $R^{-1}[m]$ is a Galois-stable set by Definition 4) and the description of $\Box\mathbf{m}$ is the set of the features which the products in $R^{-1}[m]$ have

[13] In fact, the same argument would hold more in general for any category $\Box\phi$.

in common. By definition, $b \perp z$ for every $b \in R^{-1}[m]$ and $z \in R^{-1}[m]^{\uparrow}$. The first-order correspondent of $\Box p \leq \Box\Box p$ requires that bRz for such b and z. So, while factivity corresponds to $R \subseteq \perp$, positive introspection gives the reverse inclusion restricted to products and features pertaining to 'boxed categories'. That is, the agent must be aware of the features of the products of the categories that she knows.

4 Categories as Social Constructs

In the present section, we introduce a formal account of the emergence of categories as the outcome of a process of social interaction. We consider for the sake of simplicity a setting of two agents. Accordingly, we consider the bi-modal logic $\mathcal{L}$ which is the axiomatic extension of the basic normal LE-logic for two unary normal box-type modal operators, 1 and 2, with the axioms $ip \leq p$ and $ip \leq iip$ for $1 \leq i \leq 2$. Models for this logic are structures $(\mathbb{P}, R_1, R_2, v)$ such that $\mathbb{P} = (X, A, \perp)$ is an RS-polarity, $R_i \subseteq A \times X$ for $1 \leq i \leq 2$, such that the following conditions hold:

1. $\forall x(R_i^{-1}[x]^{\uparrow\downarrow} \subseteq R_i^{-1}[x])$;
2. $\forall a(R_i[a]^{\downarrow\uparrow} \subseteq R_i[a])$;
3. $R_i \subseteq\perp$;
4. $\forall a \forall x(aR_i x \rightarrow R_i^{-1}[x]^{\uparrow} \subseteq R_i[a])$,

and v is an assignment which associates each $p \in \mathsf{PROP}$ to an element of $\mathbb{P}^+$ viewed both extensionally as $V_1(p) \subseteq A$ and intensionally as $V_2(p) \subseteq X$ in such a way that $V_1(p) = V_2(p)^{\downarrow}$ and $V_2(p) = V_1(p)^{\uparrow}$.

In this setting, a common knowledge-type construction can be performed which yields an expansion, denoted $\mathcal{L}_C$, of the bi-modal LE-logic above with a normal box-type operator C, the interpretation of which on $\mathbb{P}^+$, given the additional axioms, is given as follows: for any $u \in \mathbb{P}^+$,

$$C(u) := \bigwedge\nolimits_{s \in S} su,$$

where S is the set of all compound modalities of the forms $(ij)^n$ and $i(ji)^n$, for $1 \leq i \neq j \leq 2$ and for some $n \in \mathbb{N}$.

Lemma 5. *$C(u) \leq u$ and $C(u) \leq C(C(u))$ for any $u \in \mathbb{P}^+$.*

Let $R_C, R_s \subseteq A \times X$ for any $s \in S$ be defined as follows: $aR_s x$ iff $a \leq sx$ and $aR_C x$ iff $a \leq C(x)$. Clearly, $R_C = \bigcap_{s \in S} R_s$. In the standard setting of epistemic logic, the accessibility relations associated with agents do not directly encode the agents' knowledge but rather their uncertainty. Hence, on the relational side, the relation associated with the common knowledge operator is defined as the reflexive transitive closure of the union of the relations associated with individual agents, which is typically much bigger than those associated with individual agents. In the present setting, relations associated with agents directly encode what agents positively *know* rather than their uncertainty. Consequently, the common knowledge relation R_C is the intersection of the relations R_s encoding the finite iterations, which is typically much smaller.

As both C and every $s \in S$ are compositions of normal box-operators, they are themselves normal box-operators. Hence the relations R_C and R_s they give rise to are RS-compatible (cf. Definition 4). Thus, the correspondence reductions discussed in Sect. 2.6 apply to C and R_C, yielding:

Lemma 6. *The relation R_C defined above verifies the following conditions:*

1. $R_C \subseteq \perp$;
2. $\forall a \forall x(aR_C x \rightarrow R_C^{-1}[x]^{\uparrow} \subseteq R_C[a])$.

For any given category label ϕ, the category $C(\phi) = \bigwedge\{C(\mathbf{m}) \mid \phi^{\mathbb{P}^+} \leq \mathbf{m}\}$. For this reason, in what follows we restrict our attention to categories $C(\mathbf{m})$ for some feature $m \in X$. The members of $C(\mathbf{m})$ are the products in the set $R_C^{-1}[m] = (\bigcap_{s \in S} R_s)^{-1}[m]$, and the description of $C(\mathbf{m})$ is $R_C^{-1}[m]^{\uparrow} = ((\bigcap_{s \in S} R_s)^{-1}[m])^{\uparrow}$. These can be understood as the socially constructed categories, the membership and description of which are socially agreed upon. Clearly, there are many less of them than candidate categories, which agrees with our intuition.

5 Conclusion and Further Research

In this paper we have proposed an interpretation of RS-semantics in terms of agents' reasoning about objects, their properties and the categories induced by the accompanying relation. We have argued that this semantics is particularly well adapted to this interpretation and, conversely, that through this interpretation one could gain an intuitive understanding of the semantics.

Our proposal has a distinctly epistemic character, but one which differs from standard epistemic logic in at least two respects: firstly, the relations used to interpret the epistemic operators are intended to capture positive knowledge, rather than uncertainty; secondly, these relations relate objects to features rather than possible worlds to one another. We considered two classical principles of epistemic logic, namely factivity and positive introspection. By applying the correspondence theory of [9] we computed the relational properties corresponding to these principles, i.e. necessary and sufficient conditions on an agent's incidence relation between objects and properties for her knowledge of categories to verify these epistemic principles. Various questions for further investigation remain open here: what is the meaning of other classical epistemic principles, like e.g. negative introspection, in this setting? Are there other principles that should be included in a minimal logic of categorization? Of course, all of this depends on the reasoning abilities and level of access to reality we wish to attribute to agents. Moreover, most standard logical questions remain open: axiomatizations, proof systems, decidability, complexity, etc.

This paper is a first assay in using RS-semantics for reasoning about categorization and, as such, remains quite general in its assumptions. To be of more immediate practical relevance, the considerations here should be specialized to particular fields of enquiry where categorization plays or could play a prominent role. Below we briefly consider three such fields.

Natural Language Semantics. We have seen that the assignments of RS-models support a notion of meaning that is different from the one in classical modal logic, but is recognizably what the meaning of category labels should be: namely, a semantic category specified as the set of its members and the set of features describing it. In natural language semantics, linguistic utterances are assigned a meaning in the same spirit, which generalizes the truth-based semantics of sentences. More generally, categories or concepts are fundamental to the construction of meaning in natural language, since each noun is naturally associated with a category. Exploring systematic connections between categories and natural language semantics is a promising direction for further research.

Knowledge Representation and Formal Ontologies. Categories are central to any form of knowledge representation. Description logics [1] are one of the dominant paradigms for logical reasoning in this context. Our formalism represents a different and possibly complementary perspective on the formal ontologies, classification systems, and taxonomies studied there. In particular, the non-distributive nature of category formation and the two-level separation between objects and features are foreign to the description logics paradigm. It is natural to ask to what degree the various expressive features of description logic (like uniqueness quantification, qualified cardinality restrictions etc.) could be accommodated in our framework, and future extensions will study this question.

Categorization Theory in Management Science. As already indicated, this was one of our main sources of inspiration for the proposals of the present paper. Our formalism is a first step in the direction of a formal logical account of the real world phenomena studied by categorization theorists. There are various considerations that make it an attractive framework in which to study categorization and in which to formulate empirically testable hypotheses. We mention two of these reasons: Firstly, it allows one to study the effects of adding or removing objects with new properties and/or properties already associated with other categories, thus allowing for a fine grained analysis of the likely changes in a classification system resulting from innovations of different kinds.

Secondly, our approach gives us all potential categories “automatically”, while only some of them are real, socially agreed upon categories for economic decision-makers. It can therefore serve as a powerful instrument to better study and understand the causes and consequences of the selection of real categories from the broader set of potential ones. To start with, different real world domains could be compared with respect to the ratios of real to potential categories present in them. One reasonable conjecture seems to be that these ratios will depend a lot on competitive dynamics and the matureness of categories, while also having an effect on them. One could the go on to study changes over time in these ratios as well as the differences in ratios—and their changes—among different audiences espousing different classifications.

Extensions and Variations. In closing, we mention two of the many possible extensions of the present framework: category membership does not need to be absolute, as products can simultaneously have different grades of membership in different categories. This calls for quantitative, possibly many-valued versions of our semantics. Also, the categories in a given market do not need to be static, but can evolve and change over time as new products with new features or new combinations of existing features enter the market [41,42]. Dynamic versions of our formalism would be suitable to deal with such continuously evolving categorization systems.

Acknowledgement. The research of the first author has been made possible by the National Research Foundation of South Africa, Grant number 81309. The research of the second to fourth author has been made possible by the NWO Vidi grant 016.138.314, by the NWO Aspasia grant 015.008.054, and by a Delft Technology Fellowship awarded in 2013.

A Relational Semantics via Dual Characterization

The dual correspondence between perfect lattices and RS-polarities serves as a base to generalize the Kripkean semantics of modal logic to logics with possibly non-distributive propositional base. Analogous to the dual correspondence between Kripke frames and complete and atomic Boolean algebras with operators, one would want a dual correspondence between perfect normal lattice expansions and RS-polarities endowed with additional relations. In [9, Sect. 2], a method for computing the definition of the relations dually corresponding to normal modal operators was discussed and illustrated for a certain modal signature consisting of unary and binary modal operators.

In this subsection we will report on this method, for an expansion $\mathcal{L}$ of the basic lattice language with a unary box-modality, canonically interpreted on lattices endowed with a completely meet-preserving operation. Moreover, we will derive, by means of a dual characterization argument, its interpretation on expanded RS-polarities.

We take the connection between the satisfaction relation $\Vdash$ in Kripke frames and the interpretation of modal formulas in BAOs as our guideline: let $\mathbb{F} = (W, R)$ be a Kripke frame. From the satisfaction relation $\Vdash \subseteq W \times \mathcal{L}$ between states of $\mathbb{F}$ and formulas, an interpretation $\overline{v} : \mathcal{L} \to \mathbb{F}^+$ into the complex algebra of $\mathbb{F}$ can be defined, which is an $\mathcal{L}$-homomorphism, and is obtained as the unique homomorphic extension of the equivalent functional representation of the relation $\Vdash$ as a map $v : \mathsf{PROP} \to \mathbb{F}^+$, defined as $v(p) = \Vdash^{-1}[p]$[14]. In this way, interpretations can be derived from satisfaction relations, so that for any $a \in J^\infty(\mathbb{F}^+)$ and any formula ϕ,

$$a \Vdash \phi \quad \text{iff} \quad a \leq \overline{v}(\phi), \tag{1}$$

where, on the left-hand side, $a \in J^\infty(\mathbb{F}^+)$ is identified with a state of $\mathbb{F}$ via the isomorphism $\mathbb{F} \cong (\mathbb{F}^+)_+$. Conversely, consider a perfect lattice with completely meet-preserving operation $\mathbb{C} = (\mathbb{L}, \Box)$, and a homomorphic assignment $\overline{v} : \mathcal{L} \to \mathbb{C}$, and recall that the complete lattice $\mathbb{L}$ can be identified with the lattice $\mathbb{P}^+$ arising from some RS-polarity $\mathbb{P} = (A, X, \perp)$. We want to define a suitable relation $R = R_\Box$ and satisfaction relation $\Vdash_{\overline{v}}$ satisfying the condition (1). The method we are going to illustrate hinges on the dual characterization of $\overline{v}$ as a pair of relations $(\Vdash_{\overline{v}}, \succ_{\overline{v}})$ such that $\Vdash_{\overline{v}} \subseteq J^\infty(\mathbb{L}) \times \mathcal{L} \cong A \times \mathcal{L}$ and $\succ_{\overline{v}} \subseteq M^\infty(\mathbb{L}) \times \mathcal{L} \cong X \times \mathcal{L}$. This dual characterization is established by induction on formulas.

[14] Notice that in order for this equivalent functional representation to be well defined, we need to assume that the relation $\Vdash$ is $\mathbb{F}^+$-*compatible*, i.e. that $\Vdash^{-1}[p] \in \mathbb{F}^+$ for every $p \in \mathsf{PROP}$. In the Boolean case, every relation from W to LML is clearly $\mathbb{F}^+$-compatible, but already in the distributive case this is not so: indeed $\Vdash^{-1}[p]$ needs to be an upward- or downward-closed subset of $\mathbb{F}$. This gives rise to the persistency condition, e.g. in the relational semantics of intuitionistic logic.

The base of the induction is clear: for every $a \in J^\infty(\mathbb{P}^+)$ and every $p \in \mathsf{PROP} \cup \{0, 1\}$, we define

$$a \Vdash_{\overline{v}} p \quad \text{iff} \quad a \leq \overline{v}(p). \tag{2}$$

Now let us turn to the inductive step for the box. Since $\overline{v} : \mathcal{L} \to \mathbb{P}^+$ is a homomorphism, $\overline{v}(\Box\phi) = \Box^{\mathbb{P}^+}\overline{v}(\phi)$. Suppose that (1) holds for ϕ.

Since $\mathbb{P}^+$ is perfect, $\overline{v}(\phi) = \bigwedge\{x \in M^\infty(\mathbb{L}) \mid \overline{v}(\phi) \leq x\}$. Thus,

$$\begin{aligned} a \leq \overline{v}(\Box\phi) &\text{ iff } a \leq \Box^{\mathbb{P}^+}\overline{v}(\phi) \\ &\text{ iff } a \leq \Box^{\mathbb{P}^+} \bigwedge\{x \in M^\infty(\mathbb{P}^+) \mid \overline{v}(\phi) \leq x\} \\ &\text{ iff } a \leq \bigwedge\{\Box^{\mathbb{P}^+} x \mid x \in M^\infty(\mathbb{P}^+) \text{ and } \overline{v}(\phi) \leq x\} \\ &\text{ iff } \forall x[(x \in M^\infty(\mathbb{L}) \;\&\; \overline{v}(\phi) \leq x) \to a \leq \Box^{\mathbb{P}^+} x] \end{aligned}$$

Notice that, at the end of this chain of equivalence, we have equivalently reduced the whole information on $\Box$ to the information whether $a \leq \Box^{\mathbb{P}^+} x$ for each a and x. So this can be taken as the definition of the relation $R \subseteq A \times X$: we let aRx iff $a \leq \Box^{\mathbb{P}^+} y$.

To turn the last clause above into a satisfaction clause for $\Box$, we firstly replace $M^\infty(\mathbb{L})$ with X, which we identify via the isomorphism $\mathbb{P} \cong (\mathbb{P}^+)_+$. Secondly, we need to recall the second relation $\succ_{\overline{v}}$ between elements of X and formulas, obeying the following condition, which is to be defined by induction on the structure of the formulas in such a way that the following condition holds, analogously to (1):

$$x \succ \phi \quad \text{iff} \quad \overline{v}(\phi) \leq x. \tag{3}$$

These considerations produce the following satisfaction clause for $\Box$:

$$a \Vdash_{\overline{v}} \Box\phi \text{ iff } a \leq \overline{v}(\Box\phi) \text{ iff } \forall x[(x \in X \;\&\; x \succ \phi) \to aR_\Box x]$$

The co-satisfaction relation $\succ$ deserves some further comment: in the Boolean and distributive settings, $\succ$ is completely determined by $\Vdash$, and is hence not mentioned explicitly there. Here, in the non-distributive setting, the relation needs to be defined along with $\Vdash$. Equation (3) determines the base case:

$$y \succ \overline{v}(p) \quad \text{iff} \quad \overline{v}(p) \leq y. \tag{4}$$

Specializing the clause above to powerset algebras $\mathcal{P}(W)$, we would have $y \succ_V p$ iff $V(p) \leq y$ iff $V(p) \subseteq W/\{x\}$ for some $x \in W$ iff $\{x\} \nsubseteq V(p)$ iff $x \notin V(p)$ iff $x \nVdash p$, which shows that the relation $\succ$ can be regarded as an upside-down description of the satisfaction relation $\Vdash$, namely a *co-satisfaction*, or *refutation*.

The inductive step for the derivation of the co-satisfaction clause for $\Box$ goes as follows:

$$\begin{aligned} \overline{v}(\Box\phi) \leq x &\text{ iff } \bigvee\{a \in J^\infty(\mathbb{L}) \mid a \leq \overline{v}(\Box\phi)\} \leq x \\ &\text{ iff } \forall a[(a \in J^\infty(\mathbb{L}) \;\&\; a \leq \overline{v}(\Box\phi)) \to a \leq x] \\ &\text{ iff } \forall a[(a \in A \;\&\; a \Vdash \Box\phi) \to a \perp x]. \end{aligned}$$

The last line follows from Eq. (1) for $\Box\phi$, and the identification, via the isomorphism $\mathbb{P} \cong (\mathbb{P}^+)_+$, of $J^\infty(\mathbb{L})$ with A, and of the lattice order $\leq$ (restricted to $J^\infty(\mathbb{L}) \times M^\infty(\mathbb{L})$) with the incidence relation $\perp$ of the polarity.

B Proofs of Technical Lemmas

Proof (Lemma 1). We only prove the part concerning z. Let $x \in z{\uparrow}^{\downarrow\uparrow}$, and let us show that $z \leq x$. That is, let us fix a such that $a \perp z$, and show that $a \perp x$. Since $\perp \circ \leq \subseteq \perp$, from $a \perp z$ it follows that $\forall y(z \leq y \rightarrow a \perp y)$, which means that $a \in z{\uparrow}^{\downarrow}$. Since by assumption $x \in z{\uparrow}^{\downarrow\uparrow}$, this implies that $a \perp x$, as required.

Proof (Corollary 1). Since $z{\uparrow}$ is Galois-stable and contains z and, by definition, $z^{\downarrow\uparrow}$ is the smallest such set, $z^{\downarrow\uparrow} \subseteq z{\uparrow}$. For the converse inclusion, let $z \leq y$ and $a \perp z$. As $\perp \circ \leq \subseteq \perp$, this implies $a \perp y$, which shows that $y \in z^{\downarrow\uparrow}$, as required.

Proof (Lemma 2). Assume that aRz and $z \leq y$. To show that $y \in R[a]$, by the second compatibility condition, it is enough to show that $y \in R[a]^{\downarrow\uparrow}$. That is, let us fix $b \in R[a]^{\downarrow}$ and show that $b \perp y$. From $b \in R[a]^{\downarrow}$ and aRz it follows that $b \perp z$. This and $z \leq y$ imply that $b \perp y$, given that $\perp \circ \leq \subseteq \perp$. The remaining part is proven similarly.

Proof (Lemma 5). Clearly, $C(u) \leq 1u \leq u$, which proves the first inequality.

$$C(C(u)) = \bigwedge\nolimits_{s\in S} sC(u) = \bigwedge\nolimits_{s\in S} s(\bigwedge\nolimits_{t\in S} tu) = \bigwedge\nolimits_{s\in S} \bigwedge\nolimits_{t\in S} stu \geq \bigwedge\nolimits_{s'\in S} s'u = C(u).$$

References

1. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, Cambridge (2003)
2. Baltag, A., Moss, L.S., Solecki, S.: The logic of public announcements, common knowledge, and private suspicions. In: Proceedings of the 7th Conference on Theoretical Aspects of Rationality and Knowledge, pp. 43–56. Morgan Kaufmann Publishers Inc. (1998)
3. Bimbó, K., Dunn, J.M., et al.: Four-valued logic. Notre Dame J. Form. Log. **42**(3), 171–192 (2001)
4. Conradie, W., Craig, A.: Canonicity results for mu-calculi: an algorithmic approach. J. Log. Comput. (2015). Forthcoming, preliminary version on arXiv:1408.6367 [math.LO]
5. Conradie, W., Palmigiano, A., Sourabh, S.: Algebraic modal correspondence: Sahlqvist and beyond (Submitted)
6. Conradie, W., Craig, A., Palmigiano, A., Zhao, Z.: Constructive canonicity for lattice-based fixed point logics. Submitted, preliminary version on arXiv:1603.06547 [math.LO]
7. Conradie, W., Fomatati, Y., Palmigiano, A., Sourabh, S.: Algorithmic correspondence for intuitionistic modal mu-calculus. Theor. Comput. Sci. **564**, 30–62 (2015)
8. Conradie, W., Ghilardi, S., Palmigiano, A.: Unified correspondence. In: Baltag, A., Smets, S. (eds.) Johan van Benthem on Logic and Information Dynamics. Outstanding Contributions to Logic, vol. 5, pp. 933–975. Springer, Switzerland (2014)
9. Conradie, W., Palmigiano, A.: Algorithmic correspondence and canonicity for non-distributive logics. J. Log. Comput. forthcoming, preliminary version on arXiv:1603.08515 [math.LO]
10. Conradie, W., Palmigiano, A.: Constructive canonicity of inductive inequalities. Submitted, preliminary version on arXiv:1603.08341 [math.LO]
11. Conradie, W., Palmigiano, A.: Algorithmic correspondence and canonicity for distributive modal logic. Ann. Pure Appl. Log. **163**(3), 338–376 (2012)

12. Conradie, W., Palmigiano, A., Sourabh, S., Zhao, Z.: Canonicity and relativized canonicity via pseudo-correspondence: an application of ALBA. Submitted, preliminary version on arXiv:1511.04271 [cs.LO]
13. Conradie, W., Palmigiano, A., Zhao, Z.: Sahlqvist via translation. Submitted, preliminary version on arXiv:1603.08220 [math.LO]
14. Conradie, W., Robinson, C.: On Sahlqvist theory for hybrid logic. J. Log. Comput. exv045v1-exv045 (2015)
15. Crapo, H.: Unities and negation: on the representation of finite lattices. J. Pure Appl. Algebra **23**(2), 109–135 (1982)
16. Davey, B.A., Priestley, H.A.: Lattices and Order. Cambridge Univerity Press, Cambridge (2002)
17. Dunn, J.M.: Gaggle theory: an abstraction of Galois connections and residuation, with applications to negation, implication, and various logical operators. In: van Eijck, J. (ed.) Logics in AI. LNCS, vol. 478, pp. 31–51. Springer, Heidelberg (1991)
18. Dunn, M.J., Gehrke, M., Palmigiano, A.: Canonical extensions and relational completeness of some substructural logics. J. Symb. Log. **70**(3), 713–740 (2005)
19. Düntsch, I., Orłowska, E., Radzikowska, A., Vakarelov, D.: Relational representation theorems for some lattice-based structures. J. Relat. Methods Comput. Sci. **1**, 132–160 (2004)
20. Fagin, R., Moses, Y., Vardi, M.Y., Halpern, J.Y.: Reasoning About Knowledge. MIT press, Cambridge (2003)
21. Frittella, S., Palmigiano, A., Santocanale, L.: Dual characterizations for finite lattices via correspondence theory for monotone modal logic. J. Log. Comput. (2016). Forthcoming, preliminary version on arXiv:1408.1843 [math.LO]
22. Galatos, N., Jipsen, P., Kowalski, T., Ono, H.: Residuated Lattices: An Algebraic Glimpse at Substructural Logics. Elsevier, Amsterdam (2007)
23. Ganter, B., Wille, R.: Applied lattice theory: formal concept analysis. In: Grätzer, G. (ed.) In General Lattice Theory. Birkhäuser. Citeseer (1997)
24. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Heidelberg (1999)
25. Gehrke, M.: Generalized Kripke frames. Studia Logica **84**(2), 241–275 (2006)
26. Greco, G., Ma, M., Palmigiano, A., Tzimoulis, A., Zhao, Z.: Unified correspondence as a proof-theoretic tool. Forthcoming, preliminary version on arXiv:1603.08204 [math.LO]
27. Hartonas, T.: Modal and temporal extensions of non-distributive logics. Log. J. IGPL **24**, 156–185 (2015)
28. Hartonas, T.: Order-dual relational semantics for non-distributive al logics: a general framework (2015)
29. Hsu, G., Hannan, M.T., Pólos, L.: Typecasting, legitimation, and form emergence: a formal theory. Sociol. Theor. **29**(2), 97–123 (2011)
30. Järvinen, J., Orłowska, E.: Relational correspondences for lattices with operators. In: MacCaull, W., Winter, M., Düntsch, I. (eds.) RelMiCS 2005. LNCS, vol. 3929, pp. 134–146. Springer, Heidelberg (2006)
31. Kamide, N.: Kripke semantics for modal substructural logics. J. Log. Lang. Inf. **11**(4), 453–470 (2002)
32. Kuijken, B., Leenders, M.A.A.M., Wijnberg, N.M., Gemser, G.: The producer-consumer classification gap and its effects on music festival success (2016) (Submitted)
33. Kurtonina, N.: Categorical inference and modal logic. J. Log. Lang. Inf. **7**, 399–411 (1998)
34. Moshier, M.A., Jipsen, P.: Topological duality and lattice expansions, II: lattice expansions with quasioperators. Algebra Universalis **71**(3), 221–234 (2014)
35. Navis, C., Glynn, M.A.: How new market categories emerge: temporal dynamics of legitimacy, identity, and entrepreneurship in satellite radio, 1990–2005. Adm. Sci. Q. **55**(3), 439–471 (2010)

36. Paleo, I.O., Wijnberg, N.M.: Classification of popular music festivals: a typology of festivals and an inquiry into their role in the construction of music genres. Int. J. Arts Manag. **8**, 50–61 (2006)
37. Palmigiano, A., Sourabh, S., Zhao, Z.: Sahlqvist theory for impossible worlds. J. Log. Comput. Forthcoming, preliminary version on arXiv:1603.08202 [math.LO]
38. Palmigiano, A., Sourabh, S., Zhao, Z.: Jónsson-style canonicity for ALBA-inequalities. J. Log. Comput. exv041v1-exv041 (2015)
39. Ploščica, M.: A natural representation of bounded lattices. Tatra Mt. Math. Publ. **5**, 75–88 (1995)
40. Suzuki, T.: Canonicity results of substructural and lattice-based logics. Rev. Symb. Log. **4**, 1–42 (2011)
41. Wijnberg, N.M.: Innovation and organization: value and competition in selection systems. Organ. Stud. **25**(8), 1413–1433 (2004)
42. Wijnberg, N.M.: Classification systems and selection systems: the risks of radical innovation and category spanning. Scand. J. Manag. **27**(3), 297–306 (2011)

A Logical Approach to Context-Specific Independence

Jukka Corander[1,2], Antti Hyttinen[3], Juha Kontinen[1(✉)], Johan Pensar[4], and Jouko Väänänen[1,5]

[1] Department of Mathematics and Statistics, University of Helsinki, Helsinki, Finland
{jukka.corander,juha.kontinen,jouko.vaananen}@helsinki.fi
[2] Department of Biostatistics, University of Oslo, Oslo, Norway
[3] HIIT, Department of Computer Science, University of Helsinki, Helsinki, Finland
antti.hyttinen@helsinki.fi
[4] Department of Mathematics and Statistics, Åbo Akademi University, Turku, Finland
jopensar@abo.fi
[5] Institute for Logic, Language and Computation, University of Amsterdam, Amsterdam, The Netherlands

Abstract. Bayesian networks constitute a qualitative representation for conditional independence (CI) properties of a probability distribution. It is known that every CI statement implied by the topology of a Bayesian network G is witnessed over G under a graph-theoretic criterion called d-separation. Alternatively, all such implied CI statements have been shown to be derivable using the so-called semi-graphoid axioms. In this article we consider Labeled Directed Acyclic Graphs (LDAG) the purpose of which is to graphically model situations exhibiting context-specific independence (CSI). We define an analogue of dependence logic suitable to express context-specific independence and study its basic properties. We also consider the problem of finding inference rules for deriving non-local CSI and CI statements that logically follow from the structure of a LDAG but are not explicitly encoded by it.

1 Introduction

Dependence logic [24] adds the concept of dependence to first-order logic by means of atomic formulas

$$=(x_1, \dots, x_n) \tag{1}$$

the meaning of which is that the value of x_n is functionally determined by the values of the variables $x_1, \dots, x_{n-1}$. The area of dependence logic and its team semantics have developed and expanded rapidly during the past few years. In this article we are concerned with a variant of dependence logic called Independence Logic [8] defined in terms of independence atoms $\boldsymbol{x} \perp_{\boldsymbol{z}} \boldsymbol{y}$ instead of dependence atoms. The meaning of the atom $\boldsymbol{x} \perp_{\boldsymbol{z}} \boldsymbol{y}$ is that, when the value of $\boldsymbol{z}$ fixed, knowing the value of $\boldsymbol{x}$ does not tell us anything new about the value of $\boldsymbol{y}$.

J. Väänänen et al. (Eds.): WoLLIC 2016, LNCS 9803, pp. 165–182, 2016.
DOI: 10.1007/978-3-662-52921-8_11

Independence atoms correspond to a widely studied class of database dependencies called embedded multivalued dependencies. Furthermore, independence atoms and the notion of statistical conditional independence $\boldsymbol{X} \perp \boldsymbol{Y}|\boldsymbol{Z}$ have interesting connections as the former can be seen as a qualitative version of the latter.

Bayesian networks [14] are a popular tool for modeling complex multivariate systems. The basis of a Bayesian network is a directed acyclic graph (DAG) in which the nodes represent random variables and the directed edges represent direct dependencies between the variables. On the other hand, missing edges give rise to statements of conditional independence (CI) which can be verified directly from the graph using the graph-theoretic criterion called *d*-separation [21].

To increase the flexibility of the dependence structure associated with traditional Bayesian networks, [1] introduced and formalized the notion of context-specific independence (CSI). More specifically, they showed how certain local CSI statements are particularly convenient to include in the Bayesian network framework. A local CSI statement basically corresponds to the influence of an edge vanishing in a certain context. To generalize this idea, [22] introduced the class of labeled directed acyclic graphs (LDAGs) which can capture local CSI statements through labels assigned to the edges. Analogously to the *d*-separation criterion, (a subset of) non-local CSI statements can be verified using a concept called CSI-separation.

Conditional independence has also been given a qualitative characterization in terms of logical axioms (see Sect. 4). The semi-graphoid axioms of conditional independence are known to be sound for all distributions, and furthermore correspond exactly to d-separation in the context of Bayesian networks [6,25]. In this article we formulate a logic capable of formalizing CSI statements. For that end, we define an analogue of dependence logic suitable to express context-specific independence and study its basic properties. We also give a logical characterization for CSI-separation in LDAGs, and address the open problem of finding a complete method of deriving non-local CSI and CI statements that logically follow from the structure of a LDAG but are not explicitly encoded by it.

2 Preliminaries

2.1 Bayesian Networks

A Directed Acyclic Graph (DAG) $G = (\Delta, E)$ is specified by a set of nodes $\Delta = \{1, \ldots, n\}$ and a set of directed edges E where $(i, j) \in E$ represents a directed edge from node i to node j. The parents of a node j, denoted by Π_j, is defined as all nodes from which there is a directed edge to node j, that is, $\Pi_j = \{i \in \Delta \mid (i, j) \in E\}$. The descendants of a node i is all nodes which can be reached from node i following the direction of the edges.

In a Bayesian network, the nodes of the graph represent random variables $X_\Delta = \{X_1, \ldots, X_n\}$. As is typical in the graphical model literature, the terms node and variable will occasionally be used interchangeably. We denote by P_Δ a probability distribution over the random variables X_Δ. Each variable X_i is assumed to take values in a finite discrete set of outcomes denoted by $\mathcal{X}_i$. The joint outcome space of a set of variables X_S, where $S \subseteq \Delta$, is defined as the Cartesian product of the sets $\mathcal{X}_i$ for $i \in S$.

Definition 1 (Conditional independence (CI)). *Let A, B, and S be subsets of Δ. The variables X_A are conditionally independent of X_B given X_S if*

$$P(X_A = e_A | X_B = e_B, X_S = e_S) = P(X_A = e_A | X_S = e_S),$$

for all $(e_A, e_B, e_S) \in \mathcal{X}_A \times \mathcal{X}_B \times \mathcal{X}_S$ for which $P(X_B = e_B, X_S = e_S) > 0$. This is denoted by $X_A \perp X_B \mid X_S$.

A Bayesian network is specified by a pair (G, P_Δ) where $G = (\Delta, E)$ is a DAG and P_Δ is a probability distribution satisfying the CI statements encoded by G. The dependence structure of a DAG is characterized by the so-called local directed Markov property [21], which states that each variable X_j is conditionally independent of its non-descendants given its parents X_{Π_j}. Accordingly, the joint probability distribution P_Δ can be factorized as

$$P(X_1 = e_1, X_1 = e_2, \ldots, X_n = e_n) = \prod_{j=1}^{n} P(X_j = e_j | X_{\Pi_j} = e_{\Pi_j}), \qquad (2)$$

for any $e_\Delta \in \mathcal{X}_\Delta$. The joint distribution P_Δ can hence be thought of as being constructed from node-wise conditional distributions.

The CI statements of a graph G determined by the local directed Markov property are called *local* CIs and are denoted by $I_{loc}(G)$. However, the set $I_{loc}(G)$ implies also other *non-local* CIs which can be verified using a graph-theoretic criterion called d-separation.

Definition 2 (d-separation). *Let $G = (\Delta, E)$ be a DAG and let A, B, and S be disjoint subsets of Δ. The set A is d-separated from B by S if there is no trail in G from a node in A to a node in B along which every node that delivers an arrow (i.e., tail in either direction) is outside of S, and every node with converging arrows (i.e., heads in both directions) either is or has a descendant in S.*

For a CI statement ϕ, we write $I_{loc}(G) \models \phi$ if all distributions P_Δ that satisfy $I_{loc}(G)$ also satisfy ϕ, that is, ϕ is implied by $I_{loc}(G)$. The following result is due to [6, 25].

Theorem 1 (Soundness and completeness of d-separation). *Let $G = (\Delta, E)$ be a DAG and P_Δ a distribution satisfying every CI in $I_{loc}(G)$. Let A, B, and S be disjoint subsets of Δ. Then it holds that*

- *if A is d-separated from B by S in G, then P_Δ satisfies $X_A \perp X_B \mid X_S$.*
- *if $I_{loc}(G) \models X_A \perp X_B \mid X_S$, then A is d-separated from B by S in G.*

We will end this section with an example illustrating the use of d-separation.

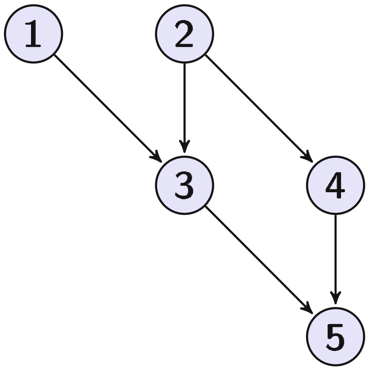


Fig. 1. DAG over five variables.

Example 1. Consider the DAG G in Fig. 1. Note first that $I_{loc}(G)$ consists of the following five CIs: $X_1 \perp \{X_2, X_4\}$, $X_2 \perp X_1$, $X_3 \perp X_4 \mid \{X_1, X_2\}$, $X_4 \perp \{X_1, X_3\} \mid X_2$ and $X_5 \perp \{X_1, X_2\} \mid \{X_3, X_4\}$, which must hold for any distribution that factorizes according to G. In addition, we can further infer that the non-local CI $X_1 \perp X_4 \mid \{X_2, X_3, X_5\}$ must hold in such a distribution since node 1 is d-separated from node 4 by nodes $\{2, 3, 5\}$.

2.2 Context-Specific Independence

The class of Labeled Directed Acyclic Graphs (LDAGs) was recently introduced in [22] as a generalization of DAGs. The purpose of the class of LDAGs is to graphically model situations exhibiting context-specific independence [1], which cannot be captured by CI-based models (see the survey [20]).

Definition 3 (Context-specific independence (CSI)). *Let A, B, C, and S be disjoint subsets of Δ. The variables X_A are contextually independent of X_B given $X_C = e_C$ and X_S if*

$$P(X_A = e_A | X_B = e_B, X_C = e_C, X_S = e_S) = P(X_A = e_A | X_C = e_C, X_S = e_S),$$

for all $(e_A, e_B, e_S) \in \mathcal{X}_A \times \mathcal{X}_B \times \mathcal{X}_S$ for which $P(X_B = e_B, X_C = e_C, X_S = e_S) > 0$. This is denoted by $X_A \perp X_B \mid X_C = e_C, X_S$.

Local CSI statements, which address only a variable X_j and its parents X_{Π_j}, can naturally be included in the Bayesian network framework. More specifically, a CSI statement is defined as local with respect to a DAG if it is of the form

$$X_j \perp X_B | X_{\Pi_j \setminus B} = e_{\Pi_j \setminus B}, \tag{3}$$

where $B \subset \Pi_j$. By Definition 3, this independence statement holds if and only if

$$P(X_j = e_j | X_B = e_B, X_{\Pi_j \setminus B} = e_{\Pi_j \setminus B}) = P(X_j = e_j | X_{\Pi_j \setminus B} = e_{\Pi_j \setminus B}),$$

for all $(e_j, e_B) \in \mathcal{X}_j \times \mathcal{X}_B$ for which $P(X_B = e_B, X_{\Pi_j \setminus B} = e_{\Pi_j \setminus B}) > 0$. In other words, a local CSI renders a variable conditionally independent of some of its parents given a certain context specified by the remaining parents. To capture such restrictions in the model structure, [22] proposed adding labels to the edges in the DAG.

Definition 4 (Labeled Directed Acyclic Graph (LDAG)). *Let $G = (\Delta, E)$ be a DAG over random variables X_Δ and let $L_{(i,j)} = \Pi_j \setminus \{i\}$. A* label *on an edge (i, j) is a subset of $\mathcal{X}_{L_{(i,j)}}$, denoted by $\mathcal{L}_{(i,j)}$, encoding a collection of local CSI statements according to*

$$X_j \perp X_i \mid X_{L_{(i,j)}} = e_{L_{(i,j)}} \text{ for all } e_{L_{(i,j)}} \in \mathcal{L}_{L_{(i,j)}}.$$

An LDAG $G_L = (\Delta, E, \mathcal{L}_E)$ is a DAG $G = (\Delta, E)$ where the edges have been assigned labels as specified by $\mathcal{L}_E = \{\mathcal{L}_{(i,j)}\}_{(i,j) \in E}$.

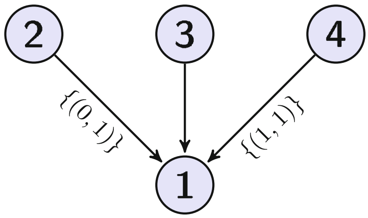


Fig. 2. LDAG over four binary variables.

Example 2. Consider the LDAG in Fig. 2 which represents the dependence structure over four binary variables with outcome space $\mathcal{X}_i = \{0, 1\}$. Note that given an ordering of the variables, the indices of the variables specifying a label need not be explicitly stated. According to Definition 4, the labels encode the local CSI statements

$$X_1 \perp X_2 \mid (X_3, X_4) = (0, 1) \text{ and } X_1 \perp X_4 \mid (X_2, X_3) = (1, 1),$$

respectively. In other words, X_1 is contextually independent of X_2 given $X_3 = 0$ and $X_4 = 1$. Moreover, X_1 is contextually independent of X_4 given $X_2 = 1$ and $X_3 = 1$.

One of the main motivations for including CSI in Bayesian networks was to reduce the number of parameters needed to specify the model distribution. The textbook way of defining the conditional probability distributions in (2) is through so-called conditional probability tables (CPTs) which simply list the conditional probabilities for each parent configuration. The number of parameters needed to specify a CPT of a node grows exponentially with the number of parents of the node. However, a local CSI statement implies that several distinct parent configurations induce the same conditional distribution which thereby needs only be defined once.

Example 2 (continued). Let us continue with the previous example concerning the LDAG in Fig. 2. A traditional CPT over variable X_1 is seen in Table 1(a). Notice that there are certain regularities in the table in form of identical distributions. These regularities correspond to the local CSI statements which are encoded by the labels. Rather than defining identical distributions several times, we can construct a reduced CPT as illustrated in Table 1(b). A star means that the variable may take on any value, for example, $(*, 0, 1) = \{(0, 0, 1), (1, 0, 1)\}$. Ultimately, a reduced CPT represents a partition of the parent outcome space. Each row represents a class in the partition such that the conditional distribution is invariant for configurations belonging to the same class.

Table 1. (a) A traditional CPT and (b) A reduced CPT over variable X_1 in Fig. 2.

X_2	X_3	X_4	$P(X_1 \mid X_{\Pi_1})$
0	0	0	p_1
0	0	1	p_2
0	1	0	p_3
0	1	1	p_4
1	0	0	p_5
1	0	1	p_2
1	1	0	p_6
1	1	1	p_6

(a)

X_2	X_3	X_4	$P(X_1 \mid X_{\Pi_1})$
0	0	0	p_1
*	0	1	p_2
0	1	0	p_3
0	1	1	p_4
1	0	0	p_5
1	1	*	p_6

(b)

As in [22], we restrict attention to so-called *maximal* and *regular* LDAGs. Maximality states that it is not possible to add a configuration $e_{L_{(i,j)}}$ to the label $\mathcal{L}_{(i,j)}$ without inducing an additional local CSI not encoded already by the original labels. Moreover, regularity simply states that each of the labels $\mathcal{L}_{(i,j)}$ of an LDAG is a strict subset of $\mathcal{X}_{L_{(i,j)}}$. Regularity and maximality together imply that a label $\mathcal{L}_{(i,j)}$ cannot induce that $X_j \perp X_i \mid X_{L_{(i,j)}} = e_{L_{(i,j)}}$ for all $e_{L_{(i,j)}} \in \mathcal{X}_{L_{(i,j)}}$, which corresponds to the CI $X_j \perp X_i \mid X_{L_{(i,j)}}$.

Analogously to the case with DAGs, an important issue with LDAGs is the derivation of non-local independence statements that logically follow from the structure of the LDAG but are not explicitly encoded by it (by the labels or the DAG-structure).

Definition 5. *Let $G_L = (\Delta, E, \mathcal{L}_E)$ be an LDAG. We denote by $I_{loc}(G_L)$ the set of local CI statements $I_{loc}(G)$, encoded by $G = (\Delta, E)$, together with the set of local CSI statements encoded by the labels $\mathcal{L}_E$.*

Already in [1] a method called *CSI-separation*, which is analogous to *d*-separation for DAGs, was introduced for the purpose of verifying non-local CSI statements.

Before defining CSI-separation in terms of LDAGs, we need to define the notion of *satisfied label*.

Definition 6. *Let $G_L = (\Delta, E, \mathcal{L}_E)$ be an LDAG, and $X_C = e_C$ a context where $C \subseteq \Delta$. A label $\mathcal{L}_{(i,j)} \subseteq \mathcal{L}_E$ is* satisfied *in the context $X_C = e_C$ if $L_{(i,j)} \cap C \neq \emptyset$ and*

$$\{e_{L_{(i,j)} \cap C} \times \mathcal{X}_{L_{(i,j)} \setminus C}\} \subseteq \mathcal{L}_{(i,j)}.$$

For an LDAG G_L and a context $X_C = e_C$, we define $G(e_C) = (\Delta, E \setminus E')$, where $E' = \{(i,j) \in E \mid \mathcal{L}_{(i,j)}$ is satisfied$\}$. Note that $G(e_C)$ is the subgraph of G that arises by removing edges whose labels are satisfied in the context $X_C = e_C$. We are now ready to define CSI-separation [22].

Definition 7 (CSI-separation). *Let $G_L = (\Delta, E, \mathcal{L}_E)$ be an LDAG and let A, B, C, S be disjoint subsets of Δ. The set X_A is CSI-separated from X_B by X_S in the context $X_C = e_C$ in G_L, if X_A is d-separated from X_B by $X_{C \cup S}$ in $G(e_C)$.*

As stated by the following result (see Theorem 5.3 in [14]), CSI-separation is a sound method for verifying non-local CSIs.

Theorem 2 (Soundness of CSI-separation). *Let $G_L = (\Delta, E, \mathcal{L}_E)$ be an LDAG, let P_Δ be a distribution satisfying $I_{loc}(G_L)$, and let A, B, C, S be disjoint subsets of Δ. If X_A is CSI-separated from X_B by X_S in the context $X_C = e_C$ in G_L, then the distribution P_Δ satisfies the CSI $X_A \perp X_B \mid X_C = e_C, X_S$.*

However, unlike *d*-separation for DAG structures, CSI-separation is not a complete method for discovering non-local independencies implied by an LDAG structure. In fact, *d*-separation is not a complete method for discovering CI statements in an LDAG. This is illustrated in the following example.

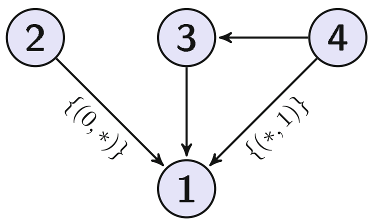


Fig. 3. LDAG over four binary variables.

Example 3. Consider the LDAG in Fig. 3, where $\{(0,*)\}$ again is a shorthand for the set $\{(0,0),(0,1)\}$. Assume that P_Δ is a joint distribution satisfying the independencies encoded by the LDAG. As discussed in [22], the underlying DAG structure does not allow us to infer the CI statement $X_2 \perp X_4 \mid \{X_1, X_3\}$ through the use of *d*-separation. Using CSI-separation, however, we can verify that $X_2 \perp X_4 \mid X_1, X_3 = 0$ and $X_2 \perp X_4 \mid X_1, X_3 = 1$. Consequently, a reasoning by cases argument allows one to conclude that $X_2 \perp X_4 \mid \{X_1, X_3\}$ holds in P_Δ (see [22] for more details).

The next theorem shows that the problem of deciding whether a CI statement ϕ is implied by an LDAG structure G_L is coNP-hard. It is worth noting that for DAGs this problem can be solved in polynomial time. The proof of the theorem can be found in the Appendix.

Theorem 3. *The problem of deciding whether an (context specific) independence is implied by an LDAG structure is coNP-hard.*

2.3 Team Semantics and Independence Logic

The syntax of independence logic, $\mathrm{FO}(\perp_c)$, extends the syntax of first-order logic (FO), defined in terms of $\vee$, $\wedge$, $\neg$, $\exists$ and $\forall$, by atomic independence formulas of the form

$$\boldsymbol{x} \perp_{\boldsymbol{z}} \boldsymbol{y}, \tag{4}$$

where $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$ are tuples of variables. The set $\mathrm{Fr}(\phi)$ of free variables of $\phi \in \mathrm{FO}(\perp_c)$ is defined analogously to first-order logic stipulating that all variable occurrences in independence atoms are free.

The semantics of independence logic is formulated using sets X of assignments called teams.

Definition 8. *Let $\mathfrak{A}$ be a model with domain A, and $\{x_1, \ldots, x_k\}$ a finite set of variables.*

- *A team X of $\mathfrak{A}$ with domain* $\mathrm{Dom}(X) = \{x_1, \ldots, x_k\}$ *is any set of assignments from the variables $\{x_1, \ldots, x_k\}$ into the set A.*
- *If s is an assignment, x a variable, and $a \in A$, then $s(a/x)$ denotes the assignment (with domain* $\mathrm{Dom}(s) \cup \{x\}$*) that agrees with s everywhere except that it maps x to a.*
- *For a function $F\colon X \to \mathcal{P}(A) \setminus \{\emptyset\}$, we define the operations of* Supplementation *$X(F/x_n)$ and* Duplication *$X(A/x_n)$ as follows:*

$$X(F/x_n) = \{s(F(s)/x_n) : s \in X \text{ and } a \in F(s)\}$$
$$X(A/x_n) = \{s(a/x_n) : s \in X \text{ and } a \in A\}.$$

We are now ready to define the semantics of independence logic. We restrict attention to formulas in a negation normal form in which negation is allowed to appear only in front of first-order atomic formulas. Below, atomic formulas and their negations are called literals, and $\mathfrak{A} \models_s \phi$ refers to satisfaction in first-order logic.

Definition 9. *Let $\mathfrak{A}$ be a model and X a team of A. The satisfaction relation $\mathfrak{A} \models_X \phi$ is defined as follows:*

- *If ϕ is a first-order literal, then $\mathfrak{A} \models_X \phi$ iff for all $s \in X$: $\mathfrak{A} \models_s \phi$.*
- *$\mathfrak{A} \models_X \boldsymbol{x} \perp_{\boldsymbol{z}} \boldsymbol{y}$ iff for all $s, s' \in X$ such that $s(\boldsymbol{z}) = s'(\boldsymbol{z})$ there is $s^* \in X$ such that $s^*(\boldsymbol{x}\boldsymbol{z}) = s(\boldsymbol{x}\boldsymbol{z})$, and $s^*(\boldsymbol{y}) = s'(\boldsymbol{y})$.*
- *$\mathfrak{A} \models_X \psi \wedge \phi$ iff $\mathfrak{A} \models_X \psi$ and $\mathfrak{A} \models_X \phi$.*

– $\mathfrak{A} \models_X \psi \vee \phi$ *iff* $X = Y \cup Z$ *such that* $\mathfrak{A} \models_Y \psi$ *and* $\mathfrak{A} \models_Z \phi$.
– $\mathfrak{A} \models_X \exists x_n \psi$ *iff* $\mathfrak{A} \models_{X(F/x_n)} \psi$ *for some* $F\colon X \to \mathcal{P}(A) \setminus \{\emptyset\}$.
– $\mathfrak{A} \models_X \forall x_n \psi$ *iff* $\mathfrak{A} \models_{X(A/x_n)} \psi$.

Above, we assume that the domain of X *contains the variables free in* ϕ*. Finally, a sentence* ϕ *is true in a model* $\mathfrak{A}$ *(abbreviated* $\mathfrak{A} \models \phi$*) if* $\mathfrak{A} \models_{\{\emptyset\}} \phi$.

One of the most basic observations about team semantics is the so-called flatness property of FO-formulas.

Theorem 4 ([24]). *Let* $\phi \in \mathrm{FO}$*. Then for all* $\mathfrak{A}$ *and* X *it holds that*

$$\mathfrak{A} \models_X \phi \Leftrightarrow \mathfrak{A} \models_s \phi \textit{ for all } s \in X.$$

Another important property of all independence logic formulas is the following locality property. For a team X and $V \subseteq \mathrm{Dom}(X)$, we define $X \upharpoonright V := \{s \upharpoonright V \mid s \in X\}$.

Theorem 5 ([4]). *Let* ϕ *be an* $\mathrm{FO}(\perp_c)$*-formula. Then for all* $\mathfrak{A}$ *and* X *it holds that*

$$\mathfrak{A} \models_X \phi \Leftrightarrow \mathfrak{A} \models_{X \upharpoonright Fr(\phi)} \phi.$$

Both dependence and independence logic are equi-expressive with existential second-order logic and are hence both non-axiomatizable. On the other hand, by restricting attention to syntactic fragments of these logics, complete axiomatization is possible [10,15].

There is an intimate connection between independence atoms of $\mathrm{FO}(\perp_c)$ and CI statements. Teams and independence atoms can be seen as qualitative analogues of probability distributions and their CI statements (so-called relational dependency models discussed, e.g., in [7]). This connection can be made explicit as follows (see [3,7]). For a set of stochastic variables X_A, $A \subseteq \Delta$, we write x_A for a tuple (in any order) consisting of first-order variables x_i for $i \in A$.

Proposition 1. *Let* P_Δ *be a distribution. Define a team* X *consisting of those assignments* $s\colon \{x_1, ..., x_n\} \to \cup_{1 \leq i \leq n} \mathcal{X}_i$ *such that*

$$X = \{s \mid P_\Delta(X_1 = s(x_1), ..., X_n = s(x_n)) > 0\}.$$

Then if P *satisfies a CI* $X_A \perp X_B | X_C$*, then* $\mathfrak{A} \models_X x_A \perp_{x_C} x_B$*, where* $\mathfrak{A} = \cup_{1 \leq i \leq n} \mathcal{X}_i$.

Proof. Let $s, s' \in X$ be such that $s(x_C) = s'(x_C)$. Now we must have $P_\Delta(X_A = s(x_A) | X_C = s(x_C)) > 0$ and $P_\Delta(X_B = s'(x_B) | X_C = s(x_C)) > 0$. Denote these non-negative probabilities by c_1 and c_2, respectively. Since $X_A \perp X_B | X_C$ holds,

$$P_\Delta(X_A = s(x_A), X_B = s'(x_B) | X_C = s(x_C)) = c_1 c_2 > 0,$$

hence it follows that there exists $s^* \in X$ such that $s^*(x_A) = s(x_A)$, $s^*(x_B) = s'(x_B)$, and $s^*(x_C) = s(x_C)$ as wanted.

3 A Logic for Expressing Context-Specific Independence

In this section we formulate a variant of independence logic that is suitable to express CSI statements and study its properties.

Definition 10 (CSI-atom). *A context-specific independence atom (CSI-atom) is a formula of the form*

$$\boldsymbol{x} \perp_{\phi(\boldsymbol{v}),\boldsymbol{u}} \boldsymbol{y},$$

where ϕ is an FO*-formula. Satisfaction for CSI-atoms is defined as follows: $\mathfrak{A} \models_X \boldsymbol{x} \perp_{\boldsymbol{u},\phi(\boldsymbol{v})} \boldsymbol{y}$ iff for all $s, s' \in X$ such that $s(\boldsymbol{uv}) = s'(\boldsymbol{uv})$ and $\mathfrak{A} \models_s \phi(\boldsymbol{v})$ there is $s^* \in X$ such that $s^*(\boldsymbol{xuv}) = s(\boldsymbol{xuv})$, and $s^*(\boldsymbol{y}) = s'(\boldsymbol{y})$.*

Definition 11. *The extension of* FO *by CSI-atoms is denoted by* $\mathrm{FO}(\perp_{\mathrm{CSI}})$.

The following observations are straightforward to prove. First of all, the locality property holds also for $\mathrm{FO}(\perp_{\mathrm{CSI}})$.

Theorem 6. *Let ϕ be an $\mathrm{FO}(\perp_{\mathrm{CSI}})$-formula. Then for all $\mathfrak{A}$ and X it holds that*

$$\mathfrak{A} \models_X \phi \Leftrightarrow \mathfrak{A} \models_{X \upharpoonright Fr(\phi)} \phi.$$

The next lemma shows that CSI-atoms can be expressed as $\mathrm{FO}(\perp_c)$-formulas.

Lemma 1. *Let $\phi \in$ FO, and let ϕ^d denote the dual of ϕ. Then the formula $\boldsymbol{x} \perp_{\phi(\boldsymbol{v}),\boldsymbol{u}} \boldsymbol{y}$ is logically equivalent to the independence logic formula*

$$\phi^d \vee (\phi \wedge \boldsymbol{x} \perp_{\boldsymbol{uv}} \boldsymbol{y}).$$

Proof. Let $\mathfrak{A}$ be a structure and X a team such that

$$\mathfrak{A} \models_X \boldsymbol{x} \perp_{\phi(\boldsymbol{v}),\boldsymbol{u}} \boldsymbol{y}. \tag{5}$$

Let $Y_1 = \{s \in X \mid \mathfrak{A} \models_s \phi\}$ and $Y_2 = \{s \in X \mid \mathfrak{A} \models_s \phi^d\}$. Now since $\phi \in$ FO it holds that $X = Y_1 \cup Y_2$ and $Y_1 \cap Y_2 = \emptyset$. Furthermore, by Theorem 4 it holds that $\mathfrak{A} \models_{Y_2} \phi^d$, and $\mathfrak{A} \models_{Y_1} \phi$. In order to show $\mathfrak{A} \models_{Y_1} \boldsymbol{x} \perp_{\boldsymbol{uv}} \boldsymbol{y}$, let $s, s' \in Y_1$ such that $s(\boldsymbol{uv}) = s'(\boldsymbol{uv})$. Since $s \in Y_1$, $\mathfrak{A} \models_s \phi(\boldsymbol{v})$ holds and hence by (5) there exists $s^* \in X$ such that $s^*(\boldsymbol{xuv}) = s(\boldsymbol{xuv})$, and $s'(\boldsymbol{y}) = s^*(\boldsymbol{y})$. Since $s^* \in Y_1$, we get $\mathfrak{A} \models_{Y_1} \boldsymbol{x} \perp_{\boldsymbol{uv}} \boldsymbol{y}$, and

$$\mathfrak{A} \models_X \phi^d \vee (\phi \wedge \boldsymbol{x} \perp_{\boldsymbol{uv}} \boldsymbol{y}).$$

The converse implication is proved analogously.

Lemma 1 implies that independence logic and $\mathrm{FO}(\perp_{\mathrm{CSI}})$ are equi-expressive.

Theorem 7. $\mathrm{FO}(\perp_{\mathrm{CSI}}) \equiv \mathrm{FO}(\perp_c)$.

Analogously to CIs and independence atoms, CSI-atoms can be viewed as qualitative analogues of CSI statements.

Proposition 2. *Let P_Δ be a distribution. Define a team X consisting those assignments $s\colon \{x_1, ..., x_n\} \to \cup_{1\le i\le n}\mathcal{X}_i$ such that*

$$X = \{s \mid P_\Delta(X_1 = s(x_1), ..., X_n = s(x_n)) > 0\}.$$

If P_Δ satisfies the CSI

$$X_A \perp X_B | X_C = e_C, X_S,$$

then $\mathfrak{A} \models_X x_A \perp_{\phi(x_C),x_S} x_B$, where $\mathfrak{A} = \cup_{1\le i\le n}\mathcal{X}_i$ and $\phi(x_C) = \wedge_{i\in C}(x_i = e_i)$ (e_i appears as a constant symbol in $\phi(x_C)$).

Proof. Analogous to the proof of Proposition 1.

4 Axiomatic Characterization of CSI-separation

In this section we define a sound extension of the semi-graphoid axioms that capture CSI-separation in LDAGs. We begin be recalling the semi-graphoid axioms [2,21] and their relation to d-separation in the context of Bayesian networks.

4.1 The Semi-graphoid Axioms and the Implication Problem of CI Statements

In the following we assume without loss of generality that the natural ordering of the nodes Δ of a DAG G agrees with the edge relation of G. Furthermore, we redefine $I_{Loc}(G)$ as follows:

$$I_{Loc}(G) = \{X_i \perp \{X_1, ..., X_{i-1}\} \setminus X_{\Pi_i} | X_{\Pi_i} : i \in \Delta\}. \quad (6)$$

Note that $\{1, ..., i-1\}$ are non-descendants of the node i in G. The equivalence of the above definition of $I_{Loc}(G)$ with the previous one follows by Theorem 8.

Definition 12 (Semi-graphoid axioms). *The following axioms are called the semi-graphoid axioms. Below X, Y, and Z denote sets of stochastic variables. The union of X and Y is denoted by XY.*

1. **Triviality**: $X \perp \emptyset | Z$,
2. **Symmetry**: $X \perp Y | Z \Rightarrow Y \perp X | Z$,
3. **Decomposition**: $X \perp YU | Z \Rightarrow X \perp Y | Z$,
4. **Weak Union**: $X \perp YU | Z \Rightarrow X \perp Y | ZU$
5. **Contraction**: $X \perp Y | ZU$ *and* $X \perp U | Z \Rightarrow X \perp YU | Z$.

The semi-graphoid axioms are known to be sound for all distributions. The following theorem shows that these axioms correspond exactly to d-separation in the context of Bayesian networks. For a finite set $\Sigma \cup \{\phi\}$ of CIs, we write $\Sigma \vdash^{sg} \phi$ with the meaning that ϕ can be derived from Σ using the semi-graphoid axioms. In other words, there exists a finite sequence $\psi_1, ..., \psi_k$ such that $\psi_k = \phi$, and $\psi_i \in \Sigma$ or ψ_i is obtained by applying one of the semi-graphoid axioms to ψ_l and ψ_t for some $l, t < i$.

Theorem 8. *[6, 25] Let $G = (\Delta, V)$ be a DAG, and let A, B, and C be disjoint subsets of Δ. Then A is d-separated from B by C if and only if $I_{Loc}(G) \vdash^{sg} X_A \perp X_B | X_C$.*

Theorems 1 and 8 together imply the following result.

Theorem 9. *[6, 25] Let $G = (\Delta, V)$ be a DAG, and let A, B, and C be disjoint subsets of Δ. Then $I_{Loc}(G) \models X_A \perp X_B | X_C$ if and only if $I_{Loc}(G) \vdash^{sg} X_A \perp X_B | X_C$.*

This result can be viewed as a complete axiomatization of a restricted version of the *implication problem* of CI statements. The implication problem of CI statements is defined as follows. Given a finite collection $\Sigma \cup \{\varphi\}$ of CI statements as input, determine whether for all P,

$$P \models \Sigma \Rightarrow P \models \varphi.$$

This problem is known not to be finitely axiomatizable [23]. Despite of this negative result, the semi-graphoid axioms are also relevant for the general implication problem of conditional independence. For example, in [9] it was shown that the axioms are complete for the implication problem of conditional independence assuming Σ consists solely of so-called *saturated* CIs. Furthermore, in [19] the semi-graphoid axioms and a certain other set of axioms are used to approximate the CI implication problem.

Independence atoms correspond to so-called embedded multivalued dependencies (EMVD) in database theory whose connections to CIs has been widely studied [16–18, 26]. In particular, the semi-graphoid axioms are sound also for EMVDs and independence atoms. For a finite set $\Sigma \cup \{\phi\}$ of independence atoms, we write $\Sigma \models \phi$ with the meaning that for all finite $\mathfrak{A}$ and X, if $\mathfrak{A} \models_X \psi$ for all $\psi \in \Sigma$, then $\mathfrak{A} \models_X \phi$. For a set of CIs Σ, Σ^* denotes the corresponding set of independence atoms. The following result is an immediate consequence of the results in [7].

Theorem 10. *Let G be a DAG, and let A, B, and C be disjoint subsets of Δ. Then the following are equivalent:*

1. $I_{Loc}(G) \models X_A \perp X_B | X_C$,
2. $I_{Loc}(G)^* \models x_A \perp_{x_C} x_B$,
3. $I_{Loc}(G) \vdash^{sg} X_A \perp X_B | X_C$.

Theorem 10 does not hold for the general implication problems of independence atoms and CIs [23]. Furthermore, the implication problem of independence atoms is know to be undecidable by the result of [13], whereas for CIs the decidabilty of the problem is still open. It is worth noting that a version of Theorem 10 holds for marginal CIs and independence atoms of the form $\boldsymbol{x} \perp \boldsymbol{y}$. Furthermore, implication problem of marginal CIs has a complete axiomatization in terms of axioms similar to the semi-graphoid axioms [5].

4.2 Axioms for Context Specific Independence

In this section we give an axiomatic characterization of CSI-separation in LDAGs. For the derivation of CSI statements, we introduce the following rule closely resembling the definition of CSI-separation. In order to apply the CSI-rule below, an input LDAG G_L as well as the outcome spaces $\mathcal{X}_i$ of the variables X_i have to be fixed. The extra assumptions $A(e_C)$ allowed in the subderivation of the CSI-rule:

$$A(e_c) = \{X_j \perp X_{j_1}, ..., X_{j_k} | X_{\Pi_j - \{j_1,...,j_k\}} \ : j \in \{1, ..., n\}\}$$

encode the information that the graph $G(e_C)$ arises from G by removing edges $(j_1, j), ..., (j_k, j)$, for $j \in \Delta$.

Definition 13 (CSI-rule for context $X_C = e_c$).

$$\left[I_{loc}(G) \cup A(e_C) \vdash^{sg} X_A \perp X_B | X_{C \cup S}\right] \quad \Rightarrow \quad X_A \perp X_B | X_C = e_C, X_S$$

The idea of the CSI-rule is that the existence of the derivation on the left-hand side (corresponding to d-separation in $G(e_C)$) justifies the conclusion on the right. The auxiliary assumptions $A(e_C)$ can only be used in the subderivation.

We will next show that the CSI-rule is sound and corresponds to CSI-separation in LDAGs. For an LDAG G_L, we write $I_{loc}(G_L) \vdash^{sg+} \phi$ with the meaning that a CI (or CSI) statement ϕ can be derived from $I_{loc}(G_L)$ using the semi-graphoid axioms and the CSI-rule.

Theorem 11. *Let $G_L = (\Delta, V, \mathcal{L}_E)$ be an LDAG, and let A, B, C, and S be disjoint subsets of Δ. Then*

$$I_{loc}(G_L) \vdash^{sg+} X_A \perp X_B | X_C = e_c, X_S$$

if and only if X_A is CSI-separated from X_B by X_S in the context $X_C = e_C$.

Proof. Recall our assumption that $I_{loc}(G)$ (and $I_{loc}(G(e_C))$) is encoded by a set of CIs of the form (6). By Theorem 8 it suffices to show that the sets $I_{loc}(G) \cup A(e_C)$ and $I_{loc}(G(e_C))$ are equivalent with respect to deductions by the semi-graphoid axioms.

Note first that $I_{loc}(G) \cup A(e_c) \vdash^{sg} \psi$, for all $\psi \in I_{loc}(G(e_C))$. This holds since $I_{loc}(G(e_C))$ consists of CIs of the form

$$X_j \perp \{X_1, ..., X_{j-1}\} \setminus X_A | X_A, \tag{7}$$

where $A = \Pi_j - \{j_1, ..., j_k\}$, and the CI in equation (7) can be derived by one application of the contraction rule applied to the CIs $X_j \perp X_{j_1}, ..., X_{j_k} | X_{\Pi_j - \{j_1,...,j_k\}}$ and $X_j \perp \{X_1, ..., X_{j-1}\} \setminus X_{\Pi_j} | X_{\Pi_j}$.

Let us then show $I_{loc}(G(e_C)) \vdash^{sg} \psi$ for all $\psi \in L_{loc}(G) \cup A(e_c)$. Assume ψ is of the form $X_j \perp X_{j_1}, ..., X_{j_k} | X_{\Pi_j - \{j_1,...,j_k\}}$. Then it can be derived by one application of the decomposition rule applied to the corresponding CI in (7). Analogously, if ψ is of the form $X_j \perp \{X_1, ..., X_{j-1}\} \setminus X_{\Pi_j} | X_{\Pi_j}$, then one application of weak union rule applied to (7) suffices.

As discussed in Example 3, CSI-separation, and hence our axioms fail to capture some non-local CIs implied by the structure of the LDAG. The following RC-rule is obviously sound for LDAGs and it addresses the problem discussed in Example 3.

Definition 14 (RC-rule).

$$X_A \perp X_B | X_C = e_C, X_S \text{ for all } e_C \in \mathcal{X}_C \quad \Rightarrow \quad X_A \perp X_B | X_{C \cup S}$$

It is worth noting that the size of the assumptions needed to apply the RC rule grows exponentially in the number of variables X_C. On the other hand, the question whether a CI holds in an LDAG is in general coNP-hard (see Theorem 3) hence the exponential blow-up might not be avoidable.

We conjecture that the following holds. Note that the implication (1) $\Rightarrow$ (2) holds by Theorem 11 and the obvious soundness of the RC-rule.

Conjecture 12. *Let $G_L = (\Delta, V, \mathcal{L}_E)$ be an LDAG, and let A, B, C, and S be disjoint subsets of Δ. Then the following are equivalent:*

1. *A CSI $X_A \perp X_B | X_C = e_c, X_S$ (CI $X_A \perp X_B | X_C$) can be derived from $I_{loc}(G_L)$ using the semi-graphoid axioms and the CSI and RC rules,*
2. *Every distribution P_Δ satisfying $I_{loc}(G_L)$ also satisfies $X_A \perp X_B | X_C = e_c, X_S$ ($X_A \perp X_B | X_C$).*

In Example 3 we showed how two labels (or CSIs) within a local structure together can imply a CI statement which cannot be inferred from the underlying graph. In addition to this type of scenario, there can also arise situations where a combination of labels from different local structures induces a non-local independence statement that holds globally. This is possible when a variable (or set of variables) specifies labels in several local structures.

Example 4. Consider the LDAG in Fig. 4. From an inspection using d-separation on the underlying DAG, it appears that $X_2 \not\perp X_4$ since there is a seemingly open trail between node 2 and node 4 via node 1. However, by combining several independence statements we can conclude that $X_2 \perp X_4$ actually holds. First, using CSI-separation it is straightforward to conclude that $X_2 \perp X_4 \mid X_3 = 0$ and $X_2 \perp X_4 \mid X_3 = 1$, which according to the RC-rule imply that $X_2 \perp$

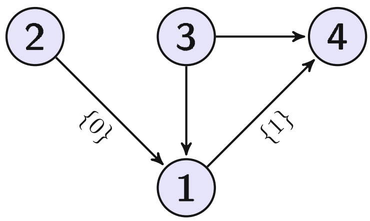


Fig. 4. LDAG over four binary variables.

$X_4 \mid X_3$. In addition, the local directed Markov property states that $X_2 \perp X_3$. By combining the two CI statements we can conclude that $X_2 \perp \{X_3, X_4\}$ (Contraction) and finally $X_2 \perp X_4$ (Decomposition). Interestingly, the graphs in Figs. 3 and 4 actually represent the same dependence structures although the underlying DAGs belong to different Markov equivalence classes (see [22] for more details).

5 Conclusion

In this article we have discussed and studied logical analogues of conditional independence and context-specific independence statements. We defined a novel version $\mathrm{FO}(\perp_{\mathrm{CSI}})$ of dependence logic suitable for formalizing qualitative versions of CSI statements, and also extended the well-known semi-graphoid axioms to logically characterize CSI-separation in LDAGs. An important question left open is whether our axioms are strong enough to characterize all independence statements implied by an LDAG structure (see Conjecture 12). However, the more complete set of implied independencies are already useful in various probabilistic inference applications.

The logic $\mathrm{FO}(\perp_{\mathrm{CSI}})$ defined in this article allows the formulation of various generalized notions of independence one particular instance being CSI atoms. Team semantics has already been successfully used to axiomatize dependencies in the database theory framework [11,12]. It is an interesting open question to formulate general axioms for the logic $\mathrm{FO}(\perp_{\mathrm{CSI}})$. In particular, it is an interesting task to identify subclasses of CSI-atoms for which the implication problem is axiomatizable or decidable.

Acknowledgements. The third author was supported by grant 292767 of the Academy of Finland. The fourth author was supported by FDPSS via grant 141318 of the Academy of Finland.

Appendix

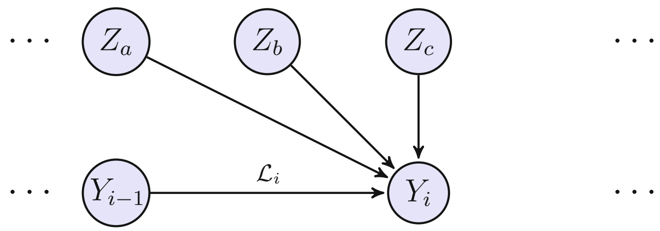


Fig. 5. LDAG for the proof of Theorem 3.

Proof of Theorem 3. We apply the proof suggested by Koller et al. (p. 196) to LDAG structures [14]. We will reduce a 3-SAT problem instance into deciding whether a CI statement is implied by an LDAG structure.

Define the corresponding LDAG to the 3-SAT instance as follows (see Fig. 5). Let binary nodes $Z_1, \cdots, Z_l$ correspond to variables in the 3-SAT instance. Let $Y_0, Y_1, \cdots, Y_k$ denote additional binary nodes of which $Y_1, \cdots, Y_k$ represent the clauses of the 3-SAT instance. Let the parents of node Y_i $(i \geq 1)$ be the node Y_{i-1}, and the Z-nodes appearing in the clause i, let us call them Z_a, Z_b, Z_c. The labels on the edge $Y_{i-1} \to Y_i$ consist of assignments to the nodes Z_a, Z_b, Z_c. Let the label $\mathcal{L}_i$ on the arc $Y_{i-1} \to Y_i$ be exactly the set of assignments to Z_a, Z_b, Z_c that *do not* satisfy the ith clause of the 3-SAT problem.

Consider different contexts e_z over variables $Z_1, \cdots, Z_l$. If e_z does not satisfy the 3-SAT instance, there is a clause i which is unsatisfied, and thus the corresponding edge $Y_{i-1} \to Y_i$ does not appear in $G(e_z)$. Thus, Y_0 and Y_k are d-separated in $G(e_z)$ and according to Theorem 2: $Y_0 \perp Y_k | Z_1, \ldots, Z_l = e_z$.

If e_z satisfies the 3-SAT instance, all clauses are satisfied and thus all edges $Y_{i-1} \to Y_i$ appear in $G(e_z)$. Thus, Y_0 and Y_k are not d-separated in $G(e_z)$. We can define a parameterization for the LDAG under which there is a dependence. Let $Y_0, Z_1, \ldots, Z_l$ be distributed uniformly. Let $Y_i = Y_{i-1}$ if Z_a, Z_b, Z_c satisfy the clause i and 0 otherwise. Now under a satisfying context e_z: $Y_k = Y_{k-1} = \cdots = Y_0$ hence $Y_0 \not\perp Y_k | Z_1, \ldots, Z_k = e_z$. Thus, $Y_0 \perp Y_k | Z_1, \ldots, Z_k = e_z$ cannot follow from the LDAG structure.

If the 3-SAT problem is satisfiable there is a context e_z such that $Y_0 \perp Y_k | Z_1, \ldots, Z_k = e_z$ does not follow from the LDAG structure, hence $Y_0 \perp Y_k | Z_1, \ldots, Z_k$ does not follow from the structure either. If the 3-SAT problem is unsatisfiable we have that for all contexts e_z: $Y_0 \perp Y_k | Z_1, \ldots, Z_k = e_z$, from which it directly follows that $Y_0 \perp Y_k | Z_1, \ldots, Z_k$. Thus, the defined LDAG structure implies independence $Y_0 \perp Y_k | Z_1, \ldots, Z_k$ if and only if the 3-SAT problem is unsatisfiable. If we could decide whether an independence is implied by an LDAG in polynomial time, we could also solve 3-SAT in polynomial time.

References

1. Boutilier, C., Friedman, N., Goldszmidt, M., Koller, D.: Context-specific independence in Bayesian networks. In: Proceedings of the Twelfth International Conference on Uncertainty in Artificial Intelligence. UAI 1996, pp. 115–123. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1996). http://dl.acm.org/citation.cfm?id=2074284.2074298
2. Dawid, A.P.: Conditional independence in statistical theory. J. Roy. Stat. Soc. Ser. B (Methodological) **41**(1), 1–31 (1979). doi:10.2307/2984718
3. Durand, A., Hannula, M., Kontinen, J., Meier, A., Virtema, J.: Approximation and dependence via multiteam semantics. In: Gyssens, M., et al. (eds.) FoIKS 2016. LNCS, vol. 9616, pp. 271–291. Springer, Heidelberg (2016). doi:10.1007/978-3-319-30024-5_15
4. Galliani, P.: Inclusion and exclusion dependencies in team semantics - on some logics of imperfect information. Ann. Pure Appl. Logic **163**(1), 68–84 (2012)

5. Geiger, D., Paz, A., Pearl, J.: Axioms and algorithms for inferences involving probabilistic independence. Inf. Comput. **91**(1), 128–141 (1991)
6. Geiger, D., Pearl, J.: On the logic of causal models. In: Proceedings of the Fourth Annual Conference on Uncertainty in Artificial Intelligence. UAI 1988, pp. 3–14. North-Holland Publishing Co., Amsterdam, The Netherlands (1990). http://dl.acm.org/citation.cfm?id=647231.719429
7. Geiger, D., Verma, T., Pearl, J.: Identifying independence in Bayesian networks. Networks **20**(5), 507–534 (1990)
8. Grädel, E., Väänänen, J.A.: Dependence and independence. Stud. Logica **101**(2), 399–410 (2013)
9. Gyssens, M., Niepert, M., Gucht, D.V.: On the completeness of the semigraphoid axioms for deriving arbitrary from saturated conditional independence statements. Inf. Process. Lett. **114**(11), 628–633 (2014). http://www.sciencedirect.com/science/article/pii/S0020019014001057
10. Hannula, M.: Axiomatizing first-order consequences in independence logic. Ann. Pure Appl. Logic **166**(1), 61–91 (2015). doi:10.1016/j.apal.2014.09.002
11. Hannula, M.: Reasoning about embedded dependencies using inclusion dependencies. In: Davis, M., Fehnker, A., McIver, A., Voronkov, A. (eds.) LPAR-20 2015. LNCS, vol. 9450, pp. 16–30. Springer, Heidelberg (2015). doi:10.1007/978-3-662-48899-7_2
12. Hannula, M., Kontinen, J.: A finite axiomatization of conditional independence and inclusion dependencies. In: Beierle, C., Meghini, C. (eds.) FoIKS 2014. LNCS, vol. 8367, pp. 211–229. Springer, Heidelberg (2014)
13. Herrmann, C.: On the undecidability of implications between embedded multivalued database dependencies. Inf. Comput. **122**(2), 221–235 (1995)
14. Koller, D., Friedman, N.: Probabilistic graphical models: principles and techniques. MIT Press, Cambridge (2009)
15. Kontinen, J., Väänänen, J.A.: Axiomatizing first-order consequences in dependence logic. Ann. Pure Appl. Logic **164**(11), 1101–1117 (2013)
16. Link, S.: Reasoning about saturated conditional independence under uncertainty: axioms, algorithms, and levesque's situations to the rescue. In: Proceedings of AAAI. AAAI Press (2013)
17. Link, S.: Sound approximate reasoning about saturated conditional probabilistic independence under controlled uncertainty. J. Appl. Logic **11**(3), 309–327 (2013)
18. Link, S.: Frontiers for propositional reasoning about fragments of probabilistic conditional independence and hierarchical database decompositions. Theor. Comput. Sci. **603**, 111–131 (2015)
19. Niepert, M., Gyssens, M., Sayrafi, B., Gucht, D.V.: On the conditional independence implication problem: a lattice-theoretic approach. Artif. Intell. **202**, 29–51 (2013). doi:10.1016/j.artint.2013.06.005
20. Nyman, H., Pensar, J., Corander, J.: Context-specific and local independence in Markovian dependence structures. In: Dependence Logic: Theory and Applications. Springer (To appear) (2016)
21. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Francisco (1988)
22. Pensar, J., Nyman, H.J., Koski, T., Corander, J.: Labeled directed acyclic graphs: a generalization of context-specific independence in directed graphical models. Data Min. Knowl. Discov. **29**(2), 503–533 (2015). doi:10.1007/s10618-014-0355-0

23. Studeny, M.: Conditional independence relations have no finite complete characterization. In: Kubik, S., Visek, J. (eds.) Transactions of the 11th Prague Conference. Information Theory, Statistical Decision Functions and Random Processes, vol. B, pp. 377–396. Kluwer, Dordrecht (1992)
24. Väänänen, J.: Dependence logic: A New Approach to Independence Friendly Logic, London Mathematical Society Student Texts, vol. 70. Cambridge University Press, Cambridge (2007)
25. Verma, T., Pearl, J.: Causal networks: semantics and expressiveness. In: Shachter, R.D., Levitt, T.S., Kanal, L.N., Lemmer, J.F. (eds.) Proceedings of the Fourth Annual Conference on Uncertainty in Artificial Intelligence, Minneapolis, MN, USA, 10–12 July 1988. UAI 1988, pp. 69–78. North-Holland (1988)
26. Wong, S., Butz, C., Wu, D.: On the implication problem for probabilistic conditional independency. IEEE Trans. Syst. Man Cybern. Part A: Syst. Hum. **30**(6), 785–805 (2000)

Descriptive Complexity of Graph Spectra

Anuj Dawar[1], Simone Severini[2], and Octavio Zapata[2(✉)]

[1] University of Cambridge Computer Laboratory, Cambridge, UK
[2] Department of Computer Science, University College London, London, UK
ocbzapata@gmail.com

Abstract. Two graphs are co-spectral if their respective adjacency matrices have the same multi-set of eigenvalues. A graph is said to be determined by its spectrum if all graphs that are co-spectral with it are isomorphic to it. We consider these properties in relation to logical definability. We show that any pair of graphs that are elementarily equivalent with respect to the three-variable counting first-order logic C^3 are co-spectral, and this is not the case with C^2, nor with any number of variables if we exclude counting quantifiers. We also show that the class of graphs that are determined by their spectra is definable in partial fixed-point logic with counting. We relate these properties to other algebraic and combinatorial problems.

Keywords: Descriptive complexity · Algebraic graph theory · Isomorphism approximations

1 Introduction

The spectrum of a graph G is the multi-set of eigenvalues of its adjacency matrix. Even though it is defined in terms of the adjacency matrix of G, the spectrum does not, in fact, depend on the order in which the vertices of G are listed. In other words, isomorphic graphs have the same spectrum. The converse is false: two graphs may have the same spectrum without being isomorphic. Say that two graphs are co-spectral if they have the same spectrum. Our aim in this paper is to study the relationship of this equivalence relation on graphs in relation to a number of other approximations of isomorphism coming from logic, combinatorics and algebra. We also investigate the definability of co-spectrality and related notions in logic.

Specifically, we show that for any graph G, we can construct a formula ϕ_G of first-order logic with counting, using only three variables (i.e. the logic C^3) so that $H \models \phi_G$ only if H is co-spectral with G. From this, it follows that elementary equivalence in C^3 refines co-spectrality, a result that also follows from [1]. In contrast, we show that co-spectrality is incomparable with elementary equivalence in C^2, or with elementary equivalence in L^k (first-order logic

We thank Aida Abiad, Chris Godsil, Robin Hirsch and David Roberson for fruitful discussions. This work was supported by CONACyT, EPSRC and The Royal Society.

J. Väänänen et al. (Eds.): WoLLIC 2016, LNCS 9803, pp. 183–199, 2016.
DOI: 10.1007/978-3-662-52921-8_12

with k variables but without counting quantifiers) for any k. We show that on strongly regular graphs, co-spectrality exactly co-incides with C^3-equivalence.

For definability results, we show that co-spectrality of a pair of graphs is definable in FPC, inflationary fixed-point logic with counting. We also consider the property of a graph G to be *determined by its spectrum*, meaning that all graphs co-spectral with G are isomorphic with G. We establish that this property is definable in *partial fixed-point logic with counting* (PFPC).

In Sect. 2, we construct some basic first-order formulas that we use to prove various results later, and we also review some well-known facts in the study of graph spectra. In Sect. 3, we make explicit the connection between the spectrum of a graph and the total number of closed walks on it. Then we discuss aspects of the class of graphs that are uniquely determined by their spectra, and establish that co-spectrality on the class of all graphs is refined by C^3-equivalence. Also, we show a lower bound for the distinguishability of graph spectra in the finite-variable logic. In Sect. 4, we give an overview of a combinatorial algorithm (named after Weisfeiler and Leman) for distinguishing between non-isomorphic graphs, and study the relationship with other algorithms of algebraic and combinatorial nature. Finally, in Sect. 5, we establish some results about the logical definability of co-spectrality and of the property of being a graph determined by its spectrum.

2 Preliminaries

Consider a first-order language $L = \{E\}$, where E is a binary relation symbol interpreted as an irreflexive symmetric binary relation called *adjacency*. Then an L-structure $G = (V_G, E_G)$ is called a *simple undirected graph*. The domain V_G of G is called the *vertex set* and its elements are called *vertices*. The unordered pairs of vertices in the interpretation E_G of E are called *edges*. Formally, a *graph* is an element of the elementary class axiomatised by the first-order L-sentence: $\forall x \forall y(\neg E(x, x) \wedge (E(x, y) \rightarrow E(y, x)))$.

The *adjacency matrix* of an n-vertex graph G with vertices $v_1, \ldots, v_n$ is the $n \times n$ matrix A_G with $(A_G)_{ij} = 1$ if vertex v_i is adjacent to vertex v_j, and $(A_G)_{ij} = 0$ otherwise. By definition, every adjacency matrix is real and symmetric with diagonal elements all equal to zero. A *permutation matrix* P is a binary matrix with a unique 1 in each row and column. Permutation matrices are orthogonal matrices so the inverse P^{-1} of P is equal to its transpose P^T. Two graphs G and H are *isomorphic* if there is a bijection h from V_G to V_H that preserves adjacency. The existence of such a map is denoted by $G \cong H$. From this definition it is not difficult to see that two graphs G and H are isomorphic if, and only if, there exists a permutation matrix P such that $A_G P = P A_H$.

The *characteristic polynomial* of an n-vertex graph G is a polynomial in a single variable λ defined as $p_G(\lambda) := det(\lambda I - A_G)$, where $det(\cdot)$ is the operation of computing the determinant of the matrix inside the parentheses, and I is the identity matrix of the same order as A_G. The *spectrum* of G is the multi-set $\mathrm{sp}(G) := \{\lambda : p_G(\lambda) = 0\}$, where each root of $p_G(\lambda)$ is considered according to its multiplicity. If $\theta \in \mathrm{sp}(G)$ then $\theta I - A_G$ is not invertible, and so there exists a nonzero vector u such that $A_G u = \theta u$. A vector like u is called an *eigenvector* of G corresponding to θ. The elements in $\mathrm{sp}(G)$ are called the *eigenvalues* of G. Two graphs are called *co-spectral* if they have the same spectrum.

The *trace* of a matrix is the sum of all its diagonal elements. By the definition of matrix multiplication, for any two matrices A, B we have $tr(AB) = tr(BA)$, where $tr(\cdot)$ is the operation of computing the trace of the matrix inside the parentheses. Therefore, if G and H are two isomorphic graphs then $tr(A_H) = tr(P^T A_G P) = tr(A_G P P^T) = tr(A_G)$ and so, $tr(A_G^k) = tr(A_H^k)$ for any $k \geq 0$.

By the spectral decomposition theorem, computing the trace of the k-th powers of a real symmetric matrix A will give the sum of the k-th powers of the eigenvalues of A. Assuming that A is an $n \times n$ matrix with (possibly repeated) eigenvalues $\lambda_1, \ldots, \lambda_n$, the *elementary symmetric polynomials* e_k in the eigenvalues are the sum of all distinct products of k distinct eigenvalues:

$$\begin{aligned} &e_0(\lambda_1, \ldots, \lambda_n) := 1; \qquad e_1(\lambda_1, \ldots, \lambda_n) := \textstyle\sum_{i=1}^n \lambda_i; \\ &e_k(\lambda_1, \ldots, \lambda_n) := \textstyle\sum_{1 \leq i_1 < \cdots < i_k \leq n} \lambda_{i_1} \cdots \lambda_{i_k} \quad \text{for } 1 \leq k \leq n. \end{aligned}$$

This expressions are the coefficients of the characteristic polynomial of A modulo a 1 or -1 factor. That is,

$$\begin{aligned} det(\lambda I - A) &= \prod_{i=1}^n (\lambda - \lambda_i) \\ &= \lambda^n - e_1(\lambda_1, \ldots, \lambda_n)\lambda^{n-1} + \cdots + (-1)^n e_n(\lambda_1, \ldots, \lambda_n) \\ &= \sum_{k=0}^n (-1)^{n+k} e_{n-k}(\lambda_1, \ldots, \lambda_n)\lambda^k. \end{aligned}$$

So if we know $s_k(\lambda_1, \ldots, \lambda_n) := \sum_{i=1}^n \lambda_i^k$ for $k = 1, \ldots, n$, then using *Newton's identities*:

$$e_k(\lambda_1, \ldots, \lambda_n) = \frac{1}{k} \sum_{j=1}^k (-1)^{j-1} e_{k-j}(\lambda_1, \ldots, \lambda_n) s_k(\lambda_1, \ldots, \lambda_n) \ \text{ for } 1 \leq k \leq n,$$

we can obtain all the symmetric polynomials in the eigenvalues, and so we can reconstruct the characteristic polynomial of A.

Proposition 1. *For n-vertex graphs G and H, the following are equivalent:*

(1) G and H are co-spectral;
(2) G and H have the same characteristic polynomial;
(3) $tr(A_G^k) = tr(A_H^k)$ for $1 \leq k \leq n$.

3 Spectra and Walks

Given a graph G, a *walk of length l* in G is a sequence $(v_0, v_1, \ldots, v_l)$ of vertices of G, such that consecutive vertices are adjacent in G. Formally, $(v_0, v_1, \ldots, v_l)$

is a walk of length l in G if, and only if, $\{v_{i-1}, v_i\} \in E_G$ for $1 \leq i \leq l$. We say that the walk $(v_0, v_1, \ldots, v_l)$ *starts* at v_0 and *ends* at v_l. A walk of length l is said to be *closed* (or *l-closed*, for short) if it starts and ends in the same vertex.

Since the ij-th entry of A_G^l is precisely the number of walks of length l in G starting at v_i and ending at v_j, by Proposition 1, we have that the spectrum of G is completely determined if we know the total number of closed walks for each length up to the number of vertices in G. Thus, two graphs G and H are co-spectral if, and only if, the total number of l-closed walks in G is equal to the total number of l-closed walks in H for all $l \geq 0$.

For an example of co-spectral non-isomorphic graphs, let $G = K_4 \cup K_1$ and $H = K_{1,4}$, where K_n is the complete n-vertex graph, $K_{n,m}$ the complete $(n+m)$-vertex bipartite graph, and "$\cup$" denotes the disjoint union of two graphs. The spectrum of both G and H is the multi-set $\{-2, 0, 0, 0, 2\}$. However, G contains an isolated vertex while H is a connected graph.

3.1 Finite Variable Logics with Counting

For each positive integer k, let C^k denote the fragment of first-order logic in which only k distinct variables can be used but we allow *counting quantifiers*: so for each $i \geq 1$ we have a quantifier $\exists^i$ whose semantics is defined so that $\exists^i x\phi$ is true in a structure if there are at least i distinct elements that can be substituted for x to make ϕ true. We use the abbreviation $\exists^{=i} x\phi$ for the formula $\exists^i x\phi \wedge \neg\exists^{i+1} x\phi$ that asserts the existence of exactly i elements satisfying ϕ. We write $G \equiv_C^k H$ to denote that the graphs G and H are not distinguished by any formula of C^k. Note that C^k-*equivalence* is the usual first-order elementary equivalence relation restricted to formulas using at most k distinct variables and possibly using counting quantifiers.

We show that for integers k, l, with $k \geq 0$ and $l \geq 1$, there is a formula $\psi_k^l(x, y)$ of C^3 so that for any graph G and vertices $v, u \in V_G$, $G \models \psi_k^l[v, u]$ if, and only if, there are exactly k walks of length l in G that start at v and end at u. We define this formula by induction on l. Note that in the inductive definition, we refer to a formula $\psi_k^l(z, y)$. This is to be read as the formula $\psi_k^l(x, y)$ with all occurrences of x and z (free or bound) interchanged. In particular, the free variables of $\psi_k^l(x, y)$ are exactly x, y and those of $\psi_k^l(z, y)$ are exactly z, y.

For $l = 1$, the formulas are defined as follows:

$$\psi_0^1(x, y) := \neg E(x, y); \quad \psi_1^1(x, y) := E(x, y);$$
$$\text{and} \quad \psi_k^1(x, y) := \text{false} \quad \text{for} \quad k > 1.$$

For the inductive case, we first introduce some notation. Say that a collection $(i_1, k_1), \ldots, (i_r, k_r)$ of pairs of integers, with $i_j \geq 1$ and $k_j \geq 0$ is an *indexed partition* of k if the $k_1, \ldots, k_r$ are pairwise distinct and $k = \sum_{j=1}^r i_j k_j$. That is, we partitioned k into $\sum_{j=1}^r i_j$ distinct parts, and there are exactly i_j parts of size k_j where $j = 1, \ldots, r$. Let K denote the set of all indexed partitions of k and note that this is a finite set.

Now, assume we have defined the formulas $\psi_k^l(x,y)$ for all values of $k \geq 0$. We proceed to define them for $l+1$

$$\psi_0^{l+1}(x,y) := \forall z(E(x,z) \rightarrow \psi_0^l(z,y))$$

$$\psi_k^{l+1}(x,y) := \bigvee_{(i_1,k_1),\ldots,(i_r,k_r)\in K} \Big(\big(\bigwedge_{j=1}^{r} \exists^{=i_j} z\ (E(x,z) \wedge \psi_{k_j}^l(z,y)\big) \wedge \exists^{=d} z\ E(x,z)\Big),$$

where $d = \sum_{j=1}^{r} i_j$. Note that without allowing counting quantification it would be necessary to use many more distinct variables to rewrite the last formula.

Given an n-vertex graph G, as noted before $(A_G^l)_{ij}$ is equal to the number of walks of length l in G from vertex v_i to vertex v_j, so $(A_G^l)_{ij} = k$ if, and only if, $G \models \psi_k^l(v_i,v_j)$. Once again, let K denote the set of indexed partitions of k. For each integer $k \geq 0$ and $l \geq 0$, we define the sentence

$$\phi_k^l := \bigvee_{(i_1,k_1),\ldots,(i_r,k_r)\in K} \Big(\bigwedge_{j=1}^{r} \exists^{=i_j} x \exists y \big(x = y \wedge \psi_k^l(x,y)\big)\Big).$$

Then we have $G \models \phi_k^l$ if, and only if, the total number of closed walks of length l in G is exactly k. Hence $G \models \phi_k^l$ if, and only if, $tr(A_G^l) = k$. Thus, we have the following proposition.

Proposition 2. *If $G \equiv_C^3 H$ then G and H are co-spectral.*

Proof. Suppose G and H are two non-cospectral graphs. Then there is some l such that $tr(A_G^l) \neq tr(A_H^l)$, i.e. the total number of closed walks of length l in G is different from the total number of closed walks of length l in H (see Proposition 1). If k is the total number of closed walks of length l in G, then $G \models \phi_k^l$ and $H \not\models \phi_k^l$. Since ϕ_k^l is a sentence of C^3, we conclude that $G \not\equiv_C^3 H$.

For any graph G and $l \geq 1$, there exists a positive integer k_l such that $tr(A_G^l) = k_l$. Since having the traces of powers of the adjacency matrix of G up to the number of vertices is equivalent to having the spectrum of G, we can define a sentence

$$\phi_G := \bigwedge_{l=1}^{n} \phi_{k_l}^l$$

of C^3 such that for any graph H, we have $H \models \phi_G$ if, and only if, $\mathrm{sp}(G) = \mathrm{sp}(H)$.

3.2 Graphs Determined by Their Spectra

We say that a graph G is *determined by its spectrum* (for short, DS) when for any graph H, if $\mathrm{sp}(G) = \mathrm{sp}(H)$ then $G \cong H$. In words, a graph is determined by its spectrum when it is the only graph up to isomorphism with a certain spectrum. In Proposition 2 we saw that C^3-equivalent graphs are necessarily co-spectral. That is, if two graphs G and H are C^3-equivalent then G and H must have the same spectrum. It thus follows that being identified by C^3 is weaker than being determined by the spectrum, so there are more graphs identified by C^3 than graphs determined by their spectra.

Observation 1. *On the class of all finite graphs, C^3-equivalence refines co-spectrality.*

In general, determine whether a graph has the DS property (i.e., the equivalence class induced by having the same spectrum coincides with its isomorphism class) is an open problem in spectral graph theory (see, e.g. [22]). Given a graph G and a positive integer k, we say that the logic C^k *identifies* G when for all graphs H, if $G \equiv_C^k H$ then $G \cong H$. Let $\mathcal{C}_n^k$ be the class of all n-vertex graphs that are identified by C^k. Since C^2-equivalence corresponds to indistinguishability by the 1-dimensional Weisfeiler-Lehman algorithm [16], from a classical result of Babai, Erdős and Selkow [3], it follows that $\mathcal{C}_n^2$ contains almost all n-vertex graphs. Let DS_n be the class of all DS n-vertex graphs.

The 1-dimensional Weisfeiler-Lehman algorithm (see Sect. 4) does not distinguish any pair of non-isomorphic regular graphs of the same degree with the same number of vertices. Hence, if a regular graph is not determined up to isomorphism by its number of vertices and its degree, then it is not in $\mathcal{C}_n^2$. However, there are regular graphs that are determined by their number of vertices and their degree. For instance, the complete graph on n vertices, which gives an example of a graph in $\mathrm{DS}_n \cap \mathcal{C}_n^2$.

Let T be a tree on n vertices. By a well-known result from Schwenk [21], with probability one there exists another tree T' such that T and T' are co-spectral but not isomorphic. From a result of Immerman and Lander [16] we know that all trees are identified by C^2. Hence T is an example of a graph in $\mathcal{C}_n^2$ and not in DS. On the other hand, the disjoint union of two complete graphs with the same number of vertices is a graph which is determined by its spectrum. That is, $2K_m$ is DS (see [22, Section 6.1]). For each $m > 2$ it is possible to construct a connected regular graph G_{2m} with the same number of vertices and the same degree as $2K_m$. Hence G_{2m} and $2K_m$ are not distinguishable in C^2 and clearly not isomorphic. This shows that co-spectrality and elementary equivalence with respect to the two-variable counting logic is incomparable.

From a result of Babai and Kučera [4], we know that a graph randomly selected from the uniform distribution over the class of all unlabeled n-vertex graphs (which has size equal to $2^{n(n-1)/2}$) is not identified by C^2 with probability equal to $(o(1))^n$. Moreover, in [18] Kučera presented an efficient algorithm for labelling the vertices of random regular graphs from which it follows that the fraction of regular graphs which are not identified by C^3 tends to 0 as the number of vertices tends to infinity. Therefore, almost all regular n-vertex graphs are in $\mathcal{C}_n^3$. Summarising, DS_n and $\mathcal{C}_n^2$ overlap and both are contained in $\mathcal{C}_n^3$.

3.3 Lower Bounds

Having established that C^3-equivalence is a refinement of co-spectrality, we now look at the relationship of the latter with equivalence in finite variable logics without counting quantifiers. First of all, we note that some co-spectral graphs can be distinguished by a formula using just two variables and no counting quantifiers.

Proposition 3. *There exists a pair of co-spectral graphs that can be distinguished in first-order logic with only two variables.*

Proof.
Let us consider the following two-variable first-order sentence:

$$\psi := \exists x \forall y \; \neg E(x, y).$$

For any graph G we have that $G \models \psi$ if, and only if, there is an isolated vertex in G. Hence $K_4 \cup K_1 \models \psi$ and $K_{1,4} \not\models \psi$. Therefore, $K_4 \cup K_1 \not\equiv^2 K_{1,4}$.

Next, we show that counting quantifiers are essential to the argument from the previous section in that co-spectrality is not subsumed by equivalence in any finite-variable fragment of first-order logic in the absence of such quantifiers. Let L^k denote the fragment of first-order logic in which each formula has at most k distinct variables.

For each $r, s \geq 0$, the *extension axiom* $\eta_{r,s}$ is the first-order sentence

$$\forall x_1 \ldots \forall x_{r+s} \left(\left(\bigwedge_{i \neq j} x_i \neq x_j \right) \rightarrow \exists y \left(\bigwedge_{i \leq r} E(x_i, y) \wedge \bigwedge_{i > r} \neg E(x_i, y) \wedge x_i \neq y \right) \right).$$

A graph G satisfies the *k-extension property* if $G \models \eta_{r,s}$ and $r + s = k$. In [17] Kolaitis and Vardi proved that if the graphs G and H both satisfy the k-extension property, then there is no formula of L^k that can distinguish them. If this happens, we write $G \equiv^k H$. Fagin [11] proved that for each $k \geq 0$, almost all graphs satisfy the k-extension property. Hence almost all graphs are not distinguished by any formula of L^k.

Let q be a prime power such that $q \equiv 1 \pmod 4$. The *Paley graph* of order q is the graph $P(q)$ with vertex set $\mathrm{GF}(q)$, the finite field of order q, where two vertices i and j are adjacent if there is a positive integer x such that $x^2 \equiv (i-j) \pmod q$. Since $q \equiv 1 \pmod 4$ if, and only if, $x^2 \equiv -1 \pmod q$ is solvable, we have that -1 is a square in $\mathrm{GF}(q)$ and so, $(j-i)$ is a square if and only if $-(i-j)$ is a square. Therefore, adjacency in a Paley graph is a symmetric relation and so, $P(q)$ is undirected. Blass et al. [6] proved that if q is greater than $k^2 2^{4k}$, then $P(q)$ satisfies the k-extension property.

Now, let $q = p^r$ with p an odd prime, r a positive integer, and $q \equiv 1 \pmod 3$. The *cubic Paley graph* $P^3(q)$ is the graph whose vertices are elements of the finite field $\mathrm{GF}(q)$, where two vertices $i, j \in \mathrm{GF}(q)$ are adjacent if and only if their difference is a cubic residue, i.e. i is adjacent to j if, and only if, $i - j = x^3$ for some $x \in \mathrm{GF}(q)$. Note that -1 is a cube in $\mathrm{GF}(q)$ because $q \equiv 1 \pmod 3$ is a prime power, so i is adjacent to j if, and only if, j is adjacent to i. In [2] it has been proved that $P^3(q)$ has the k-extension property whenever $q \geq k^2 2^{4k-2}$.

The *degree* of vertex v in a graph G is the number $d(v) := |\{\{v,u\} \in E : u \in V_G\}|$ of vertices that are adjacent to v. A graph G is *regular* of degree d if every vertex is adjacent to exactly d other vertices, i.e. $d(v) = d$ for all $v \in V_G$. So, G is regular of degree d if, and only if, each row of its adjacency matrix adds up to d. It can been shown that the Paley graph $P(q)$ is regular of degree $(q-1)/2$ [13]. Moreover, it has been proved that the cubic Paley graph $P^3(q)$ is regular of degree $(q-1)/3$ [10].

Lemma 1. *Let G be a regular graph of degree d. Then $d \in \mathrm{sp}(G)$ and for each $\theta \in \mathrm{sp}(G)$, we have $|\theta| \le d$. Here $|\cdot|$ is the operation of taking the absolute value.*

Proof. Let us denote by $\mathbf{1}$ the all-ones vector. Then $A_G\mathbf{1} = d\mathbf{1}$. Therefore, $d \in \mathrm{sp}(G)$. Now, let s be such that $|s| > d$. Then, for each row i,

$$|S_{ii}| > \sum_{j \neq i} |S_{ij}|$$

where $S = sI - A_G$. Therefore, the matrix S is strictly diagonally dominant, and so $det(sI - A_G) \neq 0$. Hence s is not an eigenvalue of G.

Lemma 2. *Let G and H be regular graphs of distinct degrees. Then G and H do not have the same spectrum.*

Proof. Suppose that G is regular of degree s and H is regular of degree t, with $s \neq t$. Then $A_G\mathbf{1} = s\mathbf{1}$ and $A_H\mathbf{1} = t\mathbf{1}$, where $\mathbf{1}$ is the all-ones vector. Therefore, s is the greatest eigenvalue in the spectrum of G and t is the greatest eigenvalue in the spectrum of H. Hence $\mathrm{sp}(G) \neq \mathrm{sp}(H)$.

Proposition 4. *For each $k \ge 1$, there exists a pair G_k, H_k of graphs which are not co-spectral, such that G_k and H_k are not distinguished by any formula of L^k.*

Proof. For any positive integer r we have that $13^r \equiv 1 \pmod 3$ and $13^r \equiv 1 \pmod 4$. For each $k \ge 1$, let r_k be the smallest integer greater than $2(k\log(4) + \log(k))/\log(13)$, and let $q_k = 13^{r_k}$. Hence $q_k > k^2 2^{4k}$. Now, let $G_k = P(q_k)$ and $H_k = P^3(q_k)$. Then G_k and H_k both satisfy the k-extension property, and so $G_k \equiv^k H_k$. Since the degree of G_k is $(13^{r_k} - 1)/2$ and the degree of H_k is $(13^{r_k} - 1)/3$, by Lemma 1 we conclude that $\mathrm{sp}(G_k) \neq \mathrm{sp}(H_k)$.

So having the same spectrum is a property of graphs that does not follows from any finite collection of extension axioms, or equivalently, from any first-order sentence with asymptotic probability 1.

4 Isomorphism Approximations

4.1 WL Equivalence

The automorphism group $\mathrm{Aut}(G)$ of G acts naturally on the set V_G^k of all k-tuples of vertices of G, and the set of orbits of k-tuples under the action of $\mathrm{Aut}(G)$ form a corresponding partition of V_G^k. The *k-dimensional Weisfeiler-Leman algorithm* is a combinatorial method that tries to approximate the partition induced by the orbits of $\mathrm{Aut}(G)$ by labelling the k-tuples of vertices of G. For the sake of completeness, here we give a brief overview of the algorithm.

The 1-dimensional Weisfeiler-Leman algorithm has the following steps: first, label each vertex $v \in V_G$ by its degree $d(v)$. The set $N(v) := \{u : \{v, u\} \in E_G\}$ is called the *neighborhood* of $v \in V_G$ and so, the degree of v is just the number of neighbours it has, i.e. $d(v) = |N(v)|$. In this way we have defined a partition $P_0(G)$ of V_G. The number of labels is equal to the number of different degrees. Hence $P_0(G)$ is the degree sequence of G. Then, relabel each vertex v with the multi-set of labels of its neighbours, so each label $d(v)$ is substituted for $\{d(v), \{d(u) : u \in N(v)\}\}$. Since these are multi-sets they might contain repeated elements. We get then a partition $P_1(G)$ of V_G which is either a refinement of $P_0(G)$ or identical to $P_0(G)$. Inductively, the partition $P_t(G)$ is obtained from the partition $P_{t-1}(G)$, by constructing for each vertex v a new multi-set that includes the labels of its neighbours, as it is done in the previous step. The algorithm halts as soon as the number of labels does not increase anymore. We denote the resulting partition of V_G by P_G^1.

Now we describe the algorithm for higher dimensions. Recall that we are working in the first-order language of graphs $L = \{E\}$. Now, for each graph G and each k-tuple $\mathbf{v}$ of vertices of G we define the *(atomic) type of* $\mathbf{v}$ in G as the set $\mathrm{tp}_G^k(\mathbf{v})$ of all atomic L-formulas $\phi(\mathbf{x})$ that are true in G when the variables of $\mathbf{x}$ are substituted for vertices of $\mathbf{v}$. More formally, for $k > 1$ we let

$$\mathrm{tp}_G^k(\mathbf{v}) := \{\phi(\mathbf{x}) : |\mathbf{x}| \leq k, G \models \phi(\mathbf{v})\}$$

where, $|\mathbf{x}|$ denotes the number of entries the tuple $\mathbf{x}$ have, and each $\phi(\mathbf{x})$ is either $x_i = x_j$ or $E(x_i, x_j)$ for $1 \leq i, j \leq k$. Essentially, the formulas of $\mathrm{tp}_G^k(\mathbf{v})$ give us the complete information about the structural relations that hold between the vertices of $\mathbf{v}$. If $u \in V_G$ and $1 \leq i \leq k$, let $\mathbf{v}_i^u$ denote the result of substituting u in the i-th entry of $\mathbf{v}$.

For each $k > 1$ the k-dimensional Weisfeiler-Leman algorithm proceeds as follows: first, label the k-tuples of vertices with their types in G, so each k-tuple $\mathbf{v}$ is labeled with $\ell_0(\mathbf{v}) := \mathrm{tp}_G^k(\mathbf{v})$; this induces a partition $P_0^k(G)$ of the k-tuples of vertices of G. Inductively, refine the partition $P_i^k(G)$ of V_G^k by relabelling the k-tuples so that each label $\ell_i(\mathbf{v})$ is substituted for $\ell_{i+1}(\mathbf{v}) := \{\ell_i(\mathbf{v}), \{\ell_i(\mathbf{v}_1^u), \ldots, \ell_i(\mathbf{v}_k^u) : u \in V_G\}\}$. The algorithm continues refining the partition of V_G^k until it gets to a step $t \geq 1$, where $P_t^k(G) = P_{t-1}^k(G)$; then it halts. We denote the resulting partition of V_G^k by P_G^k.

Notice that for any fixed $k \geq 1$, the partition P_G^k of k-subsets is obtained after at most $|V_G|^k$ steps. If the partitions P_G^k and P_H^k of graphs G and H are the same multi-set of labels obtained by the k-dimensional Weisfeiler-Leman algorithm, we say that G and H are *k-WL equivalent*. In [8], Cai, Fürer and Immerman proved that two graphs G and H are C^{k+1}-equivalent if, and only if, G and H are k-WL equivalent.

4.2 Symmetric Powers

The *k-th symmetric power* $G^{\{k\}}$ of a graph G is a graph where each vertex represents a k-subset of vertices of G, and two k-subsets are adjacent if their symmetric difference is an edge of G. Formally, the vertex set $V_{G^{\{k\}}}$ of $G^{\{k\}}$ is defined to be the set of all subsets of V_G with exactly k elements, and for every pair of k-subsets of vertices $V = \{v_1, \ldots, v_k\}$ and $U = \{u_1, \ldots, u_k\}$, we have $\{V, U\} \in E_{G^{\{k\}}}$ if, and only if, $(V \setminus U) \cup (U \setminus V) \in E_G$. The symmetric powers are related to a natural generalisation of the concept of a walk in a graph. A *k-walk* of length l in G is a sequence $(V_0, V_1, \ldots, V_l)$ of k-subsets of vertices, such that the symmetric difference of V_{i-1} and V_i is an edge of G for $1 \leq i \leq l$. A k-walk is said to be closed if $V_0 = V_l$. The connection with the symmetric powers is that a k-walk in G corresponds to an ordinary walk in $G^{\{k\}}$. Therefore, two graphs have the same total number of closed k-walks of every length if, and only if, their k-th symmetric powers are co-spectral. For each $k \geq 1$, there exist infinitely many pairs of non-isomorphic graphs G and H such that the k-th symmetric powers $G^{\{k\}}$ and $H^{\{k\}}$ are co-spectral [5].

Alzaga et al. [1] have shown that given two graphs G and H, if G and H are $2k$-WL equivalent, then their k-th symmetric powers $G^{\{k\}}$ and $H^{\{k\}}$ are co-spectral. This two facts combined allow us to deduce the following generalisation of Proposition 2.

Proposition 5. *Given graphs G and H and a positive integer k, if $G \equiv_C^{2k+1} H$ then $G^{\{k\}}$ and $H^{\{k\}}$ are co-spectral.*

4.3 Cellular Algebras

Originally, Weisfeiler and Leman [19] presented their algorithm in terms of algebras of complex matrices.Given two matrices A and B of the same order, their *Schur product* $A \circ B$ is defined by $(A \circ B)_{ij} := A_{ij}B_{ij}$. For a complex matrix A, let A^* denote the adjoint (or conjugate-transpose) of A. A *cellular algebra* W is an algebra of square complex matrices that contains the identity matrix I, the all-ones matrix J, and is closed under adjoints and Schur multiplication. Thus, every cellular algebra has a unique basis $\{A_1, \ldots, A_m\}$ of binary matrices which is closed under adjoints and such that $\sum_i A_i = J$.

The smallest cellular algebra is the one generated by the span of I and J. The cellular algebra of an n-vertex graph G is the smallest cellular algebra W_G that contains A_G. Two cellular algebras W and W' are isomorphic if there is an algebra isomorphism $h : W \to W'$, such that $h(A \circ B) = h(A) \circ h(B)$, $h(A)^* = h(A^*)$ and $h(J) = J$. It is interesting that given an isomorphism $h : W \to W'$ of cellular algebras, for all $A \in W$ we have that A and $h(A)$ are co-spectral (see Lemma 3.4 in [12]). So the next result is immediate.

Proposition 6. *Two graphs G and H are co-spectral if their corresponding cellular algebras there exists an isomorphism of W_G and W_H that maps A_G to A_H.*

In general, the converse of Proposition 6 is not true. That is, there are known pairs of co-spectral graphs whose corresponding cellular algebras are non-isomorphic (see, e.g. [5]).

The elements of the standard basis of a cellular algebra correspond to the "adjacency matrices" of a corresponding coherent configuration. Coherent configurations where introduced by Higman in [15] to study finite permutation groups. Coherent configurations are stable under the 2-dimensional Weisfeiler-Leman algorithm. Hence two graphs G and H are 2-WL equivalent if, and only if, their corresponding cellular algebras there exists an isomorphism of W_G and W_H that maps A_G to A_H.

Proposition 7. *Given graphs G and H with cellular algebras W_G and W_H, $G \equiv_C^3 H$ if, and only if, there exists an isomorphism of W_G and W_H that maps A_G to A_H.*

4.4 Strongly Regular Graphs

A *strongly regular graph* $\mathrm{srg}(n, r, \lambda, \mu)$ is a regular n-vertex graph of degree r such that each pair of adjacent vertices has λ common neighbours, and each pair of nonadjacent vertices has μ common neighbours. The numbers n, r, λ, μ are called the *parameters* of $\mathrm{srg}(n, r, \lambda, \mu)$. It can be shown that the spectrum of a strongly regular graph is determined by its parameters [13]. The complement of a strongly regular graph is strongly regular. Moreover, co-spectral strongly regular graphs have co-spectral complements. That is, two strongly regular graphs having the same parameters are co-spectral. Recall that J is the all-one matrix.

Lemma 3. *If G is a strongly regular graph then $\{I, A_G, (J - I - A_G)\}$ form the basis for its corresponding cellular algebra W_G.*

Proof. By definition, W_G has a unique basis $\mathcal{A}$ of binary matrices closed under adjoints and so that

$$\sum_{A \in \mathcal{A}} A = J.$$

Notice that I, A_G and $J - I - A_G$ are binary matrices such that $I^* = I$, $A_G^* = A_G$ and $(J - I - A_G)^* = J - I - A_G$. Furthermore,

$$I + A_G + (J - I - A_G) = J.$$

There are known pairs of non-isomorphic strongly regular graphs with the same parameters (see, e.g. [7]). These graphs are not distinguished by the 2-dimensional Weisfeiler-Leman algorithm since their corresponding cellular algebras are isomorphic. Thus, for strongly regular graphs the converse of Proposition 6 holds.

Lemma 4. *If G and H are two co-spectral strongly regular graphs, then their corresponding cellular algebras are isomorphic.*

Proof. In [12], Friedland has shown that two cellular algebras with standard bases $\{A_1, \ldots, A_m\}$ and $\{B_1, \ldots, B_m\}$ are isomorphic if, and only if, there is an invertible matrix M such that $MA_iM^{-1} = B_i$ for $1 \leq i \leq m$.

The cellular algebras W_G and W_H of G and H have standard basis $\{I, A_G, (J - I - A_G)\}$ and $\{I, A_H, (J - I - A_H)\}$, respectively. Since G and H are co-spectral, there exist an orthogonal matrix Q such that $QA_GQ^T = A_H$ and $Q(J-I-A_G)Q^T = (J-I-A_H)$. As every orthogonal matrix is invertible, we can conclude that there exists an isomorphism of W_G and W_H that maps A_G to A_H.

Proposition 8. *Given two strongly regular graphs G and H, the following statements are equivalent:*

1. $G \equiv_C^3 H$;
2. *G and H are co-spectral;*
3. *there exists an isomorphism of W_G and W_H that maps A_G to A_H.*

Proof. Proposition 2 says that for all graphs (1) implies (2). From Proposition 7, we have (1) if, and only if, (3). By Lemma 4, if (2) then (3).

5 Definability in Fixed Point Logic with Counting

In this section, we consider the definability of co-spectrality and the property DS in fixed-point logics with counting. To be precise, we show that co-spectrality is definable in *inflationary fixed-point logic with counting* (FPC) and the class of graphs that are DS is definable in *partial fixed-point logic with counting* (PFPC). It follows that both of these are also definable in the infinitary logic with counting, with a bounded number of variables (see [9, Proposition 8.4.18]). Note that it is known that FPC can express any polynomial-time decidable property of *ordered* structures and similarly PFPC can express all polynomial-space decidable properties of ordered structures. It is easy to show that co-spectrality is decidable in polynomial time and DS is in PSPACE. For the latter, note that DS can easily be expressed by a Π_2 formula of second-order logic and therefore the problem is in the second-level of the polynomial hierarchy. However, in the absence of a linear order FPC and PFPC are strictly weaker than the complexity classes P and PSPACE respectively. Indeed, there are problems in P that are not even expressible in the infinitary logic with counting. Nonetheless, it is in this context without order that we establish the definability results below.

We begin with a brief definition of the logics in question, to fix the notation we use. For a more detailed definition, we refer the reader to [9, 20].

FPC is an extension of inflationary fixed-point logic with the ability to express the cardinality of definable sets. The logic has two sorts of first-order variables: *element variables*, which range over elements of the structure on which a formula is interpreted in the usual way, and *number variables*, which range over some initial segment of the natural numbers. We usually write element variables with lower-case Latin letters $x, y, \ldots$ and use lower-case Greek letters $\mu, \eta, \ldots$ to denote number variables. In addition, we have relational variables, each of which has an arity m and an associated type from $\{\text{elem}, \text{num}\}^m$. PFPC is similarly obtained by allowing the *partial fixed point* operator in place of the inflationary fixed-point operator.

For a fixed signature τ, the atomic formulas of FPC$[\tau]$ of PFPC$[\tau]$ are all formulas of the form $\mu = \eta$ or $\mu \leq \eta$, where μ, η are number variables; $s = t$ where s, t are element variables or constant symbols from τ; and $R(t_1, \ldots, t_m)$, where R is a relation symbol (i.e. either a symbol from τ or a relational variable) of arity m and each t_i is a term of the appropriate type (either elem or num, as determined by the type of R). The set FPC$[\tau]$ of FPC *formulas* over τ is built up from the atomic formulas by applying an inflationary fixed-point operator $[\mathbf{ifp}_{R,\boldsymbol{x}}\phi](\boldsymbol{t})$; forming *counting terms* $\#_x\phi$, where ϕ is a formula and x an element variable; forming formulas of the kind $s = t$ and $s \leq t$ where s, t are number variables or counting terms; as well as the standard first-order operations of negation, conjunction, disjunction, universal and existential quantification. Collectively, we refer to element variables and constant symbols as *element terms*, and to number variables and counting terms as *number terms*. The formulas of PFPC$[\tau]$ are defined analogously, but we replace the fixed-point operator rule by the partial fixed-point: $[\mathbf{pfp}_{R,\boldsymbol{x}}\phi](\boldsymbol{t})$.

For the semantics, number terms take values in $\{0, \ldots, n\}$, where n is the size of the structure in which they are interpreted. The semantics of atomic formulas, fixed-points and first-order operations are defined as usual (c.f., e.g., [9] for details), with comparison of number terms $\mu \leq \eta$ interpreted by comparing the corresponding integers in $\{0, \ldots, n\}$. Finally, consider a counting term of the form $\#_x\phi$, where ϕ is a formula and x an element variable. Here the intended semantics is that $\#_x\phi$ denotes the number (i.e. the element of $\{0, \ldots, n\}$) of elements that satisfy the formula ϕ.

Note that, since an inflationary fixed-point is easily expressed as a partial fixed-point, every formula of FPC can also be expressed as a formula of PFPC. In the construction of formulas of these logics below, we freely use arithmetic expressions on number variables as the relations defined by such expressions can easily be defined by formulas of FPC.

In Sect. 3 we constructed sentences ϕ_k^l of C^3 which are satisfied in a graph G if, and only if, the number of closed walks in G of length l is exactly k. Our first aim is to construct a single formula of FPC that expresses this for all l and k. Ideally, we would have the numbers as parameters to the formula but it should be noted that, while the length l of walks we consider is bounded by the number n of vertices of G, the number of closed walks of length l is not bounded by any polynomial in n. Indeed, it can be as large as n^n. Thus, we cannot represent the value of k by a single number variable, or even a fixed-length tuple of number variables. Instead, we represent k as a binary relation K on the number domain. The order on the number domain induces a lexicographical order on pairs of numbers, which is a way of encoding numbers in the range $0, \ldots, n^2$. Let us write $[i, j]$ to denote the number coded by the pair (i, j). Then, a binary relation K can be used to represent a number k up to 2^{n^2} by its binary encoding. To be precise, K contains all pairs (i, j) such that bit position $[i, j]$ in the binary encoding of k is 1. It is easy to define formulas of FPC to express arithmetic operations on numbers represented in this way.

Thus, we aim to construct a single formula $\phi(\lambda, \kappa_1, \kappa_2)$ of FPC, with three free number variables such that $G \models \phi[l, i, j]$ if, and only if, the number of closed walks in G of length l is k and position $[i, j]$ in the binary expansion of k is 1. To do this, we first define a formula $\psi(\lambda, \kappa_1, \kappa_2, x, y)$ with free number variables λ, κ_1 and κ_2 and free element variables x and y that, when interpreted in G defines the set of tuples (l, i, j, v, u) such that if there are exactly k walks of length l starting at v and ending at u, then position $[i, j]$ in the binary expansion of k is 1. This can be defined by taking the inductive definition of ψ_k^l we gave in Sect. 3 and making the induction part of the formula.

We set out the definition below.

$$\begin{aligned}\psi(\lambda, \kappa_1, \kappa_2, x, y) := \mathbf{ifp}_{W,\lambda,\kappa_1,\kappa_2,x,y}[&\lambda = 1 \wedge \kappa_1 = 0 \wedge \kappa_2 = 1 \wedge E(x, y) \vee \\ &\lambda = \lambda' + 1 \wedge \mathrm{sum}(\lambda', \kappa_1, \kappa_2, x, y)]\end{aligned}$$

where W is a relation variable of type $(\mathrm{num}, \mathrm{num}, \mathrm{num}, \mathrm{elem}, \mathrm{elem})$ and the formula sum expresses that there is a 1 in the bit position encoded by (κ_1, κ_2) in the binary expansion of $k = \sum_{z:E(x,z)} k_{\lambda',z,y}$, where $k_{\lambda',z,y}$ denotes the number coded by the binary relation $\{(i, j) : W(\lambda', i, j, z, y)\}$. We will not write out the formula sum in full. Rather we note that it is easy to define inductively the sum of a set of numbers given in binary notation, by defining a sum and carry bit. In our case, the set of numbers is given by a ternary relation of type $(\mathrm{elem}, \mathrm{num}, \mathrm{num})$ where fixing the first component to a particular value z yields a binary relation coding a number. A similar application of induction to sum a set of numbers then allows us to define the formula $\phi(\lambda, \kappa_1, \kappa_2)$ which expresses that the bit position indexed by (κ_1, κ_2) is 1 in the binary expansion of $k = \sum_{x \in V} k_x$ where k_x denotes the number coded by $\{(i, j) : \psi[\lambda, i, j, x, x]\}$.

To define co-spectrality in FPC means that we can write a formula cospec in a vocabulary with two binary relations E and E' such that a structure (V, E, E') satisfies this formula if, and only if, the graphs (V, E) and (V, E') are co-spectral. Such a formula is now easily derived from ϕ. Let ϕ' be the formula obtained from ϕ by replacing all occurrences of E by E', then we can define:

$$\mathrm{cospec} := \forall \lambda, \kappa_1, \kappa_2\ \phi \Leftrightarrow \phi'.$$

Now, in order to give a definition in PFPC of the class of graphs that are DS, we need two variations of the formula cospec. First, let R be a relation symbol of type $(\mathrm{num}, \mathrm{num})$. We write $\phi(R)$ for the formula obtained from ϕ by replacing the symbol E with the relation variable R, and suitably replacing number variables with element variables. So, $\phi(R, \lambda, \kappa_1, \kappa_2)$ defines, in the graph defined by the relation R on the number domain, the number of closed walks of length λ. We write cospec_R for the formula

$$\forall \lambda, \kappa_1, \kappa_2\ \phi(R) \Leftrightarrow \phi,$$

which is a formula with a free relational variable R which, when interpreted in a graph G asserts that the graph defined by R is co-spectral with G. Similarly, we define the formula with two free second-order variables R and R'

$$\mathrm{cospec}_{R,R'} := \forall \lambda, \kappa_1, \kappa_2\ \phi(R) \Leftrightarrow \phi(R').$$

Clearly, this is true of a pair of relations iff the graphs they define are co-spectral.

Furthermore, it is not difficult to define a formula $\mathrm{isom}(R, R')$ of PFPC with two free relation symbols of type $(\mathrm{num}, \mathrm{num})$ that asserts that the two graphs defined by R and R' are isomorphic. Indeed, the number domain is ordered and any property in PSPACE over an ordered domain is definable in PFPC, so such a formula must exist. Given these, the property of a graph being DS is given by the following formula with second-order quantifiers:

$$\forall R(\mathrm{cospec}_R \Rightarrow \forall R'(\mathrm{cospec}_{R,R'} \Rightarrow \mathrm{isom}(R, R'))).$$

To convert this into a formula of PFPC, we note that second-order quantification over the number domain can be expressed in PFPC. That is, if we have a formula $\theta(R)$ of PFPC in which R is a free second-order variable of type $(\mathrm{num}, \mathrm{num})$, then we can define a PFPC formula that is equivalent to $\forall R\, \theta$. We do this by means of an induction that loops through all binary relations on the number domain in lexicographical order and stops if for one of them θ does not hold.

First, define the formula $\mathrm{lex}(\mu, \nu, \mu', \nu')$ to be the following formula which defines the lexicographical ordering of pairs of numbers:

$$\mathrm{lex}(\mu, \nu, \mu', \nu') := (\mu < \mu') \vee (\mu = \mu' \wedge \nu < \nu').$$

We use this to define a formula $\mathrm{next}(R, \mu, \nu)$ which, given a binary relation R of type $(\mathrm{num}, \mathrm{num})$, defines the set of pairs (μ, ν) occurring in the relation that is lexicographically immediately after R.

$$\begin{aligned}\mathrm{next}(R, \mu, \nu) := {} & R(\mu, \nu) \wedge \exists \mu'\nu'(\mathrm{lex}(\mu', \nu', \mu, \nu) \wedge \neg R(\mu', \nu')) \vee \\ & \vee \neg R(\mu, \nu) \wedge \forall \mu'\nu'(\mathrm{lex}(\mu', \nu', \mu, \nu) \Rightarrow R(\mu', \nu')).\end{aligned}$$

We now use this to simulate, in PFPC, second-order quantification over the number domain. Let $\bar{R}$ be a new relation variable of type $(\mathrm{num}, \mathrm{num}, \mathrm{num})$ and we define the following formula

$$\begin{aligned}\forall \alpha \forall \beta\, \mathbf{pfp}_{\bar{R},\mu,\nu,\kappa}[& (\forall \mu\nu \bar{R}(\mu, \nu, 0)) \wedge \theta(\bar{R}) \wedge \kappa = 0 \vee \\ & \vee \neg\theta(\bar{R}) \wedge \kappa \neq 0 \vee \\ & \vee \theta(\bar{R}) \wedge \mathrm{next}(\bar{R}, \mu, \nu) \wedge \kappa = 0](\alpha, \beta, 0).\end{aligned}$$

It can be checked that this formula is equivalent to $\forall R\, \theta$.

6 Conclusion

Co-spectrality is an equivalence relation on graphs with many interesting facets. While not every graph is determined upto isomorphism by its spectrum, it is a long-standing conjecture (see [22]), still open, that *almost all* graphs are DS. That is to say that the proportion of n-vertex graphs that are DS tends to 1 as n grows. We have established a number of results relating graph spectra to definability in logic and it is instructive to put them in the perspective of this open question. It is an easy consequence of the results in [17] that the proportion of graphs that are determined up to isomorphism by their L^k theory tends to 0. On the other hand, it is known that almost all graphs are determined by their C^2 theory (see [14]) and *a fortiori* by their C^3 theory. We have established that co-spectrality is incomparable with L^k-equivalence for any k; is incomparable with C^2 equivalence; and is subsumed by C^3 equivalence. Thus, our results are compatible with either answer to the open question of whether almost all graphs are DS. It would be interesting to explore further whether logical definability can cast light on this question.

References

1. Alzaga, A., Iglesias, R., Pignol, R.: Spectra of symmetric powers of graphs and the Weisfeiler-Lehman refinements. J. Comb. Theory Ser. B **100**(6), 671–682 (2010)
2. Ananchuen, W., Caccetta, L.: Cubic and quadruple paley graphs with the n-e. c. property. Discrete Math. **306**(22), 2954–2961 (2006)
3. Babai, L., Erdős, P., Selkow, S.M.: Random graph isomorphism. SIAM J. Comput. **9**(3), 628–635 (1980)
4. Babai, L., Kučera, L.: Canonical labelling of graphs in linear average time. In: 20th Annual Symposium on Foundations of Computer Science, 1979, pp. 39–46. IEEE (1979)
5. Barghi, A.R., Ponomarenko, I.: Non-isomorphic graphs with cospectral symmetric powers. Electron. J. Comb. **16**(1), R120 (2009)
6. Blass, A., Exoo, G., Harary, F.: Paley graphs satisfy all first-order adjacency axioms. J. Graph Theory **5**(4), 435–439 (1981)
7. Brouwer, A.E., Van Lint, J.H.: Strongly regular graphs and partial geometries. In: Enumeration and Design (Waterloo, Ont., 1982), pp. 85–122 (1984)
8. Cai, J., Fürer, M., Immerman, N.: An optimal lower bound on the number of variables for graph identification. Combinatorica **12**(4), 389–410 (1992)
9. Ebbinghaus, H.B., Flum, J.: Finite Model Theory, 2nd edn. Springer, Heidelberg (1999)
10. Elsawy, A.N.: Paley graphs and their generalizations (2012). arXiv preprint. arXiv:1203.1818
11. Fagin, R.: Probabilities on finite models. J. Symbolic Logic **41**(01), 50–58 (1976)
12. Friedland, S.: Coherent algebras and the graph isomorphism problem. Discrete Appl. Math. **25**(1), 73–98 (1989)
13. Godsil, C., Royle, G.F.: Algebraic Graph Theory, vol. 207. Springer, New York (2013)
14. Hella, L., Kolaitis, P.G., Luosto, K.: How to define a linear order on finite models. Ann. Pure Appl. Logic **87**(3), 241–267 (1997)
15. Higman, D.G.: Coherent configurations. Geom. Dedicata **4**(1), 1–32 (1975)
16. Immerman, N., Lander, E.: Describing graphs: a first-order approach to graph canonization. In: Selman, A.L. (ed.) Complexity Theory Retrospective, pp. 59–81. Springer, New York (1990)

17. Kolaitis, P.G., Vardi, M.Y.: Infinitary logics and 0–1 laws. Inf. Comput. **98**(2), 258–294 (1992)
18. Kučera, L.: Canonical labeling of regular graphs in linear average time. In: 28th Annual Symposium on Foundations of Computer Science, pp. 271–279. IEEE (1987)
19. Leman, A.A., Weisfeiler, B.: A reduction of a graph to a canonical form and an algebra arising during this reduction. Nauchno-Technicheskaya Informatsiya **2**(9), 12–16 (1968)
20. Libkin, L.: Elements of Finite Model Theory. Springer, Heidelberg (2004)
21. Schwenk, A.J.: Almost all trees are cospectral. In: New Directions in the Theory of Graphs, pp. 275–307 (1973)
22. Van Dam, E.R., Haemers, W.H.: Which graphs are determined by their spectrum? Linear Algebra Appl. **373**, 241–272 (2003)

Causality in Bounded Petri Nets is MSO Definable

Mateus de Oliveira Oliveira(✉)

Institute of Mathematics, Czech Academy of Sciences,
Zitná 25, 115 67 Praha 1, Czech Republic
mateus.oliveira@math.cas.cz

Abstract. In this work we show that the causal behaviour of any bounded Petri net is definable in monadic second order (MSO) logic. Our proof relies in a definability vs recognizability result for DAGs whose edges and vertices can be covered by a constant number of paths. Our notion of recognizability is defined in terms of saturated slice automata, a formalism for the specification of infinite families of graphs. We show that a family $\mathfrak{G}$ of k-coverable DAGs is recognizable by a saturated slice automaton if and only if $\mathfrak{G}$ is definable in monadic second order logic. This result generalizes Büchi's theorem from the context of strings, to the context of k-coverable DAGs.

Keywords: Partial order behaviour of Petri nets · Monadic second order logic · Recognizability · Definability

1 Introduction

Partial orders are a suitable formalism for the representation of causality on runs of concurrent systems [14,15,19,22,26]. In the realm of Petri nets [24] partial orders may be extracted from objects called Petri net processes [16]. Intuitively, a Petri net process is a directed acyclic graph (DAG) whose vertex set is partitioned into *conditions* and *events*. While condition vertices are used to keep track of each token ever created or consumed during a concurrent run, the event vertices are used to keep track of which transitions created or consumed each such token. In such a process π, an event v causally depends on the occurrence of an event v' if there is a path from v' to v in π. The partial order induced on the events of a process is called a *causal order*. The causal behavior of a Petri net N is the set $\mathcal{P}(N)$ of all causal orders derived from processes of N.

The *monadic second order logic* of partial orders extends first-order logic by allowing quantifications over sets of vertices and sets of edges. In a previous work, we have shown that for any monadic second order sentence φ in the vocabulary of partial orders, and for any bounded Petri net N, one can decide whether all causal orders of N satisfy φ [12]. In this work, we show that MSO logic is indeed powerful enough to represent the causal behaviour of any given bounded Petri net (Theorem 4.1).

J. Väänänen et al. (Eds.): WoLLIC 2016, LNCS 9803, pp. 200–214, 2016.
DOI: 10.1007/978-3-662-52921-8_13

Recall that Büchi's theorem states that a set L of finite strings is recognizable by a finite automaton if and only if L is definable in monadic second order logic [5]. Motivated by a conjecture of Courcelle [6], there has been a great amount of interest in generalizing Büchi's theorem to more general algebraic structures[1] [2,6,7,17,18,21]. Towards proving our main theorem, we will prove a definability vs recognizability result for DAGs whose edges and vertices can be covered by a constant number k of paths. We say that these DAGs are k-coverable. Our notion of recognizability is defined in terms of saturated slice automata, a formalism for the specification of infinite families of graphs. More precisely, we show that a family $\mathfrak{G}$ of k-coverable DAGs is recognizable by a saturated slice automaton if and only if $\mathfrak{G}$ is definable in monadic second order logic (Theorem 4.5, Corollary 5.1). This result, which is of independent interest from the partial order theory of Petri nets, may be regarded as a generalization of Büchi's theorem from the context of strings to the context of k-coverable DAGs (note that a string can be naturally regarded as a 1-coverable DAG).

1.1 Petri Nets

A Petri net is a tuple $N = (P, T, W, m_0)$ where P is a set of *places*, T is a set of *transitions* such that $P \cap T = \emptyset$, $W : (P \times T) \cup (T \times P) \rightarrow \mathbb{N}$ is a function that associates with each element $(x, y) \in (P \times T) \cup (T \times P)$ a weight $W(x, y)$, and $m_0 : P \rightarrow \mathbb{N}$ is a function that associates with each place $p \in P$ a non-negative integer $m_0(p)$.

A marking for N is any function of the form $m : P \rightarrow \mathbb{N}$. Intuitively, a marking m assigns a number of tokens to each place of N. The marking m_0 is called the *initial marking* of N. If m is a marking and t is a transition in T, then we say that t is enabled at m if $m(p) - W(p, t) \geq 0$ for every place $p \in P$. If this is the case, the firing of t yields the marking m' which is obtained from m by setting $m'(p) = m(p) - W(p, t) + W(t, p)$ for each place $p \in P$. A firing sequence for N is a mixed sequence of markings and transitions $m_0 \xrightarrow{t_1} m_1 \xrightarrow{t_2} ... \xrightarrow{t_n} m_n$ such that for each $i \in \{1, ..., n\}$, t_i is enabled at m_{i-1}, and m_i is obtained from m_{i-1} by the firing of t_i. We say that such a firing sequence is b-bounded if for each $i \in \{0, ..., n\}$ and each $p \in P$, $m_i(p) \leq b$. We say that N is b-bounded if each of its firing sequences is b-bounded.

1.2 The Causal Semantics of Petri Nets

In this subsection we introduce the Goltz-Reisig partial order semantics for Petri nets [16]. Within this semantics, partial orders are used to represent the causality between events in concurrent runs of a Petri net. The information about the causality between events is extracted from objects called Petri net *processes*,

[1] Proposals of proofs of Courcelle's conjecture for graphs of constant pathwidth and constant treewidth have been provided in [20] and [23] respectively. Nevertheless, both proposed proofs contained substantial gaps, and Courcelle's conjecture is regarded to be open in both cases [8].

which encode the production and consumption of tokens along a concurrent run of the Petri net in question. The definition of processes, in turn, is based on the notion of *occurrence net*.

An occurrence net is a DAG $O = (B \dot{\cup} V, F)$ where the vertex set $B \dot{\cup} V$ is partitioned into a set B, whose elements are called conditions, and a set V, whose elements are called events. The edge set $F \subseteq (B \times V) \cup (V \times B)$ is restricted in such a way that for every condition $b \in B$,

$$|\{(b,v) \mid v \in V\}| \leq 1 \quad \text{and} \quad |\{(v,b) \mid v \in V\}| \leq 1.$$

In other words, conditions in an occurrence net are unbranched. For each condition $b \in B$, we let $InDegree(b)$ denote the number of edges having b as target. A process of a Petri net N is an occurrence net whose conditions are labeled with places of N, and events are labeled with transitions of N. Processes are intuitively used to describe the token game in a concurrent execution of the net. We say that a vertex of a DAG is minimal if it has no in-neighbours, and maximal if it has no out-neighbours.

Definition 1.1 (Process [16]). *A process of a Petri net $N = (P, T, W, m_0)$ is a labeled DAG $\pi = (B \dot{\cup} V, F, \rho)$ where $(B \dot{\cup} V, F)$ is an occurrence net and $\rho : (B \cup V) \to (P \cup T)$ is a labeling function satisfying the following properties.*

1. *Places label conditions and transitions label events.*
$$\rho(B) \subseteq P \qquad \rho(V) \subseteq T$$
2. *All minimal and maximal vertices of π are conditions. Additionally, for every $p \in P$,*
$$|\{b : InDegree(b) = 0, \rho(b) = p\}| = m_0(p).$$
3. *For every $v \in V$, and every $p \in P$,*
$$|\{(b,v) \in F : \rho(b){=}p\}| = W(p, \rho(v)) \quad \textit{and}$$
$$|\{(v,b) \in F : \rho(b){=}p\}| = W(\rho(v), p)$$

Item 1 says that the conditions of a process are labeled with places, while the events are labeled with transitions. Item 2 expresses the intuition that each minimal condition of π corresponds to a token in the initial marking of N. Thus for each place p of N the process has $m_0(p)$ minimal conditions labeled with the place p. Item 3, determines that the token game of a process corresponds to the token game defined by the firing of transitions in the Petri net N. Thus if a transition t consumes $W(p,t)$ tokens from place p and produces $W(t,p)$ tokens at place p, then each event labeled with t must have $W(p,t)$ in-neighbours that are conditions labeled with p, and $W(t,p)$ out-neighbours that are conditions labeled with p (Fig. 1).

Let $R \subseteq X \times X$ be a binary relation over a set X. We denote by $\boldsymbol{tc}(R)$ the transitive closure of R. If $\pi = (B \cup V, F, \rho)$ is a process then the *causal order* of

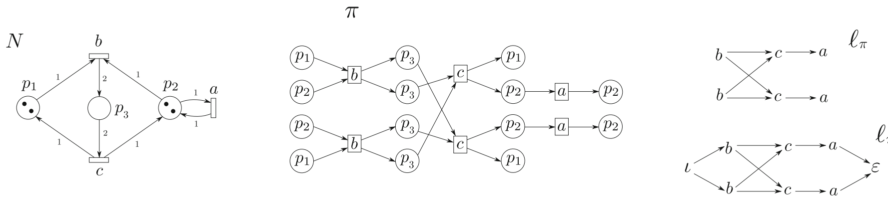


Fig. 1. A 4-bounded Petri net N. Places are denoted by circles, and transitions by rectangles. Tokens are denoted by black dots. A process π of N. The Hasse diagram of the partial order ℓ_π derived from π. The Hasse diagram of the extension ℓ_π^* of ℓ_π.

π is the partial order $\ell_\pi = (V, \boldsymbol{tc}(F)|_{V\times V}, \rho|_V)$ which is obtained by taking the transitive closure of F and subsequently by restricting $\boldsymbol{tc}(F)$ to pairs of events of V. In other words the causal order of a process π is the partial order induced by π on its events.

If $\ell = (V, <, l)$ is a partial order, then we let $\ell^* = (V', <', l')$ be the *extension* of ℓ, where $V' = V \cup \{v_\iota, v_\varepsilon\}$, $<' = < \cup (\{v_\iota\} \times V) \cup (V \times \{v_\varepsilon\}) \cup \{(v_\iota, v_\varepsilon)\}$, $l'|_V = l$, $l'(v_\iota) = \iota$ and $l'(v_\varepsilon) = \varepsilon$. In other words, ℓ' is obtained from ℓ by the addition of an element v_ι that is smaller than all elements of ℓ, and an element v_ε that is greater than all elements of ℓ. We denote by $\mathcal{P}_{cau}(N)$ the set of all extensions of partial orders derived from processes of N.

$$\mathcal{P}_{cau}(N) = \{\ell_\pi^* | \pi \text{ is a process of } N\}$$

We say that $\mathcal{P}_{cau}(N)$ is the causal language of N. We observe that several processes of N may correspond to the same partial order in $\mathcal{P}_{cau}(N)$.

2 Monadic Second Order Logic

In this section we will define two well known variants of monadic second order logic (see for instance [8]). The first, MSO_1^{po} will be used to reason about properties of partial orders. This logic extends first-order logic by allowing quantifications over sets of vertices. The second, MSO_2^{gr} will be used to reason about properties of directed acyclic graphs. This logic extends first-order logic by allowing quantifications over sets of vertices, and sets of edges.

We view a partial order ℓ as a relational structure $\ell = (V, <, l)$ where V is a set of vertices, $< \subset V \times V$ is an ordering relation, and $l \subseteq V \times T$ is a vertex labeling relation where T is a finite set of symbols (which should be regarded as the actions labeling transitions in a concurrent system). First-order variables representing individual vertices will be taken from the set $\{x_1, x_2, ...\}$ while second order variables representing sets of vertices will be taken from the set $\{X_1, X_2, ...\}$. The set of MSO_1^{po} formulas is the smallest set of formulas containing:

- the atomic formulas $x_i \in X_j$, $x_i < x_j$, $x_i = x_j$, $l(x_i, a)$ for each $i, j \in \mathbb{N}$ with $i \neq j$ and each $a \in T$,
- the formulas $\varphi \wedge \psi$, $\varphi \vee \psi$, $\neg\varphi$, $\exists x_i.\varphi(x_i)$ and $\exists X_i.\varphi(X_i)$, where φ and ψ are MSO_1^{po} formulas.

An MSO_1^{po} sentence is an MSO_1^{po} formula φ without free variables. If φ is a sentence, and $\ell = (V, <, l)$ a partial order, then we denote by $\ell \models \varphi$ the fact that ℓ satisfies φ. We let $\mathcal{P}(\varphi)$ denote the set of all partial orders satisfying φ.

We will represent a general DAG G by a relational structure $G = (V, E, s, t, l)$ where V is a set of vertices, E a set of edges, $s, t \subseteq E \times V$ are respectively the source and target relations, $l \subseteq V \times T$ is a vertex labeling relation, where T is a finite set of symbols. If e is an edge in E and v is a vertex in V then $s(e, v)$ is true if v is the source of e and $t(e, v)$ is true if v is the target of e. If $v \in V$ and $a \in T$ then $l(v, a)$ is true if v is labeled with a. First-order variables representing individual vertices will be taken from the set $\{x_1, x_2, ...\}$ and first-order variables representing edges, from the set $\{y_1, y_2, ...\}$. Second order variables representing sets of vertices will be taken from the set $\{X_1, X_2, ...\}$ and second order variables representing sets of edges, from the set $\{Y_1, Y_2, ...\}$. The set of MSO_2^{gr} formulas is the smallest set of formulas containing:

- the atomic formulas $x_i \in X_j$, $y_i \in Y_j$, $s(y_i, x_j)$, $t(y_i, x_j)$, $l(x_i, a)$ for each $i, j \in \mathbb{N}$ and $a \in T$,
- the formulas $\varphi \wedge \psi$, $\varphi \vee \psi$, $\neg\varphi$, $\exists x_i.\varphi(x_i)$ and $\exists X_i.\varphi(X_i)$, $\exists y_i.\varphi(y_i)$ and $\exists Y_i.\varphi(Y_i)$, where φ and ψ are MSO_2^{gr} formulas.

An MSO_2^{gr} sentence is a formula φ without free variables. If φ is a sentence, then we denote by $G \models \varphi$ the fact that G satisfies φ.

3 Slice Automata

In this section we define *slices* and *slice automata*. Slice automata will be used to provide a static representation of infinite families of DAGs and infinite families of partial orders. We note that slices can be related to several formalisms such as, multi-pointed graphs [13], co-span decompositions [4] and graph transformations [1,3,13,25].

A slice $\mathbf{S} = (V, E, l, s, t, [I, C, O])$ is a DAG where $V = I \,\dot\cup\, C \,\dot\cup\, O$ is a set of vertices partitioned into an in-frontier I, a center C and an out-frontier O; E is a set of edges, $s, t : E \to V$ are functions that associate with each edge $e \in E$ a source vertex e^s and a target vertex e^t, and $l : V \to T \cup \mathbb{N}$ is a function that labels the center vertices in C with elements of a finite set T, and the in- and out-frontier vertices with positive integers in such a way that $l(I) = \{1, ..., |I|\}$ and $l(O) = \{1, ..., |O|\}$. We require that each frontier-vertex v in $I \cup O$ is the endpoint of exactly one edge $e \in E$ and that the edges are directed from the in-frontier to the out frontier. More precisely, for each edge $e \in E$, we assume that $e^s \in I \cup C$ and that $e^t \in C \cup O$. For simplicity, we may omit the source and target functions s and t when specifying a slice and write simply $\mathbf{S} = (V, E, l)$. We may also speak of a slice $\mathbf{S}$ with frontiers (I, O) to indicate that the in-frontier of $\mathbf{S}$ is I and that the out-frontier of $\mathbf{S}$ is O.

A slice $\mathbf{S}_1 = (V_1, E_1, l_1)$ with frontiers (I_1, O_1) can be glued to a slice $\mathbf{S}_2 = (V_2, E_2, l_2)$ with frontiers (I_2, O_2) provided $|O_1| = |I_2|$. In this case the glueing gives rise to the slice $\mathbf{S}_1 \circ \mathbf{S}_2 = (V_3, E_3, l_3)$ with frontiers (I_1, O_2) which is obtained by taking the disjoint union of $\mathbf{S}_1$ and $\mathbf{S}_2$, and by fusing, for each $i \in \{1, ..., |O_1|\}$, the unique edge $e_1 \in E_1$ for which $l_1(e_1^t) = i$ with the unique edge $e_2 \in E_2$ for which $l_2(e_2^s) = i$. Formally, the fusion of e_1 with e_2 is performed by creating a new edge e_{12} with source $e_{12}^s = e_1^s$ and target $e_{12}^t = e_2^t$, and by deleting e_1 and e_2. Thus in the glueing process the vertices in the glued frontiers disappear.

A *unit slice* is a slice with exactly one vertex in its center. A slice is *initial* if it has empty in-frontier and *final* if it has empty out-frontier. The width of a slice $\mathbf{S}$ with frontiers (I, O) is defined as $w(\mathbf{S}) = \max\{|I|, |O|\}$. If T is a finite set of symbols, then we let $\overrightarrow{\boldsymbol{\Sigma}}(k, T)$ be the set of all unit slices of width at most k whose unique center vertex is labeled with an element of T. Observe that the alphabet $\overrightarrow{\boldsymbol{\Sigma}}(k, T)$ is finite and has asymptotically $|T| \cdot 2^{O(k \log k)}$ slices. A sequence $\mathbf{U} = \mathbf{S}_1\mathbf{S}_2...\mathbf{S}_n$ of unit slices is called a unit decomposition if $\mathbf{S}_i$ can be glued to $\mathbf{S}_{i+1}$ for each $i \in \{1, ..., n-1\}$. In this case, we let $\mathring{\mathbf{U}} = \mathbf{S}_1 \circ \mathbf{S}_2 \circ ... \circ \mathbf{S}_n$ be the DAG derived from $\mathbf{U}$, which is obtained by gluing each two consecutive slices in $\mathbf{U}$. The width of $\mathbf{U}$, denoted by $w(\mathbf{U})$, is defined as the maximum width of a slice occurring in $\mathbf{U}$.

Definition 3.1 (Slice Automaton). *Let T be a finite set of symbols and let $k \in \mathbb{N}$. A slice automaton over a slice alphabet $\overrightarrow{\boldsymbol{\Sigma}}(k, T)$ is a finite automaton $\mathcal{A} = (Q, \mathfrak{R}, q_0, F)$ where Q is a set of states, $q_0 \in Q$ is an initial state, $F \subseteq Q$ is a set of final states, and $\mathfrak{R} \subseteq Q \times \overrightarrow{\boldsymbol{\Sigma}}(k, T) \times Q$ is a transition relation such that for every $q, q', q'' \in Q$ and every $\mathbf{S} \in \overrightarrow{\boldsymbol{\Sigma}}(k, T)$:*

1. *if $(q_0, \mathbf{S}, q) \in \mathfrak{R}$ then $\mathbf{S}$ is an initial slice,*
2. *if $(q, \mathbf{S}, q') \in \mathfrak{R}$ and $q' \in F$, then $\mathbf{S}$ is a final slice,*
3. *if $(q, \mathbf{S}, q') \in \mathfrak{R}$ and $(q', \mathbf{S}', q'') \in \mathfrak{R}$, then $\mathbf{S}$ can be glued to $\mathbf{S}'$.*

Languages of a Slice Automaton. A slice automaton $\mathcal{A}$ can be used to represent three types of languages. At a syntactic level, we have the slice language $\mathcal{L}(\mathcal{A})$ which consists of the set of all unit decompositions accepted by $\mathcal{A}$ (Fig. 2).

$$\mathcal{L}(\mathcal{A}) = \{\mathbf{S}_1\mathbf{S}_2...\mathbf{S}_n \mid \mathbf{S}_1\mathbf{S}_2...\mathbf{S}_n \text{ is accepted by } \mathcal{A}\} \tag{1}$$

At a semantic level, we have the graph language $\mathcal{L}_{\mathcal{G}}(\mathcal{A})$ which consists of all DAGs represented by unit decompositions in $\mathcal{L}(\mathcal{A})$, and the partial order language $\mathcal{L}_{po}(\mathcal{A})$, which consists of all partial orders derived from DAGs in $\mathcal{L}_{\mathcal{G}}(\mathcal{A})$. If H is a DAG, we let $\boldsymbol{tc}(H)$ denote the partial order which is obtained by taking the transitive closure of H. Formally, the graph language, and the partial order languages accepted by $\mathcal{A}$ are defined as

$$\mathcal{L}_{\mathcal{G}}(\mathcal{A}) = \{\mathring{\mathbf{U}} \mid \mathbf{U} \in \mathcal{L}(\mathcal{A})\} \qquad \mathcal{L}_{po}(\mathcal{A}) = \{\boldsymbol{tc}(\mathring{\mathbf{U}}) \mid \mathring{\mathbf{U}} \in \mathcal{L}_{\mathcal{G}}(\mathcal{A})\}. \tag{2}$$

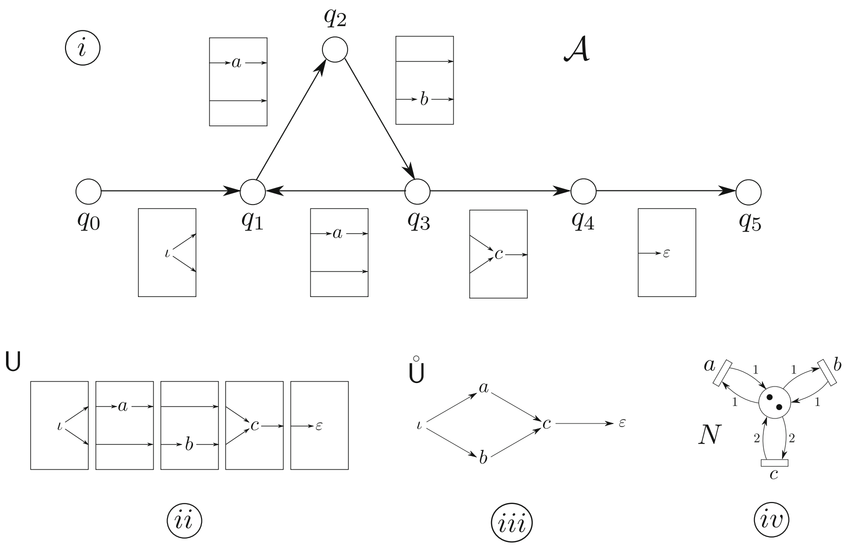


Fig. 2. (i) A slice automaton $\mathcal{A}$. (ii) A unit decomposition $\mathbf{U}$ accepted by $\mathcal{A}$. (iii) The DAG $\mathring{\mathbf{U}}$ obtained by glueing each two consecutive slices in $\mathbf{U}$. (iv) A Petri Net N. One can check that all DAGs accepted by $\mathcal{A}$ are Hasse diagrams of causal orders of N.

Saturation. Let H be a DAG whose vertices are labeled with elements from a finite set T. Then we let $\boldsymbol{ud}(H, \overrightarrow{\boldsymbol{\Sigma}}(k,T))$ denote the set of all unit decompositions $\mathbf{U}$ in $\overrightarrow{\boldsymbol{\Sigma}}(k,T)^*$ for which $\mathring{\mathbf{U}} = H$. We say that a slice automaton $\mathcal{A}$ over $\overrightarrow{\boldsymbol{\Sigma}}(k,T)$ is *saturated* if for every DAG $H \in \mathcal{L}_{\mathcal{G}}(\mathcal{A})$ we have that $\boldsymbol{ud}(H, \overrightarrow{\boldsymbol{\Sigma}}(k,T)) \subseteq \mathcal{L}(\mathcal{A})$.

4 Main Result

Our main result states that the set of partial orders representing the causal behaviour of any bounded Petri net can be defined via a monadic second order sentence. This statement is formalized in the following theorem.

Theorem 4.1 (Main Theorem). *Let N be a bounded Petri net with set of transitions T. Let $\mathcal{P}_{cau}(N)$ be the set of causal orders of N. There exists a monadic second order sentence φ_N in the vocabulary of T-labeled partial orders, such that $\mathcal{P}(\varphi_N) = \mathcal{P}_{cau}(N)$.*

The remainder of this section is dedicated to the proof of Theorem 4.1. For each $k \in \mathbb{N}$, we say that a DAG $H = (V, E)$ is k-coverable if there exists a sequence of paths $\mathfrak{p}_1 = (V_1, E_1), \mathfrak{p}_2 = (V_2, E_2), ..., \mathfrak{p}_k = (V_k, E_k)$ in H such that

$$H = \mathfrak{p}_1 \cup \mathfrak{p}_2 \cup ... \cup \mathfrak{p}_k = (\bigcup_{i=1}^{k} V_i, \bigcup_{i=1}^{k} E_i).$$

The transitive reduction of a DAG $G = (V, E, l)$ is the unique minimal subgraph $\boldsymbol{tr}(G)$ of G with the same transitive closure as G. In other words $\boldsymbol{tc}(\boldsymbol{tr}(G)) = \boldsymbol{tc}(G)$. We say that a DAG H is transitively reduced if $H = \boldsymbol{tr}(H)$. If ℓ is a partial order, then we may refer to $\boldsymbol{tr}(\ell)$ as the *Hasse diagram* of ℓ. The following lemma establishes an upper bound on the number of paths necessary to cover the Hasse diagram of a causal order of a bounded Petri net.

Lemma 4.2. *Let N be a b-bounded Petri net with n places, and let ℓ be a causal order of N. Then the Hasse diagram of ℓ^* can be covered by $b \cdot n$ paths.*

We say that a slice automaton $\mathcal{A}$ is transitively reduced if every DAG in $\mathcal{L}_{\mathcal{G}}(\mathcal{A})$ is transitively reduced. The following theorem establishes a connection between saturated, transitively reduced slice automata, and the causal behaviour of bounded Petri nets.

Theorem 4.3 (Bounded Petri Nets vs Slice Automata [9,10]**).** *Let $N = (P, T)$ be a b-bounded Petri net. Then one can construct a saturated, transitively reduced slice automaton $\mathcal{A}(N)$ over $\overrightarrow{\boldsymbol{\Sigma}}(b \cdot |P|, T)$ such that $\mathcal{L}_{po}(\mathcal{A}(N)) = \mathcal{P}_{cau}(N)$.*

We note that since $\mathcal{A}(N)$ is transitively reduced, its graph language $\mathcal{L}_{\mathcal{G}}(\mathcal{A}(N))$ is the set of Hasse diagrams of partial orders in $\mathcal{L}_{po}(\mathcal{A}(N))$. The next lemma states that if a set of k-coverable DAGs is expressible in MSO_2^{gr} logic then the set of partial orders represented by these DAGs is expressible in MSO_1^{po} logic.

Lemma 4.4. *Let k be a constant, and φ be an MSO_2^{gr} sentence defining a set of k-coverable DAGs. Then there exists an MSO_1^{po} sentence φ' such that $\ell \models \varphi'$ if and only if there exists an DAG H such that $H \models \varphi$ and $\boldsymbol{tc}(H) = \ell$.*

Proof. Any k-coverable DAG has treewidth at most k and maximum degree at most $2k$. It can be shown that if φ is an MSO_2^{gr} formula defining a set $\mathfrak{G}$ of graphs of constant treewidth and constant maximum degree, then $\mathfrak{G}$ can be defined by an MSO_1^{gr} formula φ_1, i.e., an MSO_2^{gr} formula without edge-set quantifiers [8]. Now, using the fact that transitive closure is an MSO_1^{gr} transduction (See [8], Example 1.32), such formula φ_1 can be transformed into an MSO_1^{po} formula φ' such that $\ell \models \varphi'$ iff there exists an DAG H such that $H \models \varphi_1$ and $\boldsymbol{tc}(H) = \ell$. □

As we will argue in details in Sect. 4.4, the proof of Theorem 4.1 will follow by combining Lemma 4.2, Theorem 4.3 and Lemma 4.4 with Theorem 4.5.

Theorem 4.5. *Let $\mathfrak{G}$ be a set of k-coverable DAGs. If there exists a saturated slice automaton $\mathcal{A}$ over $\overrightarrow{\boldsymbol{\Sigma}}(k, T)$ such that $\mathcal{L}_{\mathcal{G}}(\mathcal{A}) = \mathfrak{G}$, then there exists an MSO_2^{gr} sentence $\varphi_{\mathcal{A}}$ such that $\mathcal{G}(\varphi_{\mathcal{A}}) = \mathfrak{G}$.*

We will prove Theorem 4.5 in the next three subsections. The proof consists of three parts. First we note that topological orders of k-coverable DAGs can be defined in MSO logic. Subsequently, we will show that given a k-coverable DAG

G, and an ordering $\omega = (v_1, ..., v_n)$ of its vertices, one can define in MSO_2^{gr} logic a unit decomposition $\mathbf{S}_1\mathbf{S}_2...\mathbf{S}_n$ of G such that for each $i \in \{1, ..., n\}$, v_i is the center vertex of $\mathbf{S}_i$. Finally, using such MSO_2^{gr} definable unit decompositions, together with the fact that the automaton $\mathcal{A}$ is saturated, we will construct a formula $\varphi_{\mathcal{A}}$ which is true on a DAG G if and only if $G \in \mathcal{L}_{\mathcal{G}}(\mathcal{A})$. We note that this last part may be seen as a generalization of Büchi's proof that string languages recognizable by finite automata are MSO-definable.

4.1 MSO-Definable Topological Orderings

Let $\mathcal{G}$ be a class of DAGs and let $\varphi(u, v, \overrightarrow{X}, \overrightarrow{Z})$ be an MSO_2^{gr}-formula with free vertex variables u and v, free vertex-set variables $\overrightarrow{X} = (X_1, ..., X_r)$ and free edge-set variables $\overrightarrow{Z} = (Z_1, ..., Z_s)$. We say that $\varphi(u, v, \overrightarrow{X}, \overrightarrow{Z})$ orders the class $\mathcal{G}$ if for each graph $G = (V, E, \lambda)$ in $\mathcal{G}$, there exist an assignment to the variables $\overrightarrow{X}$ and $\overrightarrow{Z}$ such that the following conditions are satisfied.

- Irreflexivity: $\forall u, \neg\varphi(u, u, \overrightarrow{X}, \overrightarrow{Z})$.
- Transitivity: $\forall u, v, w,\ \varphi(u, v, \overrightarrow{X}, \overrightarrow{Z}) \wedge \varphi(v, w, \overrightarrow{X}, \overrightarrow{Z}) \rightarrow \varphi(u, w, \overrightarrow{X}, \overrightarrow{Z})$.
- Asymmetry: $\forall u, v,\ \varphi(u, v, \overrightarrow{X}, \overrightarrow{Z}) \Rightarrow \neg\varphi(v, u, \overrightarrow{X}, \overrightarrow{Z})$.
- Totality: $\forall u, v$ either $\varphi(u, v, \overrightarrow{X}, \overrightarrow{Z})$, $\varphi(v, u, \overrightarrow{X}, \overrightarrow{Z})$ or $u = v$.
- Compatibility: $\forall u, v,\ E(u, v) \rightarrow \varphi(u, v, \overrightarrow{X}, \overrightarrow{Z})$.

Intuitively, for each graph G in $\mathcal{G}$ there is an assignment of the variables $\overrightarrow{X}$ and $\overrightarrow{Z}$ such that the formula $\varphi(u, v, \overrightarrow{X}, \overrightarrow{Z})$ defines a topological ordering on the vertices of G. Note that distinct assignments to the variables $\overrightarrow{X}, \overrightarrow{Z}$ may cause $\varphi(u, v, \overrightarrow{X}, \overrightarrow{Z})$ to define distinct topological orderings on the set of vertices of G.

Lemma 4.6. *For each $k \in \mathbb{N}$, there is an* MSO_2^{gr}*-formula $\varphi_k(u, v, \overrightarrow{X}, \overrightarrow{Z})$ which orders the class of k-coverable DAGs.*

4.2 Defining Unit Decompositions in MSO Logic

Let $G = (V, E, \lambda)$ be a DAG, and let V_1 and V_2 be disjoint subsets of V. We denote by $E(V_1, V_2)$ the set of all edges with one endpoint in V_1 and another endpoint in V_2. Let $\omega : V \times V$ be a topological ordering of the vertices of G. We let $\text{Sm}(\omega, v) = \{v' \mid \omega(v', v)\}$ be the set of vertices that are strictly smaller than v according to the ordering ω. Analogously, we let $\text{Gr}(\omega, v) = \{v' \mid \omega(v, v')\}$ be the set of vertices that are strictly greater than v according to ω. The width of ω is defined as $\mathbf{w}(\omega) = \max_v |E(\text{Sm}(\omega, v) \cup \{v\}, \text{Gr}(v))|$. For each $v \in V$, define the following set.

$$E(\omega, v) \ = \ E(\text{Sm}(\omega, v), \{v\}) \ \cup \ E(\{v\}, \text{Gr}(\omega, v)) \ \cup \ E(\text{Sm}(\omega, v), \text{Gr}(\omega, v)).$$

Intuitively, $E(\omega, v)$ is the set of all edges incident with the vertex v, together with all edges with source in $\text{Sm}(\omega, v)$ and target in $\text{Gr}(\omega, v)$. We note that if ω

has width k, then $|E(\omega, v)| \leq 2k$ for each $v \in V$. We also note that if v is the minimal element of ω, or the maximal element of ω, then $E(\omega, v)$ is simply the set of edges incident with v, and in this case, $|E(\omega, v)| \leq k$.

Definition 4.7 (Well Coloring). *Let $G = (V, E, \lambda)$ be a DAG and $\omega \subseteq V \times V$ be a topological ordering of G of width k. A well coloring of G with respect to ω is a partition $\chi = (Y_1, ..., Y_r)$ of the edges of G such that for each $v \in V$, and each $l \in \{1, ..., r\}$, $|Y_l \cap E(\omega, v)| \leq 1$.*

In other words, each edge of G receives a unique color from $\{1, ..., r\}$ and for each $v \in V$, each two distinct edges in $E(\omega, v)$ have distinct colors. The following proposition states that if ω is a topological ordering of G of width k, then G can be well colored with $2k$ colors.

Proposition 4.8. *Let $G = (V, E, \lambda)$ be a DAG, and $\omega \subseteq V \times V$ be a topological ordering of G of width k. Then there exists a well coloring $\chi = (Y_1, ..., Y_{2k})$ of G with respect to ω.*

Proof. For each $v \in V$, let $\hat{E}(\omega, v) = E(\omega, v) \cup \bigcup_{v' \in \mathrm{Sm}(\omega, v)} E(\omega, v')$. We say that a partition $\chi = (Y_1, ..., Y_{2k})$ of the set $\hat{E}(\omega, v)$ is a well coloring of $\hat{E}(\omega, v)$ with respect to ω if for each $v' \in \mathrm{Sm}(\omega, v)$, and each $l \in \{1, ..., 2k\}$, $|Y_l \cap E(\omega, v')| \leq 1$. We will show that for each v, one can construct a well coloring χ_v of $\hat{E}(\omega, v)$ with respect to ω. In particular, if v is the last element of the ordering ω, then $\chi = \chi_v$ is a well coloring of G with respect to ω. The proof is by induction on the size of $\mathrm{Sm}(\omega, v)$. In the base case, $|\mathrm{Sm}(\omega, v)| = 0$ and therefore $|\hat{E}(\omega, v)| = |E(\omega, v)| \leq k$. We let $\chi_v = (Y_1^0, ..., Y_{2k}^0)$ be any partition of $E(\omega, v)$ in which each set Y_l^0 has size at most 1. Now assume that $|\mathrm{Sm}(v)| = i$ and that $\chi_v = (Y_1^i, ..., Y_{2k}^i)$ is a well coloring of $\hat{E}(\omega, v)$ with respect to ω. Let u be the vertex of V with $|\mathrm{Sm}(u)| = i + 1$. Let $\chi_u = (Y_1^{i+1}, ..., Y_{2k}^{i+1})$ be a partition of $\hat{E}(\omega, u)$ such that $Y_l^i \subseteq Y_l^{i+1}$ and $|E(\omega, u) \cap Y_l^{i+1}| \leq 1$ for each $l \in \{1, ..., 2k\}$. Such a partition χ_u exists since $|E(\omega, u)| \leq 2k$. Additionally, χ_u is by construction a well coloring of $\hat{E}(\omega, u)$ with respect to ω. □

We let $\mathrm{WellColoring}(\overrightarrow{X}, \overrightarrow{Z}, Y_1, ..., Y_{2k})$ be an MSO_2^{gr} predicate with free vertex-set variables $\overrightarrow{X}$ and free edge-set variables $\overrightarrow{Z}$ and $Y_1, ..., Y_{2k}$ which is true on a DAG G if and only if $Y_1, ..., Y_{2k}$ is a well coloring of G with respect to the topological ordering defined by the formula $\varphi_k(u, v, \overrightarrow{X}, \overrightarrow{Z})$ on the vertices of G.

Our interest in well colorings stems from the fact that given a topological ordering ω of width k of a DAG $G = (V, E, \lambda)$, together with a well coloring $\chi : E \rightarrow \{1, ..., 2k\}$ of the vertices of G, one can implicitly define in MSO_2^{gr} logic a unit decomposition $\mathbf{U}(\omega, \chi)$ of G of width k. To define such unit decomposition it is enough to define each of its slices.

Definition 4.9. *Let $G = (V, E, \lambda)$ be a DAG, ω be a topological ordering of G of width k, and $\chi : E \rightarrow \{1, ..., 2k\}$ be a well coloring of the edges of G with respect to ω. For each $v \in V$, we denote by $\mathbf{S}(v, \omega, \chi) = (V_v, E_v, \lambda_v, [I_v, C_v, O_v])$ the unique unit slice in $\overrightarrow{\mathbf{\Sigma}}(k, T)$ satisfying the following conditions.*

1. *The in-frontier I_v has $\alpha = |E(\mathrm{Sm}(\omega, v), \{v\} \cup \mathrm{Gr}(\omega, v))|$ vertices. For each $i \in \{1, ..., \alpha\}$, let x_i^I be the unique vertex of I_v with $\lambda_v(x_i^I) = i$.*
2. *The out-frontier O_v has $\beta = |E(\mathrm{Sm}(\omega, v) \cup \{v\}, \mathrm{Gr}(\omega, v))|$ vertices. For each $i \in \{1, ..., \beta\}$, we let x_i^O be the unique vertex of O_v with $\lambda_v(x_i^O) = i$.*
3. *The center C_v has a unique vertex x^C. Additionally, $\lambda_v(x^C) = \lambda(v)$.*
4. *There is a bijection $\beta : E_v \rightarrow E(\omega, v)$ satisfying the following conditions.*
   (a) $e \in E_v \ \wedge \ e^s = x^C \ \Rightarrow \ \beta(e)^s = v$.
   (b) $e \in E_v \ \wedge \ e^t = x^C \ \Rightarrow \ \beta(e)^t = v$.
   (c) For each $i, j \in [\alpha]$ with $i < j$,
   $$e_1, e_2 \in E_v \ \wedge e_1^s = x_i^I \ \wedge \ e_2^s = x_j^I \Rightarrow \chi(e_1) < \chi(e_2).$$
   (d) For each $i, j \in [\beta]$ with $i < j$,
   $$e_1, e_2 \in E_v \ \wedge e_1^t = x_i^O \ \wedge \ e_2^t = x_j^O \Rightarrow \chi(e_1) < \chi(e_2).$$

Intuitively, the ordering ω determines which edges are present in the slice $\mathbf{S}(v, \omega, \chi)$ while the coloring χ determines the numbering of the in-frontier vertices and the numbering of the out-frontier vertices of $\mathbf{S}(v, \omega, \chi)$. Now let $V = \{v_1, ..., v_n\}$ be such that v_i is the i-th vertex in the ordering ω. Then we have that

$$\mathbf{U}(\omega, \chi) = \mathbf{S}(v_1, \omega, \chi)\mathbf{S}(v_2, \omega, \chi)...\mathbf{S}(v_n, \omega, \chi)$$

is a unit decomposition of G of width k. We say that $\mathbf{U}(\omega, \chi)$ is the unit decomposition of G induced by the pair (ω, χ).

For each slice $\mathbf{S} \in \overrightarrow{\boldsymbol{\Sigma}}(k, T)$ we let $\hat{\mathbf{S}}(v, \overrightarrow{X}, \overrightarrow{Z}, Y_1, ..., Y_{2k})$ be a predicate which is true on a DAG G if and only if $\mathbf{S} = \mathbf{S}(v, \omega, \chi)$ where ω is the ordering defined by $\varphi_k(u, v, \overrightarrow{X}, \overrightarrow{Z})$ on the vertices of G, and $\chi = (Y_1, ..., Y_{2k})$ is a well coloring of the edges of G with respect to ω. Clearly, the predicate $\hat{\mathbf{S}}(v, \overrightarrow{X}, \overrightarrow{Z}, Y_1, ..., Y_{2k})$ can be defined in MSO_2^{gr} logic using Definition 4.9.

4.3 Generalizing Büchi's Theorem

Finally, in this subsection we show that if $\mathcal{A} = (Q, \mathfrak{R}, q_0, F)$ is a saturated slice automaton over $\overrightarrow{\boldsymbol{\Sigma}}(k, T)$ recognizing a set of k-coverable DAGs $\mathcal{L}_{\mathcal{G}}(\mathcal{A})$, then one can define an MSO_2^{gr} sentence $\varphi_{\mathcal{A}}$ such that $\mathcal{G}(\varphi_{\mathcal{A}}) = \mathcal{L}_{\mathcal{G}}(\mathcal{A})$. We will need the following proposition, which states that if H is a k-coverable DAG, then any unit decomposition of H has width at most k.

Proposition 4.10. *Let H be a k-coverable DAG. Then for each $k' \geq k$,*

$$\boldsymbol{ud}(H, \overrightarrow{\boldsymbol{\Sigma}}(k', T)) = \boldsymbol{ud}(H, \overrightarrow{\boldsymbol{\Sigma}}(k, T)).$$

Proof. Let $H = \mathfrak{p}_1 \cup ... \cup \mathfrak{p}_k$ where $\mathfrak{p}_i$ are paths. Let $\mathbf{U} = \mathbf{S}_1\mathbf{S}_2...\mathbf{S}_n$ be a unit decomposition of H. Each path $\mathfrak{p}_i$ crosses each frontier of each slice in $\mathbf{U}$ at most one time. Therefore, all k paths together cross each frontier of each slice in $\mathbf{U}$ at most k times. □

Intuitively, given a k-coverable DAG H, the sentence $\varphi_{\mathcal{A}}$ guesses both a topological ordering $\omega = (v_1, v_2, ..., v_n)$ of the vertices of H, and a well coloring χ of H with respect to ω. The ordering ω, and the well coloring χ uniquely determine a unit decomposition

$$\mathbf{U}(\omega, \chi) = \mathbf{S}(v_1, \omega, \chi)\mathbf{S}(v_2, \omega, \chi)...\mathbf{S}(v_n, \omega, \chi)$$

of H. Since H can be covered by k paths, Proposition 4.10 implies that any unit decomposition of H has width at most k. In particular, $\mathbf{U}(\omega, \chi)$ has width at most k. Since $\mathcal{A}$ is saturated over $\overrightarrow{\boldsymbol{\Sigma}}(k, T)$, we have that the guessed unit decomposition $\mathbf{U}(\omega, \chi)$ belongs to $\mathcal{L}(\mathcal{A})$.

Let $Q = \{q_0, q_1, ..., q_n\}$ be the set of states of $\mathcal{A}$, where q_0 is the initial state. Our formula will encode an accepting run of the slice automaton $\mathcal{A}$ on the unit decomposition $\mathbf{U}(\omega, \chi)$. More precisely, the set of vertices of H is partitioned into $|Q|$ subsets $X_{q_0}, X_{q_1}, ..., X_{q_n}$. The vertex v_j belongs to the subset X_{q_i} if and only if $\mathcal{A}$ reaches state q_i after reading the prefix

$$\mathbf{S}(v_1, \omega, \chi)\mathbf{S}(v_2, \omega, \chi)...\mathbf{S}(v_j, \omega, \chi)$$

of the unit decomposition $\mathbf{U}(\omega, \chi)$.

Below, we write $\exists\overrightarrow{X}$ to denote $\exists X_1, X_2, ..., X_r$, $\exists\overrightarrow{Z}$ to denote $\exists Z_1, ..., Z_s$, and $\exists\overrightarrow{Y}$ to denote $\exists Y_1, ..., Y_{2k}$. Let $\mathrm{Unique}(x, W_{q_0}, W_{q_1}, ..., W_{q_n})$ be a predicate that is true if the vertex x belongs to exactly one of the vertex sets

$$W_{q_0}, W_{q_1}, ..., W_{q_n}.$$

Let $\mathrm{First}(u, \overrightarrow{X}, \overrightarrow{Z})$ be a predicate that is true if u is the first vertex of the ordering defined by $\varphi_k(u, v, \overrightarrow{X}, \overrightarrow{Z})$ Analogously, let $\mathrm{Last}(u, \overrightarrow{X}, \overrightarrow{Z})$ be a predicate that is true if u is the last vertex of the ordering defined by $\varphi_k(u, v, \overrightarrow{X}, \overrightarrow{Z})$. Finally, let $\mathrm{Suc}(u, v, \overrightarrow{X}, \overrightarrow{Z})$ be a predicate that states that u is an immediate predecessor of v in the ordering $\varphi_k(u, v, \overrightarrow{X}, \overrightarrow{Z})$. Then the discussion above can be synthesized by the following MSO formula which is true on a DAG H if and only if H is accepted by $\mathcal{A}$.

$$\begin{aligned}
&\exists\overrightarrow{X}\ \exists\overrightarrow{Z}\ \exists\overrightarrow{Y}\ \exists W_{q_0}, W_{q_1}, ..., W_{q_n}\\
&\quad \mathrm{WellColoring}(\overrightarrow{X}, \overrightarrow{Z}, \overrightarrow{Y})\ \wedge\\
&\quad \forall x\quad \mathrm{Unique}(x, W_{q_0}, W_{q_1}, ..., W_{q_n}) \wedge [\mathrm{First}(x, \overrightarrow{X}, \overrightarrow{Z}) \Rightarrow W_{q_0}(x)]\ \wedge\\
&\quad \forall u\, \forall v\, \Big[\ \mathrm{Suc}(u, v, \overrightarrow{X}, \overrightarrow{Z}) \Rightarrow\ \bigvee\nolimits_{(q_i, \mathbf{S}, q_j) \in \mathfrak{R}}\ W_{q_i}(u) \wedge \hat{\mathbf{S}}(u, \overrightarrow{X}, \overrightarrow{Z}, \overrightarrow{Y}) \wedge W_{q_j}(v)\ \Big]\ \wedge\\
&\quad \forall u\, \Big[\ \mathrm{Last}(u, \overrightarrow{X}, \overrightarrow{Z}) \Rightarrow\ \bigvee\nolimits_{q_j \in F,\ (q_i, a, q_j) \in \mathfrak{R}}\ W_{q_i}(u) \wedge \hat{\mathbf{S}}(u, \overrightarrow{X}, \overrightarrow{Z}, \overrightarrow{Y})\ \Big].
\end{aligned}$$

4.4 Proof of Theorem 4.1

Let $N = (P, T)$ be b-bounded Petri Net. By Theorem 4.3, one can construct a saturated, transitively reduced slice automaton $\mathcal{A}(N)$ over the slice alphabet $\overrightarrow{\boldsymbol{\Sigma}}(b \cdot |P|, T)$ such that $\mathcal{L}_{po}(\mathcal{A}(N)) = \mathcal{P}_{cau}(N)$. Additionally, since $\mathcal{A}$ is transitively reduced, a DAG H belongs to $\mathcal{L}_{\mathcal{G}}(\mathcal{A}(N))$ if and only if H is the Hasse diagram of some partial order in $\mathcal{P}_{cau}(N)$. Now, by Theorem 4.5, we can construct from $\mathcal{A}(N)$ an MSO_2^{gr} formula $\varphi_{\mathcal{A}(N)}$ such that H satisfies $\varphi_{\mathcal{A}(N)}$ if and only if $H \in \mathcal{L}_{\mathcal{G}}(\mathcal{A}(N))$. As a last step, we use Lemma 4.4 to convert $\varphi_{\mathcal{A}(N)}$ into an MSO_1^{po} formula φ_N which is true on a partial order ℓ if and only if there exists a DAG H such that $H \models \varphi_{\mathcal{A}(N)}$ and $\boldsymbol{tc}(H) = \ell$. In other words, $\ell \models \varphi_N$ if and only if $\ell \in \mathcal{L}_{po}(\mathcal{A}(N)) = \mathcal{P}_{cau}(N)$. $\square$

5 Conclusion and Open Problems

In this work we have shown that the causal behaviour of any bounded Petri net N can be specified by an MSO logic sentence φ_N (Theorem 4.1). As a crucial step towards the proof of Theorem 4.1, we have shown that sets of k-coverable DAGs recognizable by saturated slice automata are MSO-definable (Theorem 4.5). The converse of Theorem 4.5, which states that MSO-definable sets of k-coverable graphs are recognizable by saturated slice automata has been proved in [11]. Together, these two results yield the following corollary.

Corollary 5.1. *A set $\mathfrak{G}$ of k-coverable DAGs is definable in MSO_2^{gr} logic if and only if $\mathfrak{G}$ is recognizable by a saturated slice automaton.*

Corollary 5.1 extends Büchi's theorem from the context of strings to the context of k-coverable DAGs (Note that a string may be regarded as a 1-coverable DAG). An interesting open problem is to determine whether saturated slice automata can be used to extend Büchi's theorem to larger classes of graphs, such as graphs of bounded cutwidth.

Acknowledgements. The author gratefully acknowledges financial support from the European Research Council, ERC grant agreement 339691, within the context of the project Feasibility, Logic and Randomness (FEALORA).

References

1. Bauderon, M., Courcelle, B.: Graph expressions and graph rewritings. Math. Syst. Theor. **20**(2–3), 83–127 (1987)
2. Bodlaender, H.L., Heggernes, P., Telle, J.A.: Recognizability equals definability for graphs of bounded treewidth and bounded chordality. In: Proceedings of the 7th European Conference on Combinatorics (EUROCOMB 2015) (2015)
3. Brandenburg, F.-J., Skodinis, K.: Finite graph automata for linear and boundary graph languages. Theor. Comput. Sci. **332**(1–3), 199–232 (2005)

4. Bruggink, H.S., König, B.: On the recognizability of arrow and graph languages. In: Ehrig, H., Heckel, R., Rozenberg, G., Taentzer, G. (eds.) ICGT 2008. LNCS, vol. 5214. Springer, Heidelberg (2008)
5. Büchi, J.R.: Weak second order arithmetic and finite automata. Z. Math. Logik Grundl. Math. **6**, 66–92 (1960)
6. Courcelle, B.: The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. Inf. Comput. **85**(1), 12–75 (1990)
7. Courcelle, B.: The monadic second-order logic of graphs V: on closing the gap between definability and recognizability. Theor. Comput. Sci. **80**(2), 153–202 (1991)
8. Courcelle, B., Engelfriet, J.: Graph Structure and Monadic Second-Order Logic: A Language-Theoretic Approach, vol. 138. Cambridge University Press, Cambridge (2012)
9. de Oliveira Oliveira, M.: Hasse diagram generators and Petri nets. Fundamenta Informaticae **105**(3), 263–289 (2010)
10. de Oliveira Oliveira, M.: Canonizable partial order generators. In: Dediu, A.-H., Martín-Vide, C. (eds.) LATA 2012. LNCS, vol. 7183, pp. 445–457. Springer, Heidelberg (2012)
11. de Oliveira Oliveira, M.: Subgraphs satisfying MSO properties on z-topologically orderable digraphs. In: Gutin, G., Szeider, S. (eds.) IPEC 2013. LNCS, vol. 8246, pp. 123–136. Springer, Heidelberg (2013)
12. de Oliveira Oliveira, M.: MSO logic and the partial order semantics of place/transition-nets. In: Leucker, M., Rueda, C., Valencia, F.D. (eds.) ICTAC 2015. LNCS, vol. 9399, pp. 368–387. Springer, Heidelberg (2015). doi:10.1007/978-3-319-25150-9_22
13. Engelfriet, J., Vereijken, J.J.: Context-free graph grammars and concatenation of graphs. Acta Informatica **34**(10), 773–803 (1997)
14. Gaifman, H., Pratt, V.R.: Partial order models of concurrency and the computation of functions. In: LICS 1987, pp. 72–85 (1987)
15. Gischer, J.L.: The equational theory of pomsets. Theor. Comput. Sci. **61**, 199–224 (1988)
16. Goltz, U., Reisig, W.: Processes of place/transition-nets. In: Díaz, J. (ed.) ICALP 1983. LNCS, vol. 154, pp. 264–277. Springer, Heidelberg (1983)
17. Jaffke, L., Bodlaender, H.L.: Definability equals recognizability for k-outerplanar graphs. In: Proceedings of the 10th International Symposium on Parameterized and Exact Computation, (IPEC 2015). LIPIcs, vol. 43, pp. 175–186 (2015)
18. Jaffke, L., Bodlaender, H.L.: MSOL-definability equals recognizability for Halin graphs and bounded degree k-outerplanar graphs (2015). Preprint arXiv:1503.01604
19. Jagadeesan, L.J., Jagadeesan, R.: Causality and true concurrency: a data-flow analysis of the pi-calculus. In: Alagar, V.S., Nivat, M. (eds.) AMAST 1995. LNCS, vol. 936, pp. 277–291. Springer, Heidelberg (1995)
20. Kabanets, V.: Recognizability equals definability for partial k-paths. In: Degano, P., Gorrieri, R., Marchetti-Spaccamela, A. (eds.) ICALP 1997. LNCS, vol. 1256, pp. 805–815. Springer, Heidelberg (1997)
21. Kaller, D.: Definability equals recognizability of partial 3-trees and k-connected partial k-trees. Algorithmica **27**(3–4), 348–381 (2000)
22. Langerak, R., Brinksma, E., Katoen, J.-P.: Causal ambiguity and partial orders in event structures. In: Mazurkiewicz, A., Winkowski, J. (eds.) CONCUR 1997. LNCS, vol. 1243, pp. 317–331. Springer, Heidelberg (1997)

23. Lapoire, D.: Recognizability equals monadic second-order definability for sets of graphs of bounded tree-width. In: Morvan, M., Meinel, C., Krob, D. (eds.) STACS 1998. LNCS, vol. 1373, pp. 618–628. Springer, Heidelberg (1998)
24. Petri, C.A.: Fundamentals of a theory of asynchronous information flow. In: Proceedings of IFIP Congress 62, Munchen, pp. 166–168 (1962)
25. Thomas, W.: Finite-state recognizability of graph properties. Theorie des Automates et Applications **172**, 147–159 (1992)
26. Vogler, W. (ed.): Modular Construction and Partial Order Semantics of Petri Nets. LNCS, vol. 625. Springer, Heidelberg (1992)

A Multi-type Calculus for Inquisitive Logic

Sabine Frittella[1(✉)], Giuseppe Greco[1], Alessandra Palmigiano[1,2], and Fan Yang[1]

[1] Faculty of Technology, Policy and Management, Delft University of Technology, Delft, The Netherlands
{S.S.A.Frittella,G.Greco,A.Palmigiano,F.Yang}@tudelft.nl
[2] Department of Pure and Applied Mathematics, University of Johannesburg, Johannesburg, South Africa

Abstract. In this paper, we define a multi-type calculus for inquisitive logic, which is sound, complete and enjoys Belnap-style cut-elimination and subformula property. Inquisitive logic is the logic of inquisitive semantics, a semantic framework developed by Groenendijk, Roelofsen and Ciardelli which captures both assertions and questions in natural language. Inquisitive logic adopts the so-called support semantics (also known as team semantics). The Hilbert-style presentation of inquisitive logic is not closed under uniform substitution, and some axioms are sound only for a certain subclass of formulas, called flat formulas. This and other features make the quest for analytic calculi for this logic not straightforward. We develop a certain algebraic and order-theoretic analysis of the team semantics, which provides the guidelines for the design of a multi-type environment accounting for two domains of interpretation, for flat and for general formulas, as well as for their interaction. This multi-type environment in its turn provides the semantic environment for the multi-type calculus for inquisitive logic we introduce in this paper.

1 Introduction

Inquisitive logic is the logic of inquisitive semantics, a semantic framework, introduced by Groenendijk, Roelofsen and Ciardelli in [5,13], that captures both assertions and questions in natural language. In this framework, formulas express proposals to enhance the common ground of a conversation. The inquisitive content of a formula is understood as an issue raised by a given utterance. A distinguishing feature of inquisitive logic is that formulas are evaluated on *information states*, i.e., at sets of possible worlds, rather than at single possible worlds. Inquisitive logic defines a relation of *support* between information states and formulas, the intended understanding of which is that in uttering a sentence, a speaker proposes to enhance the current common ground to one that supports the sentence.

Closely related to inquisitive logic is *dependence logic* [22], which is an extension of classical logic that characterizes the notion of "dependence" between variables using the so-called *team semantics*, introduced by Hodges [14,15]. The team

J. Väänänen et al. (Eds.): WoLLIC 2016, LNCS 9803, pp. 215–233, 2016.
DOI: 10.1007/978-3-662-52921-8_14

semantics of dependence logic builds on the basis of the notion of *team*, which, in the propositional logic context, is a *set* of assignments of atomic propositions into the 2-element Boolean algebra. As is well known, each such assignment can be identified with a set of atomic propositions. Therefore, information states providing the semantic framework for inquisitive logic can be identified with teams, providing the semantic framework for dependence logic. In fact, this is more than a peculiar coincidence: In [23], systematic connections were developed between inquisitive logic and dependence logic on the basis of the identification between information state semantics and team semantics, and it was shown that inquisitive logic is essentially a variant of propositional dependence logic [24] with the intuitionistic connectives introduced in [1]. It was further argued in [3,4] that the entailment relation of questions is a type of dependency relation considered in dependence logic.

Inquisitive logic was axiomatized in [5]. This axiomatization is not closed under uniform substitution, which is one of the main hurdles for a standard proof-theoretic treatment. No previous proposal for a sequent calculus for the inquisitive logic of [5] exists. The only relevant work of the present contribution is the labelled calculus defined in [21] for the logic of the inquisitive pair semantics introduced in [12,18]. Inquisitive pair semantics is a predecessor of inquisitive semantics and it does not give rise to the same logic as the inquisitive logic of [5]. The calculus in [21] makes use of extra linguistic labels which import pair inquisitive semantics into the calculus. This calculus is sound, complete and cut free; however, since the interpretation of the sequents is ad hoc, only a semantic proof of cut elimination is given, and this set-up is not easily transferable to the inquisitive logic of [5].

Our contribution is a calculus designed on different principles than those of [21], and for the current version of inquisitive logic, which is based on support semantics. We tackle the hurdle of the non schematicity of the Hilbert-style presentation by designing the calculus for inquisitive logic in the style of a generalization of Belnap's display calculi, the so-called *multi-type calculi*. These calculi have been introduced in [6,7], as a proposal to support a proof-theoretic semantic account of Dynamic Logics [9]. One important aspect of multi-type calculi is that various Belnap-style metatheorems have been given, which allow for a smooth syntactic proof of cut elimination.

The multi-type environment we propose is motivated by an order-theoretic analysis of the team semantics for inquisitive logic, according to which, certain maps can be defined which make it possible for the different types to interact. The non schematicity of the axioms is accounted for by assigning different types to the restricted formulas and to the general formulas. Hence, closure under arbitrary substitution holds *within each type*.

Structure of the Paper. In Sect. 2, needed preliminaries are collected on inquisitive logic. In Sect. 3, the order-theoretic analysis is given, which justifies the introduction of an expanded multi-type language, into which the original language of inquisitive logic can be embedded. In Sect. 4, the multi-type calculus for (the multi-type version of) inquisitive logic is introduced. In Sect. 5, we prove the soundness and completeness of the calculus, and we also show that the calculus is powerful enough to capture the restricted type (i.e. the flat type) proof-theoretically. In Sect. 6, we give a syntactic proof of cut elimination Belnap-style.

2 Inquisitive Logic

In the present section, we briefly recall the basics of inquisitive logic and support semantics (or team semantics). The reader is refer to [3,5] and also to [24] for an expanded treatment.

Although the support semantics (or team semantics) is originally developed for the extension of classical propositional logic with questions, for the sake of a better compatibility with the exposition in the next sections, we will first define support semantics (or team semantics) for classical propositional logic. Let us fix a set V of propositional variables and denote its elements by $p, q, r, \ldots$ possibly sub- or super-scripted. Well-formed formulas of *classical propositional logic* (**CPL**), also called *classical formulas*, are given by the following grammar:

$$\chi ::= p \mid 0 \mid \chi \wedge \chi \mid \chi \to \chi.$$

A *possible world* (or a *valuation*) is a map $v : V \to 2$, where $2 := \{0, 1\}$. An information *state* (also called a *team*) is a set of possible worlds.

Definition 1. *The* support *relation of a classical formula χ on a state S, denoted by $S \models \chi$, is defined recursively as follows:*

$$\begin{array}{rll} S \models p & \textit{iff} & v(p) = 1 \textit{ for all } v \in S \\ S \models 0 & \textit{iff} & S = \varnothing \\ S \models \chi \wedge \xi & \textit{iff} & S \models \chi \textit{ and } S \models \xi \\ S \models \chi \to \xi & \textit{iff} & \textit{for any } S' \subseteq S,\ \textit{if } S' \models \chi,\ \textit{then } S' \models \xi \end{array}$$

An easy inductive proof shows that classical formulas χ are *flat* (also called *truth conditional*); that is, for every state S,

(Flatness Property) $S \models \chi$ iff $\{v\} \models \chi$ for all $v \in S$.

Well-formed formulas ϕ of *inquisitive logic* (**InqL**) are given by expanding the language of **CPL** with the inquisitive disjunction $\vee$. Equivalently, these formulas can be defined by the following recursion:

$$\phi ::= \chi \mid \phi \wedge \phi \mid \phi \to \phi \mid \phi \vee \phi.$$

As usual, we write $\neg\chi$ for $\chi \to 0$. This two-layered presentation is slightly different but equivalent to the usual one. The reason why we are presenting it this way will be clear at the end of the following section, when we introduce a translation of **InqL**-formulas into a multi-type language.

Definition 2. *The* support relation *of an* **InqL**-*formula* ϕ *on an information state* S*, denoted by* $S \models \phi$*, is defined analogously to the support relation of classical formulas relative to the fragment shared by the two languages, and moreover:*

$$S \models \phi \vee \psi \quad \textit{iff} \quad S \models \phi \textit{ or } S \models \psi$$

We write $\phi \models \psi$ *if, for any state* S*, if* $S \models \phi$ *then* $S \models \psi$*. If both* $\phi \models \psi$ *and* $\psi \models \phi$*, then we write* $\phi \equiv \psi$*. An* **InqL**-*formula* ϕ *is* valid*, denoted by* $\models \phi$*, if* $S \models \phi$ *holds for all states* S*. The* logic **InqL** *is the set of all valid* **InqL**-*formulas.*

An easy inductive proof shows that **InqL**-formulas have the downward closure property and the empty state property:

(Downward Closure Property) If $S \models \phi$ and $S' \subseteq S$, then $S' \models \phi$.
(Empty State Property) $\varnothing \models \phi$.

CPL extended with the dependence atoms $=(p_1, \ldots, p_n, q)$ is called *propositional dependence logic* (**PD**), which is an important variant of **InqL**. The logic **PD** adopts also the support semantics (or the team semantics). It is proved in [24] that **PD** has the same expressive power as **InqL**. In particular, a constancy dependence atom $=(p)$ is semantically equivalent to the formula $p \vee \neg p$, which expresses the *polar question* 'whether p?' (denoted $?p$), and a dependence atom $=(p_1, \ldots, p_n, q)$ with multiple arguments is semantically equivalent to the entailment $?p_1 \wedge \cdots \wedge ?p_n \to ?q$ of polar questions. For more details on this connection, we refer the reader to [3,4].

Flat formulas will play an important role in this paper. Below we list some of their properties.

Lemma 3 (see [3]). *For all* **InqL**-*formulas* ϕ *and* ψ*,*

(a) If ψ *is flat, then* $\phi \to \psi$ *is flat. In particular,* $\neg\phi$ *is always flat.*
(b) The following are equivalent:
   1. ϕ *is flat.*
   2. $\phi \equiv \phi^{\mathsf{f}}$*, where* ϕ^{f} *is the classical formula obtained from* ϕ *by replacing every occurrence of* $\phi_1 \vee \phi_2$ *in* ϕ *by* $\neg\phi_1 \to \phi_2$*.*
   3. $\phi \equiv \neg\neg\phi$*.*

Below we list some meta-logical properties of **InqL**; for the proof, see [5]. For any set $\Gamma \cup \{\phi, \psi\}$ of **InqL**-formulas:

(Deduction Theorem) $\Gamma, \phi \models \psi$ if and only if $\Gamma \models \phi \to \psi$.
(Disjunction Property) If $\models \phi \vee \psi$, then either $\models \phi$ or $\models \psi$.
(Compactness) If $\Gamma \models \phi$, then there exists a finite subset Δ of Γ such that $\Delta \models \phi$.

Theorem 4 (see [3,5]). *The following Hilbert-style system of* **InqL** *is sound and complete.*

Axioms:

1. *all substitution instances of axioms of intuitionistic propositional logic* **IPL**
2. $(\chi \to (\phi \vee \psi)) \to (\chi \to \phi) \vee (\chi \to \psi)$ *whenever* χ *is a classical formula*
3. $\neg\neg\chi \to \chi$ *whenever* χ *is a classical formula*

Rule:

Modus Ponens: $\dfrac{\phi \to \psi \qquad \psi}{\psi}$ *(*MP*)*

Note that the above system of **InqL** does not have the Substitution Rule: $\dfrac{\phi}{\phi(\psi/p)}$ and the logic **InqL** is *not* closed under uniform substitution. In particular, axiom 3 in the above system is in general not valid for non-classical formulas, as, for instance, $\neg\neg?p \to ?p$ is not valid.

Each classical formula χ is equivalent to a negated formula $\neg\neg\chi$ (Theorem 3(b)). Therefore axiom 2 in the above system can be viewed as a variant of the KP axiom $(\neg p \to (q \vee r)) \to (\neg p \to q) \vee (\neg p \to r)$ of the Kreisel-Putnam logic KP [16]. There is actually an interesting connection between **InqL** and intermediate logics[1]. Let $\mathsf{L}^{\neg} = \{\phi \mid \phi^{\neg} \in \mathsf{L}\}$ be the *negative variant* of an intermediate logic L, where $\phi^{\neg}$ is obtained from ϕ by replacing any occurrence of a propositional variable p with $\neg p$. It was proved in [5] that **InqL** coincides with the negative variant of every intermediate logic that is between Maksimova's logic ND [17] and Medvedev's logic ML [19], such as the Kreisel-Putnam logic KP [16]. That is, $\mathsf{L}^{\neg} = \mathbf{InqL}$ for all L such that $\mathsf{ND} \subseteq \mathsf{L} \subseteq \mathsf{ML}$, and $\mathbf{InqL} = \mathsf{KP}^{\neg} = \mathsf{ND}^{\neg} = \mathsf{ML}^{\neg}$ in particular.

3 Order-Theoretic Analysis and Multi-type Inquisitive Logic

In the present section, building on [1,20] and using standard facts pertaining to discrete Stone and Birkhoff dualities, we give an alternative algebraic presentation of the team semantics (or support semantics). This presentation shows how two natural types emerge from the team semantics, together with natural maps connecting them. These maps will support the interpretation of additional *multi-type* connectives which will be used to define a new, multi-type language into

[1] Recall that L is an intermediate logic if $\mathbf{IPL} \subseteq \mathsf{L} \subseteq \mathbf{CPL}$.

which we will translate the original language and axioms of inquisitive logic. We will introduce a structural multi-type sequent calculus for the translated axiomatization in Sect. 4.

3.1 Order-Theoretic Analysis

In what follows, let 2^V denote the set of valuations on a fixed set V of propositional variables, and elements of 2^V are denoted by $u, v, \ldots$, possibly sub- and super-scripted. Let $\mathbb{B}$ denote the (complete and atomic) Boolean algebra $(\mathcal{P}(2^V), \cap, \cup, (\cdot)^c, \varnothing, 2^V)$. Elements of $\mathbb{B}$ (denoted by $X, Y, Z, \ldots$, possibly sub- and super-scripted) are teams (or information states). Consider the relational structure $\mathcal{F} = (\mathcal{P}(2^V), \subseteq)$. By discrete Birkhoff-type duality, a perfect Heyting algebra[2] $\mathbb{A} := (\mathcal{P}^{\downarrow}(\mathbb{B}), \cap, \cup, \Rightarrow, \varnothing, \mathcal{P}(2^V))$ with binary operator $\Rightarrow$ arises as the complex algebra of $\mathcal{F}$. Elements of $\mathbb{A}$ are downward closed collections of teams, and are denoted by the variables $\mathcal{X}, \mathcal{Y}$ and $\mathcal{Z}$, possibly sub- and super-scripted. The operation $\Rightarrow$ is defined as follows: for any $\mathcal{Y}$ and $\mathcal{Z}$,

$$\mathcal{Y} \Rightarrow \mathcal{Z} := \{X \in \mathcal{P}(2^V) \mid \text{ for all } X', \text{ if } X' \subseteq X \text{ and } X' \in \mathcal{Y}, \text{ then } X' \in \mathcal{Z}\}.$$

Any team X can be associated with the downward-closed collection of teams $\downarrow X := \{Y \mid Y \subseteq X\}$. Conversely, any (downward-closed) collection of teams $\mathcal{X}$ can be associated with the team $\mathrm{f}\mathcal{X} := \bigcup \mathcal{X} = \{v \mid v \in X \text{ for some } X \in \mathcal{X}\}$. Thirdly, for any team X, the collection of teams $\mathrm{f}^* X := \{\{v\} \mid v \in X\} \cup \{\varnothing\}$ is downward closed. These assignments respectively induce the following three natural maps between the perfect Boolean algebra $\mathbb{B}$ and the perfect Heying algebra $\mathbb{A}$ with operator:

$$\downarrow : \mathbb{B} \to \mathbb{A} \qquad \mathrm{f} : \mathbb{A} \to \mathbb{B} \qquad \mathrm{f}^* : \mathbb{B} \to \mathbb{A}.$$

The maps f^*, $\downarrow$ and f turn out to be adjoints to one another, written $\mathrm{f}^* \dashv \mathrm{f} \dashv \downarrow$ in order-theoretic notation, in the sense of the following lemma.

Lemma 5. *For all $X \in \mathbb{B}$ and $\mathcal{X} \in \mathbb{A}$,*

$$\mathrm{f}\mathcal{X} \subseteq X \quad \textit{iff} \quad \mathcal{X} \subseteq \downarrow X \qquad \textit{and} \qquad \mathrm{f}^* X \subseteq \mathcal{X} \quad \textit{iff} \quad X \subseteq \mathrm{f}\mathcal{X}. \tag{1}$$

By general order-theoretic facts, from these adjunctions it follows that $\downarrow$, f and f^* are all order-preserving (monotone). Moreover, $\downarrow$ preserves all meets of $\mathbb{B}$ (including the empty one, i.e. $\downarrow 1^{\mathbb{B}} = \top^{\mathbb{A}}$), that is, $\downarrow$ commutes with arbitrary intersections, f preserves all joins and all meets of $\mathbb{A}$, that is, f commutes with arbitrary unions and intersections, and f^* preserves all joins of $\mathbb{B}$, that is, f^* commutes with arbitrary unions. Notice also that for all $\mathcal{X} \in \mathbb{A}$ and $X, Y \in \mathbb{B}$,

[2] A Heyting algebra is *perfect* if it is complete, completely distributive and completely join-generated by its completely join-prime elements. Equivalently, any perfect algebra can be characterized up to isomorphism as the complex algebra of some partially ordered set.

$$\mathcal{X} \subseteq \downarrow \mathrm{f}(\mathcal{X}) \quad \text{and} \quad X \subseteq Y \ \text{ implies } \mathrm{f}^*(X) \subseteq \downarrow Y. \tag{2}$$

We will need the following lemma in proof of the soundness of the rule KP of the calculus to be introduced in Sect. 4.

Lemma 6. *For all* X, $\mathcal{Y}, \mathcal{Z}$,

$$\downarrow X \Rightarrow (\mathcal{Y} \cup \mathcal{Z}) \subseteq (\downarrow X \Rightarrow \mathcal{Y}) \cup (\downarrow X \Rightarrow \mathcal{Z});$$

Proof. Assume that $W \in \downarrow X \Rightarrow (\mathcal{Y} \cup \mathcal{Z})$ and $W \notin \downarrow X \Rightarrow \mathcal{Z}$. Then $W' \subseteq X$ and $W' \notin \mathcal{Z}$ for some $W' \subseteq W$. Hence $W \notin \mathcal{Z}$. To show that $W \in \downarrow X \Rightarrow \mathcal{Y}$, let $Z \subseteq W \cap X$. Then by assumption, either $Z \in \mathcal{Y}$ or $Z \in \mathcal{Z}$. However, $W \notin \mathcal{Z}$ implies that $Z \notin \mathcal{Z}$, and hence $Z \in \mathcal{Y}$, as required. □

The following lemma collects relevant properties of $\downarrow$.

Lemma 7. *For all* $X, Y \in \mathbb{B}$,

(a) $\downarrow \bot_{\mathbb{B}} = \{\varnothing\}$ *and* $\downarrow \top^{\mathbb{B}} = \top^{\mathbb{A}}$;
(b) $\downarrow(\bigcap_{i\in I} X_i) = \bigcap_{i \in I} \downarrow X_i$;
(c) $\downarrow(X^c \cup Y) = (\downarrow X) \Rightarrow (\downarrow Y)$.

Proof. (a) Immediate.

(b)
$$\begin{aligned}\downarrow(\textstyle\bigcap_{i\in I} X_i) &= \{Z \mid Z \subseteq \textstyle\bigcap_{i\in I} X_i\}\\ &= \{Z \mid Z \subseteq X_i \text{ for all } i \in I\}\\ &= \{Z \mid Z \in \downarrow X_i \text{ for all } i \in I\}\\ &= \textstyle\bigcap_{i\in I}(\downarrow X_i).\end{aligned}$$

(c)
$$\begin{aligned}(\downarrow X) \Rightarrow (\downarrow Y) &= \{Z \mid \text{for any } W, \text{ if } W \subseteq Z \text{ and } W \subseteq X \text{ then } W \subseteq Y\}\\ &= \{Z \mid \text{ if } Z \subseteq X \text{ then } Z \subseteq Y\}\\ &= \{Z \mid Z \subseteq X^c \cup Y\}\\ &= \downarrow(X^c \cup Y).\end{aligned}$$
□

3.2 Multi-type Inquisitive Logic

The existence of the maps $\downarrow$, f and f* motivates the introduction of the following language, whose formulas are given, by simultaneous recursion, in two types: Flat and General:

$$\mathsf{Flat} \ni \alpha ::= p \mid 0 \mid \alpha \sqcap \alpha \mid \alpha \rightharpoondown \alpha \qquad \mathsf{General} \ni A ::= \downarrow\alpha \mid A \wedge A \mid A \vee A \mid A \rightarrow A$$

Let $\sim\alpha$ and $\alpha \sqcup \beta$ abbreviate $\alpha \rightharpoondown 0$ and $\sim\alpha \rightharpoondown \beta$ respectively. Recall that the canonical assignment $\hat{\cdot} : V \to \mathbb{B}$ is defined as $p \mapsto \hat{p} := \{v \mid v(p) = 1\}$. This assignment can be extended to Flat-formulas as usual via the below-defined homomorphic extension $[\![\cdot]\!]_{\mathbb{B}}$: Flat $\to \mathbb{B}$, which can, in turn, be composed with $\downarrow : \mathbb{B} \to \mathbb{A}$ so as to yield a second homomorphic extension $[\![\cdot]\!]_{\mathbb{A}}$: General $\to \mathbb{A}$, defined as below:

$$
\begin{array}{rlrl}
[\![p]\!]_{\mathbb{B}} &= \hat{p} & [\![\downarrow\alpha]\!]_{\mathbb{A}} &= \downarrow[\![\alpha]\!]_{\mathbb{B}} \\
[\![0]\!]_{\mathbb{B}} &= \varnothing & [\![A \vee B]\!]_{\mathbb{A}} &= [\![A]\!]_{\mathbb{A}} \cup [\![B]\!]_{\mathbb{A}} \\
[\![\alpha \sqcap \beta]\!]_{\mathbb{B}} &= [\![\alpha]\!]_{\mathbb{B}} \cap [\![\beta]\!]_{\mathbb{B}} & [\![A \wedge B]\!]_{\mathbb{A}} &= [\![A]\!]_{\mathbb{A}} \cap [\![B]\!]_{\mathbb{A}} \\
[\![\alpha \rightharpoonup \beta]\!]_{\mathbb{B}} &= ([\![\alpha]\!]_{\mathbb{B}})^c \cup [\![\beta]\!]_{\mathbb{B}} & [\![A \to B]\!]_{\mathbb{A}} &= [\![A]\!]_{\mathbb{A}} \Rightarrow [\![B]\!]_{\mathbb{A}}. \\
[\![\alpha \sqcup \beta]\!]_{\mathbb{B}} &= [\![\alpha]\!]_{\mathbb{B}} \cup [\![\beta]\!]_{\mathbb{B}} & &
\end{array}
$$

As an immediate consequence of the above definitions and Lemma 7, we obtain the following lemma.

Lemma 8. *For all* Flat*-formulas* α *and* β*,*

$$
\begin{array}{rlrl}
[\![\downarrow p]\!]_{\mathbb{A}} &= \downarrow\hat{p} & [\![\downarrow(\alpha \sqcap \beta)]\!]_{\mathbb{A}} &= \downarrow[\![\alpha]\!]_{\mathbb{B}} \cap \downarrow[\![\beta]\!]_{\mathbb{B}} \\
[\![\downarrow 0]\!]_{\mathbb{A}} &= \{\varnothing\} & [\![\downarrow(\alpha \rightharpoonup \beta)]\!]_{\mathbb{A}} &= \downarrow[\![\alpha]\!]_{\mathbb{B}} \Rightarrow \downarrow[\![\beta]\!]_{\mathbb{B}}.
\end{array}
$$

Let us now define the multi-type counterpart of the notion of flatness:

Definition 9. *A formula* $A \in$ General *is* flat *if for every team* X*,*

$$X \models A \quad \textit{iff} \quad \{v\} \models A \ \ \textit{for every } v \in X.$$

Lemma 10. *The following are equivalent for any* $A \in$ General*:*

1. *A is flat*
2. $[\![A]\!]_{\mathbb{A}} = \downarrow\mathrm{f}([\![A]\!]_{\mathbb{A}})$

Proof. By definition, A is flat iff $[\![A]\!]_{\mathbb{A}} = \{X \mid \mathrm{f}^*(X) \subseteq [\![A]\!]_{\mathbb{A}}\}$. Moreover, the following chain of identities holds:

$$
\begin{array}{ll}
\{X \mid \mathrm{f}^*(X) \subseteq [\![A]\!]_{\mathbb{A}}\} & \\
= \{X \mid X \subseteq \mathrm{f}([\![A]\!]_{\mathbb{A}})\} & \text{(Lemma 5)} \\
= \downarrow\mathrm{f}([\![A]\!]_{\mathbb{A}}), &
\end{array}
$$

which completes the proof. □

We are now in a position to define the following translation of **InqL**-formulas into formulas of the multi-type language introduced above. **CPL**-formulas χ and ξ will be translated into Flat-formulas via τ_c, and **InqL**-formulas ϕ and ψ into General-formulas via τ_i as follows:

$$
\begin{array}{rlrl}
\tau_c(p) &= p & \tau_i(\chi) &= \downarrow\tau_c(\chi) \\
\tau_c(0) &= 0 & \tau_i(\phi \vee \psi) &= \tau_i(\phi) \vee \tau_i(\psi) \\
\tau_c(\chi \wedge \xi) &= \tau_c(\chi) \sqcap \tau_c(\xi) & \tau_i(\phi \wedge \psi) &= \tau_i(\phi) \wedge \tau_i(\psi) \\
\tau_c(\chi \to \xi) &= \tau_c(\chi) \rightharpoonup \tau_c(\xi) & \tau_i(\phi \to \psi) &= \tau_i(\phi) \to \tau_i(\psi)
\end{array}
$$

The translation above justifies the introduction of the following Hilbert-style presentation of the logic which is the natural multi-type counterpart of **InqL**:

Axioms:

(A1) **CPL** axiom schemata for Flat-formulas
(A2) **IPL** axiom schemata for General-formulas
(A3) $(\downarrow\alpha \to (A \vee B)) \to (\downarrow\alpha \to A) \vee (\downarrow\alpha \to B)$
(A4) $\neg\neg\downarrow\alpha \to \downarrow\alpha$

Rule: Modus Ponens for both Flat-formulas and General-formulas.

4 Structural Sequent Calculus for Multi-type Inquisitive Logic

In the present section, we introduce the structural calculus for our multi-type inquisitive logic.

– Structural and operational languages of type Flat and General:

Flat	General
$\alpha ::= p \mid 0 \mid \alpha \sqcap \alpha \mid \alpha \rightarrowtriangle \alpha$	$A ::= \downarrow\alpha \mid A \wedge A \mid A \vee A \mid A \to A$
$\Gamma ::= \alpha \mid \Phi \mid \Gamma \,,\, \Gamma \mid \Gamma \sqsupset \Gamma \mid \mathrm{F}X$	$X ::= A \mid \Downarrow\Gamma \mid \mathrm{F}^*\Gamma \mid X \,;\, X \mid X > X$

– Interpretation of structural Flat connectives as their operational (i.e. logical) counterparts:[3]

Structural symbols	Φ		$,$		$\sqsupset$	
Operational symbols	(1)	0	$\sqcap$	$(\sqcup)$	$(\mapsto)$	$\rightarrowtriangle$

– Interpretation of structural General connectives as their operational counterparts:

Structural symbols	$;$		$>$	
Operational symbols	$\wedge$	$\vee$	$(\rightarrowtail)$	$\to$

– Interpretation of multi-type connectives

Structural symbols	F^*		F		$\Downarrow$	
Operational symbols	(f^*)		(f)	(f)	$\downarrow$	$\downarrow$

[3] We follow the notational conventions introduced in [10], according to which each structural connective in the upper row of the synoptic tables is interpreted as the logical connective(s) in the two slots below it in the lower row. Specifically, each of its occurrences in antecedent (resp. succedent) position is interpreted as the logical connective in the left-hand (resp. right-hand) slot. Hence, for instance, the structural symbol $\sqsupset$ is interpreted as classical implication $\rightarrowtriangle$ when occurring in succedent position and as classical disimplication $\mapsto$ (i.e. $\alpha \mapsto \beta := {\sim}\alpha \sqcap \beta$) when occurring in antecedent position.

– Structural rules common to both types

$$\frac{\Gamma \vdash \alpha \qquad (\Sigma \vdash \Delta)[\alpha]^{pre}}{(\Sigma \vdash \Delta)[\Gamma/\alpha]^{pre}}\ Cut \qquad\qquad \frac{X \vdash A \qquad A \vdash Y}{X \vdash Y}\ Cut$$

$$\frac{\Gamma , \Delta \vdash \Sigma}{\Delta \vdash \Gamma \sqsupset \Sigma}\ \text{Flat res} \qquad\qquad \frac{X ; Y \vdash Z}{Y \vdash X > Z}\ \text{Gen res}$$

$$\Phi\ \frac{\Gamma \vdash \Delta}{\Phi , \Gamma \vdash \Delta} \quad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \Phi , \Delta}\ \Phi \qquad\qquad \Downarrow\Phi\ \frac{X \vdash Y}{\Downarrow\Phi ; X \vdash Y} \quad \frac{X \vdash Y}{X \vdash \Downarrow\Phi ; Y}\ \Downarrow\Phi$$

$$W\ \frac{\Gamma \vdash \Delta}{\Gamma , \Sigma \vdash \Delta} \quad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta , Z}\ W \qquad\qquad W\ \frac{X \vdash Y}{X ; Z \vdash Y} \quad \frac{X \vdash Y}{X \vdash Y ; Z}\ W$$

$$C\ \frac{\Gamma , \Gamma \vdash \Delta}{\Gamma \vdash \Delta} \quad \frac{\Gamma \vdash \Delta , \Delta}{\Gamma \vdash \Delta}\ C \qquad\qquad C\ \frac{X ; X \vdash Y}{X \vdash Y} \quad \frac{X \vdash Y ; Y}{X \vdash Y}\ C$$

$$E\ \frac{\Gamma , \Delta \vdash \Sigma}{\Delta , \Gamma \vdash \Sigma} \quad \frac{\Gamma \vdash \Delta , \Sigma}{\Gamma \vdash \Sigma , \Delta}\ E \qquad\qquad E\ \frac{X ; Y \vdash Z}{Y ; X \vdash Z} \quad \frac{X \vdash Y ; Z}{X \vdash Z ; Y}\ E$$

$$A\ \frac{\Gamma , (\Delta , \Sigma) \vdash \Pi}{(\Gamma , \Delta) , \Sigma \vdash \Pi} \quad \frac{\Gamma \vdash (\Delta , \Sigma) , \Pi}{\Gamma \vdash \Delta , (\Sigma , \Pi)}\ A \qquad A\ \frac{X ; (Y ; Z) \vdash W}{(X ; Y) ; Z \vdash W} \quad \frac{X \vdash (Y ; Z) ; W}{X \vdash Y ; (Z ; W)}\ A$$

$$\mathrm{G}\ \frac{(\Gamma \sqsupset \Delta) , \Sigma \vdash \Pi}{\Gamma \sqsupset (\Delta , \Sigma) \vdash \Pi} \quad \frac{\Pi \vdash (\Gamma \sqsupset \Delta) , \Sigma}{\Pi \vdash \Gamma \sqsupset (\Delta , \Sigma)}\ \mathrm{G} \qquad \mathrm{G}\ \frac{(X > Y) ; Z \vdash W}{X > (Y ; Z) \vdash W} \quad \frac{W \vdash (X > Y) ; Z}{W \vdash X > (Y ; Z)}\ \mathrm{G}$$

– Structural rules specific to the Flat type

$$\mathrm{Id}\ \frac{}{p \vdash p} \quad \frac{\Pi \vdash \Gamma \sqsupset (\Delta , \Sigma)}{\Pi \vdash (\Gamma \sqsupset \Delta) , \Sigma}\ \mathrm{CG}$$

– Structural rules governing the interaction between the two types:

$$\frac{\mathrm{F}^{*}\Gamma \vdash \Delta}{\Gamma \vdash \mathrm{F}\Delta}\ \text{f adj} \qquad \frac{\mathrm{F}X \vdash \Gamma}{X \vdash \Downarrow\Gamma}\ \text{d adj} \qquad \frac{\Downarrow\mathrm{F}X \vdash Y}{X \vdash Y}\ \text{d-f elim}$$

$$\frac{\Gamma \vdash \Delta}{\mathrm{F}^{*}\Gamma \vdash \Downarrow\Delta}\ \text{bal} \qquad \frac{\Gamma \vdash \Delta}{\Downarrow\Gamma \vdash \Downarrow\Delta}\ \text{d mon} \qquad \frac{X \vdash Y}{\mathrm{F}X \vdash \mathrm{F}Y}\ \text{f mon}$$

$$\frac{X \vdash \Downarrow(\Gamma \sqsupset \Delta)}{X \vdash \Downarrow\Gamma > \Downarrow\Delta}\ \text{d dis} \qquad \frac{\mathrm{F}X , \mathrm{F}Y \vdash Z}{\mathrm{F}(X ; Y) \vdash Z}\ \text{f dis}$$

$$\frac{X \vdash \Downarrow\Gamma > (Y ; Z) \qquad X \vdash \Downarrow\Gamma > (Y ; Z)}{X \vdash (\Downarrow\Gamma > Y) ; (\Downarrow\Gamma > Z)}\ \mathrm{KP}$$

– Introduction rules for pure-type logical connectives:

$$\frac{}{0 \vdash \Phi} \quad \frac{\Gamma \vdash \Phi}{\Gamma \vdash 0} \qquad\qquad \frac{A \vdash X \qquad B \vdash Y}{A \vee B \vdash X ; Y} \quad \frac{Z \vdash A ; B}{Z \vdash A \vee B}$$

$$\frac{\alpha , \beta \vdash \Gamma}{\alpha \sqcap \beta \vdash \Gamma} \quad \frac{\Gamma \vdash \alpha \qquad \Delta \vdash \beta}{\Gamma , \Delta \vdash \alpha \sqcap \beta} \qquad\qquad \frac{A ; B \vdash Z}{A \wedge B \vdash Z} \quad \frac{X \vdash A \qquad Y \vdash B}{X ; Y \vdash A \wedge B}$$

$$\frac{\Gamma \vdash \alpha \qquad \beta \vdash \Delta}{\alpha \rightarrowtriangle \beta \vdash \Gamma \sqsupset \Delta} \quad \frac{\Gamma \vdash \alpha \sqsupset \beta}{\Gamma \vdash \alpha \rightarrowtriangle \beta} \qquad\qquad \frac{X \vdash A \qquad B \vdash Y}{A \rightarrow B \vdash X > Y} \quad \frac{Z \vdash A > B}{Z \vdash A \rightarrow B}$$

– Introduction rules for $\downarrow$:

$$\frac{\Downarrow\alpha \vdash X}{\downarrow\alpha \vdash X} \qquad \frac{X \vdash \Downarrow\alpha}{X \vdash \downarrow\alpha}$$

5 Properties of the Calculus

In the present section, we discuss the soundness and completeness of the calculus introduced in Sect. 4, as well as its being able to capture flatness syntactically.

5.1 Soundness and Completeness

As is typical of structural calculi, in order to prove the soundness of the rules, structural sequents will be translated into operational sequents of the appropriate type, and operational sequents will be interpreted according to their type. Specifically, each atomic proposition $p \in V$ is assigned to the team $[\![p]\!] := \{v \in 2^V \mid v(p) = 1\}$.

In order to translate structures as operational terms, structural connectives need to be translated as logical connectives. To this effect, structural connectives are associated with one or more logical connectives, and any given occurrence of a structural connective is translated as one or the other according to its (antecedent or succedent) position, as indicated in the synoptic tables at the beginning of Sect. 4. This procedure is completely standard, and is discussed in detail in [7,9,10].

Sequents $A \vdash B$ (resp. $\alpha \vdash \beta$) will be interpreted as inequalities (actually inclusions) $[\![A]\!]_{\mathbb{A}} \leq [\![B]\!]_{\mathbb{A}}$ (resp. $[\![\alpha]\!]_{\mathbb{B}} \leq [\![\beta]\!]_{\mathbb{B}}$) in $\mathbb{A}$ (resp. $\mathbb{B}$); rules $(a_i \vdash b_i \mid i \in I)/c \vdash d$ will be interpreted as implications of the form "if $[\![a_i]\!] \leq [\![b_i]\!]$ for every $i \in I$, then $[\![c]\!] \leq [\![d]\!]$". Following this procedure, it is easy to see that:

- the soundness of (d mon) and (f mon) follows from the monotonicity of the semantic operations $\downarrow$ and f respectively (cf. discussion after Lemma 5);
- the soundness of (d-f elim) and (bal) follows from the observations in (2);
- the soundness of (d adj) and (f adj) follows from Lemma 5;
- the soundness of (f dis) follows from the fact that the semantic operation f distributes over intersections;
- the soundness of (d dis) follows from Lemma 7 (c);
- the soundness of (KP) follows from Lemma 6.

The proof of the soundness of the remaining rules is well known and is omitted.

To prove the completeness of our multi-type calculus, it suffices to prove that the calculus derives (the translation of) all theorems of the multi-type inquisitive logic defined in Sect. 3.2, which reduces to showing that the calculus derives (the translation of) all axioms of the multi-type inquisitive logic. Since our calculus contains all of the usual rules for **CPL** with respect to Flat-formulas, and all of the usual rules for **IPL** with respect to General-formulas, all **CPL** axioms for Flat-formulas and all **IPL** axioms for General-formulas can be derived by the standard derivations. We provide the derivations for axiom (A3) in Appendix II and axiom (A4) in Appendix I.

5.2 Syntactic Flatness Captured by the Calculus

Lemma 10 provided a semantic identification of flat General-formulas as those the extension of which is in the image of the semantic $\downarrow$. The following lemma provides a similar identification with syntactic means.

Lemma 11. *If a formula is of the following shape*

$$A ::= \downarrow\alpha \mid A \wedge A \mid A \rightarrow A,$$

then $A \dashv\vdash \downarrow\alpha$ *for some* α.

Proof. Base case: $A = \downarrow\alpha$.

$$\dfrac{\dfrac{\dfrac{\alpha \vdash \alpha}{\Downarrow\alpha \vdash \Downarrow\alpha}}{\downarrow\alpha \vdash \Downarrow\alpha}}{\downarrow\alpha \vdash \downarrow\alpha}$$

Inductive case 1: $A = B \wedge C = \downarrow\beta \wedge \downarrow\gamma$ by induction hypothesis.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\alpha \vdash \alpha}{\alpha\,,\beta \vdash \alpha}}{\alpha \sqcap \beta \vdash \alpha}}{\Downarrow(\alpha \sqcap \beta) \vdash \Downarrow\alpha}}{\Downarrow(\alpha \sqcap \beta) \vdash \downarrow\alpha} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\beta \vdash \beta}{\alpha\,,\beta \vdash \beta}}{\alpha \sqcap \beta \vdash \beta}}{\Downarrow(\alpha \sqcap \beta) \vdash \Downarrow\beta}}{\Downarrow(\alpha \sqcap \beta) \vdash \downarrow\beta}}{\Downarrow(\alpha \sqcap \beta)\,;\Downarrow(\alpha \sqcap \beta) \vdash \downarrow\alpha \wedge \downarrow\beta}}{\Downarrow(\alpha \sqcap \beta) \vdash \downarrow\alpha \wedge \downarrow\beta}}{\downarrow(\alpha \sqcap \beta) \vdash \downarrow\alpha \wedge \downarrow\beta}
\qquad
\dfrac{\dfrac{\dfrac{\dfrac{\text{d adj}\;\dfrac{\dfrac{\dfrac{\alpha \vdash \alpha}{\Downarrow\alpha \vdash \Downarrow\alpha}}{\downarrow\alpha \vdash \Downarrow\alpha}}{\mathrm{F}\downarrow\alpha \vdash \alpha} \qquad \text{d adj}\;\dfrac{\dfrac{\dfrac{\beta \vdash \beta}{\Downarrow\beta \vdash \Downarrow\beta}}{\downarrow\beta \vdash \Downarrow\beta}}{\mathrm{F}\downarrow\beta \vdash \beta}}{\mathrm{F}\downarrow\alpha\,,\mathrm{F}\downarrow\beta \vdash \alpha \sqcap \beta}}{\mathrm{F}(\downarrow\alpha\,;\downarrow\beta) \vdash \alpha \sqcap \beta}\;\text{f dis}}{\downarrow\alpha\,;\downarrow\beta \vdash \Downarrow(\alpha \sqcap \beta)}}{\dfrac{\downarrow\alpha\,;\downarrow\beta \vdash \downarrow(\alpha \sqcap \beta)}{\downarrow\alpha \wedge \downarrow\beta \vdash \downarrow(\alpha \sqcap \beta)}}$$

Inductive case 2: $A = B \rightarrow C = \downarrow\beta \rightarrow \downarrow\gamma$ by induction hypothesis.

$$\dfrac{\dfrac{\dfrac{\dfrac{\text{f dis}\;\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\text{d adj}\;\dfrac{\dfrac{\dfrac{\alpha \vdash \alpha}{\Downarrow\alpha \vdash \Downarrow\alpha}}{\downarrow\alpha \vdash \Downarrow\alpha}}{\mathrm{F}\downarrow\alpha \vdash \alpha} \qquad \beta \vdash \beta}{\alpha \rightarrowtriangle \beta \vdash \mathrm{F}\downarrow\alpha \sqsupset \beta}}{\Downarrow(\alpha \rightarrowtriangle \beta) \vdash \Downarrow(\mathrm{F}\downarrow\alpha \sqsupset \beta)}}{\downarrow(\alpha \rightarrowtriangle \beta) \vdash \Downarrow(\mathrm{F}\downarrow\alpha \sqsupset \beta)}}{\mathrm{F}\downarrow(\alpha \rightarrowtriangle \beta) \vdash \mathrm{F}\downarrow\alpha \sqsupset \beta}}{\dfrac{\mathrm{F}\downarrow\alpha\,,\mathrm{F}\downarrow(\alpha \rightarrowtriangle \beta) \vdash \beta}{\mathrm{F}(\downarrow\alpha\,;\downarrow(\alpha \rightarrowtriangle \beta)) \vdash \beta}}}{\downarrow\alpha\,;\downarrow(\alpha \rightarrowtriangle \beta) \vdash \Downarrow\beta}\;\text{d adj}}{\downarrow\alpha\,;\downarrow(\alpha \rightarrowtriangle \beta) \vdash \downarrow\beta}}{\dfrac{\downarrow(\alpha \rightarrowtriangle \beta) \vdash \downarrow\alpha > \downarrow\beta}{\downarrow(\alpha \rightarrowtriangle \beta) \vdash \downarrow\alpha \rightarrow \downarrow\beta}}
\qquad
\dfrac{\dfrac{\text{d adj}\;\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\alpha \vdash \alpha}{\Downarrow\alpha \vdash \Downarrow\alpha}}{\Downarrow\alpha \vdash \downarrow\alpha} \qquad \dfrac{\dfrac{\beta \vdash \beta}{\Downarrow\beta \vdash \Downarrow\beta}}{\downarrow\beta \vdash \Downarrow\beta}}{\downarrow\alpha \rightarrow \downarrow\beta \vdash \Downarrow\alpha > \Downarrow\beta}}{\downarrow\alpha \rightarrow \downarrow\beta \vdash \Downarrow(\alpha \sqsupset \beta)}}{\dfrac{\mathrm{F}(\downarrow\alpha \rightarrow \downarrow\beta) \vdash \alpha \sqsupset \beta}{\mathrm{F}(\downarrow\alpha \rightarrow \downarrow\beta) \vdash \alpha \rightarrowtriangle \beta}}}{\downarrow\alpha \rightarrow \downarrow\beta \vdash \Downarrow(\alpha \rightarrowtriangle \beta)}\;\text{d adj}}{\downarrow\alpha \rightarrow \downarrow\beta \vdash \downarrow(\alpha \rightarrowtriangle \beta)}$$

□

6 Cut Elimination

In the present section, we prove that the calculus introduced in Sect. 4 enjoys cut elimination and subformula property. Perhaps the most important feature of this calculus is that its cut elimination does not need to be proved by brute-force, but can rather be inferred from a Belnap-style cut elimination meta-theorem, proved in [8], which holds for the so called *proper multi-type calculi*, the definition of which is reported below.

6.1 Cut Elimination Meta-Theorem for Proper Multi-type Calculi

Theorem 12 *(cf.* [8, Theorem 4.1]*). Every proper multi-type calculus enjoys cut elimination and subformula property.*

Proper multi-type calculi are those satisfying the following list of conditions:

C_1: *Preservation of operational terms.* Each operational term occurring in a premise of an inference rule *inf* is a subterm of some operational term in the conclusion of *inf*.

C_2: *Shape-alikeness of parameters.* Congruent parameters (i.e. non-active terms in the application of a rule) are occurrences of the same structure.

C_2': *Type-alikeness of parameters.* Congruent parameters have exactly the same type. This condition bans the possibility that a parameter changes type along its history.

C_3: *Non-proliferation of parameters.* Each parameter in an inference rule *inf* is congruent to at most one constituent in the conclusion of *inf*.

C_4: *Position-alikeness of parameters.* Congruent parameters are either all precedent or all succedent parts of their respective sequents. In the case of calculi enjoying the display property, precedent and succedent parts are defined in the usual way (see [2]). Otherwise, these notions can still be defined by induction on the shape of the structures, by relying on the polarity of each coordinate of the structural connectives.

C_5': *Quasi-display of principal constituents.* If an operational term a is principal in the conclusion sequent s of a derivation π, then a is in display, unless π consists only of its conclusion sequent s (i.e. s is an axiom).

C_5'': *Display-invariance of axioms.* If a is principal in an axiom s, then a can be isolated by applying Display Postulates and the new sequent is still an axiom.

C_6': *Closure under substitution for succedent parts within each type.* Each rule is closed under simultaneous substitution of arbitrary structures for congruent operational terms occurring in succedent position, *within each type*.

C_7': *Closure under substitution for precedent parts within each type.* Each rule is closed under simultaneous substitution of arbitrary structures for congruent operational terms occurring in precedent position, *within each type*.

C_8': *Eliminability of matching principal constituents.* This condition requests a standard Gentzen-style checking, which is now limited to the case in which both cut formulas are *principal*, i.e. each of them has been introduced with the last rule application of each corresponding subdeduction. In this case, analogously to the proof Gentzen-style, condition C_8' requires being able to transform the given deduction into a deduction with the same conclusion in which either the cut is eliminated altogether, or is transformed in one or more applications of the cut rule, involving proper subterms of the original operational cut-term. In addition to this, specific to the multi-type setting is the requirement that the new application(s) of the cut rule be also *type-uniform* (cf. condition C_{10}' below).

C_8''': *Closure of axioms under surgicalcut.* If $(x \vdash y)([a]^{pre}, [a]^{suc})$, $a \vdash z[a]^{suc}$ and $v[a]^{pre} \vdash a$ are axioms, then $(x \vdash y)([a]^{pre}, [z/a]^{suc})$ and $(x \vdash y)([v/a]^{pre}, [a]^{suc})$ are again axioms.

C_9: *Type-uniformity of derivable sequents.* Each derivable sequent is type-uniform.[4]

C_{10}': *Preservation of type-uniformity of cut rules.* All cut rules preserve type-uniformity.

6.2 Cut Elimination for the Structural Calculus for Multi-type Inquisitive Logic

To show that the calculus defined in Sect. 4 enjoys cut elimination and subformula property, it is sufficent to show that it is a proper multi-type calculus, i.e., that verifies every condition in the list above. All conditions except C_8' are readily satisfied by inspection on the rules of the calculus. In what follows we verify C_8'.

Condition C_8' requires to check the cut elimination when both cut formulas are principal. Since principal formulas are always introduced in display, it is sufficent to show that applications of standard (rather than surgical) cuts can be either eliminated or replaced with (possibly surgical) cuts on formulas of strictly lower complexity.

Constant:

$$\dfrac{\dfrac{\dfrac{\vdots\ \pi_1}{\Gamma \vdash \Phi}}{\Gamma \vdash 0} \qquad 0 \vdash \Phi}{\Gamma \vdash \Phi} \quad \rightsquigarrow \quad \dfrac{\vdots\ \pi_1}{\Gamma \vdash \Phi}$$

Propositional variable:

$$\dfrac{p \vdash p \qquad p \vdash p}{p \vdash p} \quad \rightsquigarrow \quad p \vdash p$$

[4] A sequent $x \vdash y$ is type-uniform if x and y are of the same type.

Classical conjunction $\sqcap$:

$$
\dfrac{\dfrac{\overset{\vdots\ \pi_1}{\Gamma \vdash \alpha} \quad \overset{\vdots\ \pi_2}{\Delta \vdash \beta}}{\Gamma\,,\Delta \vdash \alpha \sqcap \beta} \qquad \dfrac{\overset{\vdots\ \pi_3}{\alpha\,,\beta \vdash \Lambda}}{\alpha \sqcap \beta \vdash \Lambda}}{\Gamma\,,\Delta \vdash \Lambda}
\quad \rightsquigarrow \quad
\dfrac{\dfrac{\dfrac{\overset{\vdots\ \pi_1}{\Gamma \vdash \alpha} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\overset{\vdots\ \pi_2}{\Delta \vdash \beta} \quad \dfrac{\overset{\vdots\ \pi_3}{\alpha\,,\beta \vdash \Lambda}}{\beta \vdash \alpha > \Lambda}}{\Delta \vdash \alpha > \Lambda}}{\alpha\,,\Delta \vdash \Lambda}}{\Delta\,,\alpha \vdash \Lambda}}{\alpha \vdash \Delta > \Lambda}}{\Gamma \vdash \Delta > \Lambda}}{\Delta\,,\Gamma \vdash \Lambda}}{\Gamma\,,\Delta \vdash \Lambda}
$$

The cases for $-\!\!\vartriangleright$, $\wedge$, $\vee$, $\rightarrow$ are standard and similar to the one above.

Downarrow $\downarrow$:

$$
\dfrac{\dfrac{\overset{\vdots\ \pi_1}{X \vdash \Downarrow\alpha}}{X \vdash \downarrow\alpha} \qquad \dfrac{\overset{\vdots\ \pi_2}{\Downarrow\alpha \vdash Y}}{\downarrow\alpha \vdash Y}}{X \vdash Y}
\quad \rightsquigarrow \quad
\dfrac{\dfrac{\dfrac{\overset{\vdots\ \pi_1}{X \vdash \Downarrow\alpha}}{FX \vdash \alpha} \qquad \overset{\vdots\ \pi_2}{\Downarrow\alpha \vdash Y}}{\Downarrow FX \vdash Y}}{X \vdash Y}
$$

7 Conclusion

The calculus introduced in the present paper is not a standard display calculus. This is due to the fact that, according to the order-theoretic analysis we gave, the axiom (A3) is not analytic inductive in the sense of [11]. Hence, it is not possible to give a proper *display* calculus to the axiomatization of the multi-type inquisitive logic introduced in Sect. 3.2. In order to encode the (A3) axiom by means of a structural rule, we made the non standard choice of allowing the structural counterpart of $\downarrow$ in antecedent position, notwithstanding the fact that it is *not* a left adjoint. As a consequence, the display property does not hold for the calculus introduced in the present paper. However, a generalization of the Belnap-style cut elimination meta-theorem holds and applies to it.

Further directions of research will address the problem of extending this calculus to propositional dependence logic.

Acknowledgements. This research has been made possible by the NWO Vidi grant 016.138.314, by the NWO Aspasia grant 015.008.054, and by a Delft Technology Fellowship awarded in 2013.

Appendix I

The derivation of (A3) $(\downarrow\alpha \to (A \vee B)) \to (\downarrow\alpha \to A) \vee (\downarrow\alpha \to B)$:

$$
\dfrac{
\dfrac{\dfrac{\dfrac{\alpha \vdash \alpha}{\Downarrow\alpha \vdash \Downarrow\alpha}\ \text{d mon}}{\downarrow\alpha \vdash \Downarrow\alpha} \qquad \dfrac{B \vdash B \qquad C \vdash C}{B \vee C \vdash B\,; C}}{\downarrow\alpha \to (B \vee C) \vdash \Downarrow\alpha > (B\,; C)}
\qquad
\dfrac{\dfrac{\dfrac{\alpha \vdash \alpha}{\Downarrow\alpha \vdash \Downarrow\alpha}\ \text{d mon}}{\downarrow\alpha \vdash \Downarrow\alpha} \qquad \dfrac{B \vdash B \qquad C \vdash C}{B \vee C \vdash B\,; C}}{\downarrow\alpha \to (B \vee C) \vdash \Downarrow\alpha > (B\,; C)}
}{\downarrow\alpha \to (B \vee C) \vdash (\Downarrow\alpha > B)\,; (\Downarrow\alpha > C)}\ \text{KP}
$$

$$
\begin{array}{c}
\downarrow\alpha \to (B \vee C) \vdash (\Downarrow\alpha > B)\,; (\Downarrow\alpha > C) \\ \hline
(\Downarrow\alpha > B) > \downarrow\alpha \to (B \vee C) \vdash \Downarrow\alpha > C \\ \hline
\Downarrow\alpha\,; ((\Downarrow\alpha > B) > \downarrow\alpha \to (B \vee C)) \vdash C \\ \hline
((\Downarrow\alpha > B) > \downarrow\alpha \to (B \vee C))\,; \Downarrow\alpha \vdash C \\ \hline
\Downarrow\alpha \vdash ((\Downarrow\alpha > B) > \downarrow\alpha \to (B \vee C)) > C \\ \hline
\downarrow\alpha \vdash ((\Downarrow\alpha > B) > \downarrow\alpha \to (B \vee C)) > C \\ \hline
((\Downarrow\alpha > B) > \downarrow\alpha \to (B \vee C))\,; \downarrow\alpha \vdash C \\ \hline
\downarrow\alpha\,; ((\Downarrow\alpha > B) > \downarrow\alpha \to (B \vee C)) \vdash C \\ \hline
(\Downarrow\alpha > B) > \downarrow\alpha \to (B \vee C) \vdash \downarrow\alpha > C \\ \hline
(\Downarrow\alpha > B) > \downarrow\alpha \to (B \vee C) \vdash \downarrow\alpha \to C \\ \hline
\downarrow\alpha \to (B \vee C) \vdash (\Downarrow\alpha > B)\,; \downarrow\alpha \to C \\ \hline
\downarrow\alpha \to (B \vee C) \vdash \downarrow\alpha \to C\,; (\Downarrow\alpha > B) \\ \hline
\downarrow\alpha \to C > \downarrow\alpha \to (B \vee C) \vdash \Downarrow\alpha > B \\ \hline
\Downarrow\alpha\,; (\downarrow\alpha \to C > \downarrow\alpha \to (B \vee C)) \vdash B \\ \hline
(\downarrow\alpha \to C > \downarrow\alpha \to (B \vee C))\,; \Downarrow\alpha \vdash B \\ \hline
\Downarrow\alpha \vdash (\downarrow\alpha \to C > \downarrow\alpha \to (B \vee C)) > B \\ \hline
\downarrow\alpha \vdash (\downarrow\alpha \to C > \downarrow\alpha \to (B \vee C)) > B \\ \hline
(\downarrow\alpha \to C > \downarrow\alpha \to (B \vee C))\,; \downarrow\alpha \vdash B \\ \hline
\downarrow\alpha\,; (\downarrow\alpha \to C > \downarrow\alpha \to (B \vee C)) \vdash B \\ \hline
\downarrow\alpha \to C > \downarrow\alpha \to (B \vee C) \vdash \downarrow\alpha > B \\ \hline
\downarrow\alpha \to C > \downarrow\alpha \to (B \vee C) \vdash \downarrow\alpha \to B \\ \hline
\downarrow\alpha \to (B \vee C) \vdash \downarrow\alpha \to C\,; \downarrow\alpha \to B \\ \hline
\downarrow\alpha \to (B \vee C) \vdash \downarrow\alpha \to B\,; \downarrow\alpha \to C \\ \hline
\downarrow\alpha \to (B \vee C) \vdash (\downarrow\alpha \to B) \vee (\downarrow\alpha \to C)
\end{array}
$$

Appendix II

The derivation of (A4) $\neg\neg{\downarrow}\alpha \rightarrow {\downarrow}\alpha$:

$$
\begin{prooftree}
\AxiomC{$\alpha \vdash \alpha$}
\UnaryInfC{$\alpha \vdash 0 \,,\, \alpha$}
\UnaryInfC{$\alpha \,,\, \Phi \vdash 0 \,,\, \alpha$}
\UnaryInfC{$\Phi \vdash \alpha \sqsupset (0 \,,\, \alpha)$}
\RightLabel{CG}
\UnaryInfC{$\Phi \vdash (\alpha \sqsupset 0) \,,\, \alpha$}
\UnaryInfC{$\Phi \vdash \alpha \,,\, (\alpha \sqsupset 0)$}
\UnaryInfC{$\alpha \sqsupset \Phi \vdash \alpha \sqsupset 0$}
\UnaryInfC{${\Downarrow}(\alpha \sqsupset \Phi) \vdash {\Downarrow}(\alpha \sqsupset 0)$}
\RightLabel{d dis}
\UnaryInfC{${\Downarrow}(\alpha \sqsupset \Phi) \vdash {\Downarrow}\alpha > {\Downarrow}0$}
\UnaryInfC{${\Downarrow}\alpha \,;\, {\Downarrow}(\alpha \sqsupset \Phi) \vdash {\Downarrow}0$}
\UnaryInfC{${\Downarrow}\alpha \,;\, {\Downarrow}(\alpha \sqsupset \Phi) \vdash {\downarrow}0$}
\UnaryInfC{${\Downarrow}(\alpha \sqsupset \Phi) \,;\, {\Downarrow}\alpha \vdash {\downarrow}0$}
\UnaryInfC{${\Downarrow}\alpha \vdash {\Downarrow}(\alpha \sqsupset \Phi) > {\downarrow}0$}
\UnaryInfC{${\downarrow}\alpha \vdash {\Downarrow}(\alpha \sqsupset \Phi) > {\downarrow}0$}
\UnaryInfC{${\Downarrow}(\alpha \sqsupset \Phi) \,;\, {\downarrow}\alpha \vdash {\downarrow}0$}
\UnaryInfC{${\downarrow}\alpha \,;\, {\Downarrow}(\alpha \sqsupset \Phi) \vdash {\downarrow}0$}
\UnaryInfC{${\Downarrow}(\alpha \sqsupset \Phi) \vdash {\downarrow}\alpha > {\downarrow}0$}
\UnaryInfC{${\Downarrow}(\alpha \sqsupset \Phi) \vdash {\downarrow}\alpha \rightarrow {\downarrow}0$}
\RightLabel{def}
\UnaryInfC{${\Downarrow}(\alpha \sqsupset \Phi) \vdash \neg{\downarrow}\alpha$}
\AxiomC{$0 \vdash \Phi$}
\RightLabel{d mon}
\UnaryInfC{${\Downarrow}0 \vdash {\Downarrow}\Phi$}
\UnaryInfC{${\downarrow}0 \vdash {\Downarrow}\Phi$}
\BinaryInfC{$\neg{\downarrow}\alpha \rightarrow {\downarrow}0 \vdash {\Downarrow}(\alpha \sqsupset \Phi) > {\Downarrow}\Phi$}
\LeftLabel{def}
\UnaryInfC{$\neg\neg{\downarrow}\alpha \vdash {\Downarrow}(\alpha \sqsupset \Phi) > {\Downarrow}\Phi$}
\RightLabel{d dis}
\UnaryInfC{$\neg\neg{\downarrow}\alpha \vdash {\Downarrow}((\alpha \sqsupset \Phi) \sqsupset \Phi)$}
\RightLabel{d adj}
\UnaryInfC{$\mathrm{F}\neg\neg{\downarrow}\alpha \vdash (\alpha \sqsupset \Phi) \sqsupset \Phi$}
\UnaryInfC{$(\alpha \sqsupset \Phi) \,,\, \mathrm{F}\neg\neg{\downarrow}\alpha \vdash \Phi$}
\LeftLabel{G}
\UnaryInfC{$\alpha \sqsupset (\Phi \,,\, \mathrm{F}\neg\neg{\downarrow}\alpha) \vdash \Phi$}
\UnaryInfC{$\Phi \,,\, \mathrm{F}\neg\neg{\downarrow}\alpha \vdash \alpha \,,\, \Phi$}
\UnaryInfC{$\mathrm{F}\neg\neg{\downarrow}\alpha \vdash \alpha \,,\, \Phi$}
\UnaryInfC{$\mathrm{F}\neg\neg{\downarrow}\alpha \vdash \alpha$}
\RightLabel{d adj}
\UnaryInfC{$\neg\neg{\downarrow}\alpha \vdash {\Downarrow}\alpha$}
\UnaryInfC{$\neg\neg{\downarrow}\alpha \vdash {\downarrow}\alpha$}
\end{prooftree}
$$

References

1. Abramsky, S., Väänänen, J.: From IF to BI. Synthese **167**(2), 207–230 (2009)
2. Belnap, N.: Display logic. J. Philos. Logic **11**, 375–417 (1982)
3. Ciardelli, I.: Questions in Logic. Ph.D. thesis, University of Amsterdam (2016)
4. Ciardelli, I.: Dependency as question entailment. In: Vollmer, H., Abramsky, S., Kontinen, J., Väänänen, J. (eds.) Dependence Logic: Theory and Application, Progress in Computer Science and Applied Logic. Birkhauser (2016, to appear)
5. Ciardelli, I., Roelofsen, F.: Inquisitive logic. J. Philos. Logic **40**(1), 55–94 (2011)
6. Frittella, S., Greco, G., Kurz, A., Palmigiano, A.: Multi-type display calculus for propositional dynamic logic. J. Logic Comput. exu064v1-exu064 (2014). Special Issue on Substructural Logic and Information Dynamics
7. Frittella, S., Greco, G., Kurz, A., Palmigiano, A., Sikimić, V.: A multi-type display calculus for dynamic epistemic logic. J. Logic Comput. exu068v1-exu068 (2014). Special Issue on Substructural Logic and Information Dynamics
8. Frittella, S., Greco, G., Kurz, A., Palmigiano, A., Sikimić, V.: Multi-type sequent calculi. In: Zawidzki, M., Indrzejczak, A., Kaczmarek, J. (eds.) Trends in Logic XIII, pp. 81–93. Lodź University Press, Łódź (2014)
9. Frittella, S., Greco, G., Kurz, A., Palmigiano, A., Sikimić, V.: A proof-theoretic semantic analysis of dynamic epistemic logic. J. Logic Comput. exu063v2-exu063 (2015). Special Issue on Substructural Logic and Information Dynamics
10. Greco, G., Kurz, A., Palmigiano, A.: Dynamic epistemic logic displayed. In: Huang, H., Grossi, D., Roy, O. (eds.) LORI. LNCS, vol. 8196, pp. 135–148. Springer, Heidelberg (2013)
11. Greco, G., Ma, M., Palmigiano, A., Tzimoulis, A., Zhao, Z.: Unified correspondence as a proof-theoretic tool. J. Logic Comput. (forthcoming)
12. Groenendijk, J.: Inquisitive semantics: two possibilities for disjunction. In: Bosch, P., Gabelaia, D., Lang, J. (eds.) TbiLLC 2007. LNCS, vol. 5422, pp. 80–94. Springer, Heidelberg (2009)
13. Groenendijk, J., Roelofsen, F.: Inquisitive semantics and pragmatics. In: Larrazabal, J.M., Zubeldia, L. (eds.) Meaning, Content, and Argument: Proceedings of the ILCLI International Workshop on Semantics, Pragmatics, and Rhetoric, pp. 41–72. University of the Basque Country Publication Service, May 2009
14. Hodges, W.: Compositional semantics for a language of imperfect information. Logic J. IGPL **5**, 539–563 (1997)
15. Hodges, W.: Some strange quantifiers. In: Mycielski, J., Rozenberg, A., Salomaa, A. (eds.) Structures in Logic and Computer Science: A Selection of Essays in Honor of A. Ehrenfeucht. LNCS, vol. 1261, pp. 51–65. Springer, Heidelberg (1997)
16. Kreisel, G., Putnam, H.: Eine Unableitbarkeitsbeweismethode für den intuitionistischen Aussagenkalkül. Archiv für Mathematische Logik und Grundlagenforschung **3**, 74–78 (1957)
17. Maksimova, L.: On maximal intermediate logics with the disjunction property. Stud. Logica **45**(1), 69–75 (1986)
18. Mascarenhas, S.: Inquisitive semantics and logic. Master's thesis, University of Amsterdam (2009)
19. Medvedev, J.T.: Finite problems. Sov. Math. Dokl. **3**(1), 227–230 (1962)
20. Roelofsen, F.: Algebraic foundations for the semantic treatment of inquisitive content. Synthese **190**, 79–102 (2013)
21. Sano, K.: Sound and complete tree-sequent calculus for inquisitive logic. In: The Sixteenth Workshop on Logic, Language, Information, and Computation (2009)

22. Väänänen, J.: Dependence Logic: A New Approach to Independence Friendly Logic. Cambridge University Press, Cambridge (2007)
23. Yang, F.: On Extensions and Variants of Dependence Logic. Ph.D. thesis, University of Helsinki (2014)
24. Yang, F., Väänänen, J.: Propositional logics of dependence. Ann. Pure Appl. Logic **167**(7), 557–589 (2016)

A Model-Theoretic Characterization of Constant-Depth Arithmetic Circuits

Anselm Haak and Heribert Vollmer(✉)

Institut für Theoretische Informatik, Leibniz Universität Hannover, Hannover, Germany
{haak,vollmer}@thi.uni-hannover.de

Abstract. We study the class $\#\mathrm{AC}^0$ of functions computed by constant-depth polynomial-size arithmetic circuits of unbounded fan-in addition and multiplication gates. No model-theoretic characterization for arithmetic circuit classes is known so far. Inspired by Immerman's characterization of the Boolean class AC^0, we remedy this situation and develop such a characterization of $\#\mathrm{AC}^0$. Our characterization can be interpreted as follows: Functions in $\#\mathrm{AC}^0$ are exactly those functions counting winning strategies in first-order model checking games. A consequence of our results is a new model-theoretic characterization of TC^0, the class of languages accepted by constant-depth polynomial-size majority circuits.

1 Introduction

Going back to questions posed by Heinrich Scholz and Günter Asser in the early 1960s, Ronald Fagin [6] laid the foundations for the areas of finite model theory and descriptive complexity theory. He characterized the complexity class NP as the class of those languages that can be defined in predicate logic by existential second-order sentences: NP = ESO. His result is the cornerstone of a wealth of further characterizations of complexity classes, cf. the monographs [5,11,12].

Fagin's Theorem has found a nice generalization: Considering first-order formulae with a free relational variable, instead of asking if there *exists* an assignment to this variable that makes the formula true (ESO), we now ask to *count* how many assignments there are. In this way, the class #P is characterized: #P = #FO [13].

But also "lower" complexity classes, defined by families of Boolean circuits, have been considered in a model-theoretical way. Most important for us is the characterization of the class AC^0, the class of languages accepted by families of Boolean circuits of unbounded fan-in, polynomial size and constant depth, by first-order formulae. This correspondence goes back to Immerman and his co-authors [2,10], but was somewhat anticipated by [8]. Informally, this may be written as $\mathrm{AC}^0 = \mathrm{FO}$; and there are two ways to make this formally correct—a non-uniform one: $\mathrm{AC}^0 = \mathrm{FO}[\mathrm{Arb}]$, and a uniform one: FO-uniform $\mathrm{AC}^0 = \mathrm{FO}[+, \times]$ (for details, see below).

Supported by DFG grant VO 630/8-1.

J. Väänänen et al. (Eds.): WoLLIC 2016, LNCS 9803, pp. 234–248, 2016.
DOI: 10.1007/978-3-662-52921-8_15

In the same way as #P can be seen as the counting version of NP, there is a counting version of AC^0, namely $\#\mathrm{AC}^0$, the class of those functions counting accepting proof-trees of AC^0-circuits. A proof-tree is a minimal sub-circuit of the original circuit witnessing that it outputs 1. Equivalently, $\#\mathrm{AC}^0$ can be characterized as those functions computable by polynomial-size constant-depth circuits with unbounded fan-in $+$ and $\times$ gates (and Boolean inputs); for this reason we also speak of *arithmetic* circuit classes.

For such arithmetic classes, no model-theoretic characterization is known so far. Our *rationale* is as follows: A Boolean circuit accepts its input if it has at least one proof-tree. An FO-formula (w.l.o.g. in prenex normal form) holds for a given input if there are Skolem functions determining values for the existentially quantified variables, depending on those variables quantified to the left. By establishing a one-one correspondence between proof-trees and Skolem functions, we show that the class $\#\mathrm{AC}^0$, defined by counting proof-trees, is equal to the class of functions counting Skolem functions, or, alternatively, winning-strategies in first-order model-checking games: $\mathrm{AC}^0 = \#\text{Skolem-FO} = \#\text{Win-FO}$. We prove that this equality holds in the non-uniform as well as in the uniform setting.

It seems a natural next step to allow first-order formulae to "talk" about winning strategies, i.e., allow access to #Win-FO-functions (like to an oracle). We will prove that in doing so, we obtain a new model-theoretic characterization of the circuit class TC^0 of polynomial-size constant-depth MAJORITY circuits.

This paper is organized as follows: In the upcoming section, we will introduce the relevant circuit classes and logics, and we state characterizations of the former by the latter known from the literature. We will also recall arithmetic circuit classes and define our logical counting classes #Skolem-FO and #Win-FO. Section 3 proves our characterization of non-uniform $\#\mathrm{AC}^0$, while Sect. 4 proves our characterization of uniform $\#\mathrm{AC}^0$. Section 5 presents our new characterization of the circuit class TC^0. Finally, Sect. 6 concludes with some open questions.

Due to space restrictions, many proofs have to be omitted and will be given in the full paper.

2 Circuit Classes, Counting Classes, and Logic

2.1 Non-uniform Circuit Classes

A relational vocabulary is a tuple $\sigma = (R_1^{a_1}, \ldots, R_k^{a_k})$, where R_i are relation symbols and a_i their arities, $1 \leq i \leq k$. We define first-order formulae over σ as usual (see, e.g., [5,11]). First-order structures fix the set of elements (the universe) as well as interpretations for the relation symbols in the vocabulary. Semantics is defined as usual. For a structure $\mathcal{A}$, $|\mathcal{A}|$ denotes its universe. We only consider finite structures here, which means their universes are finite.

Since we want to talk about languages accepted by Boolean circuits, we will use the *vocabulary*

$$\tau_{\text{string}} = (\leq^2, S^1)$$

of binary strings. A binary string is represented as a structure over this vocabulary as follows: Let $w \in \{0,1\}^*$ with $|w| = n$. Then the structure representing this string has universe $\{0,\dots,n-1\}$, $\leq^2$ is interpreted as the $\leq$-relation on the natural numbers and $x \in S$, iff the x'th bit of w is 1. The structure corresponding to string w will be called $\mathcal{A}_w$. Vice versa, structure $\mathcal{A}_w$ is simply encoded by w itself: The bits define which elements are in the S-relation—the universe and the order are implicit. This encoding can be generalized to binary encodings of arbitrary σ-structures $\mathcal{A}$. We will use the notation $\mathrm{enc}_\sigma(\mathcal{A})$ for such an encoding.

A Boolean circuit C is a directed acyclic graph (dag), whose nodes (also called gates) are marked with either a Boolean function (in our case $\wedge$ or $\vee$), a constant (0 or 1), or a (possibly negated) query of a particular position of the input. Also, one gate is marked as the output gate. On any input x, a circuit computes a Boolean function f_C by evaluating all gates according to what they are marked with. The value of the output gate gives then the result of the computation of C on x, i.e., $f_C(x)$.

A single circuit computes only a finite Boolean function. When we want circuits to work on different input lengths, we have to consider families of circuits: A family contains one circuit for any input length $n \in \mathbb{N}$. Families of circuits allow us to talk about languages being accepted by circuits: A circuit family $\mathcal{C} = (C_n)_{n\in\mathbb{N}}$ is said to accept (or decide) the language L, if it computes its characteristic function c_L:

$$C_{|x|}(x) = c_L(x) \text{ for all } x.$$

Since we will describe Boolean circuits by FO-formulae, we define the vocabulary

$$\tau_{\mathrm{circ}} = (E^2, G^1_\wedge, G^1_\vee, B^1, r^1),$$

the *vocabulary of Boolean circuits*. The relations are interpreted as follows:

- $E(x,y)$: y is a child of x
- $G_\wedge(x)$: gate x is an and-gate
- $G_\vee(x)$: gate x is an or-gate
- $B(x)$: Gate x is a true leaf of the circuit
- $r(x)$: x is the root of the circuit

The definition from [11] is more general because it allows negations to occur arbitrary in a circuit. Here we only consider circuits in *negation normal form*, i.e., negations are only applied to input bits. This restriction is customary for arithmetic circuits like for the class $\#\mathrm{AC}^0$ to be defined below.

The complexity classes in circuit complexity are classes of languages that can be decided by circuit families with certain restrictions on their depth or size. The depth here is the length of a longest path from any input gate to the output gate of a circuit and the size is the number of non-input gates in a circuit. Depth and size of a circuit family are defined as functions accordingly.

Definition 1. The class AC^0 is the class of all languages decidable by Boolean circuit families of constant depth and polynomial size.

In this definition we do not have any restrictions on the computability of the function $n \mapsto \langle C_n \rangle$, i.e., the function computing (an encoding of) the circuit for a given input length. This phenomenon is referred to as *non-uniformity*, and it leads to undecidable problems in AC^0. In first-order logic there is a class that has a similar concept, the class FO[Arb], to be defined next.

For arbitrary vocabularies τ, we consider formulae over $\tau_{\text{string}} \cup \tau$ and our input structures will always be τ_{string}-structures $\mathcal{A}_w$ for a string $w \in \{0,1\}^*$. To evaluate a formula we additionally specify a (non-uniform) family $I = (I_n)_{n \in \mathbb{N}}$ of interpretations of the relation symbols in τ. For $\mathcal{A}_w$ and I as above we now evaluate $\mathcal{A}_w \models_I \varphi$ by using the universe of $\mathcal{A}_w$ and the interpretations from both $\mathcal{A}_w$ and $I_{|w|}$. The language defined by a formula φ and a family of interpretations I is

$$L_I(\varphi) =_{\text{def}} \{w \in \{0,1\}^* \mid \mathcal{A}_w \models_I \varphi\}$$

This leads to the following definition of FO[Arb] (equivalent to the one given in [14]):

Definition 2. A language L is in FO[Arb], if there are an arbitrary vocabulary τ, a first-order sentence φ over $\tau_{\text{string}} \cup \tau$ and a family $I = (I_n)_{n \in \mathbb{N}}$ of interpretations of the relation symbols in τ such that:

$$L_I(\varphi) = L$$

It is known that the circuit complexity class AC^0 and the model theoretic class FO[Arb] are in fact the same:

Theorem 3 (see, e.g., [14]). $\mathrm{AC}^0 = \mathrm{FO}[\mathrm{Arb}]$.

2.2 Uniform Circuit Classes

As already stated, non-uniform circuits are able to solve undecidable problems, even when restricting size and depth of the circuits dramatically. Thus, the non-uniformity somewhat obscures the real complexity of problems. There are different notions of uniformity to deal with this problem: The computation of the circuit $C_{|x|}$ from x must be possible within certain bounds, e.g. polynomial time, logarithmic space, logarithmic time. Since we are dealing with FO-formulae, the type of uniformity we will need is first-order uniformity, to be defined in this section.

In the logical languages, "uniformity" means we now remove the non-uniform family of interpretations from the definition of FO[Arb], and replace it with two special symbols for arithmetic, a 3-ary relation $+$ (with the intended interpretation $+(i,j,k)$ iff $i+j=k$) and a 3-ary relation $\times$ (with the intended interpretation $\times(i,j,k)$ iff $i \cdot j = k$).

Definition 4. A language L is in $\mathrm{FO}[+,\times]$, if there is a first-order sentence φ over $\tau_{\text{string}} \cup \{+,\times\}$ such that

$$\mathcal{A}_w \models_I \varphi \Leftrightarrow w \in L,$$

where I interprets $+$ and $\times$ in the intended way.

In the circuit world, as mentioned, "uniformity" means we can access from any given input structure $\mathcal{A}_w$ also the circuit $C_{|w|}$. The way we achieve this is via *FO-interpretations*.

In the following, for any vocabulary σ, STRUC$[\sigma]$ denotes the set of all structures over σ.

Definition 5. Let σ, τ be vocabularies, $\tau = (R_1^{a_1}, \ldots, R_r^{a_r})$, and let $k \in \mathbb{N}$. A *first-order interpretation* (or *FO-interpretation*)

$$I \colon \mathrm{STRUC}[\sigma] \to \mathrm{STRUC}[\tau]$$

is given by a tuple of FO-formulae $\varphi_0, \varphi_1, \ldots, \varphi_r$ over the vocabulary σ. φ_0 has k free variables and φ_i has $k \cdot a_i$ free variables for all $i \geq 1$. For each structure $\mathcal{A} \in \mathrm{STRUC}[\sigma]$, these formulae define the structure

$$I(\mathcal{A}) = \big(|I(\mathcal{A})|, R_1^{I(\mathcal{A})}, \ldots, R_r^{I(\mathcal{A})}\big) \in \mathrm{STRUC}[\tau],$$

where the universe is defined by φ_0 and the relations are defined by $\varphi_1, \ldots, \varphi_r$ in the following way:

$$|I(\mathcal{A})| = \big\{\langle b^1, \ldots, b^k\rangle \bigm| \mathcal{A} \vDash \varphi_0(b^1, \ldots, b^k)\big\} \text{ and}$$

$$R_i^{I(\mathcal{A})} = \big\{(\langle b_1^1, \ldots, b_1^k\rangle, \ldots, \langle b_{a_i}^1, \ldots, b_{a_i}^k\rangle) \in |I(\mathcal{A})|^{a_i} \bigm| \mathcal{A} \vDash \varphi_i(b_1^1, \ldots, b_{a_i}^k)\big\}$$

The name FO-interpretations was used, e.g., in [4]. Sometimes they are also referred to as first-order queries, see, e.g., [11]. They are not to be confused with interpretations of relation symbols as in Sect. 2.1. It is customary to use the same symbol I in both cases.

Analogously, FO$[+, \times]$-interpretations are interpretations given by tuples of FO$[+, \times]$-formulae.

Definition 6. A circuit family $\mathcal{C} = (C_n)_{n\in\mathbb{N}}$ is said to be FO$[+, \times]$-*uniform* if there is an FO$[+, \times]$-interpretation

$$I \colon \mathrm{STRUC}[\tau_{\mathrm{string}}] \to \mathrm{STRUC}[\tau_{\mathrm{circ}}]$$

mapping from an input word w given as a structure $\mathcal{A}_w$ over τ_{string} to the circuit $C_{|w|}$ given as a structure over the vocabulary τ_{circ}.

Now we can define the FO-uniform version of AC^0:

Definition 7. FO$[+, \times]$-uniform AC^0 is the class of all languages that can be decided by FO$[+, \times]$-uniform AC^0 circuit families.

Thus, if $\mathcal{C} = (C_n)_{n\in\mathbb{N}}$ is an FO$[+, \times]$-uniform circuit family, we can define from any given input structure $\mathcal{A}_w$ also the circuit $C_{|w|}$ in a first-order way.

Interestingly, uniform AC^0 coincides with FO with built-in arithmetic:

Theorem 8 (see, e.g., [11]). FO$[+, \times]$-*uniform* $\mathrm{AC}^0 = \mathrm{FO}[+, \times]$.

Alternatively, we can replace $+$ and $\times$ in the above theorem by the binary symbol BIT with the meaning $\mathrm{BIT}(i, j)$ iff the ith bit in the binary representation of j is 1, see also [11].

2.3 Counting Classes

Building on the previous definitions we want to define next counting classes. The objects counted on circuits are proof trees: A *proof tree* is a minimal subtree showing that a circuit evaluates to true for a given input. For this, we first unfold the given circuit into tree shape, and we further require that it is in negation normal form. A proof tree then is a tree we get by choosing for any $\vee$-gate exactly one child and for any $\wedge$-gate all children, such that every leaf which we reach in this way is a true literal.

Now, $\#\mathrm{AC}^0$ is the class of functions that "count proof trees of AC^0 circuits":

Definition 9. (FO[$+, \times$]-uniform) $\#\mathrm{AC}^0$ is the class of all functions $f\colon \{0,1\}^* \to \mathbb{N}$ for which there is a (FO[$+, \times$]-uniform) circuit family $\mathcal{C} = (C_n)_{n\in\mathbb{N}}$ such that for any word x, $f(x)$ equals the number of proof trees of $C_{|x|}(x)$.

It is the aim of this paper to give model-theoretic characterizations of these classes. The only model-theoretic characterization of a counting class that we are aware of is the following: In [13], a counting version of FO was defined, inspired by Fagin's characterization of NP: Functions in this class count assignments to free relational variables in FO-formulas. However, it is known that $\#\mathrm{P} = \#\mathrm{FO}$, i.e., this counting version of FO coincides with the much higher counting class $\#\mathrm{P}$ of functions counting accepting paths of nondeterministic polynomial-time Turing machines. It is known that $\#\mathrm{AC}^0 \subsetneq \#\mathrm{P}$. Thus, we need some weaker notion of counting.

Suppose we are given a τ_{string}-formula φ in prenex normal form,

$$\varphi \triangleq \exists y_1 \forall z_1 \exists y_2 \forall z_2 \ldots \exists y_{k-1} \forall z_{k-1} \exists y_k \; \psi(\overline{y}, \overline{z})$$

for quantifier-free ψ. If we want to satisfy φ in a word model $\mathcal{A}_w$, we have to find an assignment for y_1 such that for all z_1 we have to find an assignment for $y_2 \ldots$ such that ψ with the chosen variables holds in $\mathcal{A}_w$. Thus, the number of ways to satisfy ϕ consists in the number of picking the suitable y_i, depending on the universally quantified variables to the left, such that ψ holds, in other words, the number of Skolem functions for the existentially quantified variables.

Definition 10. A function $g\colon \{0,1\}^* \to \mathbb{N}$ is in the class #Skolem-FO[Arb] if there is a vocabulary τ, a sequence of interpretations $I = (I_n)_{n\in\mathbb{N}}$ for τ and a first-order sentence φ over $\tau_{\text{string}} \cup \tau$ in prenex normal form

$$\varphi \triangleq \exists y_1 \forall z_1 \exists y_2 \forall z_2 \ldots \exists y_{k-1} \forall z_{k-1} \exists y_k \; \psi(\overline{y}, \overline{z})$$

such that for all $w \in \{0,1\}^*$, $g(w)$ is equal to the number of tuples $(f_1, \ldots, f_k)$ of functions such that

$$\mathcal{A}_w \;\models_I\; \forall z_1 \ldots \forall z_{k-1} \; \psi(f_1, f_2(z_1), \ldots, f_k(z_1, \ldots, z_{k-1}), z_1, \ldots, z_{k-1})\}$$

This means that #Skolem-FO[Arb] contains those functions that, for a fixed FO-formula, map an input w to the number of Skolem functions on $\mathcal{A}_w$.

A different view on this counting class is obtained by recalling the well-known game-theoretic approach to first-order model checking: Model checking for FO-formulae (in prenex normal form) can be characterized using a two player game: The verifier wants to show that the formula evaluates to true, whereas the falsifier wants to show that it does not. For each quantifier, one of the players chooses an action: For an existential quantifier, the verifier chooses which element to take (because he needs to prove that there is an element). For a universal quantifier, the falsifier chooses which element to take (because he needs to prove that there is a choice falsifying the formula following after the quantifier). When all quantifiers have been addressed, it is checked whether the quantifier-free part of the formula is true or false. If it is true, the verifier wins. Else, the falsifier wins. Now the formula is fulfilled by a given model, iff there is a winning strategy (for the verifier).

Definition 11. A function f is in #Win-FO[Arb], if there are a vocabulary τ, a sequence of interpretations $I = (I_n)_{n\in\mathbb{N}}$ for τ and a first-order sentence φ in prenex normal form over $\tau_{\text{string}} \cup \tau$ such that for all $w \in \{0,1\}^*$, $f(w)$ equals the number of winning strategies for the verifier in the game for $\mathcal{A}_w \vDash_I \varphi$.

The correspondence between Skolem functions and winning strategies has been observed in far more general context, see, e.g., [7]. In our case, this means that

$$\#\text{Skolem-FO[Arb]} = \#\text{Win-FO[Arb]}.$$

Analogously we define the uniform version (where we only state using the notion of the model checking games):

Definition 12. A function f is in #Win-FO$[+,\times]$, if there is a first-order sentence φ in prenex normal form over $\tau_{\text{string}} \cup \{+,\times\}$ such that for all $w \in \{0,1\}^*$, $f(w)$ equals the number of winning strategies for the verifier in the game for $\mathcal{A}_w \vDash_I \varphi$, where I interprets $+$ and $\times$ in the intended way.

We will use $\#\text{Win}(\varphi, \mathcal{A}, I)$ ($\#\text{Win}(\varphi, \mathcal{A})$, resp.) to denote the number of winning strategies for φ evaluated on the structure $\mathcal{A}$ and the interpretation I (the structure $\mathcal{A}$ and the intended interpretation of $+$ and $\times$, resp.).

In the previous two definitions we could again have replaced $+$ and $\times$ by BIT.

In the main result of this paper, we will show that the thus defined logical counting classes equal the previously defined counting classes for constant-depth circuits.

3 A Model-Theoretic Characterization of $\#\mathbf{AC^0}$

We first note that there is a sort of a closed formula for the number of winning strategies of FO-formulae on given input structures:

Lemma 13. *Let τ_1, τ_2 be vocabularies and I an interpretation of τ_2. Let φ be an FO-formula in prenex normal form over the vocabulary $\tau_1 \cup \tau_2$ of the form*

$$\varphi \triangleq Q_1 x_1 \dots Q_n x_n \psi,$$

where $Q_i \in \{\exists, \forall\}$.
Let $\mathcal{A}$ be a τ_1-structure and I a sequence of interpretations for τ_2 . Then the number of winning strategies of $\mathcal{A} \vDash_I \varphi$ is the following:

$$\#\mathrm{Win}(\varphi, \mathcal{A}, I) = \Delta_1 \Delta_2 \cdots \Delta_n([\mathcal{A} \vDash_I \varphi(a_1 \dots a_n)]),$$

where

$$\Delta_i = \begin{cases} \sum\limits_{a_i \in |\mathcal{A}|} & \text{, if } Q_i = \exists \\ \prod\limits_{a_i \in |\mathcal{A}|} & \text{, if } Q_i = \forall \end{cases}$$

and $[\mathcal{A} \vDash_I \varphi(a_1, \dots, a_n)]$ is interpreted as either 0 or 1 depending on its truth value.

In the uniform case, $\#\mathrm{Win}(\varphi, \mathcal{A})$ is the special case of $\#\mathrm{Win}(\varphi, \mathcal{A}, I)$ where I interprets $+$ and $\times$ in the intended way.

Our main theorem can now be stated as follows:

Theorem 14. $\#\mathrm{AC}^0 = \#\mathrm{Win}\text{-}\mathrm{FO}[\mathrm{Arb}]$

The rest of this section is devoted to a proof of this theorem.

Proof. $\subseteq$: Let f be a function in $\#\mathrm{AC}^0$ and $\mathcal{C} = (C_n)_{n \in \mathbb{N}}$ a AC^0-circuit witnessing this. Assume that all C_n are already trees and all leaves have the same depth (the latter can be achieved easily by adding and-gates with only one input). Also, we can assume that the circuit always uses and- and or-gates alternating beginning with an and-gate in the root. This can be achieved by doubling the depth of the circuit: For every layer of the old circuit we use an and-gate followed by an or-gate. If the original gate in that layer was an and-gate, we just put an (one-input) or-gate on top of every child and connect those or-gates to the and-gate. If the original gate in that layer was an or-gate, we put an and-gate above it with the or-gate as its only child.

Let $w \in \{0,1\}^*$ be an input, r be the root of $C_{|w|}$ and k the depth of C_n for all n. The value $f(w)$ can be given as follows:

$$f(w) = \prod_{\substack{y_1 \text{ is a} \\ \text{child of } r}} \sum_{\substack{y_2 \text{ is a} \\ \text{child of } y_1}} \cdots \bigcirc_{\substack{y_k \text{ is a} \\ \text{child of } y_{k-1}}} \begin{cases} 1 & \text{, if } y_k \text{ is a true literal} \\ 0 & \text{, if } y_k \text{ is a false literal} \end{cases},$$

where $\bigcirc = \begin{cases} \prod & \text{, if } k \text{ is odd,} \\ \sum & \text{, if } k \text{ is even.} \end{cases}$

We will now build an FO-sentence φ over $\tau_{\mathrm{string}} \cup \tau_{\mathrm{circ}}$ such that for any input $w \in \{0,1\}^*$, the number of winning strategies to verify $\mathcal{A}_w \vDash_{\mathcal{C}} \varphi$ equals the number of proof trees of the circuit $C_{|w|}$ on input w. Note that the circuit family $\mathcal{C}$ as a τ_{circ}-structure can directly be used as the non-uniform family of interpretations for the evaluation of φ. Since only one universe is used for evaluation and it is determined by the input structure $\mathcal{A}_w$, the gates in this τ_{circ}-structure have to be tuples of variables ranging over the universe of $\mathcal{A}_w$. To simplify the presentation, we assume in the following that we do not need tuples—a single element of the universe already corresponds to a gate. The proof can be generalized to the case where this assumption is dropped.

The sentence φ over $\tau_{\text{string}} \cup \tau_{\text{circ}}$ can be given as

$$\begin{aligned}\varphi :=& \exists y_0 \forall y_1 \exists y_2 \dots Q_k y_k \\ & r(y_0) \wedge \left(\left(\left(\bigwedge_{1 \le i \le k} E(y_i, y_{i-1}) \right) \wedge B(y_k) \right) \right. \\ & \left. \vee \bigvee_{\substack{1 \le i \le k, \\ i \text{ odd}}} \left(\bigwedge_{1 \le j < i} (E(y_j, y_{j-1})) \wedge \neg E(y_i, y_{i-1}) \wedge \bigwedge_{i < j \le k} y_j = r \right) \right),\end{aligned}$$

where $Q_k = \begin{cases} \exists & \text{, if } k \text{ is odd,} \\ \forall & \text{, if } k \text{ is even.} \end{cases}$

We now need to show that the number of winning strategies for $\mathcal{A}_w \vDash_I \varphi$ is equal to the number of proof trees of the circuit $C_{|w|}$ on input w. For this, let

$$\begin{aligned}\varphi^{(n)}(y_1, \dots, y_n) :=& Q_{n+1} y_{n+1} \dots Q_k y_k \\ & \left(\bigwedge_{1 \le i \le k} \big(E(y_i, y_{i-1})\big) \wedge B(y_k) \right) \vee \\ & \bigvee_{\substack{n+1 \le i \le k, \\ i \text{ odd}}} \left(\bigwedge_{1 \le j < i} \big(E(y_j, y_{j-1})\big) \wedge \neg E(y_i, y_{i-1}) \wedge \right. \\ & \left. \bigwedge_{i < j \le k} y_j = r \right),\end{aligned}$$

where $Q_{n+1}, \dots, Q_{k-1}$ are the quantifiers preceding Q_k. Note that the start of the index i on the big "or" changed compared to φ. Also, r is notation for the root of the circuit, although we formally do not use constants. In the following we will use the abbreviation

$$\#w(\varphi) = \#\text{Win}(\varphi, \mathcal{A}_w, I).$$

We now show by induction that:

$$\#w(\varphi) = \prod_{\substack{y_1 \text{ is a} \\ \text{child of } r}} \sum_{\substack{y_2 \text{ is a} \\ \text{child of } y_1}} \cdots \bigcirc_{\substack{y_n \text{ is a} \\ \text{child of } y_{n-1}}} \#w(\varphi^{(n)}[y_0/r]).$$

Replacing y_0 by r is only done for simplicity.

Induction basis ($n = 0$): The induction hypothesis here simply states

$$\#w(\varphi) = \#w(\varphi^{(0)}[y_0/r]),$$

which holds by definition.

Induction step ($n \to n+1$): We can directly use the induction hypothesis here:

$$\#w(\varphi) = \prod_{\substack{y_1 \text{ is a} \\ \text{child of } r}} \sum_{\substack{y_2 \text{ is a} \\ \text{child of } y_1}} \cdots \bigcirc_{\substack{y_n \text{ is a} \\ \text{child of } y_{n-1}}} \#w(\varphi^{(n)}[y_0/r]),$$

so it remains to show that

$$\#w(\varphi^{(n)}[y_0/r]) = \bigcirc_{\substack{y_{n+1} \text{ is a} \\ \text{child of } y_n}} \#w(\varphi^{(n+1)}[y_0/r])$$

We distinguish two cases: Depending on whether $n+1$ is even or odd, the $(n+1)$-st quantifier is either an existential or a universal quantifier. In the same way all gates of that depth in the circuits from $\mathcal{C}$ are either or- or and-gates.

Case 1: $n+1$ is odd, so the $(n+1)$-st quantifier is a universal quantifier. Thus, from $\#w(\varphi^{(n)}[y_0/r])$ we get a $\prod$-operator, which is the same we get for an and-gate in the corresponding circuit. We now need to check over which values of y_{n+1} the product runs:

The big conjunction may only be true if y_{n+1} is a child of y_n.

The big disjunction may become true for values of y_{n+1} which are no children of y_n only if all variables quantified after y_{n+1} are set to r (the choice of r here is arbitrary and was only made because r is the only constant in the circuit). Also, the disjunct for $i = n$ can only be made true if y_{n+1} is not a child of y_n, so we can drop it if y_{n+1} is a child of y_n. Since for all values of y_{n+1} that are not children of y_n we fix all variables quantified afterwards, we get:

$$\begin{aligned} \prod_{y_{n+1} \in |\mathcal{A}_w|} \#w(\varphi^{(n+1)}[y_0/r]) &= \prod_{\substack{y_{n+1} \in |\mathcal{A}_w|, \\ y_{n+1} \text{is a child of } y_n}} \#w(\varphi^{(n+1)}[y_0/r]) \cdot \prod_{\substack{y_{n+1} \in |\mathcal{A}_w|, \\ y_{n+1} \text{is not a child of } y_n}} 1 \\ &= \prod_{\substack{y_{n+1} \in |\mathcal{A}_w|, \\ y_{n+1} \text{is a child of } y_n}} \#w(\varphi^{(n+1)}[y_0/r]). \end{aligned}$$

Thus, we get a product only over the children of y_n and can drop the disjunct for $i = n$ from the formula for the next step.

Case 2: $n + 1$ is even, so the $(n + 1)$-st quantifier is an existential quantifier. Therefore, we get a $\sum$-operator from $\#w(\varphi^{(n)}[y_0/r])$, which is the same we get for an or-gate in the corresponding circuit. We now need to check over which values of y_{n+1} the sum runs:

The big conjunction can only be true if y_{n+1} is a child of y_n.

The big disjunction can also only be true if y_{n+1} is a child of y_n.

Thus, we directly get the sum

$$\sum_{\substack{y_{n+1} \in |\mathcal{A}_w|, \\ y_{n+1} \text{ is a child of } y_n}} \#w(\varphi^{(n+1)}[y_0/r]).$$

Here, $\varphi^{(n+1)}$ does not drop a disjunct. This concludes the induction.

For $\varphi^{(k)} = \text{trueLiteral}(y_k)$, we get

$$\#w(\varphi^{(k)}) = \begin{cases} 1 & \text{if } \mathcal{A}_w \vDash_I \varphi^{(k)} \\ 0 & \text{else} \end{cases} = \begin{cases} 1 & \text{if } y_k \text{ is a true literal,} \\ 0 & \text{else.} \end{cases}$$

$\supseteq$: Let f be a function in #Win-FO[Arb]. Let τ be a vocabulary and φ be a formula over $\tau \cup \tau_{\text{string}}$ together with the non-uniform family $I = (I_n)_{n \in \mathbb{N}}$ of interpretations of the relation symbols in τ a witness for $f \in$ #Win-FO[Arb]. Let k be the length of the quantifier prefix of φ. We now sketch how to construct a circuit family $\mathcal{C} = (C_n)_{n \in \mathbb{N}}$ that shows $f \in \#\text{AC}^0$. The gates of the circuit are $\langle a_1, \ldots, a_i \rangle$ with $1 \leq i \leq k$ and $a_j \in |\mathcal{A}_w|$ for all j. Each such gate has the meaning that we set the first i quantified variables to the values $a_1, \ldots, a_i$. Therefore, for the i'th quantifier of φ and for any choice of $a_1, \ldots, a_{i-1}$, $\langle a_1, \ldots, a_{i-1} \rangle$ is an and-gate if the quantifier was $\forall$ and an or-gate if the quantifier was $\exists$. Also, if $i \leq k$ we add as children to each such gate $\langle a_1, \ldots, a_{i-1}, a_i \rangle$ for all $a_i \in |\mathcal{A}_w|$.

On the lowest layer, where values have been assigned to all quantified variables, we add to every gate a circuit evaluating the quantifier-free part of φ for the choices made on that specific path. This can be done with a DNF which in each disjunct accesses the same variables. Thus, for any input there is exactly one proof tree for each DNF, if that DNF is true (and none otherwise). The size of these DNFs is constant, since they directly result from the quantifier-free part of φ. Handling the non-uniform family of interpretations I does not lead to problems, because for each fixed circuit C_n the input length and thus the specific interpretation I_n is fixed and thus only a Boolean function depending on the input bits has to be computed. The non-uniformity of the circuit family $\mathcal{C}$ is used to build the different C_n depending on the different I_n.

Now by Lemma 13 it is clear that counting proof trees on this circuit family leads to the same function as counting winning strategies for the verifier for $\mathcal{A}_w \vDash_I \varphi$. □

4 The Uniform Case

Next we want to transfer this result to the uniform setting. In the direction from right to left we will have to show that the constructed circuit is uniform, which is straightforward. On the other hand, the following important point changes in the direction from left to right: We have to actually replace queries to $C_{|w|}$ in the FO-sentence by the corresponding FO-formulae we get from the FO-interpretation which shows uniformness of $\mathcal{C}$. Since we introduce new quantifiers by this, we have to show how we can keep the counted value the same. That this is possible follows from the following lemma, proving that #Win-FO[$+, \times$] is closed under FO-reductions (exact definitions follow).

Lemma 15. *Let φ be an* FO[$+, \times$]*-formula over some vocabulary τ, and let $I\colon STRUC[\sigma] \to STRUC[\tau]$ be an* FO[$+, \times$]*-interpretation. Then there is an* FO[$+, \times$]*-formula φ' over σ such that for all $\mathcal{A} \in STRUC[\sigma]$,*

$$\#\text{Win}(\varphi', \mathcal{A}) = \#\text{Win}(\varphi, I(\mathcal{A})).$$

As already mentioned, this lemma yields an interesting closure property as a corollary, that is, closure under FO-reductions:

Definition 16. Let $f, g : \{0,1\}^* \to \mathbb{N}$. We say that f is (many-one) first-order reducible to g, in symbols: $f \leq^{\mathrm{fo}} g$, if there are vocabularies σ, τ and an FO$[+,\times]$ interpretation $I : \mathrm{STRUC}[\sigma] \to \mathrm{STRUC}[\tau]$ such that for all $\mathcal{A} \in \mathrm{STRUC}[\sigma]$:

$$f(\mathrm{enc}_\sigma(\mathcal{A})) = g(\mathrm{enc}_\tau(I(\mathcal{A}))).$$

Corollary 17. *On ordered structures with* BIT, *#Win-FO is closed under first-order reductions, that is, if f, g are functions such that $g \in$ #Win-FO and $f \leq^{\mathrm{fo}} g$, then $f \in$ #Win-FO.*

Using Lemma 15 we can now establish the desired result in the FO-uniform setting.

Theorem 18. FO$[+,\times]$*-uniform* $\#\mathrm{AC}^0 =$ *#Win-FO*$[+,\times]$.

Proof (Sketch). $\subseteq$: Let $f \in$ FO-uniform $\#\mathrm{AC}^0$ via the circuit family $\mathcal{C} = (C_n)_{n\in\mathbb{N}}$ and the FO-interpretation I showing its uniformness. With the formula φ from the proof of $\#\mathrm{AC}^0 \subseteq \#\mathrm{Win\text{-}FO[Arb]}$ this means we have for all w:

$$\begin{aligned} f(w) &= \text{number of proof trees of } C_{|w|} \text{ on input } w \\ &= \#\mathrm{Win}(\varphi, C_{|w|}(w)), \end{aligned}$$

where $C_{|w|}(w)$ is given as a τ_{circ}-structure. From φ and I by Lemma 15 we get φ' over vocabulary τ_{string} such that for all $\mathcal{A}_w \in \mathrm{STRUC}[\tau_{\mathrm{string}}]$:

$$\begin{aligned} \#\mathrm{Win}(\varphi', \mathcal{A}_w) &= \#\mathrm{Win}(\varphi, \underbrace{I(\mathcal{A}_w)}_{C_{|w|}(w)}) \\ &= f(w) \end{aligned}$$

$\supseteq$: We can prove this analogously to $\#\mathrm{AC}^0 \supseteq \#\mathrm{Win\text{-}FO[Arb]}$. The only difference is that we need to show FO-uniformity of the circuit. Let f be a function in #Win-FO with witness φ. We now need the formulae $\varphi_{\mathrm{universe}}, \varphi_{G_\wedge}, \varphi_{G_\vee}, \varphi_E, \varphi_I$ and φ_r defining the circuit. We do this by encoding gates in the circuit by suitable k-tuples over $\{0, \ldots, n\}$, where k can be chosen to be the number of quantifiers in φ. The technical details will be given in the full paper. $\square$

5 A Model-Theoretic Characterization of $\mathbf{TC^0}$

We will now introduce the oracle class $\mathrm{AC}^{0^{\#\mathrm{AC}^0}}$ as well as FOCW[Arb], which is a variant of FO with counting. From the known connections between TC^0 and $\#\mathrm{AC}^0$ and from the new connection between $\#\mathrm{AC}^0$ and #Win-FO[Arb] we will then get a new model theoretic characterization for TC^0, the class of all languages accepted by Boolean circuits of polynomial size and constant depth with unbounded fan-in AND, OR, and MAJORITY gates, see [14]. First we want to define the above classes:

Definition 19. $\mathrm{AC}^{0\#\mathrm{AC}^0}$ is the complexity class containing all languages decidable by AC^0-circuit families that may use gates computing bits of a fixed function from $\#\mathrm{AC}^0$. More precisely, for each circuit family we fix a $f \in \#\mathrm{AC}^0$ and can use gates that are labeled with $\#_i$. Such a gate computes the Boolean function

$$\begin{aligned} f_i\colon \{0,1\}^* &\to \{0,1\} \\ \mathrm{bin}(x) &\mapsto \mathrm{BIT}(i, f(x)) \end{aligned}$$

The main result of this section will be a new characterization of the circuit class TC^0 using a certain two-sorted logic.

Definition 20. Given a vocabulary σ, a σ-structure for FOCW[Arb] is a structure of the form

$$\langle \{a_0, \dots, a_{n-1}\}, \{0, \dots, n-1\}, (R_i)^{\mathcal{A}}, +, \times, \underline{\min}, \underline{\max} \rangle,$$

where $\langle \{a_0, \dots, a_{n-1}\}, (R_i)^{\mathcal{A}} \rangle \in \mathrm{STRUC}[\sigma]$, $+$ and $\times$ are the ternary relations corresponding to addition and multiplication in $\mathbb{N}$ and $\underline{\min}, \underline{\max}$ denote 0 and $n-1$, respectively. We assume that the two universes are disjoint. Formulas can have free variables of two sorts.

This logic extends the syntax of first order logic as follows:

- terms of the second sort: $\underline{\min}$, $\underline{\max}$
- formulae:
  (1) if t_1, t_2, t_3 are terms of the second sort, then the following are (atomic) formulae: $+(t_1, t_2, t_3)$, $\times(t_1, t_2, t_3)$
  (2) if $\varphi(x, i)$ is a formula, then also $\exists i \varphi(x, i)$ (binding the second-sort variable i)
  (3) if Q is a quantifier prefix quantifying the first-sort variables $\overline{x}$ and the second-sort variables $\overline{i}$, $\varphi(\overline{x}, \overline{i})$ is a quantifier-free formula and $\overline{j}$ a tuple of second-sort variables, then the following is a formula: $\#_{Q\varphi}(\overline{j})$

The semantics is clear except for $\#_{Q\varphi}(\overline{j})$. Let $\mathcal{A}$ be an input structure and $\overline{j}_0$ an assignment for $\overline{j}$. Then

$$\mathcal{A} \vDash \#_{Q\varphi}(\overline{j}_0) \quad \Longleftrightarrow_{\mathrm{def}} \quad \text{the } \mathrm{val}(\overline{j}_0)\text{-th bit of } \#\mathrm{Win}(Q\varphi, \mathcal{A}) \text{ is } 1$$

Here, $\mathrm{val}(\overline{j}_0)$ denotes the numeric value of the vector $\overline{j}_0$ under an appropriate encoding of the natural numbers as tuples of elements from the second sort.

The types (1) and (2) of formulae in our definition are the same as in Definition 8.1 in [12, p. 142]. Additionally, our definition allows new formulae $\#_{Q\varphi}(\overline{j})$. These allow us to talk about the number of winning strategies for subformula φ. Note that these numbers can be exponentially large, hence polynomially long in binary representation; therefore we can only talk about them using some form of BIT predicate. Formulas of type (3) are exactly this: a BIT predicate applied to a number of winning strategies.

Our logic FOCW[Arb] thus gives FO with access to number of winning strategies, i.e., in FOCW[Arb] we can count in an exponential range. Libkin's logic

$\mathrm{FO(Cnt)}_{\mathsf{All}}$ can count in the range of input positions, i.e., in a linear range. Nevertheless we will obtain the maybe somewhat surprising result that both logics are equally expressive on finite structures: both correspond to the circuit class TC^0.

Theorem 21. *On ordered structures,*

$$\mathrm{TC}^0 = \mathrm{FOCW}[\mathrm{Arb}] = \mathrm{AC}^{0^{\#\mathrm{AC}^0}}.$$

Proof. A central ingredient of the proof is the known equality $\mathrm{TC}^0 = \mathrm{PAC}^0$ from [1]. Here, PAC^0 is defined to be the class of languages L for which there exist functions $f, h \in \#\mathrm{AC}^0$ such that for all x, $x \in L$ iff $f(x) > h(x)$.

The proof then consists of establishing the inclusions

$$\mathrm{TC}^0 \subseteq \mathrm{PAC}^0 \subseteq \mathrm{FOCW}[\mathrm{Arb}] \subseteq \mathrm{AC}^{0^{\#\mathrm{AC}^0}} \subseteq \mathrm{TC}^0,$$

and will be given in the full paper. $\square$

Remark 22. In the full paper we will show that this result also holds in the uniform world. A central ingredient in the proof is that division, and thus iterated multiplication, can be done in uniform TC^0 due to [9].

6 Conclusion

Arithmetic classes are of current focal interest in computational complexity, but no model-theoretic characterization for any of these was known so far. We addressed the maybe most basic arithmetic class $\#\mathrm{AC}^0$ and gave such a characterization, and, based on this, a new characterization of the (Boolean) class TC^0.

This immediately leads to a number of open problems:

- We mentioned the logical characterization of #P in terms of counting assignments to free relations. We here count assignments to free function variables. Hence both characterizations are of a similar spirit. Can this be made more precise? Can our class #Win-FO be placed somewhere in the hierarchy of classes from [13]?
- Can larger arithmetic classes be defined in similar ways? The next natural candidate might be $\#\mathrm{NC}^1$ which corresponds to counting paths in so called non-uniform finite automata [3]. Maybe this will lead to a descriptive complexity characterization.
- Still the most important open problem in the area of circuit complexity is the question if $\mathrm{TC}^0 = \mathrm{NC}^1$. While we cannot come up with a solution to this, it would be interesting to reformulate the question in purely logical terms, maybe making use of our (or some other) logical characterization of TC^0.

Acknowledgements. We are grateful to Lauri Hella (Tampere) and Juha Kontinen (Helsinki) for helpful discussion, leading in particular to Definition 20. We also thank the anonymous referees for helpful comments.

References

1. Agrawal, M., Allender, E., Datta, S.: On TC^0, AC^0, and arithmetic circuits. J. Comput. Syst. Sci. **60**(2), 395–421 (2000)
2. Barrington, D.A.M., Immerman, N., Straubing, H.: On uniformity within NC^1. J. Comput. Syst. Sci. **41**, 274–306 (1990)
3. Caussinus, H., McKenzie, P., Thérien, D., Vollmer, H.: Nondeterministic NC^1 computation. J. Comput. Syst. Sci. **57**, 200–212 (1998)
4. Dawar, A.: The nature and power of fixed-point logic with counting. ACM SIGLOG News **2**(1), 8–21 (2015)
5. Ebbinghaus, H., Flum, J., Thomas, W.: Mathematical Logic. Undergraduate Texts in Mathematics. Springer, New York (1994)
6. Fagin, R.: Generalized first-order spectra and polynomial-time recognizable sets. In: Karp, R.M. (ed.) Complexity of Computation, vol. 7, pp. 43–73. SIAM-AMS Proceedings (1974)
7. Grädel, E.: Model-checking games for logics of imperfect information. Theoret. Comput. Sci. **493**, 2–14 (2013)
8. Gurevich, Y., Lewis, H.: A logic for constant-depth circuits. Inf. Control **61**, 65–74 (1984)
9. Hesse, W.: Division is in uniform TC^0. In: Orejas, F., Spirakis, P.G., Leeuwen, J. (eds.) ICALP 2001. LNCS, vol. 2076, pp. 104–114. Springer, Heidelberg (2001)
10. Immerman, N.: Languages that capture complexity classes. SIAM J. Comput. **16**, 760–778 (1987)
11. Immerman, N.: Descriptive Complexity. Graduate Texts in Computer Science. Springer, New York (1999)
12. Libkin, L.: Elements of Finite Model Theory. Springer, New York (2012)
13. Saluja, S., Subrahmanyam, K.V., Thakur, M.N.: Descriptive complexity of #P functions. J. Comput. Syst. Sci. **50**(3), 493–505 (1995)
14. Vollmer, H.: Introduction to Circuit Complexity - A Uniform Approach. Texts in Theoretical Computer Science. An EATCS Series. Springer, New York (1999)

True Concurrency of Deep Inference Proofs

Ozan Kahramanoğulları[1,2]

[1] Department of Mathematics, University of Trento, Trento, Italy
[2] The Micrososft Research - University of Trento Centre for Computational and Systems Biology, Rovereto, Italy

Abstract. We give an event structures based true-concurrency characterization of deep inference proofs. The method is general to all deep inference systems that can be expressed as term rewriting systems. This delivers three consequences in a spectrum from theoretical to practical: the event structure characterization (*i*) provides a qualification of proof identity akin to proof nets for multiplicative linear logic and to atomic flows for classical logic; (*ii*) provides a concurrency theoretic interpretation for applications in logic programming; (*iii*) reduces the length of the proofs, and thereby extends the margin of proof search applications.

1 Introduction

Deep inference [4] proofs are sequences of inference rule instances, which are essentially term rewrites of a rewriting system. The sequential construction of deep inference derivations imposes a total order structure that is beneficial in simplifying certain aspects of the proof theoretical analysis, for example, in an inductive argument for proving cut-elimination. However, such a bureaucratic view of derivations [6] also veils other aspects of proofs, and this has important implications. For example, at the more theoretical end of the spectrum, the causal dependence of the deduction components becomes hidden by the total order structure, and as a result of this, the derivations that are identical in terms of their deductive essence become presented as distinct syntactic objects.

To see this on an example, consider the following three proofs of the same multiplicative linear logic formula. While the first two proofs are identical with respect to their inference steps, they differ in the order of these steps, and thus they are depicted as syntactically distinct objects. On the other hand, the third proof is shorter, although it results in the same proof net [2] as the first two proofs. Moreover, from a computational point of view, the sequentiality of some of the rule instances in the first two proofs is redundant as the sequentiality does not imply a causal dependence; for example, the instances of the rule $\mathsf{ai}\!\downarrow$ are causally independent from each other.

J. Väänänen et al. (Eds.): WoLLIC 2016, LNCS 9803, pp. 249–264, 2016.
DOI: 10.1007/978-3-662-52921-8_16

$$\mathsf{s}\,\dfrac{\mathsf{ai}\!\downarrow\dfrac{\mathsf{u}_2\!\downarrow\dfrac{\mathsf{ai}\!\downarrow\dfrac{1}{[a \parr \bar{a}]}}{([a \parr \bar{a}] \otimes 1)}}{([a \parr \bar{a}] \otimes [b \parr \bar{b}])}}{[a \parr (\bar{a} \otimes [b \parr \bar{b}])]} \qquad \mathsf{ai}\!\downarrow\dfrac{\mathsf{s}\,\dfrac{\mathsf{u}_2\!\downarrow\dfrac{\mathsf{ai}\!\downarrow\dfrac{1}{[a \parr \bar{a}]}}{([a \parr \bar{a}] \otimes 1)}}{[a \parr (\bar{a} \otimes 1)]}}{[a \parr (\bar{a} \otimes [b \parr \bar{b}])]} \qquad \mathsf{ai}\!\downarrow\dfrac{\mathsf{u}_2\!\downarrow\dfrac{\mathsf{ai}\!\downarrow\dfrac{1}{[a \parr \bar{a}]}}{[a \parr (\bar{a} \otimes 1)]}}{[a \parr (\bar{a} \otimes [b \parr \bar{b}])]}$$

In this respect, the lack of a formal mechanism that can identify the independence and causality in the derivations poses a problem. This problem was previously addressed by generalizing the deep inference formalism syntactically to partially capture some of the concurrency in derivations [6]. Here, we present an alternative approach via a labelled event structure (LES) characterization that provides a true-concurrency interpretation of deep inference derivations.

Event structures is a model of concurrency [14,16], where the concurrent events are described with a partial order relation that formalizes their causal dependency, and nondeterminism is captured by a conflict relation, which is a symmetric irreflexive relation of events. In a proof search perspective, this corresponds to inference rules that are applicable in the same state, but are in conflict with each other in the sense that application of one of them instead of the other results in a different state space ahead. Figure 1 depicts in the middle the event structure for the example derivations above. There, the conflict-free structure on the left characterizes the two proofs on the left as well as two others, whereas the one on the right describes the single proof on the right and no other.

In the following, by relaxing the total order in deep inference derivations at incremental steps and by generalizing the approach in [7], we associate each formula an event structure, and show the correspondence between their conflict-free sub-structures and deep inference derivations. For the presentation, we use the multiplicative linear logic system MLS, which is the simplest meaningful deep inference system. However, the methods should generalize to all the deep inference systems that can be expressed as term rewriting systems.

Our results provide a qualification of identity of proofs with respect to event structures, where all the derivations that only differ with respect to permutations of their inference rules are identified by a unique structure. Moreover, the event structure interpretation of derivations makes it possible to consider appli-

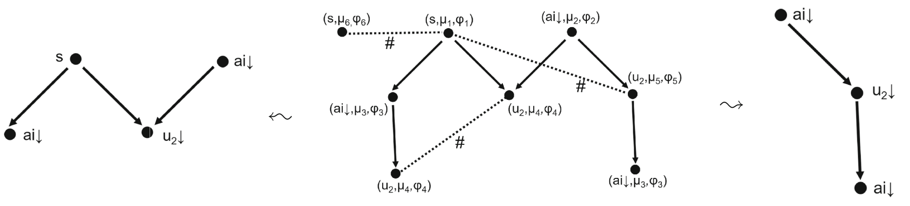


Fig. 1. The labelled event structure $\mathsf{LES}[\![[a \parr (\bar{a} \otimes [b \parr \bar{b}])]]\!]$. The symbol # denotes the conflict relation and the events are abbreviated by their actions. The conflict-free structures on the left and right characterize the example derivations in the introduction.

cations in logic programming that can exploit the true-concurrent nature of the derivations. This has also implications in proof search as a controlled use of concurrency becomes instrumental in reducing the length of the proofs, and thereby extends the margin of proof search applications.

2 Deep Inference

We use the term rewriting notation of deep inference systems modulo equational theories [10]. Other common representations for deep inference systems and derivations can be trivially obtained from the term rewriting notation without any loss of information.

Formulae (or structures) are defined in the usual way. For example, the multiplicative linear logic *formulae* P, Q, R, ... are generated by

$$R ::= a \mid \bar{a} \mid 1 \mid \bot \mid [R \parr R] \mid (R \otimes R),$$

where a stands for any atom; negation is defined on the atoms as a (non-identical) involution $\bar{\cdot}$, thus dual atom occurences, as a and $\bar{a}$, can appear in the formulae. 1 and $\bot$ are the units *one* and *bottom*, which are special atoms. Different kind of brackets are used to enhance readability, and they can be ignored.

Formulae are considered to be equivalent modulo a congruence relation. Within the term rewriting setting, term rewriting rules are applied modulo this relation. For multiplicative linear logic, we use the smallest congruence relation induced by the equational system consisting of the equations for *associativity* and *commutativity* for multiplicative disjunction and multiplicative conjunction.

Remark 1. We define negation only on atoms. This is not a limitation because of De Morgan laws. In deep inference systems in the literature, often the congruence relation includes equalities for the units of the logic. Here, we carry these equalities to the inference system to make their role in deduction more explicit.

Example 1. With respect to the congruence relation on the formulae, we have $[(\bar{b} \otimes (\bar{a} \otimes \bar{c})) \parr [b \parr [a \parr c]]] \approx [[b \parr ((\bar{a} \otimes \bar{c}) \otimes \bar{b})] \parr [c \parr a]]$ and we can denote both formulae with $[(\bar{a} \otimes \bar{b} \otimes \bar{c}) \parr a \parr b \parr c]$.

Inference rules are rewriting rules. We define system MLS for multiplicative linear logic as the term rewiting system below, where r, t, u are generic terms that can match any formula, and x is a special term that can only match atoms.

$$\begin{array}{llll} \mathsf{s} : [(r \otimes t) \parr u] \rightarrow ([r \parr u] \otimes t) & & \mathsf{u}_1{\downarrow} : [\bot \parr r] \rightarrow r \\ \mathsf{ai}{\downarrow} : \quad [x \parr \bar{x}] \quad \rightarrow 1 & & \mathsf{u}_2{\downarrow} : (1 \otimes r) \rightarrow r \end{array}$$

A rule instance of the form $R \overset{(\rho,\mu,\phi)}{\rightarrow} T$ is defined by a rule ρ, a function μ that uniquely indicates the position of the redex, and a function ϕ that assigns a substitution σ. The rule instance is then given by an application of a rule ρ at the redex uniquely identified by μ with a substitution σ such that σ provides a matching for the redex subterm R' of R with the left-hand side of the rule. That is, with $\rho : l \to r$ and $\mu(R) = R'$ and $R' = l\sigma$, we get the contractum $\mu(T) = r\sigma$. We call the triple (ρ, μ, ϕ) an *action*. We denote actions with $\mathsf{a}, \mathsf{b}, \mathsf{c}, \ldots$

A *derivation* Δ is a formula or a finite chain of instances of rule instances. The left-most structure in a derivation is the *premise*, and the right-most formula is the *conclusion*. A derivation Δ whose premise is T, conclusion is R, and inference rules are in $\mathscr{S}$ is written as $R \xrightarrow{\Delta} T$. A MLS *proof* Π is a derivation whose premise is the unit $\mathsf{1}$. A derivation can also be written as the pair of the conclusion and the sequence of actions of the derivations, e.g., $(R, \langle \mathsf{a_1}; \ldots; \mathsf{a_k} \rangle)$.

Example 2. Consider the proof of the formula $R = [a \parr (\bar{a} \otimes [b \parr \bar{b}])]$ below, where we denote the redexes with shading.

$$\begin{array}{lll} [a \parr (\bar{a} \otimes \boxed{[b \parr \bar{b}]})] & \to [a \parr \boxed{(\bar{a} \otimes \mathsf{1})}] & (\mathsf{ai}{\downarrow}, \mu_1, \phi_1),\ \sigma_1 = \{x \mapsto b\} \\ & \to \boxed{[a \parr \bar{a}]} & (\mathsf{u_2}{\downarrow}, \mu_2, \phi_2),\ \sigma_2 = \{r \mapsto \bar{a}\} \\ & \to \mathsf{1} & (\mathsf{ai}{\downarrow}, \mu_3, \phi_3),\ \sigma_3 = \{x \mapsto \bar{a}\} \end{array}$$

We then write this derivation as $(\ R,\ \langle (\mathsf{ai}{\downarrow}, \mu_1, \phi_1);\ (\mathsf{u_2}{\downarrow}, \mu_2, \phi_2);\ (\mathsf{ai}{\downarrow}, \mu_3, \phi_3) \rangle\ )$.

Below, we associate to every formula a *labelled event structure*. Events correspond to instances of inference rules. In a LES events are partially ordered and there is a conflict relation amongst the events. The partial order relation provides a representation of independence and causality between these events, which can be, for example, due to resource production and consumption relationships, or modifications of structures as it is the case below. The conflict relation here represents the nondeterminism in the system. Events that are not in conflict can take place in a way, which respects the order determined by the partial order. Labelled event structures that we formally define below thus provide a computational model for logical expressions and their derivations. This results in a characterization of concurrency, given by the partial order and conflict relations that deliver the independence and causality relationships in the inference steps.

Definition 1. *A* labelled event structure *is a structure* $(E, \leq, \#, \mathcal{L}, \ell)$, *where*

(i) E *is a set of events;*
(ii) $\leq \subseteq E^2$ *is a partial order such that for every* $\mathsf{e} \in E$ *the set* $\{\mathsf{e}' \in E \mid \mathsf{e}' \leq \mathsf{e}\}$ *is finite;*
(iii) the conflict relation $\# \subseteq E^2$ *is a symmetric and irreflexive relation such that if* $\mathsf{e} \# \mathsf{e}'$ *and* $\mathsf{e}' \leq \mathsf{e}''$, *then* $\mathsf{e} \# \mathsf{e}''$, *for every* $\mathsf{e}, \mathsf{e}', \mathsf{e}'' \in E$;
(iv) $\mathcal{L}$ *is a set of labels;*
(v) $\ell : E \to \mathcal{L}$ *is a labeling function.*

Example 3. Figure 1 depicts the LES for the formula $[a \parr (\bar{a} \otimes [b \parr \bar{b}])]$.

3 Event Structures of Proofs

We associate to every formula an event structure that characterizes the independence and causality of the rule instances. We first associate to each formula a labelled transition system. We define labelled transition systems in the usual way, and denote states with s and transitions with t. Given a transition system, a path with length k is a sequence $\langle t_1; t_2; \ldots; t_k \rangle$ of transitions such that $t_i = s_{i-1} \overset{a}{\twoheadrightarrow} s_i$ or equivalently $t_i = (s_{i-1}, s_i, a)$ for $i = 1, 2, \ldots$ and the initial state $s_I = s_0$. We distinguish the transitions from rewrites with the notation $\twoheadrightarrow$; rewrites and derivations are denoted with $\rightarrow$ and $\longrightarrow$.

Definition 2. *Given a formula R and a set $\mathcal{A}$ of actions of system* MLS *as defined above,* $\mathsf{TS}[\![R]\!] = (\mathcal{S}, s_I, \mathcal{A}, \twoheadrightarrow)$ *is the reachable transition system with the set of states $\mathcal{S}$ such that $(\Delta, \Delta', \mathsf{a}) \in \twoheadrightarrow$ where $\Delta, \Delta' \in \mathcal{S}$ iff*

- *$s_I = R \in \mathcal{S}$ is the initial state;*
- *for some structure T, derivation Δ has the shape $R \overset{\Delta}{\longrightarrow} T$;*
- *for some structure Q, there exists $T \overset{\mathsf{a}}{\rightarrow} Q$ with $\mathsf{a} = (\rho, \sigma, \mu) \in \mathcal{A}$;*
- *Δ' is the derivation $R \overset{\Delta}{\longrightarrow} T \overset{\mathsf{a}}{\rightarrow} Q$.*

We then write $\Delta \overset{\mathsf{a}}{\twoheadrightarrow} \Delta'$.

For any formula R, $\mathsf{TS}[\![R]\!]$ is acyclic, because transitions result in bigger derivations. For a formula R, $\mathsf{TS}[\![R]\!]$ overlaps with the state space of the derivations with R in the conclusion; each state of $\mathsf{TS}[\![R]\!]$ is a reachable derivation.

As a first step towards observing the independence and the causality in the derivations, we consider two derivations equivalent if they have the same premise and conclusion. The following definition serves this purpose.

Definition 3. *Let $\mathcal{D}$ be the set of derivations, and R and T be formulae. $\approx \subset \mathcal{D}^2$ is the least equivalence relation such that $\Delta \approx \Delta'$ iff we have that*

$$R \overset{\Delta}{\longrightarrow} T \quad \textit{and} \quad R \overset{\Delta'}{\longrightarrow} T \ .$$

$[\Delta]_{\approx}$ denotes the equivalence class of the derivation Δ under $\approx$. The set $\mathcal{D}/\!\approx$, the set of equivalence classes of derivations under $\approx$, is called the set of abstract derivations*. The elements of $\mathcal{D}/\!\approx$ are denoted by δ.*

Example 4. Consider the two derivations Δ and Δ' below with $\Delta \approx \Delta'$.

$$\begin{array}{llll} \Delta : & [a \parr (\bar{a} \otimes [b \parr \bar{b}])] \overset{\mathsf{ai}\downarrow}{\rightarrow} & [a \parr (\bar{a} \otimes 1)] & \overset{\mathsf{s}}{\rightarrow} ([a \parr \bar{a}] \otimes 1) \\ \Delta' : & [a \parr (\bar{a} \otimes [b \parr \bar{b}])] \overset{\mathsf{s}}{\rightarrow} & ([a \parr \bar{a}] \otimes [b \parr \bar{b}]) & \overset{\mathsf{ai}\downarrow}{\rightarrow} ([a \parr \bar{a}] \otimes 1) \end{array}$$

Proposition 1. *For any two states Δ and Δ' of $\mathsf{TS}[\![R]\!]$, if $\Delta \approx \Delta'$ then for all $\Delta \overset{\mathsf{a}}{\twoheadrightarrow} \Delta''$ in $\mathsf{TS}[\![R]\!]$, there exists a transition $\Delta' \overset{\mathsf{a}}{\twoheadrightarrow} \Delta'''$ in $\mathsf{TS}[\![R]\!]$ with $\Delta'' \approx \Delta'''$.*

Proof. Because Δ'' and Δ''' have the same premises, same inference rules can be applied to the premises of these two derivations.

We now redefine transition systems that are associated with the formulae such that they respect the equivalence of derivations induced by the relation $\approx$.

Definition 4. *Given a formula R and a $\mathsf{TS}[\![R]\!] = (\mathcal{S}, s_I, \mathscr{A}, \twoheadrightarrow)$, let $\mathsf{TS}_{\approx}[\![R]\!] = (\mathcal{S}_{\approx}\,, \, s_{I\approx}, \, \mathscr{A}, \, \twoheadrightarrow_{\approx})$ be the transition system such that (i) $s_{I\approx} = R$; (ii) $\mathcal{S}_{\approx} = \mathcal{S}/\approx$; (iii) $[\Delta]_{\approx} \stackrel{\mathsf{a}}{\twoheadrightarrow} [\Delta']_{\approx}$ iff $\Delta \stackrel{\mathsf{a}}{\twoheadrightarrow} \Delta'$ where $\mathsf{a} \in \mathscr{A}$.*

As a result of this definition, transition systems are not trees anymore, but they are graphs. For the case of system MLS, because each inference rule results in an incremental step in a terminating computation, the $\mathsf{TS}_{\approx}[\![R]\!]$ graphs are acyclic. However, $\mathsf{TS}_{\approx}[\![R]\!]$ graphs can in general be cyclic, for example, if the deductive system involves a cut rule or a contraction rule.

Definition 5. *Let R be a formula and $\tau = \langle t_1; \ldots; t_h \rangle$ be a finite path in $\mathsf{TS}_{\approx}[\![R]\!]$. τ is an* abstract path yielding δ_h*, if, for all $1 \leq i \leq h$, $t_i = (\delta_{i-1}, \delta_i, a_i)$.*

The intuition behind abstract paths can be better understood from the point of view of their transitions: given $\delta \stackrel{\mathsf{a}}{\twoheadrightarrow} \delta'$, we have that δ is the equivalence class of derivations with a premise T and conclusion R, and we have $T \stackrel{\mathsf{a}}{\rightarrow} Q$ such that δ' is the equivalence class of derivations with a premise Q and conclusion R. Because $\mathsf{TS}[\![R]\!]$ is by definition reachable, $\mathsf{TS}_{\approx}[\![R]\!]$ is also reachable.

Definition 6. *Given a formula R and $\mathsf{TS}_{\approx}[\![R]\!]$, let the relation $\diamondsuit \subseteq \mathscr{A}^2 \times \mathcal{S}_{\approx}^4$ be such that $(\mathsf{a}, \mathsf{a}', \delta, \delta', \delta'', \delta''') \in \diamondsuit$ iff $(\delta, \delta', \mathsf{a})$, $(\delta, \delta'', \mathsf{a}')$, $(\delta', \delta''', \mathsf{a}')$, and $(\delta'', \delta''', \mathsf{a})$ are in $\mathsf{TS}_{\approx}[\![R]\!]$. We call $\diamondsuit$ the* diamond relation *of $\mathsf{TS}_{\approx}[\![R]\!]$.*

The diamond relation corresponds to the permutability of the inference rules over each other in the standard deep inference notation, e.g., [15]. With this definition, we do not distinguish anymore the derivations that differ only in the permutations of their inference rules. Below, we thus propagate the diamond relation to paths, and this way establish an equivalence relation on the paths.

Example 5. For an MLS derivation with formula R in the conclusion, let δ, δ', δ'' and δ''' be equivalence classes of derivations such that we have the graph in Fig. 2 with $\mathsf{a} = (\mathsf{ai}{\downarrow}, \mu_1, \phi_1)$ and $\mathsf{a}' = (\mathsf{s}, \mu_2, \phi_2)$. Then we have $(\mathsf{a}, \mathsf{a}', \delta, \delta', \delta'', \delta''')$, where ϕ_1 maps σ_1 to $\{x \mapsto b\}$, and ϕ_2 maps σ_2 to $\{r \mapsto \bar{a}, t \mapsto \mathsf{1}, u \mapsto a\}$ and $\{r \mapsto \bar{a}, t \mapsto [b \parr \bar{b}], u \mapsto a\}$ on the left and on the right, respectively.

With the following definition we propagate the diamond relation from derivations to paths, and this way prepare the grounds for defining a transition system whose states are paths rather than derivations.

Definition 7. *Given $\mathsf{TS}_{\approx}[\![R]\!] = (\mathcal{S}_{\approx}, R, \mathscr{A}, \rightarrow_{\approx})$ and its diamond relation $\diamondsuit$, the relation $\backsimeq$ is the least equivalence relation such that, for any two paths*

$$\tau_1 = \langle\, t\,;\, (\delta, \delta', \mathsf{a})\,;\, (\delta', \delta''', \mathsf{a}')\,;\, t' \rangle \quad \textit{and} \quad \tau_2 = \langle t\,;\, (\delta, \delta'', \mathsf{a}')\,;\, (\delta'', \delta''', \mathsf{a})\,;\, t' \rangle\,,$$

if $(\mathsf{a}, \mathsf{a}', \delta, \delta', \delta'', \delta''') \in \diamondsuit$, then $\tau_1 \backsimeq \tau_2$.

$$
\begin{tikzcd}[column sep=tiny]
& \delta \;=\; [\; R \longrightarrow [a \parr (\bar{a} \otimes [b \parr \bar{b}])]\;]_{\approx} \arrow[dl, "\mathsf{a}"'] \arrow[dr, "\mathsf{a}'"] & \\
\delta' \;=\; [\; R \longrightarrow [a \parr (\bar{a} \otimes 1)]\;]_{\approx} \arrow[dr, "\mathsf{a}'"'] & & \delta'' \;=\; [\; R \longrightarrow ([a \parr \bar{a}] \otimes [b \parr \bar{b}])\;]_{\approx} \arrow[dl, "\mathsf{a}"] \\
& \delta''' \;=\; [\; R \longrightarrow ([a \parr \bar{a}] \otimes 1)\;]_{\approx} &
\end{tikzcd}
$$

Fig. 2. The diamond relation of two paths induced by the formula $[a \parr (\bar{a} \otimes [b \parr \bar{b}])]$ in the transition system $\mathsf{TS}_{\approx}\llbracket R \rrbracket$ of an MLS formula R.

Proposition 2. *Given finite paths τ_1 and τ_2 in $\mathsf{TS}_{\approx}\llbracket R \rrbracket$, if $\tau_1 \backsimeq \tau_2$, then they both yield the same state in $\mathsf{TS}_{\approx}\llbracket R \rrbracket$.*

The labelled event structure for a formula R is obtained from a transition system defined on paths such that all paths reaching a certain state belong to the same equivalence class induced by $\backsimeq$. Below we define this transition system based on the $\mathsf{TS}_{\approx}\llbracket R \rrbracket$ and the equivalence relation $\backsimeq$ on its paths.

Definition 8. *Given an MLS formula R and $\mathsf{TS}_{\approx}\llbracket R \rrbracket = (\mathcal{S}_{\approx}, s_{I\approx}, \mathscr{A}, \rightarrow_{\approx})$, $\mathsf{TS}_{\simeq}\llbracket R \rrbracket = (\mathcal{S}_{\simeq}, s_{I\simeq}, \mathscr{A}, \rightarrow_{\simeq})$ is the transition system such that*

- *(i) $\mathcal{S}_{\simeq} = \mathcal{T}/\backsimeq$, where $\mathcal{T}$ is the set of finite paths in $\mathsf{TS}_{\approx}\llbracket R \rrbracket$ and $\simeq$ is the equivalence relation on its paths induced by the diamond relation $\Diamond$ of $\mathsf{TS}_{\approx}\llbracket R \rrbracket$. Elements of $\mathcal{S}_{\simeq}$ are denoted by π ;*
- *(ii) $s_{I\simeq} = [\circ]_{\simeq}$;*
- *(iii) $[\tau]_{\simeq} \xrightarrow{\mathsf{a}}_{\simeq} [\tau']_{\simeq}$ iff $\tau' \simeq \langle\, \tau\,;\, (\delta, \delta', \mathsf{a})\,\rangle$ where $(\delta, \delta', \mathsf{a}) \in \rightarrow_{\approx}$.*

Proposition 3. *For every formula R, $\mathsf{TS}_{\simeq}\llbracket R \rrbracket$ is reachable and acyclic.*

Proof. $\mathsf{TS}_{\simeq}\llbracket R \rrbracket$ is obtained from $\mathsf{TS}_{\approx}\llbracket R \rrbracket$ which is reachable. Because each transition transforms an abstract path to a syntactically bigger abstract path, $\mathsf{TS}_{\simeq}\llbracket R \rrbracket$ is acyclic.

Example 6. The transition system $\mathsf{TS}_{\simeq}$ associated to the formula $[a \parr (\bar{a} \otimes [b \parr \bar{b}])]$ is depicted in Fig. 3.

A LES of a formula R is obtained from $\mathsf{TS}_{\simeq}\llbracket R \rrbracket$ by extracting the transitions denoting the same events. To obtain this information, we propagate the diamond relation of the transition systems $\mathsf{TS}_{\approx}$ to the transition systems $\mathsf{TS}_{\simeq}$, that is, from the equivalence classes of derivations to the equivalence classes of paths.

Definition 9. *Given a formula R and the diamond relation $\Diamond$ of $\mathsf{TS}_{\approx}\llbracket R \rrbracket$, we define $\Diamond_{\simeq} \subset \mathscr{A}^2 \times \mathcal{S}_{\simeq}^4$ for $\mathsf{TS}_{\simeq}\llbracket R \rrbracket$ as the relation such that, for some abstract paths $\tau, \tau', \tau'', \tau'''$, we have $(\mathsf{a}, \mathsf{a}', [\tau]_{\simeq}, [\tau']_{\simeq}, [\tau'']_{\simeq}, [\tau''']_{\simeq}) \in \Diamond_{\simeq}$ iff*

$$
\tau' \simeq \langle\, \tau\,; (\delta, \delta', \mathsf{a})\,\rangle\,, \qquad \tau'' \simeq \langle\, \tau\,; (\delta, \delta'', \mathsf{a}')\,\rangle\,,
$$

$$
\tau''' \simeq \langle\, \tau'\,; (\delta', \delta''', \mathsf{a}')\,\rangle \qquad \tau''' \simeq \langle\, \tau''\,; (\delta'', \delta''', \mathsf{a})\,\rangle
$$

and $(\mathsf{a}, \mathsf{a}', \delta, \delta', \delta'', \delta''') \in \Diamond$ for some states $\delta, \delta', \delta''$ and δ''' of $\mathsf{TS}_{\approx}\llbracket R \rrbracket$.

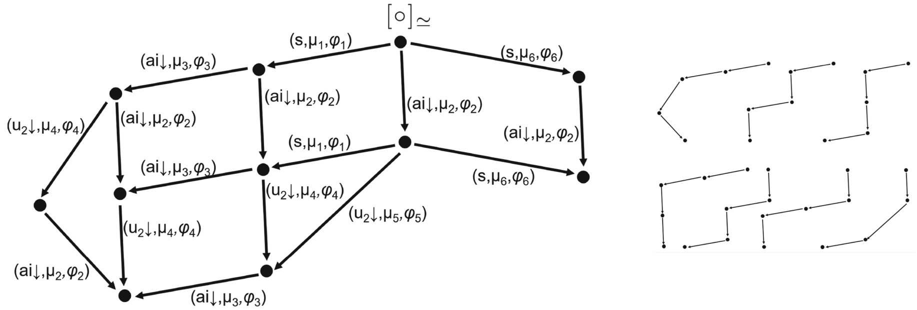


Fig. 3. The transition system $\mathsf{TS}_{\simeq}[\![[a \bindnasrepma (\bar{a} \otimes [b \bindnasrepma \bar{b}])]]\!]$ and the pathways that result in seven distinct MLS proofs.

Definition 10. *Given* $\mathsf{TS}_{\simeq}[\![R]\!] = (\mathcal{S}_{\simeq}, R, \mathscr{A}, \rightarrow_{\simeq})$ *and its diamond relation* $\diamondsuit_{\simeq}$ *, the relation* $\sim$ *is the least equivalence relation on* $t, t' \in \rightarrow_{\simeq}$ *such that*

$$t \sim t' \text{ iff } t = (\pi, \pi', \mathsf{a}),\ t' = (\pi'', \pi''', \mathsf{a})$$

and there exists $\mathsf{a}' \in \mathscr{A}$ *such that* $(\mathsf{a}, \mathsf{a}', \pi, \pi', \pi'', \pi''') \in \diamondsuit_{\simeq}$ *.*

Intuitively, two transitions are in $\sim$ if they represent the same event.

Example 7. Let τ be an abstract path that leads to a derivation with the formula $[a \bindnasrepma (\bar{a} \otimes [b \bindnasrepma \bar{b}])]$ at the premise. Then we have

$$\tau' \simeq \langle\, \tau\,; ([\,R \longrightarrow [a \bindnasrepma (\bar{a} \otimes [b \bindnasrepma \bar{b}])]\,]_{\approx}, [\,R \longrightarrow [a \bindnasrepma (\bar{a} \otimes 1)]\,]_{\approx}, (\mathsf{ai}{\downarrow}, \mu_2, \phi_2))\,\rangle\ ,$$
$$\tau'' \simeq \langle \tau; ([R \longrightarrow [a \bindnasrepma (\bar{a} \otimes [b \bindnasrepma \bar{b}])]]_{\approx}, [R \longrightarrow ([a \bindnasrepma \bar{a}] \otimes [b \bindnasrepma \bar{b}])]_{\approx}, (\mathsf{s}, \mu_1, \phi_1))\rangle,$$
$$\tau''' \simeq \langle \tau'; ([\,R \longrightarrow [a \bindnasrepma (\bar{a} \otimes 1)]]_{\approx}, [R \longrightarrow ([a \bindnasrepma \bar{a}] \otimes 1)]_{\approx}, (\mathsf{s}, \mu_1, \phi_1))\rangle,$$
$$\tau''' \simeq \langle \tau''; ([\,R \longrightarrow ([a \bindnasrepma \bar{a}] \otimes [b \bindnasrepma \bar{b}])]_{\approx}, [R \longrightarrow ([a \bindnasrepma \bar{a}] \otimes 1)]_{\approx}, (\mathsf{ai}{\downarrow}, \mu_2, \phi_2))\rangle.$$

We then have $(\mathsf{a}, \mathsf{a}', [\tau]_{\simeq}, [\tau']_{\simeq}, [\tau'']_{\simeq}, [\tau''']_{\simeq}) \in \diamondsuit_{\simeq}$. Then we have $t \sim t'$ for

$$t = ([\,R \longrightarrow [a \bindnasrepma (\bar{a} \otimes [b \bindnasrepma \bar{b}])]\,]_{\approx},\ [\,R \longrightarrow [a \bindnasrepma (\bar{a} \otimes 1)]\,]_{\approx},\ (\mathsf{ai}{\downarrow}, \mu_2, \phi_2)),\ \text{and}$$
$$t' = ([\,R \longrightarrow ([a \bindnasrepma \bar{a}] \otimes [b \bindnasrepma \bar{b}])\,]_{\approx},\ [\,R \longrightarrow ([a \bindnasrepma \bar{a}] \otimes 1)\,]_{\approx},\ (\mathsf{ai}{\downarrow}, \mu_2, \phi_2))\ .$$

Definition 11. *Given a formula* R *and* $\mathsf{TS}_{\simeq}[\![R]\!] = (\mathcal{S}_{\simeq}, R, \mathscr{A}, \ell)$*, let* $\mathsf{LES}[\![R]\!] = (E, \leq, \#, \mathscr{A}, \ell)$ *be the labelled event structure such that*

(i) $E = \rightarrow_{\simeq} / \sim$ *;*

(ii) $\leq$ *is the reflexive closure of* $<$*, which is defined as follows: for all* $\mathsf{e}, \mathsf{e}' \in E$*,* $\mathsf{e} < \mathsf{e}'$ *iff* $\mathsf{e} = [t]_{\sim}$ *and* $\mathsf{e}' = [t']_{\sim}$*, and for every path* τ *in* $\mathsf{TS}_{\simeq}[\![R]\!]$ *and for every* $t''' \in \rightarrow_{\simeq}$ *such that* $\langle\, \tau\,;\, t'''\,\rangle$ *is a path and* $t''' \sim t'$*, there exists* $t'' \sim t$ *such that* $\tau = \langle\, \tau'; t''; \tau''\,\rangle$ *for some* τ', τ'' *;*

(iii) $[t]_\sim \# [t']_\sim$ *iff for every path* τ *in* $\mathsf{TS}_\simeq[\![R]\!]$ *and for every* $t'', t''' \in \rightarrow_\simeq$ *such that* $t \sim t''$ *and* $t' \sim t'''$*, if* t'' *appears in* τ*, then* t''' *does not appear in* τ*;*

(iv) $\ell([(\pi, \pi', \mathsf{a})]_\sim) = \mathsf{a}$ *.*

The partial order relation of a LES provides a representation of independence and causality between different events. The events that are left unordered by $\leq$ are independent, thus their ordering in execution with respect to each other does not affect the behavior of the system. In deep inference derivations, this independence corresponds to the permutability of the instances of the inference rules: the independent rule instances can be applied to a formula in any order as their instances do not create a conflict for their mutual applicability. In contrast, the events that are ordered by the relation $\leq$ follow a chain of causality, that is, for an event e, all events $\mathsf{e}' < \mathsf{e}$, the execution of e is impossible without the prior execution of e'. For the deep inference proofs such a causal dependence is a consequence of the structural relations that are modified by the instances of the inference rules that make a rule instance necessary prior to another one.

The relation $\#$ is an irreflexive relation on events that expresses conflicting situations in execution. If $e \# e'$, then event e and e' are competing for resources, thus execution of e conflicts with the execution of e', and vice versa, which requires a choice of one over the other. In the deep inference proofs, this corresponds to the different choices in the construction of derivations due to multiple rule instances that can be applicable to a formula at each inference step.

Example 8. The LES associated to the formula $[a \parr (\bar{a} \otimes [b \parr \bar{b}])]$ is depicted in Fig. 1, where we abbreviate events with their labels.

The notions of LES result in a concurrent model of all the possible derivations of formula, which we characterize with the definitions below.

Definition 12. *Given a* LES $(E, \leq, \#, \mathscr{A}, \ell)$*, for an event* $\mathsf{e} \in E$*,* $\lfloor \mathsf{e} \rfloor$ *denotes the set* $\{\mathsf{e}' \in E \mid \mathsf{e}' < \mathsf{e}\}$ *of* causes *of event* e*.*

The causes of an event e is the set that collects those events that event e requires in order to take place. In the deep inference derivations, the causes of a rule instance at an inference step is the set of rule instances that modify the formula in such a way that makes that rule instance possible. Configurations, that we define below, collect such causally related events that are not in conflict, while preserving the information on the independence of different causes.

Definition 13. *Given a* LES $(E, \leq, \#, L, \ell)$*,* $\mathscr{C} \subseteq E$ *is a* configuration *iff*

(i) for all $\mathsf{e} \in \mathscr{C}$ *we have that* $\lfloor \mathsf{e} \rfloor \subset \mathscr{C}$*;*
(ii) for all $\mathsf{e}, \mathsf{e}' \in \mathscr{C}$*, it is not the case that* $\mathsf{e} \# \mathsf{e}'$*.*

Definition 14. *Given a* LES $(E, \leq, \#, L, \ell)$*, and one of its configurations* $\mathscr{C}$*, we say that* event e is enabled at $\mathscr{C}$ *(denoted by* $\mathscr{C} \rhd \mathsf{e}$*) if and only if*
(i) $\mathsf{e} \notin \mathscr{C}$*;* *(ii)* $\lfloor \mathsf{e} \rfloor \subseteq \mathscr{C}$*;* *(iii)* $\mathsf{e}' \# \mathsf{e}$ *implies* $\mathsf{e}' \notin \mathscr{C}$*.*

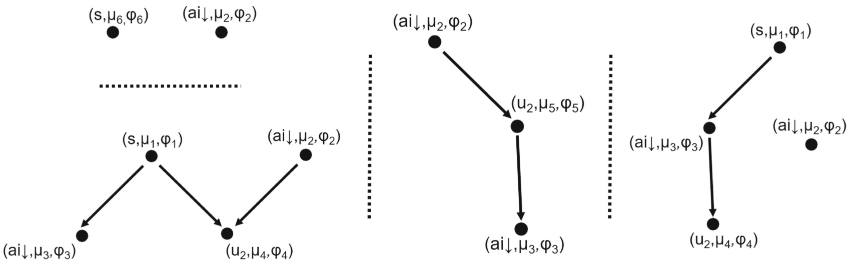


Fig. 4. Four configurations obtained from the LES depicted in Fig. 1.

Example 9. Consider the LES in Fig. 1. Let us abbreviate the events with their labels. We have that $(\mathsf{ai}{\downarrow}, \mu_2, \psi_2) \in \lfloor(\mathsf{u}_2{\downarrow}, \mu_4, \psi_4)\rfloor$, and there exists a configuration $\mathscr{C} = \{(\mathsf{s}, \mu_1, \psi_1), (\mathsf{ai}{\downarrow}, \mu_2, \psi_2)\}$ such that $\mathscr{C} \rhd (\mathsf{u}_2{\downarrow}, \mu_4, \psi_4)$.

Informally, an event e is enabled at a configuration $\mathscr{C}$ if it is not in $\mathscr{C}$, all the events on which it depends are in $\mathscr{C}$ and it does not conflict with any event in $\mathscr{C}$. Let us now define securings, which are serializations of events in configurations.

Definition 15. *Given a* LES $(E, \leq, \#, L, \ell)$ *and a finite sequence of events* $\mathsf{S} = \langle e_1; \ldots; e_h\rangle$, S *is a* securing *for* $\mathscr{C}$ *if and only if* $\mathscr{C} = \{e_1, \ldots, e_h\}$ *is a configuration and, for all* $1 \leq i \leq h$, $\{e_1, \ldots, e_{i-1}\} \rhd e_i$.

Example 10. From the LES in Fig. 1, we can read nine derivations, which can be read as the securings in $\mathsf{LES}[\![[a \bindnasrepma (\bar{a} \otimes [b \bindnasrepma \bar{b}])]]\!]$, obtained from four configurations in Fig. 4. The first two derivations do not result in a proof.

$$
\begin{aligned}
\Delta_1 &= \langle(\mathsf{s}, \mu_6, \phi_6); (\mathsf{ai}{\downarrow}, \mu_2, \phi_2)\rangle\\
\Delta_2 &= \langle(\mathsf{ai}{\downarrow}, \mu_2, \phi_2); (\mathsf{s}, \mu_6, \phi_6)\rangle\\
\Delta_3 &= \langle(\mathsf{ai}{\downarrow}, \mu_2, \phi_2); (\mathsf{u}_2{\downarrow}, \mu_5, \phi_5); (\mathsf{ai}{\downarrow}, \mu_3, \phi_3)\rangle\\
\Delta_4 &= \langle(\mathsf{s}, \mu_1, \phi_1); (\mathsf{ai}{\downarrow}, \mu_2, \phi_2); (\mathsf{ai}{\downarrow}, \mu_3, \phi_3); (\mathsf{u}_2{\downarrow}, \mu_4, \phi_4)\rangle\\
\Delta_5 &= \langle(\mathsf{s}, \mu_1, \phi_1); (\mathsf{ai}{\downarrow}, \mu_2, \phi_2); (\mathsf{u}_2{\downarrow}, \mu_4, \phi_4); (\mathsf{ai}{\downarrow}, \mu_3, \phi_3)\rangle\\
\Delta_6 &= \langle(\mathsf{s}, \mu_1, \phi_1); (\mathsf{ai}{\downarrow}, \mu_3, \phi_3); (\mathsf{u}_2{\downarrow}, \mu_4, \phi_4); (\mathsf{ai}{\downarrow}, \mu_2, \phi_2)\rangle\\
\Delta_7 &= \langle(\mathsf{s}, \mu_1, \phi_1); (\mathsf{ai}{\downarrow}, \mu_3, \phi_3); (\mathsf{ai}{\downarrow}, \mu_2, \phi_2); (\mathsf{u}_2{\downarrow}, \mu_4, \phi_4)\rangle\\
\Delta_8 &= \langle(\mathsf{ai}{\downarrow}, \mu_2, \phi_2); (\mathsf{s}, \mu_1, \phi_1); (\mathsf{u}_2{\downarrow}, \mu_4, \phi_4); (\mathsf{ai}{\downarrow}, \mu_3, \phi_3)\rangle\\
\Delta_9 &= \langle(\mathsf{ai}{\downarrow}, \mu_2, \phi_2); (\mathsf{s}, \mu_1, \phi_1); (\mathsf{ai}{\downarrow}, \mu_3, \phi_3); (\mathsf{u}_2{\downarrow}, \mu_4, \phi_4)\rangle
\end{aligned}
$$

The following results are analogous to the results on search spaces of multiset rewriting encodings in multiplicative exponential linear logic in deep inference [8], following the discussions above with respect to the ideas presented in [3, 14, 16]. They demonstrate, for any MLS formula R, the formal correspondence between the transition systems $\mathsf{TS}[\![R]\!]$ and the $\mathsf{LES}[\![R]\!]$.

Theorem 1. *Given a formula* R, $\mathsf{LES}[\![R]\!] = (E, \leq, \#, \mathscr{A}, \ell)$ *and a securing* S *in* $\mathsf{LES}[\![R]\!]$, *there is a path* $R \overset{\ell(\mathsf{S})}{\twoheadrightarrow} \Delta$ *in* $\mathsf{TS}[\![R]\!]$.

Theorem 2. *Given a formula R and a path $R \xrightarrow{\mathsf{a}_1} \Delta_1 \xrightarrow{\mathsf{a}_2} \cdots \xrightarrow{\mathsf{a}_\mathsf{h}} \Delta_h$ in $\mathsf{TS}[\![R]\!]$, there is a securing S in $\mathsf{LES}[\![R]\!] = (E, \leq, \#, \mathscr{A}, \ell)$ such that $\ell(\mathsf{S}) = \langle \mathsf{a}_1; \ldots; \mathsf{a}_\mathsf{h} \rangle$.*

For an exposure of the relationship between transition systems, labelled event structures, and other models for concurrency, we refer the reader to [14,16].

4 From Deep Inference Derivations to Configurations

A configuration in a LES of a formula can be considered as a canonical representation of a set of derivations with the same premise and conclusion, which however differ in their orders of the rule instances due to permutations. Conversely, inference rules in a derivation can be permuted to obtain other derivations that have the same premise and conclusion. Below by exploiting this observation, we introduce an algorithm for obtaining configurations from MLS derivations.

Definition 16. *We denote labels that range over actions with letters $\mathsf{a}, \mathsf{b}, \mathsf{c}, \ldots$ Let Δ be an MLS derivation such that each subformula in Δ is labeled with the special action ϵ as a subscript and each inference rule modifies these labels as described below. The function ψ on Δ is defined as follows. If Δ is a formula then $\psi(\Delta) = \emptyset$. Otherwise, if Δ is of the form $R \xrightarrow{\rho} R' \xrightarrow{\Delta'} T$ and the instance of the rule ρ with its action a is an instance of*

- s *with* $[(r \otimes t)_\mathsf{b} \parr u]_\mathsf{c} \to ([r \parr u]_\mathsf{a} \otimes t)_\mathsf{a}$ *then* $\psi(\Delta) = \{(\mathsf{b}, \mathsf{a}), (\mathsf{c}, \mathsf{a})\} \cup \psi(\Delta')$;
- $\mathsf{ai}{\downarrow}$ *with* $[x \parr \bar{x}]_\mathsf{b} \to 1_\mathsf{a}$ *then* $\psi(\Delta) = \{(\mathsf{b}, \mathsf{a})\} \cup \psi(\Delta')$;
- $\mathsf{u}_1{\downarrow}$ *with* $[\bot \parr r_\mathsf{b}]_\mathsf{c} \to r_\mathsf{b}$ *then* $\psi(\Delta) = \{(\mathsf{b}, \mathsf{a}), (\mathsf{c}, \mathsf{a})\} \cup \psi(\Delta')$;
- $\mathsf{u}_2{\downarrow}$ *with* $(1_\mathsf{b} \otimes r_\mathsf{c})_\mathsf{d} \to r_\mathsf{c}$ *then* $\psi(\Delta) = \{(\mathsf{d}, \mathsf{a}), (\mathsf{b}, \mathsf{a})\} \cup \psi(\Delta')$.

Given a derivation Δ with a formula R in the conclusion, a constraint set of Δ for R ($\mathcal{C}_{R,\Delta}$) is given with $\psi(\Delta)$.

Function ψ extracts the production and consumption relationships between rule instances, and this way provides a canonical representation of all derivations that are different from the input derivation only with respect to rule permutations. This is because each action at each inference step modifies, annihilates or produces a subformula that has been introduced by another. By keeping track of these relationships as a representation of causality, we obtain a configuration.

Example 11. We label the derivation below according to Definition 16 and apply function ψ. We obtain the set $\mathcal{C} = \mathcal{C}_4$, which provides a canonical representation of four derivations. Any of these four derivations delivers the same set $\mathcal{C}$.

$$
\begin{array}{lll}
[a \parr (\bar{a} \otimes [b \parr \bar{b}]_\epsilon)_\epsilon]_\epsilon & & \\
\quad \to ([a \parr \bar{a}]_\mathsf{a} \otimes [b \parr \bar{b}]_\epsilon)_\mathsf{a}\,, & \mathsf{a} = (\mathsf{s}, \mu_1, \phi_1), & \mathcal{C}_1 = \{(\epsilon, \mathsf{a})\} \\
\quad \to ([a \parr \bar{a}]_\mathsf{a} \otimes 1_\mathsf{b})_\mathsf{a}\,, & \mathsf{b} = (\mathsf{ai}{\downarrow}, \mu_2, \phi_2), & \mathcal{C}_2 = \mathcal{C}_1 \cup \{(\epsilon, \mathsf{b})\} \\
\quad \to [a \parr \bar{a}]_\mathsf{a}\,, & \mathsf{c} = (\mathsf{u}_2{\downarrow}, \mu_3, \phi_3), & \mathcal{C}_3 = \mathcal{C}_2 \cup \{(\mathsf{a}, \mathsf{c}), (\mathsf{b}, \mathsf{c})\} \\
\quad \to 1_\mathsf{d}\,, & \mathsf{d} = (\mathsf{ai}{\downarrow}, \mu_3, \phi_3), & \mathcal{C}_4 = \mathcal{C}_3 \cup \{(\mathsf{a}, \mathsf{d})\}
\end{array}
$$

Remark 2. There can be cases in derivations due to identical subformulae with different labels, which correspond to different configurations. For example, consider the instance below, which can result in two different constraint sets due to the choice of the redex of the $\mathsf{ai}\downarrow$ rule. This nondeterminism is due to the choice of the redex of the rule instance at the inference step.

$$([a \parr \bar{a}]_{\mathsf{b}} \otimes [a \parr \bar{a}]_{\mathsf{c}}) \begin{array}{l} \nearrow (1_{\mathsf{a}} \otimes [a \parr \bar{a}]_{\mathsf{a}}) \Rightarrow (\mathsf{b},\mathsf{a}) \in \mathcal{C} \\ \searrow ([a \parr \bar{a}]_{\mathsf{b}} \otimes 1_{\mathsf{a}}) \Rightarrow (\mathsf{c},\mathsf{a}) \in \mathcal{C} \end{array}$$

Proposition 4. *For any derivation Δ, $\psi(\Delta)$ terminates in linear time in the number of atoms in Δ.*

Proposition 5. *Let R be a multiplicative linear logic formula with a derivation Δ and $\mathcal{C}_{R,\Delta}$ be their constraint set. $\mathcal{C}_{R,\Delta}$ is irreflexive and antisymetric.*

Proof. Follows from an inspection of the steps of function ψ in Definition 16: none of the actions introduces a constraint of the form (x, x), and the inductive steps of ψ introduce a new action as the second of a pair (x, y) at each step.

Definition 17. *Let R be a multiplicative linear logic formula with a derivation Δ and $\mathcal{C}_{R,\Delta}$ be the constraint set. The* concurrent derivation *of Δ for R, and denoted with $\mathsf{Con}_{R,\Delta}$, is the transitive reflexive closure of $\mathcal{C}_{R,\Delta}$.*

Remark 3. For any constraint set $\mathcal{C}_{R,\Delta}$, $\mathsf{Con}_{R,\Delta}$ is a partial order.

Definition 18. *A linearization Lin of a concurrent derivation $\mathsf{Con}_{R,\Delta}$ is a strict total order of the actions of $\mathsf{Con}_{R,\Delta}$ such that $\mathsf{Con}_{R,\Delta} \subseteq \mathsf{Lin}$. Given a linearization Lin of $\mathsf{Con}_{R,\Delta}$, the derivation induced by Lin is the derivation with R in the conclusion, constructed by applying the actions of Lin with respect to their order.*

Theorem 3. *For a derivation $R \xrightarrow{\Delta} T$ in MLS, any derivation Δ' induced by a linearization Lin of $\mathsf{Con}_{R,\Delta}$ has the premise T. In other words, if Δ' is a derivation induced by a linearization Lin of $\mathsf{Con}_{R,\Delta}$ then it has the premise T.*

Proof. Proof by induction on the length of Δ. If $k = 0$, then $\mathcal{C} = \emptyset$, hence $\mathsf{Lin} = \emptyset$. Then the only linearization of $\mathsf{Con}_{R,\Delta}$ is $R = T$.

For the inductive case, let Δ be of the form $R \xrightarrow{\Delta_k} T' \xrightarrow{\rho} T$. Let $\psi(\Delta_k) = \mathcal{C}_k$ and $\mathsf{Con}_{R,\Delta_k}$ be the corresponding concurrent derivation. By induction hypothesis, any derivation Δ'_k induced by a linearization Lin_k of $\mathsf{Con}_{R,\Delta_k}$ has the premise T' and by applying ρ to T' we obtain T. Let Δ'_k be $(R, \langle \mathsf{a}_1; \ldots; \mathsf{a}_k \rangle)$. For any $\rho \in \mathsf{MLS}$ and its action a, we have that, for $\mathsf{a}_i, \mathsf{a}_j \in \{\mathsf{a}_1, \ldots, \mathsf{a}_k\}$, $\mathcal{C}$ is given by

(a) either $\psi(\Delta) = \{(\mathsf{a}_i, \mathsf{a}), (\mathsf{a}_j, \mathsf{a})\} \cup \mathcal{C}_k$,

(b) or $\psi(\Delta) = \{(\mathsf{a}_i, \mathsf{a})\} \cup \mathcal{C}_k$.

We proceed as in case (a) since the case (b) follows from (a). Let $\mathsf{Con}_{R,\Delta}$ be the concurrent derivation obtained from $\mathcal{C}$, that is, $\mathsf{Con}_{R,\Delta}$ is the reflexive transitive

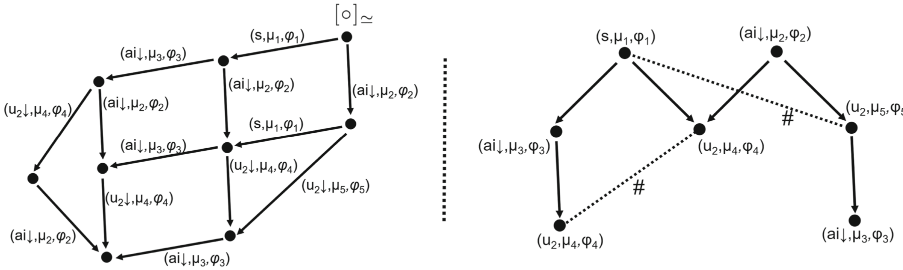


Fig. 5. The transition system $\mathsf{TS}'_{\simeq}(R)$ and $\mathsf{LES}'(R)$ obtained by using system MLSi instead of system MLS, which prune the derivations that do not result in proofs.

closure of $\mathcal{C}$. By induction hypothesis, for any linearization of $\mathsf{Con}_{R,\Delta_k}$, we have that $(R, \langle \mathsf{a}_1; \ldots; \mathsf{a}_k; \mathsf{a}\rangle)$ is a derivation with T at the premise.

In any linearization Lin of $\mathsf{Con}_{R,\Delta}$ we have that either $(\mathsf{a}_i, \mathsf{a}_\mathsf{j})$ or $(\mathsf{a}_j, \mathsf{a}_\mathsf{i})$. Let us assume the former (since otherwise we have an analogous case). Given that $(\mathsf{a}_j, \mathsf{a}) \in \mathcal{C}$, the redex of a is not modified by any $\mathsf{a}_n \in \{\mathsf{a}_{j+1}, \ldots, \mathsf{a}_k\}$, which implies that for any derivation $(R, \langle \mathsf{a}_1; \ldots; \mathsf{a}_j; \ldots; \mathsf{a}_k\rangle)$ induced by a linearization of $\mathsf{Con}_{R,\Delta_k}$, the linearizations induced by $\mathsf{Con}_{R,\Delta}$ are enumerated by all the derivations where a is applied anywhere after a_j. Because no action $\mathsf{a}_n \in \{\mathsf{a}_{j+1}, \ldots, \mathsf{a}_k\}$ modifies the redex of a, the derivation has the premise T.

Corollary 1. *For any* MLS *formula* R *and derivation* $R \xrightarrow{\Delta} T$, $\mathsf{Con}_{R,\Delta}$ *is a configuration in* $\mathsf{LES}[\![R]\!]$.

5 From Concurrent Derivations to Proof Search

Because of the exponential blow up in proof search that is a consequence of hard complexity bounds [7,11], the margin of successful applications are determined by the interplay between the breadth of the search space, nondeterminism and length of proofs. In general, deep inference provides short proofs due to a more immediate access to subformulae [1,9]. However, the greater nondeterminism and resulting large breadth of search space hinders broad proof search applications that benefit from these short proofs. In [9], we have provided a formal method that reduces nondeterminism in deep inference proof search, and this way provides a more immediate access to shorter proofs. Below, we demonstrate that this method together with the true-concurrency characterization above provides a means to simultaneously reduce nondeterminism and proof length. This is because an untamed introduction of concurrency can result in an excessive increase in the breadth of the search space. However, a formal mechanism such as the following can provide control on search-space-breadth due to concurrency.

Definition 19. *Given a formula* R, $\mathsf{at}\,R$ *is the set of the atoms in* R.

Example 12. For $R = [a \parr \bar{a} \parr b \parr (\bar{a} \otimes \bot) \parr (a \otimes \bar{b})]$, we have $\mathsf{at}\, R = \{a, \bar{a}, b, \bar{b}, \bot\}$.

Definition 20. [9] *Consider the switch rule.*

$$\mathsf{s} \,:\, [(r \otimes t) \parr u] \rightarrow ([r \parr u] \otimes t)$$

We say that an instance of switch is an instance of interaction switch (is) *iff*

(i) r and u are matched to formulae R and U such that $\mathsf{at}\,\overline{R} \cap \mathsf{at}\, U \neq \emptyset$, that is, R and U contain complementary atoms.
(ii) u is matched to formula that is a conjunction or an atom different from $\bot$.
(iii) r is matched to formula that is a disjunction or an atom different from $\mathsf{1}$.

Definition 21. [9] *System* MLSi *is the system obtained by replacing the switch rule in system* MLS *with the interaction switch rule.*

Example 13. Consider the formula $R = [a \parr (\bar{a} \otimes [b \parr \bar{b}])]$ with the transition system $\mathsf{TS}_{\simeq}[\![R]\!]$ depicted in Fig. 3 and $\mathsf{LES}[\![R]\!]$ depicted in Fig. 1. By using system MLSi instead of system MLS we obtain $\mathsf{TS}'_{\simeq}(R)$ and $\mathsf{LES}'(R)$ depicted in Fig. 5, which prune the two MLS derivations that do not result in a proof.

Theorem 4. [9] *Systems* MLS *and* MLSi *prove the same formulae, that is, system* MLSi *is complete for multiplicative linear logic.*

Example 14. In systems MLS and MLSi the shortest proof of $[a \parr (\bar{a} \otimes [a \parr \bar{a}])]$ has length three as illustrated in Fig. 1. However, by resorting to the causally independent true-concurrent characterization of derivations we can obtain proofs of length two by composing rules in parallel as hinted in Fig. 5. Below we first apply the composition of the causally independent rules s and $\mathsf{ai}\!\downarrow$, and then the composition of the rules $\mathsf{u_2}\!\downarrow$ and $\mathsf{ai}\!\downarrow$.

$$[a \parr (\bar{a} \otimes [b \parr \bar{b}])] \overset{\mathsf{s}|\mathsf{ai}\downarrow}{\rightarrow} ([a \parr \bar{a}] \otimes \mathsf{1}) \overset{\mathsf{u_2}\downarrow|\mathsf{ai}\downarrow}{\rightarrow} \mathsf{1}$$

6 Discussion

The partial order representation of derivations provides a canonical representation of the derivations that differ only with respect to permutation of inference rules, but that are in essence identical. Such a characterization provides a qualification of identity of proofs with respect to rule permutations, while distinguishing proofs that differ in the inference steps that they take. A similar characterization is provided, for example, by proof nets for the case of multiplicative linear logic [2]. However, proof nets lack any information about the deductive steps performed in the proof. In contrast, in our approach, proofs that have identical proof nets can be distinguished with respect to their different configurations that result from distinct inference strategies. This aspect, which is illustrated in Fig. 4 and Example 10, is one of the contributions of this paper. Such considerations are also addressed by atomic flows for classical logic [5,6].

The event structure characterization of derivations exploits the independence and causality of rule instances such that the rule instances that are not causally dependent become partially ordered. This partial order characterization relaxes their total order representation in the standard deep inference syntax, which is in fact recognized as a bureaucratic constraint rather than a logical requirement [6]. As a result of the event structure characterization, our method reveals the true-concurrent nature of derivations, which should find applications in logic programming. Within the computation as proof search paradigm that uses deductive systems as a framework for logic programming [12,13], such a true concurrent interpretation of proof construction should broaden the potential applications.

As we have demonstrated in the previous section, the true-concurrency characterization of the derivations provides a point of view of the rule instances that exposes their independence in proof search. This in return provides a means for concurrent application of inference rules in a way that reduces the length of the proofs. Although uncontrolled use of concurrency can result in an excessive increase in the breadth of search space, introducing control by means of other orthogonal methods for reducing nondeterminism [9] should result in improvement in proof search applications.

Topics of further investigation include carrying these methods to other logics with deep inference systems within a more general framework, and their exploration with respect to applications in logic programming and proof search.

References

1. Bruscoli, P., Guglielmi, A.: On the proof complexity of deep inference. ACM Trans. Computat. Logic **2**(14), 1–34 (2009)
2. Girard, J.-Y.: Linear logic: Its syntax and semantics. In: Girard, J.-Y., Lafont, Y., Regnier, L. (eds.) Advances in Linear Logic (Proceedings of the Workshop on Linear Logic, Cornell University), vol. 222. Cambridge University Press (1995)
3. Guglielmi, A.: Abstract Logic Programming in Linear Logic Independence and Causality in a First Order Calculus. Ph.D. thesis, Universita di Pisa (1996)
4. Guglielmi, A.: A system of interaction and structure. ACM Trans. Comput. Logic **8**(1), 1–64 (2007)
5. Guglielmi, A., Gundersen, T.: Normalisation control in deep inference via atomic flows. Log. Methods Comput. Sci. **4**(1:9), 1–36 (2008)
6. Guglielmi, A., Gundersen, T., Parigot, M.: A proof calculus which reduces syntactic bureaucracy. In: Proceedings of the International Conference on Rewriting Techniques and Applications 2010 (Edinburgh), pp. 135–150. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik 2010 LIPIcs (2010)
7. Kahramanoğulları, O.: System BV is NP-complete. Ann. Pure Appl. Logic **152**(1–3), 107–121 (2008)
8. Kahramanoğulları, O.: On linear logic planning and concurrency. Inform. Comput. **207**(11), 1229–1258 (2009)
9. Kahramanoğulları, O.: Interaction and depth against nondeterminism in proof search. Log. Methods Comput. Sci. **10**(2:5), 1–49 (2014)
10. Kahramanoğulları, O.: Maude as a platform for designing and implementing deep inference systems. In: Proceedings of the Eighth International Workshop on Rule-Based Programming, RULE 2007. ENTCS, vol. 219, pp. 35–50. Elsevier (2008)

11. Kanovich, M.: The multiplicative fragment of linear logic is NP-complete. Technical Report X-91-13, Institute for Language, Logic, and Information (1991)
12. Miller, D.: Forum: a multiple-conclusion specification logic. Theor. Comput. Sci. **165**, 201–232 (1996)
13. Miller, D.: Overview of linear logic programming. In: Ehrhard, T., Girard, J.-Y., Ruet, P., Scott, P. (eds.) Linear Logic in Computer Science. London Mathematical Society Lecture Note, vol. 316. Cambridge University Press, Cambridge (2004)
14. Sassone, V., Nielsen, M., Winskel, G.: Models for concurrency: towards a classification. Theor. Comput. Sci. **170**(1–2), 297–348 (1996)
15. Strassburger, L., Guglielmi, A.: A system of interaction and structure IV: The exponentials anddecomposition. ACM Trans. Comp. Logic **12**(4), 1–39 (2011)
16. Winskel, G., Nielsen, M.: Models for concurrency. In: Abramsky, S., Gabbay, D.M., Maibaum, T.S.E. (eds.) Handbook of Logic in Computer Science, vol. 4, pp. 1–148. Oxford University Press, Oxford (1995)

On the Complexity of the Equational Theory of Residuated Boolean Algebras

Zhe Lin[1(✉)] and Minghui Ma[2]

[1] Institute of Logic and Cognition, Sun Yat-sen University, Guangzhou, China
pennyshaq@gmail.com
[2] Institute for Logic and Intelligence, Southwest University, Chongqing, China
mmh.thu@gmail.com

Abstract. Residuated boolean algebras are introduced by Jonsson and Tsinakis [10] as generalizations of relation algebras. Jispen [12] proved that the equational theory of residuated boolean algebras with unit, and that of many relative classes of algebras are decidable. Buszkowski [2] showed the finite embeddability property for residuated boolean algebras, which yields the decidability of the universal theory of residuated boolean algebras. In this paper, we study the complexity of the equational theory of residuated boolean algebras. The main result is that the equational theory of residuated boolean algebras is PSPACE-complete.

1 Introduction

A *residuated Boolean algebra*, or *r-algebra*, is an algebra $\mathbf{A} = (\mathrm{A}, \wedge, \vee, ', \top, \bot, \cdot, \backslash, /)$ where $(\mathrm{A}, \wedge, \vee, ', \top, \bot)$ is a Boolean algebra, and $\cdot, \backslash$ and $/$ are binary operators on A satisfying the following residuation property: for any $a, b, c \in \mathrm{A}$,

$$a \cdot b \leq c \quad \text{iff} \quad b \leq a \backslash c \quad \text{iff} \quad a \leq c/b$$

The operators $\backslash$ and $/$ are called *right* and *left* residuals of $\cdot$ respectively.

The left and right conjugates of $\cdot$ are binary operators on A defined by setting

$$a \rhd c = (a \backslash c')' \text{ and } c \rhd b = (c'/b)'.$$

The following conjugation property holds for any $a, b, c \in \mathrm{A}$:

$$a \cdot b \leq c' \quad \text{iff} \quad a \rhd c \leq b' \quad \text{iff} \quad c \lhd b \leq a'$$

Equivalently, a r-algebra can also be defined as a Boolean algebra with a binary operation $\cdot$ and its left and right conjugates. We prefer to choose $\backslash$ and $/$ as basic operations.

A *unital r-algebra*, or *ur-algebra*, is a r-algebra $\mathbf{A} = (\mathrm{A}, \wedge, \vee, ', \top, \bot, \cdot, \backslash, /, 1)$ enriched with a unit element 1 with respect to $\cdot$, i.e., an element 1 satisfying

The work of the first author is supported by Chinese national fund for social sciences (grant No. 13&ZD186).

J. Väänänen et al. (Eds.): WoLLIC 2016, LNCS 9803, pp. 265–278, 2016.
DOI: 10.1007/978-3-662-52921-8_17

the condition $1 \cdot a = a \cdot 1 = a$ for all $a \in A$. A *residuated Boolean monoid*, or *rm-algebra*, is a ur-algebra in which $\cdot$ is associative. We always use boldface $\mathbf{A}$ for an algebra, and plain A for its base set. Classes of algebras are denoted by a kind of blackboard bold capital letters. The varieties of all r-algebras, all ur-algebras and all rm-algebras are denoted by $\mathbb{RBA}$, $\mathbb{URA}$ and $\mathbb{RMA}$ respectively.

Tarski [20] introduced relation algebras, and Maddux [17] introduced nonassociative relation algebras by weakening the monoid in a relation algebra to a groupoid with identity. The varieties of all relation algebras and all nonassociative relation algebras are denoted by $\mathbb{RA}$ and $\mathbb{NA}$ respectively. These algebras are generalized to residuated Boolean algebras by Jónsson and Tsinakis [11]. Both relation algebras and nonassociative relation algebras are characterized as special r-algebras by some equations. Algebraic studies on r-algebras can be found in literatures [8,9,12].

Let $\mathbb{K}$ be any class of algebras. The equational theory of $\mathbb{K}$, denoted by $Eq(\mathbb{K})$, is the set of all equations of the form $s = t$ that are valid in $\mathbb{K}$. The universal theory of $\mathbb{K}$ is the set of all first-order universal sentences that are valid in $\mathbb{K}$. Németi [18] proved that $Eq(\mathbb{NA})$ is decidable. Jispen [12] proved that the equational theories of the variety of all ur-algebras and some of its subvarieties are decidable. Buszkowski [2] proved the finite embeddability property (FEP) for r-algebras. If a class of algebras $\mathbb{K}$ has the FEP, then the universal theory of $\mathbb{K}$ is decidable, provided that $\mathbb{K}$ is finitely axiomatizable. Then the universal theory of r-algebras is decidable, and the same result is obtained by Kaminski and Francez [13] using different methods. Kurucz, Nemeti, Sain and Simon [15] proved that the equational theory of all Boolean algebras with an associative operator is undecidable. It follows that $Eq(\mathbb{RMA})$ is undecidable.

The aim of the present paper is to study the complexity of the decision problem of the equation validity in $\mathbb{RBA}$. Our approach is to reduce the decidability of $Eq(\mathbb{RBA})$ to the decidability of a sequent calculus for $\mathbb{RBA}$. A sequent calculus for r-algebras, called *Boolean nonassociative Lambek calculus* (BFNL), was developed by Buszkowski [2] and Kaminski and Francez [13]. It is an extension of non associative Lambek calculus (NL) that was introduced by Lambek [16] as a calculus for nonassociave residauted groupoids. BFNL is sound and complete with respect to $\mathbb{RBA}$, i.e., a sequent is derivable in BFNL if and only if it is valid in $\mathbb{RBA}$. Thus the decidability of $Eq(\mathbb{RBA})$ is reduced to the decidability of the derivability in BFNL. Consequently, the complexity of the decision problem of $Eq(\mathbb{RBA})$ is equal to the complexity of the decision problem of the derivability in BFNL.

In the following sections, we will analyze the complexity of the decision problem of BFNL. Our main result is that BFNL is PSPACE-complete. PSPACE-hardness of BFNL is obtained by a polynomial reduction from the minimal normal modal logic K, which is PSPACE-complete, to BFNL. That BFNL is in PSPACE is shown by a polynomial reduction from BFNL to the minimal bi-tense logic $\mathrm{K}^{\mathrm{t}}_{1,2}$. We show that $\mathrm{K}^{\mathrm{t}}_{1,2}$ is in PSACE by a polynomial reduction from it to the minimal tense logic K.t which is PSPACE-complete. As a result, $Eq(\mathbb{RBA})$ is PSPACE-complete. Our result also yields that the equational theory of residuated distributive lattice is in PSPACE.

2 Boolean Nonassociative Lambek Calculus

We recall some basic notions for BFNL. The language of BFNL $\mathcal{L}_{\mathsf{BFNL}}(\mathsf{Prop})$ is built from the set of propositional variables Prop by Lambek connectives $/, \backslash, \cdot$ and propositional connectives $\wedge, \vee, \bot, \top$ and $\neg$. The set of all $\mathcal{L}_{\mathsf{BFNL}}(\mathsf{Prop})$-*formulae* is defined inductively by the following rule:

$$A ::= p \mid \bot \mid \top \mid (A \cdot A) \mid (A/A) \mid (A\backslash A) \mid (A \wedge A) \mid (A \vee A) \mid \neg A$$

where $p \in \mathbf{Prop}$. We write p, q, r etc. for propositional variables, and A, B, C etc. for formulae. The set of all *formula trees* is defined inductively by the following rule:

$$\Gamma ::= A \mid \Gamma \circ \Gamma$$

where A is an $\mathcal{L}_{\mathsf{BFNL}}(\mathsf{Prop})$-formula. Each formula tree Γ is associated with a formula $f(\Gamma)$ defined inductively by: $f(A) = A$ and $f(\Gamma \circ \Delta) = f(\Gamma) \cdot f(\Delta)$. A *context* is a formula tree containing one occurrence of special atom $-$ (a place for substitution). If $\Gamma[-]$ is a context, then $\Gamma[\Delta]$ denotes the substitution of Δ for $-$ in Γ. Sequents are of the form $\Gamma \Rightarrow A$ where Γ is a formula tree and A is a formula. A *sequent* is an expression of the form $\Gamma \Rightarrow A$ where Γ is a formula tree and A is a formula. The sequent system BFNL consists the following axioms and rules:

$$\text{(Id) } A \Rightarrow A, \quad \text{(D) } A \wedge (B \vee C) \Rightarrow (A \wedge B) \vee (A \wedge C),$$

$$(\bot)\ \Gamma[\bot] \Rightarrow A, \quad (\top)\ \Gamma \Rightarrow \top,$$

$$(\neg 1)\ A \wedge \neg A \Rightarrow \bot, \quad (\neg 2)\ \top \Rightarrow A \vee \neg A,$$

$$(\backslash \mathrm{L})\ \frac{\Delta \Rightarrow A \quad \Gamma[B] \Rightarrow C}{\Gamma[\Delta \circ (A\backslash B)] \Rightarrow C}, \quad (\backslash \mathrm{R})\ \frac{A \circ \Gamma \Rightarrow B}{\Gamma \Rightarrow A\backslash B},$$

$$(/\mathrm{L})\ \frac{\Gamma[A] \Rightarrow C \quad \Delta \Rightarrow B}{\Gamma[(A/B) \circ \Delta] \Rightarrow C}, \quad (/\mathrm{R})\ \frac{\Gamma \circ B \Rightarrow A}{\Gamma \Rightarrow A/B},$$

$$(\cdot \mathrm{L})\ \frac{\Gamma[A \circ B] \Rightarrow C}{\Gamma[A \cdot B] \Rightarrow C}, \quad (\cdot \mathrm{R})\ \frac{\Gamma \Rightarrow A \quad \Delta \Rightarrow B}{\Gamma \circ \Delta \Rightarrow A \cdot B}, \quad (\mathrm{Cut})\ \frac{\Delta \Rightarrow A; \quad \Gamma[A] \Rightarrow B}{\Gamma[\Delta] \Rightarrow B},$$

$$(\wedge \mathrm{L})\ \frac{\Gamma[A_i] \Rightarrow B}{\Gamma[A_1 \wedge A_2] \Rightarrow B}, \quad (\wedge \mathrm{R})\ \frac{\Gamma \Rightarrow A \quad \Gamma \Rightarrow B}{\Gamma \Rightarrow A \wedge B},$$

$$(\vee \mathrm{L})\ \frac{\Gamma[A_1] \Rightarrow B \quad \Gamma[A_2] \Rightarrow B}{\Gamma[A_1 \vee A_2] \Rightarrow B}, \quad (\vee \mathrm{R})\ \frac{\Gamma \Rightarrow A_i}{\Gamma \Rightarrow A_1 \vee A_2}.$$

In $(\wedge\mathrm{L})$ and $(\vee\mathrm{R})$, the subscript i equals 1 or 2.

By $\vdash_{\mathrm{BFNL}} \Gamma \Rightarrow A$ we mean that the sequent $\Gamma \Rightarrow A$ is *derivable* in BFNL. The notation $A \Leftrightarrow B$ stands for $A \Rightarrow B$ and $B \Rightarrow A$.

Fact 1. *The following sequents are derivable in* BFNL*:*

(1) $C \cdot (A \vee B) \Leftrightarrow (C \cdot A) \vee (C \cdot B)$,
(2) $(A \vee B) \cdot C \Leftrightarrow (A \cdot C) \vee (B \cdot C)$,
(3) $C \cdot (A \wedge B) \Rightarrow (C \cdot A) \wedge (C \cdot B)$,
(4) $(A \wedge B) \cdot C \Rightarrow (A \cdot C) \wedge (B \cdot C)$,
(5) $\neg A \vee \neg B \Leftrightarrow \neg(A \wedge B)$,
(1) $\neg(A \vee B) \Leftrightarrow \neg A \wedge \neg B$,
(7) $\neg\bot \Leftrightarrow \top$ *and* $\neg\top \Leftrightarrow \bot$,
(8) $A \wedge (B \vee C) \Leftrightarrow (A \wedge B) \vee (A \wedge C)$,
(9) $A \vee (B \wedge C) \Leftrightarrow (A \vee B) \wedge (A \vee C)$,

Fact 2. *For any $\mathcal{L}_{\mathsf{BFNL}}(\mathsf{Prop})$-formulae A and B,*

(1) *if* $\vdash_{\mathrm{BFNL}} A \Rightarrow B$, *then* $\vdash_{\mathrm{BFNL}} \neg B \Rightarrow \neg A$,
(2) $\vdash_{\mathrm{BFNL}} A \Rightarrow B$ *if and only if* $\vdash_{\mathrm{BFNL}} \top \Rightarrow \neg A \vee B$.

BFNL has Kripke semantics ([13]). A *ternary frame* is a pair (W, S) such that W is a nonempty set and S is an arbitrary ternary relation on W. A *ternary model* $\mathfrak{J} = (W, S, \sigma)$ consists of a ternary frame (W, S) and a valuation function $\sigma : \mathsf{Prop} \to \mathcal{P}(W)$ from Prop to the powerset of W. The *satisfiability relation* $\mathfrak{J}, w \models A$ between a ternary model $\mathfrak{J}$ with a state $w \in W$ and an $\mathcal{L}_{\mathsf{BFNL}}(\mathsf{Prop})$-formula A is recursively defined as below:

$\mathfrak{J}, u \models p$ iff $u \in \sigma(p)$.
$\mathfrak{J}, u \not\models \bot$.
$\mathfrak{J}, u \models \top$.
$\mathfrak{J}, u \models A \cdot B$ iff there are $v, w \in W$ such that $S(u, v, w)$, $\mathfrak{J}, v \models A$ and $\mathfrak{J}, w \models B$.
$\mathfrak{J}, u \models A/B$ iff for all $v, w \in W$ with $S(w, u, v)$, if $\mathfrak{J}, v \models B$, then $\mathfrak{J}, w \models A$
$\mathfrak{J}, u \models A\backslash B$ iff for all $v, w \in W$ with $S(v, w, u)$, if $\mathfrak{J}, w \models A$, then $\mathfrak{J}, v \models B$.
$\mathfrak{J}, u \models A \wedge B$ iff $\mathfrak{J}, u \models A$ and $\mathfrak{J}, u \models B$,
$\mathfrak{J}, u \models A \vee B$ iff $\mathfrak{J}, u \models A$ or $\mathfrak{J}, u \models B$.
$\mathfrak{J}, u \models \neg A$ iff $\mathfrak{J}, u \not\models A$.

An $\mathcal{L}_{\mathsf{BFNL}}(\mathsf{Prop})$-formula A is *satisfiable* if $\mathfrak{J}, u \models A$ for some ternary model $\mathfrak{J} = (W, R, \sigma)$ and some $u \in W$. We say that A is *true* in $\mathfrak{J}$ (notation: $\mathfrak{J} \models A$), if $\mathfrak{J}, u \models A$ for all $u \in W$. For any sequent $\Gamma \Rightarrow A$, we say that $\Gamma \Rightarrow A$ is true at a state u in a ternary model $\mathfrak{J}$ (notation: $\mathfrak{J}, u \models \Gamma \Rightarrow A$), if $\mathfrak{J}, u \models f(\Gamma)$ implies $\mathfrak{J}, u \models A$. A sequent $\Gamma \Rightarrow A$ is *true* in $\mathfrak{J}$ (notation: $\mathfrak{J} \models \Gamma \Rightarrow A$), if $\mathfrak{J}, u \models \Gamma \Rightarrow A$ for all $u \in W$. By $\models_{\mathbf{BFNL}} \Gamma \Rightarrow A$ we mean that $\Gamma \Rightarrow A$ is true in all ternary models.

The soundness and completeness of the Hilbert-style presentation PNL of BFNL under Kripke semantics are settled in [13]. Let us recall this theorem.

Theorem 3. *For any formula A, $\vdash_{\mathrm{PNL}} A$ if and only if $\models_{\mathrm{PNL}} A$.*

The relation between BFNL and PNL is as below: for any $\mathcal{L}_{\mathsf{BFNL}}(\mathsf{Prop})$-formula A, we have $\vdash_{\mathrm{PNL}} A$ if and only if $\vdash_{\mathrm{BFNL}} \top \Rightarrow A$. Consequently, by Fact 2 (2), for any $\mathcal{L}_{\mathsf{BFNL}}(\mathsf{Prop})$-formula $A \supset B$, $\vdash_{\mathrm{PNL}} A \supset B$ if and only if $\vdash_{\mathrm{BFNL}} A \Rightarrow B$, where $\supset$ is the Boolean implication. Note that the Kripke semantics of BFNL defined here is the same as PNL in [13]. Therefore we have the following theorem:

Theorem 4. *For any sequent $\Gamma \Rightarrow A$, $\vdash_{\mathrm{BFNL}} \Gamma \Rightarrow A$ if and only if $\models_{\mathrm{BFNL}} \Gamma \Rightarrow A$.*

3 Modal and Tense Logics

The language of modal logic $\mathcal{L}_{\mathrm{M}}$ consists of a set of propositional variables Prop, propositional connectives $\bot, \supset$ and a unary modal operator $\Diamond$. The set of all modal formulas is defined inductively by the following rule:

$$A ::= p \mid \bot \mid (A \supset A) \mid \Diamond A, \text{ where } p \in \mathsf{Prop}.$$

Define $\neg A := A \supset \bot$. Other propositional connectives $\top, \wedge, \vee$ and $\leftrightarrow$ are defined as usual. The dual of $\Diamond$ is defined by $\Box A := \neg\Diamond\neg A$.

The language of tense logic $\mathcal{L}_t$ is an extension of $\mathcal{L}_{\mathrm{M}}$ by adding a unary operator $\Box^{\downarrow}$. The set of all tense formulas is defined inductively by the following rule:

$$A ::= p \mid \bot \mid (A \supset A) \mid \Diamond A \mid \Box^{\downarrow} A, \quad p \in \mathsf{Prop}.$$

Define $\Diamond^{\downarrow} A := \neg\Box^{\downarrow}\neg A$.

Definition 1. *The minimal modal system* K *consists of the following axiom schemata and inference rules:*

(1) *All instances of classical propositional tautologies.*
(2) $\Box(A \supset B) \supset (\Box A \supset \Box B)$
(3) *MP: from A and $A \supset B$ infer B.*
(4) *Nec$\Box$: from A infer $\Box A$.*

By $\vdash_{\mathrm{K}} A$ we mean that the modal formula A is provable in K.

Definition 2. *The minimal tense system* K.t *consists of the following axiom schemata and inference rules:*

(1) *All instances of classical propositional tautologies.*
(2) $\Box(A \supset B) \supset (\Box A \supset \Box B)$.
(3) $\Box^{\downarrow}(A \supset B) \supset (\Box^{\downarrow} A \supset \Box^{\downarrow} B)$.
(4) $A \supset \Box\Diamond^{\downarrow} A$.
(5) $A \supset \Box^{\downarrow}\Diamond A$.
(6) *MP: from A and $A \supset B$ infer B.*
(7) *Nec$\Box$: from A infer $\Box A$.*
(8) *Nec$\Box^{\downarrow}$: from A infer $\Box^{\downarrow} A$.*

By $\vdash_{\mathrm{K.t}} A$ *we mean that the tense formula* A *is provable in* K.t*.*

We give the following facts without proof that will be used to prove the embedding result in Theorem 15.

Fact 5. *The following hold in* K.t*:*

(1) *if* $\vdash_{\mathrm{K.t}} A \supset B$*, then* $\vdash_{\mathrm{K.t}} (C \supset A) \supset (C \supset B)$*.*
(2) *if* $\vdash_{\mathrm{K.t}} (A \wedge B) \supset C$*, then* $\vdash_{\mathrm{K.t}} A \supset (B \supset C)$*.*
(3) *if* $\vdash_{\mathrm{K.t}} A \supset B$ *and* $\vdash_{\mathrm{K.t}} B \supset C$*, then* $\vdash_{\mathrm{K.t}} A \supset C$*.*
(4) $\vdash_{\mathrm{K.t}} (A \supset (B \supset C)) \supset ((A \supset B) \supset (A \supset C))$*.*
(5) *if* $\vdash_{\mathrm{K.t}} A \supset B$*, then* $\vdash_{\mathrm{K.t}} \Box A \supset \Box B$*.*
(6) *if* $\vdash_{\mathrm{K.t}} A \supset B$*, then* $\vdash_{\mathrm{K.t}} \Box^{\downarrow} A \supset \Box^{\downarrow} B$*.*
(7) *if* $\vdash_{\mathrm{K.t}} \Diamond A \supset B$*, then* $\vdash_{\mathrm{K.t}} A \supset \Box^{\downarrow} B$*.*
(8) *if* $\vdash_{\mathrm{K.t}} \Diamond^{\downarrow} A \supset B$*, then* $\vdash_{\mathrm{K.t}} A \supset \Box B$*.*

A *binary frame* is a pair $\mathfrak{F} = (W, R)$ where W is a nonempty set of states, and R is a binary relation over W. A *binary model* $\mathfrak{M} = (W, R, \sigma)$ consists of a binary frame (W, R) and a valuation $\sigma : \mathsf{Prop} \to \mathcal{P}(W)$. The *satisfiability relation* $\mathfrak{M}, w \models A$ between a model $\mathfrak{M}$ with a state $w \in W$ and a tense formula A is defined as below:

$\mathfrak{M}, w \models p$ iff $w \in \sigma(p)$.
$\mathfrak{M}, w \not\models \bot$.
$\mathfrak{M}, w \models A \supset B$ iff $\mathfrak{M}, w \not\models A$ or $\mathfrak{M}, w \models B$.
$\mathfrak{M}, w \models \Diamond A$ iff there exists $u \in W$ with $R(w, u)$ and $\mathfrak{M}, u \models A$.
$\mathfrak{M}, w \models \Box^{\downarrow} A$ iff for every $u \in W$, if $R(u, w)$, then $\mathfrak{M}, u \models A$.

Then we have (i) $\mathfrak{M}, w \models \Diamond^{\downarrow} A$ iff there exists $u \in W$ with $R(u, w)$ and $\mathfrak{M}, u \models A$; and (ii) $\mathfrak{M}, w \models \Box A$ iff for every $u \in W$, if $R(w, u)$, then $\mathfrak{M}, u \models A$. Note that the Kripke semantics for the modal language is given by the semantic clauses for $\Diamond$ and $\Box$.

The notions of satisfiability and validity are defined as usual. By $\models_{\mathrm{K.t}} A$ (respectively $\models_{\mathrm{K}} A$) we mean that the tense formula A (respectively the modal formula A) is valid in all binary frames. By the standard canonical model construction, it is easy to show the soundness and completeness of K.t and K, i.e., (1) for any tense formula A, $\vdash_{\mathrm{K.t}} A$ if and only if $\models_{\mathrm{K.t}} A$; (2) for any modal formula A, $\vdash_{\mathrm{K}} A$ if and only if $\models_{\mathrm{K}} A$.

The bi-tense language is defined like the tense language but with two pair of tense operators $\Diamond_1$, $\Box_1^{\downarrow}$ and $\Diamond_2$, $\Box_2^{\downarrow}$ with their duals $\Box_1$, $\Diamond_1^{\downarrow}$ and $\Box_2$, $\Diamond_2^{\downarrow}$, respectively. The minimal bi-tense system $\mathrm{K}^{\mathrm{t}}_{1,2}$ is defined exactly like the minimal tense system K.t but the axiom schemata and inference rules are given for two pairs of tense operators. By $\vdash_{\mathrm{K}^{\mathrm{t}}_{1,2}} A$ we mean that the bi-tense formula A is provable in $\mathrm{K}^{\mathrm{t}}_{1,2}$.

A *bi-tense frame* is a triple $\mathfrak{F} = (W, R_1, R_2)$ where W is a nonempty set of states, and R_1, R_2 are binary relations over W. A *bi-tense model* $\mathfrak{M} = (W, R_1, R_2, \sigma)$ consists of a bi-tense frame (W, R_1, R_2) and a valuation $\sigma : \mathsf{Prop} \to \wp(W)$. The notions of satisfiability and validity are defined as usual.

By $\models_{\mathrm{K}^{\mathrm{t}}_{1,2}} A$ we mean that A is valid in all bi-tense frames. We also have the soundness and completeness of the bi-tense system $\mathrm{K}^{\mathrm{t}}_{1,2}$, i.e., for every bi-tense formula A, $\vdash_{\mathrm{K}^{\mathrm{t}}_{1,2}} A$ if and only if $\models_{\mathrm{K}^{\mathrm{t}}_{1,2}} A$ (cf. [1]).

4 PSPACE-Hard Decision Problem of BFNL

We reduce the validity problem of K, which is PSPACE-complete, to the decision problem of BFNL in P-time so that we prove the PSPACE-hardness of the decision problem of BFNL. Now let us consider an embedding from modal logic K into BFNL.

Definition 6. *Let* $\mathsf{P} \subseteq \mathsf{Prop}$ *and* $m \notin \mathsf{P}$ *be a distinguished propositional variable. Define a translation (mapping)* $(.)^\dagger\colon \mathcal{L}_{\mathrm{K}}(\mathsf{P}) \to \mathcal{L}_{\mathrm{BFNL}}(\mathsf{P} \cup \{m\})$ *recursively as follows:*

$$p^\dagger = p, \qquad \bot^\dagger = \bot,$$
$$(A \supset B)^\dagger = A^\dagger \supset B^\dagger, \qquad (\Diamond A) = m \cdot A^\dagger.$$

Let $\mathfrak{M} = (W, R, \sigma)$ be a binary model. Define a ternary model $\mathfrak{J}^{\mathfrak{M}} = (W', R', \sigma')$ from $\mathfrak{M}$ as follows:

(1) if $w \in W$, then put two copies w_1, w_2 of w into W',
(2) if $R(w, u)$, then $R'(w_1, w_2, u_1)$,
(3) $w_i \in \sigma'(p)$ iff $w \in \sigma(p)$, for all $p \in \mathsf{P}$ and $i \in \{1, 2\}$; and $\sigma'(m) = W'$.

For each state w in the original binary model, we make two copies w_1 and w_2. Then define a ternary relation among these copies according to the original binary relation.

Lemma 7. *Suppose that* $\mathfrak{M} = (W, R, \sigma)$ *is a binary model and* $\mathfrak{J}^{\mathfrak{M}} = (W', R', \sigma')$ *is defined from* $\mathfrak{M}$ *as in Definition 6. Then for any* $w \in W$ *and modal formula* A, $\mathfrak{M}, w \models A$ *if and only if* $\mathfrak{J}^{\mathfrak{M}}, w_1 \models A^\dagger$.

Proof. By induction on the complexity of A. The atomic and Boolean cases are easy by the construction of $\mathfrak{J}^{\mathfrak{M}}$ and the induction hypothesis. For $A = \Diamond B$, assume $\mathfrak{M}, w \models \Diamond B$. Then there exists $u \in W$ such that $R(w, u)$ and $\mathfrak{M}, u \models B$. Since $R(w, u)$, we get $R'(w_1, w_2, u_1)$. By the induction hypothesis, $\mathfrak{J}^{\mathfrak{M}}, u_1 \models B^\dagger$. Hence $\mathfrak{J}^{\mathfrak{M}}, w_1 \models m \cdot B^\dagger$. Conversely, assume $\mathfrak{J}^{\mathfrak{M}}, w_1 \models m \cdot B^\dagger$. Then there exist $k, z \in W'$ such that $R'(w_1, k, z)$, $\mathfrak{J}^{\mathfrak{M}}, k \models m$ and $\mathfrak{J}^{\mathfrak{M}}, z \models B^\dagger$. By the construction $k = w_2$ and $z = u_1$ for some $u \in W$. By the induction hypothesis, $\mathfrak{M}, u \models B$. By the construction of $\mathfrak{J}^{\mathfrak{M}}$, we get $R(w, u)$. Hence $\mathfrak{M}, w \models \Diamond B$. □

Lemma 8. *For any modal formula* A, *if* $\vdash_{\mathrm{BFNL}} \top \Rightarrow A^\dagger$, *then* $\vdash_{\mathrm{K}} A$.

Proof. Assume $\nvdash_{\mathrm{K}} A$. Then there is a binary model $\mathfrak{M}$ such that $\mathfrak{M} \not\models A$. By Lemma 7, $\mathfrak{J}^{\mathfrak{M}} \not\models A^\dagger$. So $\mathfrak{J}^{\mathfrak{M}} \not\models \top \Rightarrow A^\dagger$. By Theorem 4, we get $\nvdash_{\mathrm{BFNL}} \top \Rightarrow A^\dagger$. □

Lemma 9. *For any modal formula A, if $\vdash_{\mathrm{K}} A$, then $\vdash_{\mathrm{BFNL}} \top \Rightarrow A^\dagger$.*

Proof. We proceed by induction on the length of proofs in K. It suffices to show that all axioms and inference rules of K are admissible in BFNL with respect to the translation †. Obviously the translations of all tautologies of classical propositional logic are provable in BFNL. Consider $(\Box(A \supset B) \supset (\Box A \supset \Box B))^\dagger = (m \cdot (A^\dagger \wedge \neg B^\dagger)) \vee (m \cdot (\neg A^\dagger)) \vee (\neg(m \cdot (\neg B^\dagger))$. Since $A^\dagger \wedge B^\dagger \Rightarrow B^\dagger$, by Fact 2 (1) and 1 (5), one gets $\vdash_{\mathrm{BFNL}} \neg B^\dagger \Rightarrow (\neg A^\dagger \vee \neg B^\dagger)$. Hence by monotonicity of $\cdot$, one gets $\vdash_{\mathrm{BFNL}} m \cdot (\neg B^\dagger) \Rightarrow (m \cdot (\neg A^\dagger \vee \neg B^\dagger))$. Then by Fact 2 (2) one gets $\vdash_{\mathrm{BFNL}} \top \Rightarrow \neg(m \cdot (\neg B^\dagger)) \vee (m \cdot (\neg A^\dagger \vee \neg B^\dagger))$. Since $\vdash_{\mathrm{BFNL}} (A^\dagger \vee \neg A^\dagger) \Leftrightarrow \top$, one gets $\vdash_{\mathrm{BFNL}} (m \cdot (\neg A^\dagger \vee \neg B^\dagger)) \Leftrightarrow m \cdot ((A^\dagger \vee \neg A^\dagger) \wedge (\neg A^\dagger \vee \neg B^\dagger))$. By Fact 1 (9) and monotonicity of $\cdot$, one gets $\vdash_{\mathrm{BFNL}} m \cdot ((A^\dagger \vee \neg A^\dagger) \wedge (\neg A^\dagger \vee \neg B^\dagger)) \Leftrightarrow m \cdot ((A^\dagger \wedge \neg B^\dagger) \vee \neg A^\dagger)$. Again, by Fact 1 (1) one can prove that $\vdash_{\mathrm{BFNL}} m \cdot ((A^\dagger \wedge \neg B^\dagger) \vee \neg A^\dagger) \Leftrightarrow (m \cdot (A^\dagger \wedge \neg B^\dagger)) \vee ((m \cdot (\neg A^\dagger)))$. Hence one gets $\vdash_{\mathrm{BFNL}} \top \Rightarrow (m \cdot (A^\dagger \wedge \neg B^\dagger)) \vee (m \cdot (\neg A^\dagger)) \vee (\neg(m \cdot (\neg B^\dagger)))$.

Consider the rule (MP). Assume that $\vdash_{\mathrm{BFNL}} \top \Rightarrow A^\dagger$ and $\vdash_{\mathrm{BFNL}} \top \Rightarrow (A \supset B)^\dagger$, which is equal to $\vdash_{\mathrm{BFNL}} \top \Rightarrow \neg A^\dagger \vee B^\dagger$. We need to show $\vdash_{\mathrm{BFNL}} \top \Rightarrow B^\dagger$. By ($\neg$1), ($\bot$) and (Cut), one gets $\vdash_{\mathrm{BFNL}} A^\dagger \wedge \neg A^\dagger \Rightarrow B^\dagger$. By ($\wedge$L), one gets $\vdash_{\mathrm{BFNL}} A^\dagger \wedge B^\dagger \Rightarrow B^\dagger$. By ($\vee$L), one gets $\vdash_{\mathrm{BFNL}} (A^\dagger \wedge \neg A^\dagger) \vee (A^\dagger \wedge B^\dagger) \Rightarrow B^\dagger$. Then by (D) and (Cut), one gets $\vdash_{\mathrm{BFNL}} A^\dagger \wedge (\neg A^\dagger \vee B^\dagger) \Rightarrow B^\dagger$. Clearly, by assumptions and ($\wedge$R), one gets $\vdash_{\mathrm{BFNL}} \top \Rightarrow A^\dagger \wedge (\neg A^\dagger \vee B^\dagger)$, which yields $\vdash_{\mathrm{BFNL}} \top \Rightarrow B^\dagger$ by (Cut).

Finally we consider the rule (Nec). Assume $\vdash_{\mathrm{BFNL}} \top \Rightarrow A^\dagger$. We need to show $\vdash_{\mathrm{BFNL}} \top \Rightarrow \neg(m \cdot (\neg A^\dagger))$. By Fact 2 (1) and the assumption, one gets $\vdash_{\mathrm{BFNL}} \neg(A^\dagger) \Rightarrow \bot$. By monotonicity, one gets $\vdash_{\mathrm{BFNL}} m \cdot \neg(A^\dagger) \Rightarrow m \cdot \bot$. Then by ($\bot$), ($\cdot$L) and (Cut), one gets $\vdash_{\mathrm{BFNL}} m \cdot \neg(A^\dagger) \Rightarrow \bot$. Hence by Fact 2 (1), one gets $\vdash_{\mathrm{BFNL}} \top \Rightarrow \neg(m \cdot (\neg A^\dagger))$. □

Theorem 10. *For any modal formula A, $\vdash_{\mathrm{K}} A$ if and only if $\vdash_{\mathrm{BFNL}} \top \Rightarrow A^\dagger$.*

Proof. By Lemmas 8 and 9. □

A normal modal logic is a set S of modal formulae such that all theorems of K belongs to S and S is closed under MP, Nec and uniform substitution. The PSPACE-hardness of the validity problems of some modal logics were settled first by Lander [19]. Let us recall this theorem from [19].

Theorem 11 (Lander's Theorem). *If S is a normal modal logic such that* $\mathrm{K} \subseteq S \subseteq \mathrm{S4}$, *then S has a PSPACE-hard satisfiability problem. Moreover, S has PSPACE-hard validity problem.*

Obviously the reduction $(.)^\dagger$ is in polynomial time. By Lardner's theorem (Theorem 11), one gets the following result:

Theorem 12. BFNL *is PSPACE-hard.*

5 BFNL is in PSPACE

Kurtonina [14] proved that the nonassociative Lambek calculus NL is faithfully embedded into the minimal bi-tense system $\mathrm{K}^{\mathrm{t}}_{1,2}$, i.e., for any formulas A and B in the language of NL, $\vdash_{\mathrm{NL}} A \Rightarrow B$ if and only if $\vdash_{\mathrm{K}^{\mathrm{t}}_{1,2}} A^{\#} \supset B^{\#}$, where $\#$ is a mapping from the language of NL to the bi-tense language. This result can be extended to an embedding from BFNL into $\mathrm{K}^{\mathrm{t}}_{1,2}$.

Definition 3. *The translation* $(.)^{\#} : \mathcal{L}_{\mathrm{BFNL}}(\mathsf{Prop}) \to \mathcal{L}_{\mathrm{K}^{\mathrm{t}}_{1,2}}(\mathsf{Prop})$ *is defined as below:*

$$
\begin{aligned}
p^{\#} &= p, & \top^{\#} &= \top, \quad \bot^{\#} = \bot,\\
(\neg A)^{\#} &= \neg A^{\#}, & (A \wedge B)^{\#} &= A^{\#} \wedge B^{\#},\\
(A \vee B)^{\#} &= A^{\#} \vee B^{\#}, & (A \cdot B)^{\#} &= \Diamond_1(\Diamond_1 A^{\#} \wedge \Diamond_2 B^{\#}),\\
(A\backslash B)^{\#} &= \Box^{\downarrow}_2(\Diamond_1 A^{\#} \supset \Box^{\downarrow}_1 B^{\#}), & (A/B)^{\#} &= \Box^{\downarrow}_1(\Diamond_2 B^{\#} \supset \Box^{\downarrow}_1 A^{\#}).
\end{aligned}
$$

Theorem 13. *For any $\mathcal{L}_{\mathrm{BFNL}}$-sequent $\Gamma \Rightarrow D$, $\vdash_{\mathrm{BFNL}} \Gamma \Rightarrow D$ if and only if $\vdash_{\mathrm{K}^{\mathrm{t}}_{1,2}} (f(\Gamma))^{\#} \supset D^{\#}$.*

Proof. We can confine ourselves to the relation $\vdash_{\mathrm{BFNL}} A \Rightarrow B$, since every sequent $\Gamma \Rightarrow D$ is deductively equivalent in BFNL to $f(\Gamma) \Rightarrow D$. The left-to-right direction is shown by induction on the proof of $A \Rightarrow B$ in BFNL. Notice that it is easy to show that the translations of axioms in BFNL are theorems in $\mathrm{K}^{\mathrm{t}}_{1,2}$. For instance, consider $(\neg 1)$ $A \wedge \neg A \Rightarrow \bot$. Its translation $(A^{\#} \wedge \neg A^{\#}) \supset \bot$ is a theorem of $\mathrm{K}^{\mathrm{t}}_{1,2}$. Rules of BFNL are checked regularly. We demonstrate only one typical case. Let us consider $(\backslash\mathrm{L})$. It suffices to show that the following rule is admissible under the translation:

$$\frac{A \Rightarrow B \quad C \Rightarrow D}{A \cdot (B\backslash C) \Rightarrow D}$$

Assume that $\vdash_{\mathrm{K}^{\mathrm{t}}_{1,2}} A^{\#} \supset B^{\#}$ and $\vdash_{\mathrm{K}^{\mathrm{t}}_{1,2}} C^{\#} \supset D^{\#}$. It suffices to show that $\vdash_{\mathrm{K}^{\mathrm{t}}_{1,2}} \Diamond_1(\Diamond_1 A^{\#} \wedge \Diamond_2 \Box^{\downarrow}_2(\Diamond_1 B^{\#} \supset \Box^{\downarrow}_1 C^{\#})) \supset D^{\#}$. For any bi-tense model $\mathfrak{M} = (W, R_1, R_2, V)$ and $u \in W$, assume that $\mathfrak{M}, u \models \Diamond_1(\Diamond_1 A^{\#} \wedge \Diamond_2 \Box^{\downarrow}_2(\Diamond_1 B^{\#} \supset \Box^{\downarrow}_1 C^{\#}))$. It suffices to show $\mathfrak{M}, u \models D^{\#}$. By assumptions, we have $\mathfrak{M} \models A^{\#} \supset B^{\#}$ and $\mathfrak{M} \models C^{\#} \supset D^{\#}$. Again by the assumption, we get $\mathfrak{M}, v \models \Diamond_1 A^{\#} \wedge \Diamond_2 \Box^{\downarrow}_2(\Diamond_1 B^{\#} \supset \Box^{\downarrow}_1 C^{\#})$ for some $v \in W$ such that $R_1(u, v)$. Then there exist $w, z \in W$ such that $R_1(v, w)$, $R_2(v, z)$, $\mathfrak{M}, w \models A^{\#}$ and $\mathfrak{M}, z \models \Box^{\downarrow}_2(\Diamond_1 B^{\#} \supset \Box^{\downarrow}_1 C^{\#})$. Since $\mathfrak{M} \models A^{\#} \supset B^{\#}$, we get $\mathfrak{M}, w \models B^{\#}$. Hence $\mathfrak{M}, v \models \Diamond_1 B^{\#}$. Then by $\mathfrak{M}, z \models \Box^{\downarrow}_2(\Diamond_1 B^{\#} \supset \Box^{\downarrow}_1 C^{\#})$ and $R_2(v, z)$, we get $\mathfrak{M}, v \models \Box^{\downarrow}_1 C^{\#}$. Since $R_1(u, v)$, we get $\mathfrak{M}, u \models C^{\#}$. By the assumption $\mathfrak{M} \models C^{\#} \supset D^{\#}$, we get $\mathfrak{M}, u \models D^{\#}$. Hence we get $\vdash_{\mathrm{K}^{\mathrm{t}}_{1,2}} (A \cdot (B\backslash C))^{\#} \Rightarrow D^{\#}$.

For the other direction, assume that $\not\vdash_{\mathrm{BFNL}} A \Rightarrow B$. By completeness of BFNL, there exists a ternary model $\mathfrak{M} = (W, R, \sigma)$ and a state $k \in W$ such that $\mathfrak{M}, k \models A$ but $\mathfrak{M}, k \not\models B$. Then we construct a bi-tense model $\mathfrak{M}^* = (W^*, R_1, R_2, \sigma^*)$ from $\mathfrak{M}$ satisfying the following conditions:

- $k \in W^*$.
- if $R(u,v,w)$, then put a fresh state x and u,v,w into W^* such that $R_1(u,x)$, $R_1(x,v)$ and $R_2(x,w)$.
- set for all $u \in W \cap W^*$ and $p \in \mathsf{Prop}$, $u \in \sigma^*(p)$ iff $u \in \sigma(p)$.

We may show that for any $u \in W \cap W^*$ and $\mathcal{L}_{\mathrm{BFNL}}$-formula A,

$$(\mho)\ \ \mathfrak{M}, u \models A \text{ iff } \mathfrak{M}^*, u \models A^{\#}.$$

By induction on the length of A. The basic case is a direct consequence of the definition of $\mathfrak{M}^*$. We demonstrate only one typical clause of the inductive step. Let us consider the case $A = B \cdot C$. Then $B \cdot C = \Diamond_1(\Diamond_1 B^{\#} \wedge \Diamond_2 C^{\#})$. Assume that $\mathfrak{M}, u \models B \cdot C$. Then there exist $v, w \in W$ such that $R(u,v,w)$, $\mathfrak{M}, v \models B$ and $\mathfrak{M}, w \models C$. By the induction hypothesis, $\mathfrak{M}^*, v \models B^{\#}$ and $\mathfrak{M}^*, w \models C^{\#}$. By the construction, there exists $x \in W^*$ such that $R_1(u,x)$, $R_1(x,v)$ and $R_2(x,w)$. Hence $\mathfrak{M}^*, x \models \Diamond_1 B^{\#} \wedge \Diamond_2 C^{\#}$. Hence $\mathfrak{M}^*, u \models \Diamond_1(\Diamond_1 B^{\#} \wedge \Diamond_2 C^{\#})$. Conversely, assume that $\mathfrak{M}^*, u \models \Diamond_1(\Diamond_1 B^{\#} \wedge \Diamond_2 C^{\#})$. Then there exist $x, v, w \in W^*$ such that $R_1(u,x)$, $R_1(x,v)$, $R_2(x,w)$, and $\mathfrak{M}^*, v \models B^{\#}$ and $\mathfrak{M}^*, w \models C^{\#}$. By the construction, we get $R(u,v,w)$. By the induction hypothesis, we get $\mathfrak{M}, v \models B$ and $\mathfrak{M}, w \models C$. Therefore, $\mathfrak{M}, u \models B \cdot C$.

Finally, by $(\mho)$, $\mathfrak{M}^*, k \models A^{\#}$ but $\mathfrak{M}^*, k \not\models B^{\#}$. Hence $A^{\#} \supset B^{\#}$ is refuted in $\mathfrak{M}^*$. So $\nvdash_{\mathrm{K}^{\mathrm{t}}_{1,2}} A^{\#} \supset B^{\#}$. ◻

It is easy to see that the reduction from BFNL to $\mathrm{K}^{\mathrm{t}}_{1,2}$ is in polynomial time. Then by the fact that $\mathrm{K}^{\mathrm{t}}_{1,2}$ is in PSPACE (Theorem 16), we get the following theorem:

Theorem 14. BFNL *is in PSPACE.*

Now we show that the validity problem for $\mathrm{K}^{\mathrm{t}}_{1,2}$ is in PSPACE. Our method is to show that $\mathrm{K}^{\mathrm{t}}_{1,2}$ is embedded into K.t in polynomial time.

Definition 4. *Let* $\mathsf{P} \subseteq \mathsf{Prop}$ *and* $x \notin \mathsf{P}$ *be a distinguished propositional variable. Define a translation* $(.)^* : \mathcal{L}_{\mathrm{K}^{\mathrm{t}}_{12}}(\mathsf{P}) \to \mathcal{L}_{\mathrm{K.t}}(\mathsf{P} \cup \{x\})$ *recursively as follows:*

$$\begin{aligned}
p^* &= p, & \bot^* &= \bot,\\
(\Diamond_1 A)^* &= \neg x \wedge \Diamond(\neg x \wedge A^*), & (\Diamond_2 A)^* &= \Diamond(x \wedge \Diamond A^*),\\
(\Box_1^{\downarrow} A)^* &= \neg x \supset \Box^{\downarrow}(\neg x \supset A^*), & (\Box_2^{\downarrow} A)^* &= \Box^{\downarrow}(x \supset \Box^{\downarrow} A^*),\\
(A \supset B)^* &= A^* \supset B^*.
\end{aligned}$$

Then we have

$$\begin{aligned}
(\Diamond_1^{\downarrow} A)^* &= \neg x \wedge \Diamond^{\downarrow}(\neg x \wedge A^*), & (\Diamond_2^{\downarrow} A)^* &= \Diamond^{\downarrow}(x \wedge \Diamond^{\downarrow} A^*),\\
(\Box_1 A)^* &= \neg x \supset \Box(\neg x \supset A^*), & (\Box_2 A)^* &= \Box(x \supset \Box A^*).
\end{aligned}$$

Theorem 15. *For any bi-tense formula* A, $\vdash_{\mathrm{K}^{\mathrm{t}}_{1,2}} A$ *if and only if* $\vdash_{\mathrm{K.t}} A^*$.

Proof. (i) By induction on the proof of A in $\mathrm{K}^{\mathrm{t}}_{1,2}$, we show that $\vdash_{\mathrm{K}^{\mathrm{t}}_{1,2}} A$ implies $\vdash_{\mathrm{K.t}} A^*$. We show only that the translations of axioms and rules of $\mathrm{K}^{\mathrm{t}}_{1,2}$ hold in K.t. The cases for propositional tautologies and (MP) are obvious. Let us consider other cases.

Case 1. The translation of $\Box_1(A \supset B) \supset (\Box_1 A \supset \Box_1 B)$ is $(\neg x \supset \Box(\neg x \supset (A^* \supset B^*))) \supset ((\neg x \supset \Box(\neg x \supset A^*)) \supset (\neg x \supset \Box(\neg x \supset B^*)))$. First, $\vdash_{\mathrm{K.t}} (\neg x \supset (A^* \supset B^*)) \supset ((\neg x \supset A^*) \supset (\neg x \supset B^*))$ by Fact 5 (4). Then $\vdash_{\mathrm{K.t}} \Box(\neg x \supset (A^* \supset B^*)) \supset (\Box(\neg x \supset A^*) \supset \Box(\neg x \supset B^*))$ by Fact 5 (5) and distributivity of $\Box$ over implications. Then by Fact 5 (1) and (4), we get the required theorem in K.t.

Case 2. The translation of $\Box_2(A \supset B) \supset (\Box_2 A \supset \Box_2 B)$ is $\Box(x \supset \Box(A^* \supset B^*)) \supset (\Box(x \supset \Box A^*) \supset \Box(x \supset \Box B^*))$. Since $\vdash_{\mathrm{K.t}} \Box(A^* \supset B^*) \supset (\Box A^* \supset \Box B^*)$, by Fact 5 (1), we get $\vdash_{\mathrm{K.t}} (x \supset \Box(A^* \supset B^*)) \supset (x \supset (\Box A^* \supset \Box B^*))$. Since $\vdash_{\mathrm{K.t}} (x \supset (\Box A^* \supset \Box B^*)) \supset ((x \supset \Box A^*) \supset (x \supset \Box B^*))$, we obtain $\vdash_{\mathrm{K.t}} (x \supset \Box(A^* \supset B^*)) \supset ((x \supset \Box A^*) \supset (x \supset \Box B^*))$ by Fact 5 (3). Hence by Fact 5 (5), we get $\vdash_{\mathrm{K.t}} \Box(x \supset \Box(A^* \supset B^*)) \supset \Box((x \supset \Box A^*) \supset (x \supset \Box B^*))$. Since $\vdash_{\mathrm{K.t}} \Box((x \supset \Box A^*) \supset (x \supset \Box B^*)) \supset (\Box(x \supset \Box A^*) \supset \Box(x \supset \Box B^*))$, by Fact 5 (3), we obtain $\vdash_{\mathrm{K.t}} \Box(x \supset \Box(A^* \supset B^*)) \supset (\Box(x \supset \Box A^*) \supset \Box(x \supset \Box B^*))$.

Case 3. For $\Box^{\downarrow}_1(A \supset B) \supset (\Box^{\downarrow}_1 A \supset \Box^{\downarrow}_1 B)$, the proof is similar to Case 1.

Case 4. For $\Box^{\downarrow}_2(A \supset B) \supset (\Box^{\downarrow}_2 A \supset \Box^{\downarrow}_2 B)$, the proof is similar to Case 2.

Case 5. The translation of $A \supset \Box^{\downarrow}_2 \Diamond_2 A$ is $A^* \supset \Box^{\downarrow}(x \supset \Box^{\downarrow}\Diamond(x \wedge \Diamond A^*))$. Since $\vdash_{\mathrm{K.t}} \Diamond(x \wedge \Diamond A^*) \supset \Diamond(x \wedge \Diamond A^*)$, by Fact 5 (7), we get $(x \wedge \Diamond A^*) \supset \Box^{\downarrow}\Diamond(x \wedge \Diamond A^*)$, Then by Fact 5 (2), we obtain $\vdash_{\mathrm{K.t}} \Diamond A^* \supset (x \supset \Box^{\downarrow}\Diamond(x \wedge \Diamond A^*))$. Finally by Fact 5 (7), we get $\vdash_{\mathrm{K.t}} A^* \supset \Box^{\downarrow}(x \supset \Box^{\downarrow}\Diamond(x \wedge \Diamond A^*))$.

Case 6. For $A \supset \Box_2 \Diamond^{\downarrow}_2 A$, the proof is quite similar to Case 5.

Case 7. Let us consider the translation of $A \supset \Box_1 \Diamond^{\downarrow}_1 A$, which is $A^* \supset (\neg x \supset \Box(\neg x \supset (\neg x \wedge \Diamond^{\downarrow}(\neg x \wedge A^*))))$. Since $(\neg x \wedge \Diamond^{\downarrow}(\neg x \wedge A^*)) \supset (\neg x \wedge \Diamond^{\downarrow}(\neg x \wedge A^*))$ is a propositional tautology, $\vdash_{\mathrm{K.t}} \Diamond^{\downarrow}(\neg x \wedge A^*) \supset (\neg x \supset (\neg x \wedge \Diamond^{\downarrow}(\neg x \wedge A^*)))$ by Fact 5 (2). Hence by Fact 5 (8), one obtains $\vdash_{\mathrm{K.t}} (\neg x \wedge A^*) \supset \Box(\neg x \supset (\neg x \wedge \Diamond^{\downarrow}(\neg x \wedge A^*)))$. Finally by Fact 5 (2), one obtains $\vdash_{\mathrm{K.t}} A^* \supset (\neg x \supset \Box(\neg x \supset (\neg x \wedge \Diamond^{\downarrow}(\neg x \wedge A^*))))$.

Case 8. For the axiom $A \supset \Box^{\downarrow}_1 \Diamond_1 A$, the proof is quite similar to Case 7.

Case 9. For Nec$\Box_1$, assume that $\vdash_{\mathrm{K.t}} A^*$. Then by MP and propositional tautology $A^* \supset (\neg x \supset A^*)$, we get $\vdash_{\mathrm{K.t}} \neg x \supset A^*$. By Nec$\Box$, we get $\vdash_{\mathrm{K.t}} \Box(\neg x \supset A^*)$. Again, we obtain $\vdash_{\mathrm{K.t}} \neg x \supset \Box(\neg x \supset A^*)$. The proofs for the cases of other Nec-rules are quite similar.

(ii) For the other direction, assume that $\nvdash_{\mathrm{K}^{\mathrm{t}}_{1,2}} A$. Let $\mathfrak{M} = (W, R_1, R_2, \sigma)$ be a bi-tense model and $k \in W$ such that $\mathfrak{M}, k \not\models A$. Let us construct a binary model $\mathfrak{M}' = (W', R', \sigma')$ satisfying the following conditions:

- $k \in W'$.
- if $R_1(u, v)$, then put $u, v \in W'$ and let $R'(u, v)$.
- if $R_2(u, v)$, then take a fresh state $w \notin W$, put u, v, w into W' and let $R'(u, w)$ and $R'(w, v)$.
- set for all $u \in W \cap W'$, $u \in \sigma'(p)$ iff $u \in \sigma(p)$, for each $p \in \mathsf{P}$.
- set for all $u \in W' \backslash W$, $u \in \sigma'(x)$.

We show that for any $u \in W \cap W'$ and bi-tense formula C,

$$(\mho)\mathfrak{M}, u \models C \text{ iff } \mathfrak{M}', u \models C^*.$$

Now by induction on the length of C. The cases of propositional variables and boolean connectives are easy. Let us consider other cases. In the following proofs and later on we often employ the following obvious facts: (1) if $u, v \in \sigma'(\neg x)$ and $R'(u, v)$, then $R_1(u, v)$; (2) for any $w, u, v \in W'$, if $w \in W' \backslash W$, $R'(u, w)$ and $R'(w, v)$, then $u, v \in W$ and $R_2(u, v)$.

Case 1. $C = \Diamond_1 B$. Assume that $\mathfrak{M}, u \models \Diamond_1 B$. Then $R_1(u, v)$ and $\mathfrak{M}, v \models B$ for some $v \in W$. By the induction hypothesis, we get $\mathfrak{M}', v \models B^*$. Since $u, v \in W \cap W'$, we get $\mathfrak{M}', u \models \neg x$ and $\mathfrak{M}', v \models \neg x$. Hence $\mathfrak{M}', v \models \neg x \wedge B^*$. By $R_1(u, v)$, we get $R'(u, v)$. Then $\mathfrak{M}', u \models \neg x \wedge \Diamond(\neg x \wedge B^*)$, i.e., $\mathfrak{M}', u \models (\Diamond_1 B)^*$. Conversely, assume that $\mathfrak{M}', u \models \neg x \wedge \Diamond(\neg x \wedge B^*)$. Then $\mathfrak{M}', u \models \neg x$, and there exists $v \in W'$ such that $R'(u, v)$ and $\mathfrak{M}', v \models \neg x$ and $\mathfrak{M}', v \models B^*$. Then $u, v \in W$ and $R_1(u, v)$. By the induction hypothesis, $\mathfrak{M}, v \models B$. Then $\mathfrak{M}, u \models \Diamond_1 B$.

Case 2. $C = \Box_1^{\downarrow} B$. Assume that $\mathfrak{M}', u \not\models (\Box_1^{\downarrow} B)^*$. By definition, we have $\mathfrak{M}', u \models \neg x \wedge \Diamond^{\downarrow}(\neg x \wedge \neg B^*)$. Hence $\mathfrak{M}', u \models \neg x$, and there exists $v \in W'$ such that $R'(v, u)$, $\mathfrak{M}', v \models \neg x$ and $\mathfrak{M}', v \models \neg B^*$. Then $u, v \in W$ and $R_1(v, u)$. By the induction hypothesis, $\mathfrak{M}, v \models \neg B$. Thus $\mathfrak{M}, u \not\models \Box_1^{\downarrow} B$. Conversely, assume that $\mathfrak{M}, u \not\models \Box_1^{\downarrow} B$. Then there exists $v \in W$ such that $R_1(v, u)$ and $\mathfrak{M}, v \models \neg B$. Since $u, v \in W \cap W'$, by the induction hypothesis and arguments for $\neg$, we get $\mathfrak{M}', v \models \neg B^*$. By the construction, $R'(v, u)$, $\mathfrak{M}', u \models \neg x$ and $\mathfrak{M}', v \models \neg x$. Hence $\mathfrak{M}', u \models \neg x \wedge \Diamond^{\downarrow}(\neg x \wedge \neg B^*)$. Hence $\mathfrak{M}', u \not\models \neg x \supset \Box^{\downarrow}(\neg x \supset B^*)$, i.e., $\mathfrak{M}', u \not\models (\Box_1^{\downarrow} B)^*$.

Case 3. $C = \Diamond_2 B$. Assume that $\mathfrak{M}, u \models \Diamond_2 B$. Then there exists $v \in W$ such that $R_2(u, v)$ and $\mathfrak{M}, v \models B$. By the induction hypothesis, $\mathfrak{M}', v \models B^*$. By the construction, there exists $w \in W' \setminus W$ such that $R'(u, w)$ and $R'(w, v)$. Then $\mathfrak{M}', w \models \Diamond B^*$ and $\mathfrak{M}', w \models x$. Hence $\mathfrak{M}, u \models \Diamond(x \wedge \Diamond B^*)$, i.e., $\mathfrak{M}, u \models (\Diamond_2 B)^*$. Conversely, assume that $\mathfrak{M}', u \models (\Diamond_2 B)^*$, i.e., $\mathfrak{M}', u \models \Diamond(x \wedge \Diamond B^*)$. Then there exist $v, w \in W'$ such that $R'(u, w)$, $R'(w, v)$, $\mathfrak{M}', w \models x$ and $\mathfrak{M}', v \models B^*$. So $w \in W' \backslash W$. By the construction, we get $u, v \in W \cap W'$ and $R_2(u, v)$. By the induction hypothesis, $\mathfrak{M}, v \models B$. Hence $\mathfrak{M}, u \models \Diamond_2 B$.

Case 4. $C = \Box_2^{\downarrow} B$. Assume that $\mathfrak{M}', u \not\models (\Box_2^{\downarrow} B)^*$, i.e., $\mathfrak{M}', u \models \Diamond^{\downarrow}(x \wedge \Diamond^{\downarrow} \neg B^*)$. There exist $w, v \in W'$ such that $R'(v, w)$, $R'(w, u)$, $\mathfrak{M}', w \models x$ and $\mathfrak{M}', v \models \neg B^*$. Then $w \in W' \backslash W$, $v \in W' \cap W$ and so by the construction, we have $R_2(v, u)$. By the induction hypothesis, $\mathfrak{M}, v \models \neg B$. So $\mathfrak{M}, u \not\models \Box_2^{\downarrow} B$. Conversely, assume that $\mathfrak{M}, u \not\models \Box_2^{\downarrow} B$. Then there exists $v \in W$ such that $R_2(v, u)$ and $\mathfrak{M}, v \not\models B$. By the construction, there exists $w \in W' \setminus W$ such that $R'(v, w)$, $R'(w, u)$ and $\mathfrak{M}', w \models x$. By the induction hypothesis, $\mathfrak{M}', v \not\models B^*$. So $\mathfrak{M}', v \models \neg B^*$ and $\mathfrak{M}', w \models x \wedge \Diamond^{\downarrow} \neg B^*$. Hence $\mathfrak{M}', u \models \Diamond^{\downarrow}(x \wedge \Diamond^{\downarrow} \neg B^*)$. Hence $\mathfrak{M}', u \not\models (\Box_2^{\downarrow} B)^*$.

The completes the proof of $(\mho)$. Since $\mathfrak{M}, k \not\models A$, by $(\mho)$, we have $\mathfrak{M}', k \not\models A^*$. Therefore $\not\vdash_{\mathrm{K.t}} A^*$. $\square$

It is known that the validity problem of K.t is in PSPACE ([6,7]). Since $\mathrm{K}^{t}_{1,2}$ is embedded into K.t in polynomial time, by Theorem 15 we get the following result:

Theorem 16. $\mathrm{K}^{t}_{1,2}$ *is in PSPACE.*

6 PSPACE-Completeness

Theorem 17. BFNL *is PSPACE-complete.*

Proof. By Theorem 12. □

Since the complexity of the equational theory of r-algebras is equal to the complexity of BFNL, we get the following theorem:

Theorem 18. *Eq*($\mathbb{RBA}$) *is PSPACE-complete*

One can also consider the complexity problem for the subclasses of r-algebras, e.g. the variety of all residuated distributive lattices $\mathbb{RDL}$. Since every residuated distributive lattice can be expanded to a r-algebra (see [3]), the equational theory of $\mathbb{RDL}$ is in PSPACE. The universal theory of $\mathbb{RDL}$ is known to be PSPACE-hard (see [4]). This result can also be proved by reducing the decision problem of the universal theory of $\mathbb{RDL}$ to the decision problem of the consequence relation of the sequent calculus for $\mathbb{RDL}$. However, this does not yield the PSPACE-hardness of the equational theory of $\mathbb{RDL}$. It would be an interesting question to determine whether the equational theory of $\mathbb{RDL}$ is PSPACE-hard or not.

Acknowledgements. Thanks are given to Prof. Wojciech Buszkowski (Poland) and anonymous reviewers for their helpful comments.

References

1. Burgess, J.P.: Basic tense logic. In: Handbook of Philosophical Logic. Synthese Library, vol. 165, pp. 83–133. Springer, Netherlands (1984)
2. Buszkowski, W.: Interpolation and FEP for logics of residuated algebras. Logic J. IGPL **19**(3), 437–454 (2011)
3. Buszkowski, W.: Multi-sorted residuation. In: Casadio, C., Coecke, B., Moortgat, M., Scott, P. (eds.) Categories and Types in Logic, Language, and Physics. LNCS, vol. 8222, pp. 136–155. Springer, Heidelberg (2014)
4. Buszkowski, W.: Some syntactic interpretations in different systems of full Lambek calculus. In: Ju, S., Liu, H., Ono, H. (eds.) Modality, Semantics and Interpretations, pp. 23–48. Springer, Heidelberg (2015)
5. Collinson, M., Mcdonald, K., Pym, D.: A substructural logic for layered graphs. J. Logic Comput. **24**(4), 953–998 (2014)
6. Goré, R.: Tableau methods for modal and temporal logics. In: D'Agostino, M., Gabbay, D.M., Hähnle, R., Posegga, J. (eds.) Handbook of Tableau Methods, pp. 297–396. Kulwer Academic Publishers, Dordrecht (1999)

7. Granko, K.: Temporal logics of computation. In: Proceeding of the 12th European Summer School in Logic, Language and Information (2000)
8. Jipsen, P., Jónsson, B., Rafter, J.: Adjoining units to residuated Boolean algebras. Algebra Universalis **34**(1), 118–127 (1995)
9. Jónsson, B.: A survey of Boolean algebras with operators. In: Rosenberg, I.G., Sabidussi, G. (eds.) Algebras and Orders, pp. 239–286. Springer, The Netherlands (1993)
10. Jónsson, B., Tsinakis, C.: Relation algebras as residuated Boolean algebras. Algebra Universalis **30**(4), 469–478 (1993)
11. Jónsson, B., Tsinakis, C.: Relation algebras as residuated Boolean algebras. Algebra Universalis **30**, 469–478 (1993)
12. Jipsen, P.: Computer aided investigations of relation algebras. Ph.D. Dissertation, Vanderbilt University (1992)
13. Kaminski, M., Francez, N.: Relational semantics of the Lambek calculus extended with classical propositional logic. Stud. Logica. **102**(3), 479–497 (2014)
14. Kurtonina, N.: Frames and labels: a modal analysis of categorical inference. Ph.D. thesis, Universiteit Utrecht (1994)
15. Kurucz, Á., Németi, I., Sain, I., Simon, A.: Undecidable varieties of semilattice-ordered semigroups, of Boolean algebras with operators, and logics extending Lambek calculus. Logic J. IGPL **1**(1), 91–98 (1993)
16. Lambek, J.: On the calculus of syntactic types. In: Jakobson, R. (ed.) Structure of Language and its Mathematical Aspects, pp. 168–178. American Mathematical Society, Providence (1961)
17. Maddux, R.D.: Some varieties containing relation algebras. Trans. Am. Math. Assoc. **272**(2), 501–526 (1982)
18. Németi, I.: Decidability of relation algebras with weakened associativity. Proc. Amer. Math. Soc. **100**, 340–344 (1987)
19. Ladner, R.: The computational complexity of provability in systems of modal propositional logic. SIAM J. Comput. **6**(3), 467–480 (1977)
20. Tarski, A.: On the calculus of relations. J. Symbol. Logic **6**, 73–89 (1941)

Semantic Equivalence of Graph Polynomials Definable in Second Order Logic

Johann A. Makowsky[1(✉)] and Elena V. Ravve[2]

[1] Department of Computer Science,
Technion - Israel Institute of Technology, Haifa, Israel
janos@cs.technion.ac.il
[2] Department of Software Engineering, ORT-Braude College, Karmiel, Israel

Abstract. We study semantic equivalence of multivariate graph polynomials via their distinctive power introduced in (Makowsky, Ravve, Blanchard 2014) under the name of d.p.-equivalence. There we studied only univariate graph polynomials. In this paper we extend our study to multivariate graph polynomials. We use the characterization from the previous paper of d.p.-equivalence of two graph polynomials in terms of computing their respective coefficients. To make our graph polynomials combinatorially meaningful we require them to be definable in Second Order Logic SOL. The location of zeros in the multivariate case is captured by various versions of halfplane properties, also known as stability or Hurwitz stability. Our main application shows that every multivariate SOL-definable graph polynomial $P(G; X_1, X_2, ... X_k)$ is d.p.-equivalent to a substitution instance of a stable (Hurwitz stable) SOL-definable graph polynomial $Q(G; Y, X_1, X_2, ... X_k)$. In other words, two d.p.-equivalent SOL-definable multivariate graph polynomials can also have very different behavior concerning their halfplane properties.

1 Introduction

A graph $G = (V(G), E(G))$ is given by the set of vertices $V(G)$ and a symmetric edge-relation $E(G)$. We denote by $n(G)$ the number of vertices, by $m(G)$ the number of edges, by $k(G)$ the number of connected components of a graph G, and by $\mathcal{G}$ the class of finite graphs.

Graph polynomials are graph invariants with values in a polynomial ring $\mathcal{R}$, usually $\mathbb{Z}[\mathbf{X}]$ with $\mathbf{X} = (X_1, \ldots, X_\ell)$. Let $P(G; \mathbf{X})$ be a graph polynomial of the form

$$P(G; \mathbf{X}) = \sum_{i_1, \ldots, i_\ell = 0}^{d(G)} c_{i_1, \ldots, i_\ell}(G) X_1^{i_1} \cdot \ldots \cdot X_\ell^{i_\ell}$$

where $\mathbf{X} = (X_1, \ldots, X_\ell)$, $d(G)$ is a graph parameter with non-negative integers as its values, and

$$c_{i_1, \ldots, i_\ell}(G) : i_1, \ldots, i_\ell \leq d(G)$$

are integer valued graph parameters.

J.A. Makowsky—Partially supported by a grant of Technion Research Authority. Work done in part while the author was visiting the Simons Institute for the Theory of Computing in Spring 2016.

J. Väänänen et al. (Eds.): WoLLIC 2016, LNCS 9803, pp. 279–296, 2016.
DOI: 10.1007/978-3-662-52921-8_18

This paper is written for the logically minded and is a continuation of our analysis of notions used in the literature on graph polynomials. In particular, we were bothered by the question whether the location of the roots of univariate graph polynomials is a combinatorially meaningful statement about its underlying graph. The notion of "combinatorially meaningful" is made precise by our definition of a *semantic (aka graph theoretic) property of graph polynomials*. In [40] we have given our analysis of this question for the graph theory audience. Here we want to stress the *logical and foundational aspects* of this analysis, and extend the results of [40] to multivariate graph polynomials.

In the first part of this paper we justify, from a foundational point of view, the definitions we have introduced in [40]. This concerns the restrictions of graph polynomials to graph polynomials definable in Second Order Logic SOL, the various notions of equivalence of graph polynomials, the notions of syntactic and semantic properties of graph polynomials. We also paraphrase the main results of [40]. These results are all of the form:

> (*) Let U be a subset of the complex numbers, such as the reals, an open disk, the lower or upper halfplane, or the complement thereof. Given a univariate SOL-definable graph polynomial $P(G;X)$, there exists a semantically equivalent SOL-definable graph polynomial $Q(G;X)$ with all its roots in U.

They show, in a precise sense, that the location of the roots of a univariate graph polynomial is *not a semantic property*. They are more of a *normal form property*: Every univariate SOL-definable graph polynomial $P(G;X)$ can be put into a semantically equivalent form with prescribed location of its roots.

The proofs in [40] have two parts: Finding $Q(G;X)$, and showing that this $Q(G;X)$ is SOL-definable. Finding $Q(G;X)$ often uses some "dirty trick" from analysis, whereas showing SOL-definability, only sketched in [40], needs more efforts in the details.

In the second part of the paper we extend results of [40] to multivariate graph polynomials $P(G;\mathbf{X})$. We show that various versions of the"halfplane property" in higher dimensions of multivariate graph polynomials are also not semantic properties of the underlying graph in the sense of (*). This is interesting for two reasons: First, these halfplane properties were studied in the recent literature on graph polynomials, and, second, the proofs that the constructed $Q(G;\mathbf{X})$ is SOL-definable is much more complex. For space reasons we have to omit many examples already discussed in [40]. However, in this paper we provide details in proving SOL-definability for the more difficult case of multivariate graph polynomials and the various halfplane properties.

1.1 Why SOL-Definability

There are too many graph polynomials in the way we defined them above. We can impose more restrictions by imposing *computability* and *definability* requirements.

$P(G; \mathbf{X})$ is *computable* if both

(i) the coefficients of $P(G; \mathbf{X}) \in \mathbb{Z}[\mathbf{X}]$ are computable from G, and,
(ii) given a polynomial $s(\mathbf{X})$ it is decidable whether there is a graph G_s such that $P(G_s; \mathbf{X}) = s(\mathbf{X})$.

The second condition is needed to make Theorem 1 work.

Imposing complexity theoretic restrictions poses some serious problems, and is studied in [36]. However, it is not the subject of this paper. We only briefly discuss some problems involved in defining the complexity of graph polynomials in the conclusions.

It is more natural to impose definability restrictions. In [3,20,31,34] the class of graph polynomials definable in Second Order Logic, SOL, is studied, which requires that $d(G)$ and $c_\imath(G) = c(G; \imath)$, $\imath = (i_1, \ldots, i_\ell)$, are definable in SOL. With very few exceptions, the graph polynomials studied in the literature are SOL-definable[1]. For readers not familiar with SOL-definability have to refer the reader to [35] or [31].

Requiring that the graph polynomials are SOL-definable also garantees that their coefficients are the result of counting *combinatorially meaningful* SOL-definable configurations in the underlying graph.

1.2 Why Study Graph Polynomials?

The first graph polynomial, the chromatic polynomial, was introduced in 1912 by G. Birkhoff to study the Four Color Conjecture, [6]. The emergence of the Tutte polynomial can be seen as an attempt to generalize the chromatic polynomial, cf. [7,18,50]. The characteristic polynomial and the matching polynomial were introduced with applications from chemistry in mind, cf. [4,5,11,16,49]. Physicists study various partition functions in statistical mechanics, in percolation theory and in the study of phase transitions, cf. [42]. It turns out that many partition functions are incarnations of the Tutte polynomial. Another incarnation of the Tutte polynomial is the Jones polynomial in Knot Theory, cf. [29] and again [7]. The various incarnations of the Tutte polynomial have triggered an interest in other graph polynomials. These graph polynomials are studied for various reasons:

– Graph polynomials can be used to distinguish non-isomorphic graphs. A graph polynomial is complete if it distinguishes all non-isomorphic graphs. The quest for a complete graph polynomial which is also easy to compute failed so far for two reasons. Either there were too many non-isomorphic graphs which could not be distinguished, and/or the proposed graph polynomial was more difficult to compute than just checking graph isomorphism.

[1] Many are even definable in Monadic Second Order Logic MSOL, [35]. The exceptions are in [43]. The algorithmic advantages of MSOL-definability, [14] are of no importance in this paper.

– New graph polynomials may appear when we model behavior of physical, chemical or biological systems. The arguments whether a graph polynomial is interesting, depends on its success in predicting the behavior of the modeled systems. Also the particular choice of the representation is dictated by the modeling process. That the modeled process gives, in this case, rise to a particular graph polynomial, is secondary, and the properties of the graph polynomial reflect more properties of the physical or chemical process modeled, than properties of the underlying graph.
– New graph polynomials are also studied as part of graph theory proper. Here one is interested in the interrelationship between various graph parameters without particular applications in mind. A graph polynomial is considered interesting *from a graph theoretic point of view*, if many graph parameters can be (easily) derived from it.
– Graph polynomials are sometimes studied as a way of generating families of polynomials, irrespective of their graph theoretic meaning. H. Wilf, [56] e.g., asked the question how to characterize the polynomials which do occur as instances of chromatic polynomials of graphs as a family of polynomials. We have addressed this approach to graph polynomials in [39].

This paper deals only with the graph theoretic and logical aspects of graph polynomials, discarding the graph isomorphisms problem and discarding the modeling of systems describing phenomena in the natural sciences. It ultimately asks the question: When is a newly introduced graph polynomial interesting from a graph theoretic point of view and deserves to be studied, and what aspects are more rewarding in this study than others. In particular we scrutinize the role of the location of the roots of specific graph polynomials in terms of other graph theoretic properties.

1.3 Outline of the Paper

In Sect. 2 we discuss the foundational aspects of comparing graph polynomials, and introduce the notion of semantic properties of graph polynomials. In Sect. 3 we discuss properties of univariate graph polynomials related to the oocation or multiplicity of their roots. In Sect. 4 we look at the multivariate version of the location of roots, the various halfplane properties, also called stability properties, and prove that stability is also not a semantic property of multivariate graph polynomials. In Sect. 5 we draw our conclusions and formulate several open problems.

2 How to Compare Graph Polynomials?

Once the graph theorists started to study several graph polynomials, the need of comparing them naturally arises.

For $\mathcal{R} \in \{\mathbb{R}, \mathbb{C}, \mathbb{Z}\}$ we denote by $\mathfrak{GP}_{\mathcal{R},m}$ the set of graph polynomials in m indeterminates with coefficients in $\mathcal{R}$, and let $\mathbf{X} = (X_1, \ldots, X_m)$ be m indeterminates. Let $P(G) = P(G; \mathbf{X})$ and $Q(G) = Q(G; \mathbf{X})$ be two graph polynomials.

The following statements appear frequently with a *recognizable meaning*, but *without a general definition*:

(i) $Q(G)$ is *a substitution instance* of $P(G)$.
(ii) Q and P are really the same, up to a *prefactor*. For example the various versions of the Tutte polynomial are said to be the same *up to a prefactor*, [48], and the same holds for the various versions of the matching polynomial, [32].
(iii) Q is *at least as expressive* than P.
(iv) The coefficients of $P(G)$ can be determined, or even computed, from the coefficients of $Q(G)$.

Usually these statements are understood to be uniform in the graphs G, but this uniformity can take various forms. In [40] we have given these statements precise meanings, and we have initiated the analysis of their relationship.

2.1 Equivalence of Graph Polynomials

A graph G is P-unique if for all graphs G' the polynomial identity $P(G;\mathbf{X}) = P(G';\mathbf{X})$ implies that G is isomorphic to G'. As a graph invariant $P(G;\mathbf{X})$ can be used to check whether two graphs are not isomorphic. For P-unique graphs G and G' the polynomial $P(G;\mathbf{X})$ can also be used to check whether they are isomorphic. One usually compares graph polynomials by their *distinctive power*.

Definition 1. *Let* $P \in \mathfrak{GP}_{\mathcal{R},m_1}$ *and* $Q \in \mathfrak{GP}_{\mathcal{R},m_2}$.

(i) Two graphs G_1 *and* G_2 *are called* similar *if they have the same number of vertices, edges and connected components.*
(ii) A graph parameter or a graph polynomial is a similarity function *if it is invariant under graph similarity.*
(iii) P *is* d.p.-reducible to Q, *written as* $P(G;\mathbf{X}) \preceq_{d.p.} Q(G;\mathbf{Y})$, *if for every two similar graphs* G_1 *and* G_2 *with* $Q(G_1;\mathbf{X}) = Q(G_2;\mathbf{X})$ *we also have* $P(G_1;\mathbf{Y}) = P(G_2;\mathbf{Y})$.
(iv) $P(G;\mathbf{X})$ *is* prefactor reducible to $Q(G;\mathbf{Y})$, *written as* $P(G;\mathbf{X}) \preceq_{prefactor} Q(G;\mathbf{Y})$, *if there are similarity functions* $f(G;\mathbf{X})$ *and* $g_1(G;\mathbf{X}), \ldots, g_{m_2}(G;\mathbf{X})$ *such that* $P(G;\mathbf{X}) = f(G;\mathbf{X}) \cdot Q(G; g_1(G;\mathbf{X}), \ldots, g_{m_2}(G;\mathbf{X}))$.
(v) The graph polynomial $P(G;\mathbf{X})$ *is* substitutions reducible to $Q(G;\mathbf{Y})$, *written as* $P(G;\mathbf{X}) \preceq_{subst} Q(G;\mathbf{Y})$, *if* $f(G;\mathbf{X})$ *is the constant function with value* 1.
(vi) Two graph polynomials $P(G;\mathbf{X})$ *and* $Q(G;\mathbf{Y})$ *are* d.p.-equivalen (prefactor equivalent, substitution equivalent) *if the relationship holds in both directions.*

It follows that $P(G;\mathbf{X}) \preceq_{subst} Q(G;\mathbf{Y})$ implies $P(G;\mathbf{X}) \preceq_{prefactor} Q(G;\mathbf{Y})$, and $P(G;\mathbf{X}) \preceq_{prefactor} Q(G;\mathbf{Y})$ implies $P(G;\mathbf{X}) \preceq_{d.p.} Q(G;\mathbf{Y})$, etc.

Comments for Logicians: Our notion of similarity is extracted from the literature on graph polynomials: It is implicitly used frequently both in claims that two polynomials are "really the same", or "the same up to a prefactor". From a logical point of view one would rather define a more general notion: Let Σ be a finite set of graph parameters. Two graphs G, H are Σ-similar if they have the same values $s(G) = s(H)$ for all $s \in \Sigma$. It is easy, but currently of little use, to rewrite the definitions of various forms of equivalence of graph polynomials using Σ-similarity rather than similarity as we defined it in this paper.

Theorem 1. *Let* $P(G; \mathbf{X}) \in \mathfrak{GP}_{\mathcal{R},m_1}$ *and* $Q(G, \mathbf{Y}) \in\in \mathfrak{GP}_{\mathcal{R},m_2}$. *The following are equivalent:*

(i) P *is* d.p.-reducible to Q.
(ii) *There is a function*

$$F : \mathcal{R}[\mathbf{Y}] \to \mathcal{R}[\mathbf{X}]$$

such that for all graphs G *we have*

$$F(n(G), m(G), k(G), Q(G)) = P(G)$$

(iii) *If, furthermore both* $P(G; \mathbf{X})$ *and* $Q(G; \mathbf{Y})$ *are computable. then* F *can be made computable, too.*

The equivalence of (i) and (ii) was proved in [40] and (iii) follows from the definition of computability of graph polynomials. We note here that (ii) is useful for proving d.p.-reducibility, whereas (i) is more useful to prove its negation. Theorem 1 shows that our definition of d.p.-equivalence of graph polynomials is mathematically equivalent to the definition proposed in [41].

Comments for Logicians: The notion of d.p.-equivalence (having the same distinguishing power) of graph polynomials evolved very slowly, mostly in implicit arguments. Originally, a graph polynomial such as the chromatic or characteristic polynomial had a *unique* definition which both determined its algebraic presentation and its semantic content. The need to spell out semantic equivalence emerged when the various forms of the Tutte polynomial had to be compared. As was to be expected, some of the presentations of the Tutte polynomial had more convenient properties than others, and some of the properties of one form got completely lost when passing to another semantically equivalent form. Two d.p.-equivalent polynomials carry the same combinatorial information about the underlying graph, independently of their presentation as polynomials. This situation is analogous to the situation in Linear Algebra: Similar matrices represent the same linear operator under two different bases. The choice of a suitable basis, however, may be useful for numeric evaluations. Here d.p.-equivalent graph polynomials represent the same combinatorial information under two different polynomial representations. The choice of a particular polynomial representation $P(G; \mathbf{X})$ may carry more numeric information about a particular graph parameter $p(G)$ determined by $P(G; \mathbf{X})$.

2.2 Syntactic vs Semantic Properties of Graph Polynomials

An *n-ary property of graph polynomials* Φ, aka a *GP-property*, is a subset of $\mathfrak{GP}^n_{\mathcal{R},m}$. Φ is a *semantic property* if it is closed under d.p.-equivalence. Semantic properties are independent of the particular presentation of its members. Consequently, we call a property Φ, which does depend on the presentation of its members, a *syntactic (aka algebraic) property*. Let us make this definition clearer via examples:

Example 2. *(i) The GP-property which says that for every graph G the polynomial $P(G, \mathbf{X})$ is P-unique, is a semantic property.*

(ii) The unary GP-properties of univariate graph polynomials that for each graph G the polynomials $P(G; X)$ is monic[2]*, or that its coefficients are unimodal*[3]*, is not a semantic GP-property, because, by applying Theorem 1, multiplying each coefficient by a fixed integer gives a d.p.-equivalent graph polynomial.*

(iii) The GP-property that the multiplicity of a certain value a as a root of $P(G; X)$ coincides with the value of a graph parameter $p(G)$ with values in $\mathbb{N}$, is not a semantic property. For example, the multiplicity of 0 as a root of the Laplacian polynomial is the number of connected components $k(G)$ of G, [11, Chap. 1.3.7]. However, stating that for two graphs G_1, G_2 with $P(G_1; X) = P(G_2; X)$ we also have $p(G_1) = p(G_2)$, is a semantic property.

(iv) Similarly, proving that the leading coefficient of $P(G; \mathbf{X})$ equals the number of vertices of G is not a semantic property, for the same reason. However, proving that two graphs G_1, G_2 with $P(G_1; \mathbf{X}) = P(G_2; \mathbf{X})$ have the same number of vertices is semantically meaningful.

(v) In similar vain, the classical result of [21]*, that the characteristic polynomial of a forest equals the (acyclic) matching polynomial of the same forest, is a syntactic coincidence, or reflects a clever choice in the definition of the acyclic matching polynomial, but it is not a semantic GP-property. The semantic GP-property of this result says that if we restrict our graphs to forests, then the characteristic and the matching polynomials (in all its versions) have the same distinctive power on trees of the same size. We discuss this and similar examples further in* [40]*.*

Comments for Logicians: To prove a semantic GP-property it is sometimes easier to prove a stronger non-semantic version. From the above examples, (iii), (iv) and (v) are illustrative cases for this.

3 Roots of Graph Polynomials

The literature on graph polynomials mostly got its inspiration from the successes in studying the chromatic polynomial and its many generalizations and

[2] A univariate polynomial is monic if the leading coefficient equals 1.

[3] A sequence of numbers $a_i : i \leq m$ is unimodal if there is $k \leq m$ such that $a_i \leq a_j$ for $i < j < k$ and $a_i \geq a_j$ for $k \leq i < j \leq m$.

the characteristic polynomial of graphs. In both cases the roots of graph polynomials are given much attention and are meaningful when these polynomials model physical reality.

A complex number $z \in \mathbb{C}$ is a root of a univariate graph polynomial $P(G;X)$ if there is a graph G such that $P(G;z) = 0$. It is customary to study the location of the roots of univariate graph polynomials. Prominent examples, besides the chromatic polynomial, the matching polynomial and the characteristic polynomial and its Laplacian version, are the independence polynomial, the domination polynomial and the vertex cover polynomial.

For a fixed univariate graph polynomial $P(G;X)$ typical statements about roots are:

(i) For every G the roots of $P(G;X)$ are real. This is the univariate version of stability or Hurwitz stability for real polynomials. It is true for the characteristic and the matching polynomial, [16,32]. Similarly, for every claw-free graph G the roots of the independence polynomial are real, [33]. Incidentally, by a classical theorem of I. Newton, if all the roots of a polynomial with positive coefficients are real, then its coefficients are unimodal.
(ii) Assuming that all roots of $P(G;X)$ are real, the (second) largest root has an interesting combinatorial interpretation. This is true for the characteristic polynomial where the second largest eigenvalue is related to the Cheeger constant, [1,11, Chap. 4].
(iii) The multiplicity of a certain value a as a root of $P(G;X)$ has an interesting interpretation. For example, the multiplicity of 0 as a root of the Laplacian polynomial is the number of connected components of G, [11, Chap. 1.3.7].
(iv) For every G all real roots of $P(G;X)$ are positive (negative) or the only real root is 0. The real roots are positive in the case of the chromatic polynomial and the clique polynomial, and negative for the independence polynomial, [12,17,22,26,27].
(v) For every G the roots of $P(G;X)$ are contained in a disk of radius $\rho(d(G))$, where $d(G)$ is the maximal degree of the vertices of G. This is true for the characteristic polynomial and its Laplacian version, [11, Chap. 3]. This is also the case for the chromatic polynomial, [17,46], but the proof of this is far from trivial.
(vi) For every G the roots of $P(G;X)$ are contained in a disk of constant radius. This is the case for the edge-cover polynomial, [15]. For the unit disk this is the univariate version of Schur-stability.
(vii) The roots of $P(G;X)$ are dense in the complex plane. This is again true for the chromatic polynomial, the dominating polynomial and the independence polynomial, [12,17,27,47].

In [40] we showed that the precise location of roots of univariate SOL-definable graph polynomials is not a graph theoretic (semantic) property of graphs. In the next section we investigate whether stability, the multivariate analog the location of zeros, of multivariate SOL-definable graph polynomials is a semantic property.

4 Stable Graph Polynomials

Multivariate analogs of location of zeros of polynomials are the various halfplane properties aka stability properties.

4.1 Why are Stable Multivariate Polynomials Interesting?

A multivariate polynomial is *stable*[4] if the imaginary part of its zeros is negative, and it is *Hurwitz-stable* if the real part of its zeros is negative. Analogously, it is *Schur-stable* if all its roots are in the open unit ball. Recently, stable and Hurwitz-stable polynomials have attracted the attention of combinatorial research. In [13] the study of graph and matroid invariants and their various stability properties was initiated. The more recent paper [25] does the same for knot and link invariants. Due mainly to the recent work of J. Borcea and P. Brändén [8], see also [53], a very successful multivariate generalization of stability of polynomials has been developed. To quote from the abstract of [52]:

> Problems in many different areas of mathematics reduce to questions about the zeros of complex univariate and multivariate polynomials. Recently, several significant and seemingly unrelated results relevant to theoretical computer science have benefited from taking this route: they rely on showing, at some level, that a certain univariate or multivariate polynomial has no zeros in a region. This is achieved by inductively constructing the relevant polynomial via a sequence of operations which preserve the property of not having roots in the required region.

Further on, [52] gives the following applications of stable polynomials to theoretical computer science: A new proof of the van der Waerden conjecture about the permanent of doubly stochastic matrices, [23]; various applications to the traveling salesman problem, [44,51]; applications to the Lee-Yang theorem in statistical physics that shows the lack of phase transition in the Ising model, [45], and more. [9] discuss various sampling problems and show, among other things, that the generating polynomial of spanning trees of a graph is stable, see also [2].

[4] In engineering and stability theory, a square matrix A is called stable matrix (or sometimes Hurwitz matrix) if every eigenvalue of A has strictly negative real part. These matrices were first studied in the landmark paper [28] in 1895. The Hurwitz stability matrix plays a crucial part in control theory. A system is stable if its control matrix is a Hurwitz matrix. The negative real components of the eigenvalues of the matrix represent negative feedback. Similarly, a system is inherently unstable if any of the eigenvalues have positive real components, representing positive feedback. In the engineering literature, one also considers Schur-stable univariate polynomials, which are polynomials such that all their roots are in the open unit disk, see for example [55].

4.2 Stable Polynomials

Let $m, n \in \mathbb{N}$ be indices. Let $\mathbf{X} = (X_1, \ldots, X_n)$ and $\mathbf{Y} = (Y_1, \ldots, Y_m)$ be $n+m$ indeterminates and $f(\mathbf{X}, \mathbf{Y}) \in \mathbb{C}[\mathbf{X}, \mathbf{Y}]$. Let $\mathcal{H}_u = \{a \in \mathbb{C} : \Im(a) > 0\}$ and $\mathcal{H}_r = \{a \in \mathbb{C} : \Re(a) > 0\}$ be the upper, respectively right half-plane of $\mathbb{C}$.

Definitions 3. *(i) f is homogeneous if all its monomials have the same degree.*

(ii) f is multiaffine if each indeterminate occurs at most to the first power in f.

(iii) $f \in \mathbb{C}[\mathbf{X}, \mathbf{Y}]$ is stable *if $f \equiv 0$ or, whenever $\mathbf{a} \in \mathcal{H}_u^{n+m}$, then $f(\mathbf{a}) \neq 0$. If additionally $f(\mathbf{X}) \in \mathbb{R}[\mathbf{X}, \mathbf{Y}]$, it is real stable.*

(iv) f is Hurwitz-stable *if $f \equiv 0$ or, whenever $\mathbf{a} \in \mathcal{H}_r^{n+m}$, then $f(\mathbf{a}) \neq 0$.*

(v) f is stable with respect to $\mathbf{X}$ *if for every $\mathbf{b} \in \mathcal{H}^m$ either $f(\mathbf{X}, \mathbf{b}) \equiv 0$ or whenever $\mathbf{a} \in \mathcal{H}_u^n$ then $f(\mathbf{a}, \mathbf{b}) \neq 0$.*

(vi) Let $\mathcal{K}$ be class of finite graphs. A graph polynomial $P(G; \mathbf{X})$ is stable on $\mathcal{K}$ *if for every graph $G \in \mathcal{K}$ the polynomial $P(G; \mathbf{X}) \in \mathbb{C}[\mathbf{X}]$ is stable.*

Remark 1. If $f(\mathbf{X}, \mathbf{Y})$ is stable, it is stable with respect to $\mathbf{X}$, but not conversely.

Example 4. *(i) Univariate polynomials are stable iff they have only real roots.*

(ii) The characteristic polynomial P_{cc} and its Laplacian version P_L are stable because they have only real roots.

(iii) Let $\mathrm{Tree}(G; X) = \sum_{T \subseteq E(G)} \prod_{e \in T} X_e$*, be the tree polynomial, where T ranges over all trees of $G = (V(G), E(G))$.* $\mathrm{Tree}(G; X)$ *is Hurwitz-stable,* [13].

(iv) Let $G = (V(G), E(G))$ be a graph and let $\mathbf{X}_E = (X_e : e \in E(G))$ be commutative indeterminates. Let S be a family of subsets of $E(G)$, i.e., $S \subset \wp(E(G))$ and let $\mathrm{P}_S(G; \mathbf{X}_E) = \sum_{A \in S} \prod_{e \in A} X_e$. If S is the family of trees of $E(G)$ then $\mathrm{P}_S(G; \mathbf{X}_E)$ is a multivariate version of the tree polynomial, which is also Hurwitz-stable, cf. [48, Theorem 6.2].

(v) In [13, Question 1.3] it is asked for which S is the polynomial $\mathrm{P}_S(G; \mathbf{X}_E)$ Hurwitz-stable. Actually they ask the corresponding question for matroids

$$M = (E(M), S(M)).$$

(vi) In [25, Sect. 16] the stability of multivariate knot polynomials is studied.

4.3 Sufficient Conditions for Stability

The characteristic polynomial of a symmetric real matrix is stable. Stable polynomials are often determinant like in the following sense:

Theorem 5 (Criteria for Stability). *Let $\mathbf{X} = (X_1, \ldots, X_m)$ be indeterminates, and $\mathcal{X}$ be the diagonal matrix of n indeterminates with $(\mathcal{X})_{i,i} = X_i$.*

(i) [8, Proposition 2.4] For $i \in [m]$ let each A_i be a positive semi-definite Hermitian $(n \times n)$-matrix and let B be Hermitian. Then

$$f(\mathbf{X}) = \det(X_1 A_1 + \ldots + X_m A_m + B) \in \mathbb{R}[\mathbf{X}]$$

is stable.

(ii) *[24, Theorem 2.2] For* $m = 2$ *and* $f(X_1, X_2) \in \mathbb{R}[X_1, X_2]$ *we have* $f(X_1, X_2)$ *is stable iff there are Hermitian matrices* A_1, A_2, B *with* A_1, A_2 *positive semi-definite such that*

$$f(X_1, X_2) = \det(X_1 A_1 + X_2 A_2 + B).$$

(iii) *[10, after Theorem 4.2] If* A *is a Hermitian* $(m \times m)$ *matrix then the polynomials* $\det(\mathcal{X} + A)$ *and* $\det(\mathrm{I} + A \cdot \mathcal{X})$ *are real stable.*

Theorem 6 (Criteria for Hurwitz-stability)

(i) [54] *If* $f(\mathbf{X}) \in \mathbb{R}[\mathbf{X}]$ *is a real homogeneous then* $f(\mathbf{X})$ *is stable iff* $f(\mathbf{X})$ *is Hurwitz stable.*

(ii) *[13, Theorem 8.1] Let* A *be a complex* $(r \times m)$*-matrix,* A^* *be its Hermitian conjugate, then the polynomial in* m*-indeterminates*

$$Q(\mathbf{X}) = \det(A\mathcal{X}A^*)$$

is multiaffine, homogeneous and Hurwitz-stable.

(iii) *[10, after Theorem 4.2] If* B *is a skew-Hermitian* $(n \times n)$ *matrix then* $\det(\mathcal{X} + B)$ *and* $\det(\mathrm{I} + B \cdot \mathcal{X})$ *are Hurwitz stable.*

(iv) *[13, Theorem 10.2] Let* A *be a real* $(r \times m)$*-matrix with non-negative entries. Then the polynomial in* m*-indeterminates*

$$Q(\mathbf{X}) = \mathrm{per}(AX) = \sum_{S \subseteq [m], |S| = r} \mathrm{per}(A \mid_S) \prod_{i \in S} X_i$$

is Hurwitz-stable.

4.4 Making Graph Polynomials Stable

We first consider graph polynomials with a fixed number of indeterminates m.

Let $P(G; \mathbf{X})$ be a graph polynomial with integer coefficients and with SOL-definition

$$P(G; \mathbf{X}) = \sum_{\phi} \prod_{\psi_1} X_1 \cdot \ldots \cdot \prod_{\psi_m} X_m,$$

with coefficients $(c_{i_1, \ldots, i_m} : i_j \leq d(G), j \in [m])$

$$P(G; \mathbf{X}) = \sum_{i_1, \ldots, i_m} c_{i_1, \ldots, i_m} X_1^{i_1} X_2^{i_2} \ldots X_m^{i_m} \in \mathbb{N}[\mathbf{X}],$$

such that in each indeterminate the degree of $P(G, \mathbf{X})$ is less than $d(G)$. We put $M(G) = d(G)^m$ which serves as a bound on the number of relevant coefficients, some of which can be 0.

Theorem 7. *There is a stable graph polynomial* $Q^s(G; Y, \mathbf{X})$ *with integer coefficients such that*

(i) the coefficients of $Q^s(G)$ can be computed uniformly[5] *in polynomial time from the coefficients of $P(G)$;*
(ii) there is $a_0 \in \mathbb{N}$ such that $Q^s(G; a_0, \mathbf{X})$ is d.p.-equivalent to $P(G; \mathbf{X})$;
(iii) $Q^s(G; Y, \mathbf{X})$ is SOL*-definable and its* SOL*-definition can be computed uniformly in polynomial time from $\phi, \psi_1, \ldots, \psi_m$.*

Theorem 8. *If additionally, $P(G; \mathbf{X})$ has only non-negative coefficients, there is a Hurwitz-stable graph polynomial $Q^h(G; Y, \mathbf{X})$ with non-negative integer coefficients and one more indeterminate Y such that*

(i) The coefficients of $Q^h(G)$ can be computed uniformly in polynomial time from the coefficients of $P(G)$;
(ii) there is $\mathbf{a} \in \mathbb{N}^{M-n}$ such that $Q^h(G; \mathbf{a}, \mathbf{X})$ is d.p.-equivalent to $P(G; \mathbf{X})$;
(iii) $Q^h(G; Y, \mathbf{X})$ is SOL*-definable and its* SOL*-definition can be computed uniformly in polynomial time from $\phi, \psi_1, \ldots, \psi_m$.*

In [13,48] the authors also consider graph polynomials where the number of indeterminates depends on the graph $G = (V(G), E(G))$, as in Example 4(iv). We will not give the most general definition here, but restrict ourselves to the case the indeterminates X_e are labeled by the edges $E(G)$ of G. We put $m(G)$ to be the cardinality of $E(G)$.

Let $S(G; \mathbf{X})$ be a multiaffine graph polynomial with non-negative integer coefficients and with SOL-definition

$$S(G; \mathbf{X}) = \sum_{\phi(A)} \prod_{\psi_1(A,e)} X_e \cdot \ldots \cdot \prod_{\psi_m(A,e)} X_e,$$

and coefficients $(c_{i_1,\ldots,i_m} : i_j \in \{0,1\}, j \in [m(G)])$

$$S(G; \mathbf{X}) = \sum_{i_1,\ldots,i_m} c_{i_1,\ldots,i_m} X_1^{i_1} X_2^{i_2} \ldots X_m^{i_{m(G)}} \in \mathbb{N}[\mathbf{X}],$$

such that in each indeterminate the degree of $S(G, \mathbf{X})$ is less than $d(G)$. We put $M(G) = 2^{m(G)}$ which serves as a bound on the number of relevant coefficients, some of which can be 0. Let $\mathbf{X}_G = (X_e : e \in E(G))$.

Theorem 9. *There are graph polynomials $T^s(G; \mathbf{X}_G)$ and $T^h(G; \mathbf{X}_G)$ with non-negative integer coefficients such that*

(i) $T^s(G; \mathbf{X}_G)$ is stable and $T^h(G; \mathbf{X}_G)$ is Hurwitz-stable;
(ii) Both the coefficients of $T^s(G)$ and of $T^h(G)$ can be computed uniformly in polynomial time from the coefficients of $S(G)$;
(iii) Both $T^s(G; \mathbf{X}_G)$ and $T^h(G; \mathbf{X}_G)$ are d.p.-equivalent to $S(G; \mathbf{X}_G)$;
(iv) Both $T^s(G; \mathbf{X}_G)$ and $T^h(G; \mathbf{X}_G)$ are SOL*-definable and its* SOL*-definition can be computed uniformly in polynomial time from $\phi, \psi_1, \ldots, \psi_m$.*

[5] There is a polynomial time computable function $F : \mathbb{Z}[\mathbf{X}] \to \mathbb{Z}[Y, \mathbf{X}]$ such that for all graphs G we have $F(P(G; \mathbf{X})) = Q^s(G; Y, \mathbf{X})$.

4.5 Proofs

Proof of Theorem 7. We use Theorem 5(i). Let $\alpha : \mathbb{N}^m \to \mathbb{N}$ which maps $(i_1, \ldots i_m) \in \mathbb{N}^m$ into its position in the lexicographic order of $\mathbb{N}^m$. We relabel the coefficients of $P(G; \mathbf{X})$ such that $d_i = c_{i_1,\ldots,i_m}$ with $\alpha(i_1, \ldots, i_m) = i, i \in [M]$ and $M = d(G)^m$.

We put B to be the $(M \times M)$ diagonal matrix with $B_{i,i} = d_i \cdot Y_i$ and $A_1 = A_2 = \ldots = A_m$ to be the $(M \times M)$ identity matrix. The identity matrix is both Hermitian and positive semi-definite. Furthermore, $B \mid_{Y=a} = B(a)$ being a diagonal matrix, is Hermitian for every $a \in \mathbb{C}$. Hence,

$$Q^s_a(G; a, \mathbf{X}) = \det(B(a) + \sum_{i=1}^{M} X_i \cdot A_i) = \prod_{i=1}^{M}(d_i + \sum_{i=1}^{M} X_i)$$

is stable for every $a \in \mathbb{C}$.

We have to verify (i)–(iii).

(i) All the matrices can be computed in polynomial time in $\mathbb{Z}[Y, \mathbf{X}]$.

(ii) We use Theorem 1: $Q^s \preceq_{d.p} P$ follows from (i). We have to show that there is $a_0 \in \mathbb{N}$ with $P \preceq_{d.p} Q^s_{a_0}$. The function α can be easily inverted. To recover the coefficients of $P(G)$ from the coefficients of $Q^s(G)$, we note that

$$\sum_{i=0}^{M} d_i(G) \cdot Y^i$$

is the coefficient of $(\sum_{\ell=1}^{m} X_\ell)^{M-1}$ of $Q^s(G; Y, \mathbf{X})$. This can be computed in polynomial time from the coefficients of Q^s. To find a_0 we let $a_0 \in \mathbb{N}$ be bigger than $1 + 2 \cdot |d_i(G)|$, as $d_i(G)$ could be negative. Now $\sum_{i=0}^{M} d_i(G) \cdot a_0^i$ can be viewed is natural number written in base a_0, and the digits $d_i(G)$ can be uniquely determined.

(iii) To prove that $Q^s(G; Y, \mathbf{X})$ is SOL-definable we need a few lemmas from [19,30,31].

The first lemma is part of the definition of SOL-definability.

Lemma 1. *Finite sums and products of* SOL-*definable polynomials are* SOL-*definable.*

Lemma 2. *Let $G_< = (V(G), E(G), < (G))$ be a graph with an ordering $< (G)$ on the vertices. Let $Q(G; \mathbf{X})$ be a graph polynomial with non-negative integer coefficients and with* SOL-*definition*

$$Q(G; \mathbf{X}) = \sum_{A \subseteq V^r : \phi(A)} \prod_{\mathbf{v}_1 \in A : \psi_1(A, \mathbf{v}_1)} X_1 \cdot \ldots \cdot \prod_{\mathbf{v}_m \in A : \psi_1(A, \mathbf{v}_m)} X_m$$

with coefficients $(c_{i_1,\ldots,i_m} : i_j \leq d(G), j \in [m])$

$$Q(G; \mathbf{X}) = \sum_{i_1,\ldots,i_m} c_{i_1,\ldots,i_m} X_1^{i_1} X_2^{i_2} \ldots X_m^{i_m} \in \mathbb{N}[\mathbf{X}].$$

such that in each indeterminate the degree of $P(G, \mathbf{X})$ is less than $d(G)$. Let $s(G)$ be such that $|V(G)|^{s(G)} \geq d(G)$ and extend the ordering $<(G)$ to the lexicographic ordering of $|V(G)|^{s(G)}$. For $\mathbf{v} \in V(G)^{s(G)}$ we define $Init(G; \mathbf{v})$ to be the set of predecessors of $\mathbf{v}$ in this lexicographic ordering.

The coefficients $c_{i_1,\ldots,i_m}$ of $Q(G; \mathbf{X})$ are SOL*-definable by*

$$c(\mathbf{v}_1, \ldots, \mathbf{v}_m) = \sum_{A \subseteq V^r} 1$$

where A ranges over all subsets satisfying $\phi(A)$ and for each $\ell \in [m]$ the set $Init(G; \mathbf{v}_\ell)$ is of the same size as i_ℓ and as

$$\{\mathbf{w}_\ell \in V^r : (V(G), E(G), <(G), A, \mathbf{w}_\ell) \models \phi(A) \wedge \psi(A, \mathbf{w}_\ell)\}$$

Proof. We only have to note that the equicardinality requirement is expressible in SOL.

Lemma 3. *The polynomial*

$$Q_a^s(G; a, \mathbf{X}) = \prod_{i=1}^{M}(d_i + \sum_{i=1}^{M} X_i) = \prod_{\mathbf{v}_1,\ldots,\mathbf{v}_m}\left(c(\mathbf{v}_1, \ldots, \mathbf{v}_m) + \sum_{i=1}^{M} X_i\right)$$

is SOL*-definable.*

Proof of Theorem 8. Now all the coefficients of $P(G; \mathbf{X})$ are non-negative. We want to use Theorem 6(i) together with Theorem 5(i). We repeat the proof of Theorem 7 with the following changes: Let D be the diagonal $(M \times M)$-matrix of the coefficients, and Y a new indeterminate. Instead of $B(a)$ we use $D \cdot Y$ where Y is now a scalar. D is now a diagonal matrix with non-negative coefficients, so it is positive semi-definite. We put

$$Q(G; Y, \mathbf{X}) = \det(D \cdot Y + \sum_{i \in [m]} A_i \cdot X_i).$$

The resulting polynomial $Q(G; Y, \mathbf{X})$ is homogeneous and has integer coefficients. So we can apply Theorem 6(i) together with Theorem 5(i) to make to see that $Q(G; Y, \mathbf{X})$ is both stable and Hurwitz stable. In particular, for each $a \in \mathbb{N}$ $Q(G; a, \mathbf{X})$ is Hurwitz stable. To see that $Q(G; Y, \mathbf{X})$ is SOL-definable we again use $a \in \mathbb{N}$ large enough as in the proof of Theorem 8.

Proof of Theorem 9. The proof is the same as the proof of Theorem 8, where the number of indeterminates equals the number $m(G) = | E(G) |$.

5 Conclusion

5.1 Interpretation of Our Results

In [38,40] we initiated the study of semantic equivalence of univariate graph polynomials without focusing on definability or complexity. We showed there that the location of the roots are not a semantic property.

In this paper we have extended these studies to multivariate graph polynomials. We have also extended our framework threefold:

(i) We have imposed computability restriction on our framework. To have a workable framework it does not suffice that the coefficients of a graph polynomial have to be computable from the graph, but that one needs to require that the inverse problem be decidable as well. This additional requirement was not used in [36], where we were more concerned with complexity issues of evaluating graph polynomials.
(ii) We have restricted our discussion to SOL-definable graph polynomials. This means that the d.p.-equivalent polynomial with stability properties has to be SOL-definable as well. In the univariate cases discussed in [38,40] the additional definability requirement is not too difficult to be established. In the multivariate case, this is considerably more complicated.
(iii) We have studied stability and Hurwitz-stability (aka the half-plane property) of multivariate graph polynomials. We have chosen this topic, because various graph polynomials arising from modeling natural phenomena turn out to be stable or Hurwitz-stable. Our study shows that these stability properties do not really reflect properties of the underlying graphs proper, but are the result of extraneous requirements arising from the particular modeling process of the natural phenomena in question.

Our work shows that to justify the study of the location of the zeroes of a graph polynomial, the particular choice of the coefficients of the graph polynomial has to be taken into account. If the only purpose of the graph polynomial is to encode purely graph theoretic properties, the location of its zeroes is irrelevant.

Acknowledgment. The authors would like to thank Petter Brändén for guiding us to the literature of stable polynomials, and Jason Brown and four anonymous readers of an earlier version of this paper for valuable comments. Thanks also to Jingcheng Lin for pointing out the references [2,9]. We want to acknowledge that some of the definitions and examples were taken verbatim from our [40]. D.p-equivalence was first characterized in [37].

References

1. Alon, N., Milman, V.: λ 1, isoperimetric inequalities for graphs, and superconcentrators. J. Comb. Theory Ser. B **38**(1), 73–88 (1985)
2. Anari, N., Gharan, S.O., Rezaei, A.: Monte carlo markov chains for sampling strongly rayleigh distributions and determinantal point processes (2016). arXiv preprint arXiv:1602.05242
3. Averbouch, I., Godlin, B., Makowsky, J.A.: An extension of the bivariate chromatic polynomial. Eur. J. Comb. **31**(1), 1–17 (2010)
4. Balaban, A.T.: Solved and unsolved problems in chemical graph theory. quo vadis, graph theory? Ann. Discrete Math. **35**, 109–126 (1993)

5. Balaban, A.T.: Chemical graphs: looking back and glimpsing ahead. J. Chem. Inf. Comput. Sci. **35**, 339–350 (1995)
6. Birkhoff, G.D.: A determinant formula for the number of ways of coloring a map. Ann. Math. **14**, 42–46 (1912)
7. Bollobás, B.: Modern Graph Theory. Springer, New York (1998)
8. Borcea, J., Brändén, P., et al.: Applications of stable polynomials to mixed determinants: johnson's conjectures, unimodality, and symmetrized fischer products. Duke Math. J. **143**(2), 205–223 (2008)
9. Borcea, J., Brändén, P., Liggett, T.: Negative dependence and the geometry of polynomials. J. Am. Math. Soc. **22**(2), 521–567 (2009)
10. Brändén, P.: Polynomials with the half-plane property and matroid theory. Adv. Math. **216**(1), 302–320 (2007)
11. Brouwer, A.E., Haemers, W.H.: Spectra of Graphs. Universitext. Springer, New York (2012)
12. Brown, J.I., Hickman, C.A., Nowakowski, R.J.: On the location of roots of independence polynomials. J. Algebraic Comb. **19**(3), 273–282 (2004)
13. Choe, Y.B., Oxley, J.G., Sokal, A.D., Wagner, D.G.: Homogeneous multivariate polynomials with the half-plane property. Adv. Appl. Math. **32**(1), 88–187 (2004)
14. Courcelle, B., Makowsky, J.A., Rotics, U.: On the fixed parameter complexity of graph enumeration problems definable in monadic second order logic. Discrete Appl. Math. **108**(1–2), 23–52 (2001)
15. Csikvári, P., Oboudi, M.R.: On the roots of edge cover polynomials of graphs. Eur. J. Comb. **32**(8), 1407–1416 (2011)
16. Cvetković, D.M., Doob, M., Sachs, H.: Spectra of Graphs, 3rd edn. Johann Ambrosius Barth, Heidelberg (1995)
17. Dong, F.M., Koh, K.M., Teo, K.L., Polynomials, C.: Chromaticity of Graphs. World Scientific, Singapore (2005)
18. Ellis-Monaghan, J.A., Merino, C.: Graph polynomials and their applications i: the tutte polynomial. In: Dehmer, M. (ed.) Structural Analysis of Complex Networks, pp. 219–255. Springer, Heidelberg (2011)
19. Fischer, E., Kotek, T., Makowsky, J.A.: Application of logic to combinatorial sequences and their recurrence relations. In: Grohe, M., Makowsky, J.A. (eds.) Model Theoretic Methods in Finite Combinatorics (Contemporary Mathematics), vol. 558, pp. 1–42. American Mathematical Society, Providence (2011)
20. Godlin, B., Katz, E., Makowsky, J.A.: Graph polynomials: from recursive definitions to subset expansion formulas. J. Log. Comput. **22**(2), 237–265 (2012)
21. Godsil, C.D., Gutman, I.: On the theory of the matching polynomial. J. Graph Theory **5**, 137–144 (1981)
22. Goldwurm, M., Santini, M.: Clique polynomials have a unique root of smallest modulus. Inf. Process. Lett. **75**(3), 127–132 (2000)
23. Gurvits, L.: Hyperbolic polynomials approach to van der waerden/schrijver-valiant like conjectures: sharper bounds, simpler proofs and algorithmic applications. In: Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing, pp. 417–426. ACM (2006)
24. Helton, J.W., Vinnikov, V.: Linear matrix inequality representation of sets. Commun. Pure Appl. Math. **60**(5), 654–674 (2007)
25. Hirasawa, M., Murasugi, K.: Various stabilities of the alexander polynomials of knots and links (2013). arXiv preprint arXiv:1307.1578
26. Hoede, C., Li, X.: Clique polynomials and independent set polynomials of graphs. Discrete Math. **125**, 219–228 (1994)

27. Hoshino, R.: Independence polynomials of circulant graphs. Ph.D. thesis, Dalhousie University, Halifax, Nova Scotia (2007)
28. Hurwitz, A.: Ueber die bedingungen, unter welchen eine gleichung nur wurzeln mit negativen reellen theilen besitzt. Math. Ann. **46**(2), 273–284 (1895)
29. Jaeger, F.: Tutte polynomials and link polynomials. Proc. Am. Math. Soc. **103**, 647–654 (1988)
30. Kotek, T.: Definability of combinatorial functions: Ph.D. thesis, Technion - Israel Institute of Technology, Haifa, Israel, March 2012
31. Kotek, T., Makowsky, J.A., Zilber, B.: On counting generalized colorings. In: Grohe, M., Makowsky, J.A. (eds.) Model Theoretic Methods in Finite Combinatorics (Contemporary Mathematics), vol. 558, pp. 207–242. American Mathematical Society, Providence (2011)
32. Lovász, L., Plummer, M.D., Theory, M.: Matching Theory, Annals of Discrete Mathematics, vol. 29. North Holland Publishing, North Holland (1986)
33. Seymour, P., Chudnovsky, M.: The roots of the independence polynomial of a clawfree graph. J. Comb. Theory Ser. B **97**(3), 350–357 (2007)
34. Makowsky, J.A.: Algorithmic uses of the Feferman-Vaught theorem. Ann. Pure Appl. Log. **126**(1–3), 159–213 (2004)
35. Makowsky, J.A.: From a zoo to a zoology: towards a general theory of graph polynomials. Theory Comput. Syst. **43**, 542–562 (2008)
36. Makowsky, J.A., Kotek, T., Ravve, E.V.: A computational framework for the study of partition functions and graph polynomials. In: Proceedings of the 12th Asian Logic Conference 2011, pp. 210–230. World Scientific (2013)
37. Makowsky, J.A., Ravve, E.V.: Logical methods in combinatorics, Lecture 11. Course given in 2009 under the number 236605, Advanced Topics, Lecture Notes. http://www.cs.technion.ac.il/janos/
38. Makowsky, J.A., Ravve, E.V.: On the location of roots of graph polynomials. Electron. Notes Discrete Math. **43**, 201–206 (2013)
39. Makowsky, J.A., Ravve, E.V.: On sequences of polynomials arising from graph invariants, preprint (2016)
40. Makowsky, J.A., Ravve, E.V., Blanchard, N.K.: On the location of roots of graph polynomials. Eur. J. Comb. **41**, 1–19 (2014)
41. Merino, C., Noble, S.D.: The equivalence of two graph polynomials and a symmetric function. Comb. Probab. Comput. **18**(4), 601–615 (2009)
42. Nešetřil, J., Winkler, P. (eds.): Graphs, Morphisms and Statistical Physics. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 63. AMS, New York (2004)
43. Noble, S.D., Welsh, D.J.A.: A weighted graph polynomial from chromatic invariants of knots. Ann. Inst. Fourier Grenoble **49**, 1057–1087 (1999)
44. Pemantle, R.: Hyperbolicity and stable polynomials in combinatorics and probability (2012). arXiv preprint arXiv:1210.3231
45. Sinclair, A., Srivastava, P.: Lee-Yang theorems and the complexity of computing averages. In: Proceedings of the forty-fifth annual ACM symposium on Theory of computing, pp. 625–634. ACM (2013)
46. Sokal, A.D.: Bounds on the complex zeros of (di)chromatic polynomials and potts-model partition functions. Comb. Prob. Comput. **10**(1), 41–77 (2001)
47. Sokal, A.D.: Chromatic roots are dense in the whole complex plane. Comb. Prob. Comput. **13**(2), 221–261 (2004)
48. Sokal, A.D.: The multivariate Tutte polynomial (alias Potts model) for graphs and matroids. In: Survey in Combinatorics, 2005. London Mathematical Society Lecture Notes, vol. 327, pp. 173–226 (2005)

49. Trinajstić, N.: Chemical Graph Theory, 2nd edn. CRC Press, Boca Raton (1992)
50. Tutte, W.T.: A contribution to the theory of chromatic polynomials. Can. J. Math. **6**, 80–91 (1954)
51. Vishnoi, N.K.: A permanent approach to the traveling salesman problem. In: 2012 IEEE 53rd Annual Symposium on Foundations of Computer Science (FOCS), pp. 76–80. IEEE (2012)
52. Vishnoi, N.K.: Zeros of polynomials and their applications to theory: a primer, Preprint. Microsoft Research, Bangalore (2013)
53. Wagner, D.G.: Multivariate stable polynomials: theory and applications. Bull. Am. Math. Soc. **48**(1), 53–84 (2011)
54. Wagner, D.G., Wei, Y.: A criterion for the half-plane property. Discrete Math. **309**(6), 1385–1390 (2009)
55. Wang, K., Michel, A.N., Liu, D.: Necessary and sufficient conditions for the hurwitz and schur stability of interval matrices. IEEE Trans. Autom. Control **39**(6), 1251–1255 (1994)
56. Wilf, H.S.: Which polynomials are chromatic. In: Proceedings of Colloquium Combinatorial Theory, Rome (1973)

Sheaves of Metric Structures

Maicol A. Ochoa[1] and Andrés Villaveces[2(✉)]

[1] Department of Chemistry, 261A Cret Wing, University of Pennsylvania, 231 S. 34 Street, Philadelphia, PA 19104, USA
maicol@sas.upenn.edu

[2] Departamento de Matemáticas, Universidad Nacional de Colombia, Bogotá 111321, Colombia
avillavecesn@unal.edu.co

Abstract. We introduce *sheaves of metric structures* and develop their basic model theory. The metric sheaves defined here provide a way to construct new metric models on sheaves (a strong generalization of the ultraproduct construction), with the additional property of having the theory of the resulting model controlled by the topology of a given space. More specifically, a metric sheaf $\mathfrak{A}$ is defined on a topological space X such that each fiber is a metric model. A new model, the generic metric model, is obtained as the quotient space of the sheaf through an appropriate filter of open sets. Semantics in the generic model is completely controlled and understood by the forcing rules in the sheaf and Theorem 3. This work extends early constructions due to Comer [5] and Macintyre [12] and later developments due to Caicedo [3], to the context of continuous logic. We illustrate these concepts by studying the metric sheaf of the continuous cyclic flow on tori.

1 Introduction

Sheaves have played an interesting, albeit under-developed, role in model theory (see Macintyre [13] for an interesting discussion of this issue). They are both supports for cohomology constructions and systems of variable structures themselves. Their limits (also called "generic models") are both generalizations of ultraproducts and of models obtained via model theoretic forcing. Limits of sheaves of structures are in many ways the optimal combination of geometrical (topological) control of limiting processes (supports of cohomology theories) and more general topos theoretic approaches.

The logic of sheaves of structures has had a rather non-linear progression. It harks back, in implicit format, to the work of Grothendieck [9] in his "Kansas Paper" of 1958. Explicit developments of the internal logic on topoi, and of the connections between model theoretic forcing and various logics on sheaves were successively developed during the decade of 1970 by Carson [4],

A. Villaveces—The second author was partially supported by Colciencias (Departamento Administrativo de Ciencia, Tecnología e Innovación) for the research presented here.

J. Väänänen et al. (Eds.): WoLLIC 2016, LNCS 9803, pp. 297–315, 2016.
DOI: 10.1007/978-3-662-52921-8_19

Comer [5], Ellerman [6], Macintyre [12] and others. Specifically, Ellerman extracted a version of a "ultrastalk" theorem linking truth on a generic fiber with model theoretic forcing over open sets and Macintyre studied various model completions of theories of rings by means of specific constructions of sheaves of rings. In his 1995 paper, Caicedo [3] introduced a more general way to construct sheaves of models by generalizing the previous constructions to sheaves of first order structures over arbitrary topological spaces.

Our work is partially motivated by Caicedo's results [3]. For the sake of completeness, we will now briefly review a few crucial aspects of Caicedo's work. Given two topological spaces X and E, a sheaf over X is defined as the pair (E, p) where $p : E \to X$ is a local homeomorphism. For each $x \in X$, the fiber $E_x = p^{-1}(x)$ is the universe of a first order structure in a language $\mathcal{L}$. A section σ is a continuous function defined from an open set $U \subset X$ in E such that $\sigma \circ p$ is the identity map in U. As a consequence of these definitions, the image set $\mathrm{Im}(\sigma)$ is an open set in E, and sections are in one-to-one correspondance with their image sets. Thus the satisfaction relation on each fiber can be extended transversally along the sheaf, i.e. from fiber to fiber, by defining a forcing relation that describes a semantics in the same language $\mathcal{L}$ and where the variables can be interpreted in the family of sections. Another important property of this construction is that whenever a statement is forced in a fiber E_x, one can always find a neighborhood U of x, such that for every $y \in U$ the same statement is forced in E_y. In addition a new $\mathcal{L}$-structure (the generic model) is obtained as a quotient space and the satisfaction relation is determined by the forcing relation defined in the sheaf.

Sheaves of first order structures are generalizations of the previous constructions (and versions of a "Generic Model Theorem") due to Ellerman and Macintyre, in which Caicedo distilled a substratum of the logic of topoi that has several direct applications to classical Model Theory. While other constructions made extensive use of the theory of topoi, Caicedo's presentation is substantially simpler, as it does not make explicit use of elements of category theory. As sketched above, every model-theoretical concept in these sheaves can be understood by means of the topological properties of the sheaf and an appropriate forcing relation. Both the fibers and the structures of sections are endowed in a natural way with classical structures, but Caicedo's construction provides the most natural way of dealing with model theoretic forcing, and ends up linking forcing over points and open sets of the topological space with classical truth on the "generic" structure.

Further research in this area was undertaken by Caicedo [2], Forero [8], Montoya [14] and the second author of this article (sheaf-models of Set Theory and generalizing classical forcing over partially ordered sets to forcing over arbitrary topological spaces in constructing generic models - see [16]).

In this paper we present a construction that takes these ideas to the realm of continuous logic. In brief, we construct sheaves of metric structures as understood and studied in the model theory developed by Ben Yaacov et al. [1]. We believe that this work can lead to further progress in the interactions between Model Theory and Geometry, can also contribute to the model-theoretic study of dynamical systems and, in particular, the study of classical and quantum mechanical systems.

A contrasting approach to metric sheaves and their logic appears in [11]. There the author focuses on the continuous model theory analysis of *reduced products* and then derives a semantics for sheaves. The emphasis is placed on the analysis of the semantics obtained directly in the products, rather than on forcing or generic models as here.

For the sake of completeness, we briefly describe a few elements of continuous logic (following [1]).

Logical connectives in metric structures are continuous functions from $[0,1]^n$ to $[0,1]$ and the supremum and infimum play the role of quantifiers. Semantics differs from that in classical structures by the fact that the satisfaction relation is defined on $\mathcal{L}$-conditions rather than on $\mathcal{L}$-formulas, where $\mathcal{L}$ is a metric signature. If $\phi(x)$ and $\psi(y)$ are $\mathcal{L}$-formulas, expressions of the form $\phi(x) \leq \psi(y)$, $\phi(x) < \psi(y)$, $\phi(x) \geq \psi(y)$, $\phi(x) > \psi(y)$ are $\mathcal{L}$-conditions. In addition, if ϕ and ψ are sentences then we say that the condition is closed.

The set $\mathcal{F} = \{0, 1, x/2, \dot{-}\}$, where 0 and 1 are taken as constant functions, $x/2$ is the function taking half of its input and $\dot{-}$ is the truncated subtraction, is uniformly dense in the set of all connectives [1]. We may therefore restrict the set of connectives that we use in building formulas to the set $\mathcal{F}$. These constitute the set of $\mathcal{F}$-restricted formulas.

In Sect. 2 we define the sheaf of metric structures, introduce pointwise and local forcing on sections and show how to define a metric space in some families of sections. In Sect. 3 we show how to construct the metric generic model from a metric sheaf. We also show how the semantics of the generic model can be understood by the forcing relation and the topological properties of the base space of the sheaf. Finally, we illustrate some of our results by means of a simple example.

2 The Metric Sheaf and Forcing

Consider a topological space X. A sheafspace over X is a pair (E, p), where E is a topological space and p is a local homeomorphism from E into X. A section σ is a function from an open set U of X to E such that $p \circ \sigma = Id_U$. We say that the section is global if $U = X$. Sections are determined by their images, as p is their common continuous inverse function. Besides, images of sections form a basis for the topology of E. We will refer indistinctly to the image set of a section and the function itself.

In what follows we assume that a metric language $\mathcal{L}$ is given and we omit the prefix $\mathcal{L}$ when talking about $\mathcal{L}$-formulas, $\mathcal{L}$-conditions and others.

Definition 1 (Sheaf of metric structures). *Let X be a topological space. A sheaf of metric structures (or, for short, a "metric sheaf") $\mathfrak{A}$ over X consists of:*

1. *A sheafspace (E, p) over X.*
2. *For all x in X we associate a metric structure* $(\mathfrak{A}_x, d) = \Big(E_x, \{R_i^{(n_i)}\}_x, \{f_j^{(m_j)}\}_x, \{c_k\}_x, \Delta_{R_{i,x}}, \Delta_{f_{i,x}}, d, [0,1]\Big)$, *where E_x is the fiber $p^{-1}(x)$ over x, and the following conditions hold:*

(a) (E_x, d_x) *is a complete, bounded metric space of diameter* 1.
(b) *For all* i, $R_i^{\mathfrak{A}} = \bigcup_{x\in X} R_i^{\mathfrak{A}_x}$ *is a continuous function according to the topology of* $\bigcup_{x\in X} E_x^{n_j}$.
(c) *For all* j, *the function* $f_j^{\mathfrak{A}} = \bigcup_x f_j^{\mathfrak{A}_x} : \bigcup_x E_x^{m_j} \to \bigcup_x E_x$ *is a continuous function according to the topology of* $\bigcup_{x\in X} E_x^{m_j}$.
(d) *For all* k, *the function* $c_k^{\mathfrak{A}} : X \to E$, *given by* $c_k^{\mathfrak{A}}(x) = c_k^{\mathfrak{A}_x}$, *is a continuous global section.*
(e) *We define the premetric function* $d^{\mathfrak{A}}$ *by* $d^{\mathfrak{A}} = \bigcup_{x\in X} d_x : \bigcup_{x\in X} E_x^2 \to [0,1]$, *where* $d^{\mathfrak{A}}$ *is a continuous function according to the topology of* $\bigcup_{x\in X} E_x^2$.
(f) *For all* i, $\Delta_{R_i}^{\mathfrak{A}} = \inf_{x\in X}(\Delta_{R_i}^{\mathfrak{A}_x})$ *with the condition that* $\inf_{x\in X} \Delta_{R_i}^{\mathfrak{A}_x}(\varepsilon) > 0$ *for all* $\varepsilon > 0$.
(g) *For all* j, $\Delta_{f_j}^{\mathfrak{A}} = \inf_{x\in X}(\Delta_{f_i}^{\mathfrak{A}_x})$ *with the condition that* $\inf_{x\in X} \Delta_{f_i}^{\mathfrak{A}_x}(\varepsilon) > 0$ *for all* $\varepsilon > 0$.
(h) *The closed interval* $[0,1]$ *is a second sort and is provided with the usual metric.*

The space $\bigcup_x E_x^n$ has as open sets the image of sections given by $\langle \sigma_1, \ldots, \sigma_n\rangle = \sigma_1 \times \cdots \times \sigma_n \cap \bigcup_x E_x^n$. These are the sections of a sheaf over X with local homeomorphism p^* defined by $p^*\langle\sigma_1(x), \ldots, \sigma_n(x)\rangle = x$. We drop the symbol * from our notation when talking about this local homeomorphism but it must be clear that this local homeomorphism differs from the function p used in the definition of the topological sheaf.

The induced function $d^{\mathfrak{A}}$ is not necessarily a metric nor a pseudometric. Thus, we cannot expect the sheaf just defined to be a metric structure, in the sense of continuous logic. Indeed, we want to build the local semantics on the sheaf so that for a given sentence ϕ, if ϕ is true at some $x \in X$, then we can find a neighborhood U of x such that for every y in U, ϕ is also true. In order to accomplish this task, first note that semantics in continuous logic is not defined on formulas but on conditions. Since the truth of the condition "$\phi < \varepsilon$" for ε small can be thought as a good approximation to the notion of ϕ being true in a first order model, one may choose this as the condition to be forced in our metric sheaf. Therefore, for a given real number $\varepsilon \in (0,1)$, we consider conditions of the form $\phi < \varepsilon$ and $\phi > \varepsilon$. Our first result comes from investigating to what extent the truth in a fiber "spreads" onto the sheaf.

Lemma 1 (Truth continuity in restricted cases).

- *Let* ε *be a real number,* $x \in X$, ϕ *an* $\mathcal{L}$*-formula composed only of the logical metric connectives and perhaps the quantifier* $\inf$. *If* $\mathfrak{A}_x \models \phi(\sigma(x)) < \varepsilon$, *then there exists an open neighborhood* U *of* x, *such that for every* y *in* U, $\mathfrak{A}_y \models \phi(\sigma(y)) < \varepsilon$.
- *Let* ε *be a real number,* $x \in X$, ϕ *an* $\mathcal{L}$*-formula composed only of the logical metric connectives and perhaps the quantifier* $\sup$. *If* $\mathfrak{A}_x \models \phi(\sigma(x)) > \varepsilon$, *then there exists an open neighborhood* U *of* x, *such that for every* y *in* U, $\mathfrak{A}_y \models \phi(\sigma(y)) > \varepsilon$.

Proof. In atomic cases, use the fact that $d^{\mathfrak{A}}$ and $R^{\mathfrak{A}}$ are continuous with respect to the topology defined by sections. For logical connectives this is a simple consequence of the fact that every connective is a continuous function. Thus, a formula $\phi(x_1, ..., x_2)$ constructed inductively only from connectives and atomic formulas is a composition of continuous functions and therefore continuous. If $\mathfrak{A}_x \models \phi(\sigma_1(x), \ldots, \sigma_n(x)) < \varepsilon$, then $p\left(\langle \sigma_1, \ldots, \sigma_2 \rangle \cap \phi^{-1}[0, \varepsilon)\right)$ is an open set in X satisfying the condition that for all y in it $\mathfrak{A}_y \models \phi(\sigma_1(y), \ldots, \sigma_n(y)) < \varepsilon$.

In particular, the above lemma is true for $\mathcal{F}$-restricted sentences. We may consider a different proof by induction using density. This alternative approach provides a better setting to define the point-forcing relation on conditions.

Definition 2 (Point Forcing). *Given a metric sheaf* $\mathfrak{A}$ *over a topological space* X*, we define the relation* $\Vdash_x$ *on the set of all conditions of the form* $\phi < \varepsilon$ *and* $\phi > \varepsilon$ *(where* ϕ *is an* $\mathcal{L}$*-statement,* ε *is an arbitrary real number in* $(0, 1)$ *and* $x \in X$*). Furthermore, where in our definition* $\phi = \phi(v_1, \cdots, v_n)$ *has free variables, the forcing at* x *will depend on specifying* local sections $\sigma_1, \cdots, \sigma_n$ *of the sheaf defined on open sets around the point* x*. Where necessary (for atomic formulas and the quantifier stage) we will indicate this.*

Our definition is by induction on the complexity of $\mathcal{L}$*-statements, and given for every* $\varepsilon \in (0, 1)$ *simultaneously.*

Atomic formulas

- $\mathfrak{A} \Vdash_x d(\sigma_1, \sigma_2) < \varepsilon \iff d_x(\sigma_1(x), \sigma_2(x)) < \varepsilon$
- $\mathfrak{A} \Vdash_x R(\sigma_1, \ldots, \sigma_n) < \varepsilon \iff R^{\mathfrak{A}_x}(\sigma_1(x), \ldots, \sigma_n(x)) < \varepsilon$
- *similar to the previous two, but with* $>$ *instead of* $<$

Logical connectives

- $\mathfrak{A} \Vdash_x \max(\phi, \psi) < \varepsilon \iff \mathfrak{A} \Vdash_x \phi < \varepsilon$ *and* $\mathfrak{A} \Vdash_x \psi < \varepsilon$
- $\mathfrak{A} \Vdash_x \max(\phi, \psi) > \varepsilon \iff \mathfrak{A} \Vdash_x \phi > \varepsilon$ *or* $\mathfrak{A} \Vdash_x \psi > \varepsilon$
- $\mathfrak{A} \Vdash_x \min(\phi, \psi) < \varepsilon \iff \mathfrak{A} \Vdash_x \phi < \varepsilon$ *or* $\mathfrak{A} \Vdash_x \psi < \varepsilon$
- $\mathfrak{A} \Vdash_x \min(\phi, \psi) > \varepsilon \iff \mathfrak{A} \Vdash_x \phi > \varepsilon$ *and* $\mathfrak{A} \Vdash_x \psi > \varepsilon$
- $\mathfrak{A} \Vdash_x 1 \dot{-} \phi < \varepsilon \iff \mathfrak{A} \Vdash_x \phi > 1 - \varepsilon$
- $\mathfrak{A} \Vdash_x 1 \dot{-} \phi > \varepsilon \iff \mathfrak{A} \Vdash_x \phi < 1 - \varepsilon$
- $\mathfrak{A} \Vdash_x \phi \dot{-} \psi < \varepsilon \iff \mathfrak{A}_x \models \psi = 1$ *or* $\mathfrak{A}_x \models \psi = r$ *for some* $r \in (0, 1)$ *and one of the following holds:*
  (i) $\mathfrak{A} \Vdash_x \phi < r$
  ii) $\mathfrak{A} \nVdash_x \phi < r$ *and* $\mathfrak{A} \nVdash_x \phi > r$
  (iii) $\mathfrak{A} \Vdash_x \phi > r$ *and* $\mathfrak{A} \Vdash_x \phi < r + \delta$ *for some* $\delta \in (0, \varepsilon)$.
- $\mathfrak{A} \Vdash_x \phi \dot{-} \psi > \varepsilon \iff \mathfrak{A} \Vdash_x \phi > r + \varepsilon$ *with* r *such that* $\mathfrak{A}_x \models \psi = r$

Quantifiers

- $\mathfrak{A} \Vdash_x \inf_\sigma \phi(\sigma) < \varepsilon \iff$ *There exists a section* μ *such that* $\mathfrak{A} \Vdash_x \phi(\mu) < \varepsilon$.
- $\mathfrak{A} \Vdash_x \inf_\sigma \phi(\sigma) > \varepsilon \iff$ *There exist an open set* $U \ni x$ *and a real number* $\delta_x > 0$ *such that for every* $y \in U$ *and every section* μ *defined on* y*,* $\mathfrak{A} \Vdash_y \phi(\mu) > \varepsilon + \delta_x$

- $\mathfrak{A} \Vdash_x \sup_\sigma \phi(\sigma) < \epsilon \iff$ *There exist an open set* $U \ni x$ *and a real number* δ_x *such that for every* $y \in U$ *and every section* μ *defined on* y $\mathfrak{A} \Vdash_y \phi(\mu) < \varepsilon - \delta_x$.
- $\mathfrak{A} \Vdash_x \sup_\sigma \phi(\sigma) > \epsilon \iff$ *There exists a section* μ *defined on* x *such that* $\mathfrak{A} \Vdash_x \phi(\mu) > \varepsilon$

The above definition and the previous lemma lead to the equivalence between $\mathfrak{A} \Vdash_x \inf_\sigma(1 \dot{-} \phi) > 1 \dot{-} \varepsilon$ and $\mathfrak{A} \Vdash_x \sup_\sigma \phi < \varepsilon$. More importantly, we can state the truth continuity lemma for the forcing relation on sections as follows.

Lemma 2. *Let* $\phi(\sigma)$ *be an* $\mathcal{F}-$*restricted formula. Then*

1. $\mathfrak{A} \Vdash_x \phi(\sigma) < \varepsilon$ *iff there exists* U *open neighborhood of* x *in* X *such that* $\mathfrak{A} \Vdash_y \phi(\sigma) < \varepsilon$ *for all* $y \in U$.
2. $\mathfrak{A} \Vdash_x \phi(\sigma) > \varepsilon$ *iff there exists* U *open neighborhood of* x *in* X *such that* $\mathfrak{A} \Vdash_y \phi(\sigma) > \varepsilon$ *for all* $y \in U$.

We can also define the point-forcing relation for non-strict inequalities by

- $\mathfrak{A} \Vdash_x \phi \leq \varepsilon$ iff $\mathfrak{A} \nVdash_x \phi > \varepsilon$ and
- $\mathfrak{A} \Vdash_x \phi \geq \varepsilon$ iff $\mathfrak{A} \nVdash_x \phi < \varepsilon$,

for $\mathcal{F}-$restricted formulas. This definition allows us to show the following proposition.

Proposition 1. *Let* $0 < \varepsilon' < \varepsilon$ *be real numbers. Then*

1. *If* $\mathfrak{A} \Vdash_x \phi(\sigma) \leq \varepsilon'$ *then* $\mathfrak{A} \Vdash_x \phi(\sigma) < \varepsilon$.
2. *If* $\mathfrak{A} \Vdash_x \phi(\sigma) \geq \varepsilon$ *then* $\mathfrak{A} \Vdash_x \phi(\sigma) > \varepsilon'$.

Proof. By induction on the complexity of formulas.

The fact that sections may have different domains brings additional difficulties to the problem of defining a metric function with the triangle inequality holding for an arbitrary triple. However, we do not need to consider the whole set of sections of a sheaf but only those whose domain is in a filter of open sets (as will be evident in the construction of the "Metric Generic Model" below). One may consider a construction of such a metric by defining the ultraproduct and the ultralimit for an ultrafilter of open sets. However, the ultralimit may not be unique since E is not always a compact set in the topology defined by the set of sections. In fact, it would only be compact if each fiber was finite. Besides, it may not be the case that the ultraproduct is complete. Thus, we proceed in a different way by observing that a pseudometric can be defined for the set of sections with domain in a given filter.

Lemma 3. *Let* $\mathbb{F}$ *be a filter of open sets. For all sections* σ *and* μ *with domain in* $\mathbb{F}$, *let the family* $\mathbb{F}_{\sigma\mu} = \{U \cap \mathrm{dom}(\sigma) \cap \mathrm{dom}(\mu) | \mathrm{U} \in \mathbb{F}\}$. *Then the function*

$$\rho_{\mathbb{F}}(\sigma, \mu) = \inf_{U \in \mathbb{F}_{\sigma\mu}} \sup_{x \in U} d_x(\sigma(x), \mu(x))$$

is a pseudometric in the set of sections σ *such that* $\mathrm{dom}(\sigma) \in \mathbb{F}$.

Proof. We prove the triangle inequality. Let σ_1, σ_2 and σ_3 be sections with domains in $\mathbb{F}$, and let V be the intersection of their domains. Then it is true that

$$\sup_{x\in V} d_x(\sigma_1(x),\sigma_2(x)) \leq \sup_{x\in V} d_x(\sigma_1(x),\sigma_3(x)) + \sup_{x\in V} d_x(\sigma_3(x),\sigma_2(x)),$$

and since $\sup_{x\in A} f(x) \leq \sup_{x\in B} f(x)$ whenever $A \subset B$, we have

$$\begin{aligned}\inf_{W\in\mathbb{F}_{\sigma_1\sigma_2}} &\sup_{x\in W} d_x(\sigma_1(x),\sigma_2(x)) \leq \\ &\leq \inf_{W\in\mathbb{F}} \left(\sup_{x\in W} d_x(\sigma_1(x),\sigma_3(x)) + \sup_{x\in W} d_x(\sigma_3(x),\sigma_2(x))\right).\end{aligned}$$

Given $\varepsilon > 0$, there exist W' and W'' such that

$$\begin{aligned}\sup_{x\in W'} d_x(\sigma_1(x),\sigma_3(x)) &< \inf_{W\in\mathbb{F}_{\sigma_1,\sigma_3}} \sup_{x\in W} d_x(\sigma_1(x),\sigma_3(x)) + \varepsilon/2\\ \sup_{x\in W''} d_x(\sigma_3(x),\sigma_2(x)) &< \inf_{W\in\mathbb{F}_{\sigma_2,\sigma_3}} \sup_{x\in W} d_x(\sigma_3(x),\sigma_2(x)) + \varepsilon/2.\end{aligned}\tag{1}$$

Therefore,

$$\begin{aligned}\sup_{x\in W'\cap W''} d_x(\sigma_1(x),\sigma_3(x)) + \sup_{x\in W'\cap W''} d_x(\sigma_3(x),\sigma_2(x)) <\\ \inf_{W\in\mathbb{F}_{\sigma_1\sigma_2}} \sup_{x\in W} d_x(\sigma_1(x),\sigma_3(x)) + \inf_{W\in\mathbb{F}_{\sigma_2\sigma_3}} \sup_{x\in W} d_x(\sigma_3(x),\sigma_2(x)) + \varepsilon.\end{aligned}\tag{2}$$

Since $W' \cap W''$ is in $\mathbb{F}$ and ε was chosen arbitrarily, the triangle inequality holds for $\rho_{\mathbb{F}}(\sigma,\mu)$.

In the following, whenever we talk about a filter $\mathbb{F}$ in X we will be considering a filter of open sets. For any pair of sections σ, μ with domains in a filter, we define $\sigma \sim_{\mathbb{F}} \mu$ if and only if $\rho_{\mathbb{F}}(\sigma,\mu) = 0$. This is an equivalence relation, and the quotient space is therefore a metric space under $d_{\mathbb{F}}([\sigma],[\mu]) = \rho_{\mathbb{F}}(\sigma,\mu)$. The quotient space provided with the metric $d_{\mathbb{F}}$ is the metric space associated with the filter $\mathbb{F}$. If $\mathbb{F}$ is principal and the topology of the base space X is given by a metric, then the associated metric space of that filter is complete. In fact completeness is a trivial consequence of the fact that sections are continuous and bounded in the case of a σ-complete filter (if X is a metric space). However, principal filters are not interesting from the semantic point of view and σ-completeness might not hold for filters or even ultrafilters of open sets. The good news is that we can still guarantee completeness in certain kinds of ultrafilters.

Theorem 1. *Let $\mathfrak{A}$ be a sheaf of metric structures defined over a regular topological space X. Let $\mathbb{F}$ be an ultrafilter of regular open sets. Then, the induced metric structure in the quotient space $\mathfrak{A}[\mathbb{F}]$ is complete under the induced metric.*

In order to prove this theorem we need to state a few lemmas.

Lemma 4. *Let A and B be two regular open sets. If $A\setminus B \neq \emptyset$ then $\mathrm{int(A\setminus B)} \neq \emptyset$.*

Proof. If $x \in A \setminus B$ and $\mathrm{int(A \setminus B)} = \emptyset$, then $x \in \overline{B}$ and $A \subset \overline{B}$. Therefore $A \subset \mathrm{int(\overline{B})} = \mathrm{B}$ which is in contradiction to the initial hypothesis.

Lemma 5. *Let $\mathbb{F}$ be a filter and $\{\sigma_n\}$ be a Cauchy sequence of sections according to the pseudometric $\rho_{\mathbb{F}}$ with all of them defined in an open set U in $\mathbb{F}$. Then*

1. *There exists a limit function μ_∞ not necessarily continuous defined on U such that* $\lim_{n\to\infty} \rho_{\mathbb{F}}(\sigma_n, \mu_\infty) = 0$.
2. *If X is a regular topological space and* $\mathrm{int}(\mathrm{ran}(\mu_\infty)) \neq \emptyset$, *there exists an open set $V \subset U$, such that $\mu_\infty \restriction V$ is continuous.*

Proof. 1. This follows from the fact that $\{\sigma_n(x)\}$ is a Cauchy sequence in the complete metric space (E_x, d_x). Then let $\mu_\infty(x) = \lim_{n\to\infty} \sigma_n(x)$ for each $x \in U$.
2. Consider the set of points e in μ_∞ such that there exists a section η defined in some open neighborhood U of x, with $\eta(x) = e$ and $\eta \subset \sigma_\infty$. Let V be the projection set in X of that set of points e. This is an open subset of U and $\mu_\infty \restriction V$ is a section.

We can now prove Theorem 1.

Proof. Let $\{[\sigma^m] | m \in \omega\}$ be a Cauchy sequence in the associated metric space of an ultrafilter of regular open sets $\mathbb{F}$. If the limit exists, it is unique and the same for every subsequence. Thus, we define the subsequence $\{[\mu^k] | k \in \omega\}$ by making $[\mu^k]$ equal to $[\sigma^m]$ for the minimum m such that for all $n \geq m$, $d_{\mathbb{F}}([\sigma^m], [\sigma^n]) < k^{-1}$. Since $d_{\mathbb{F}}([\mu^k], [\mu^{k+1}]) < k^{-1}$, for every pair $(k, k+1)$, there exists an open set $U_k \in \mathbb{F}$, such that

$$\sup_{x \in U_k} d_x(\mu^k(x), \mu^{k+1}(x)) < k^{-1}.$$

Let $W_1 = U_1$, $W_m = \cap_{i=1}^m U_k$ and define a function μ_∞ on W_1 as follows.

– If $x \in W_k \setminus W_{k+1}$ for some k, let $\mu_\infty(x) = \mu^k(x)$.
– Otherwise, if $x \in W_k$ for all k, we can take $\mu_\infty(x) = \lim_{k\to\infty} \mu^k(x)$.

The function μ_∞ might not be a section but, based on the above construction, one can find a suitable restriction σ_∞ that is indeed a section but defined on a smaller domain. We show this by analyzing different cases.

1. If $W_1 = W_k$ for all k, $\bigcap_k W_k = W_1$ then for all x in W_1, $\sigma_\infty(x) = \lim_{k\to\infty} \mu^k(x)$.
   (a) Suppose $\mathrm{int}(\mu_\infty) = \emptyset$. Let $\tilde{B}_1 = W_1$. For every x in $\tilde{B}_1$ choose a section η_x, such that $\eta_x(x) = \mu_\infty(x)$; by the continuity of $d^{\mathfrak{A}}$ in the sheaf of structures, the set $\tilde{B}_k = p(\langle \eta_x, \mu^k \rangle \cap (d^{\mathfrak{A}})^{-1}[0, k^{-1}))$ for $k \geq 2$ is an open neighborhood of x. Consider $\bigcap_{k\in\omega} \tilde{B}_k$. It is clear that this set is not empty and that $\mathrm{int}(\bigcap_{\mathrm{k}\in\omega} \tilde{\mathrm{B}}_\mathrm{k}) = \emptyset$, as we assumed that $\mathrm{int}(\mu_\infty) = \emptyset$. Since the base space is regular, there exists a local basis on x consisting of regular open sets. We can define a family $\{B_k\}$ of open regular sets so that

– $B_1 := \tilde{B}_1$
– $B_k \subset \tilde{B}_k$
– $B_{k+1} \subset B_k$
– $x \in B_k$

Let $C_1 := B_1 = W_1$. For all $k \geq 1$, define $C_{k+1} \subset C_k \cap B_{k+1}$ with the condition that C_{k+1} is a regular open set and let $V_k \subset C_k \setminus C_{k+1}$ be some regular open set such that

$$\overline{V}_k \ \cap \ \overline{C_k \setminus C_{k+1}} \setminus \mathrm{int}(\mathrm{C_k} \setminus \mathrm{C_{k+1}}) = \emptyset, \tag{3}$$

(if $C_{k+1} \subsetneq C_k$, this is possible by Lemma 4; if $C_{k+1} = C_k$ let $V_k = \emptyset$) - i.e., the closure of V_k does not contain any point in the boundary of $C_k \setminus C_{k+1}$ (Use Lemma 4 and the fact that X is regular). Then $\bigcap_{k\in\omega} C_k \supset \{x\}$. Now, if necessary, we renumber the family V_k so that all the empty choices of V_k are removed from this listing. Let $\Gamma = \Gamma_{odd} := \bigcup_{k=1}^{\infty} V_{2k-1}$ and observe that this is an open regular set:

$$\begin{aligned} \overline{\Gamma} &= \overline{\bigcup_{k\in\omega} V_{2k-1}} = \bigcup_{k\in\omega} \overline{V_{2k-1}} \\ \mathrm{int}\big(\overline{\Gamma}\big) &= \mathrm{int}\Big(\bigcup_{k\in\omega} \overline{V_{2k-1}}\Big) = \bigcup_{k\in\omega} \mathrm{int}(\overline{V_{2k-1}}) = \Gamma. \end{aligned} \tag{4}$$

For the first equality observe that if $z \in \overline{\bigcup_{k\in\omega} V_{2k-1}}$ then $z \in \overline{V}_{2l-1}$ for only one l since $\overline{V_n} \cap \overline{V_m} = \emptyset$ for $m \neq n$. In the second line, if $z \in \mathrm{int}\big(\bigcup_{k\in\omega} \overline{V_{2k-1}}\big)$, then every open set containing z is a subset of $\bigcup_{k\in\omega} \overline{V_{2k-1}}$ and again since $\overline{V_n} \cap \overline{V_m} = \emptyset$ for $m \neq n$, there exists at least an open neighborhood that is a proper subset of a unique $\overline{V_{2l-1}}$.

If Γ is an element of $\mathbb{F}$, then we can define the section σ_∞ in the open regular set Γ by $\sigma_\infty \restriction V_{2k-1} := \mu^{2k-1} \restriction V_{2k-1}$. This is a limit section of the original Cauchy sequence.

Now, consider the family $\mathbb{G}$ of all Γs that can be defined as above for the same element x in W_1 and for the same family $\{C_k\}$. This family includes the alternative choice $\Gamma_{even} := \bigcup_{k=1}^{\infty} V_{2k}$. $\mathbb{G}$ is partially ordered by inclusion. Consider a chain $\{\Gamma_i\}$ in $\mathbb{G}$. Observe that $\bigcup_i \Gamma_i$ is an upper bound for this and that $\bigcup_i \Gamma_i$ is regular, since

$$\begin{gathered} \bigcup_i \Gamma_i \subset \bigcup_i \mathrm{int}\Big(\overline{\mathrm{C_{2i-1}} \setminus \mathrm{C_{2i}}}\Big) \\ \overline{\mathrm{C}_{2i-1} \setminus \mathrm{C}_{2i}} \ \cap \ \overline{\mathrm{C}_{2i+1} \setminus \mathrm{C}_{2i+2}} = \emptyset \end{gathered} \tag{5}$$

Thus, by Zorn Lemma there is a maximal element Γ_{max}. This intersects every element in the ultrafilter and therefore is an element of the same. Otherwise we would be able to construct $\tilde{\Gamma}_{max} \supsetneq \Gamma_{max}$. Suppose that there exists $A \in \mathbb{F}$ such that $\Gamma_{max} \cap A = \emptyset$, then we could repeat the above arguments taking an element x' in $W_1' := A \cap W_1 = A \cap C_1$, finding $\Gamma' \subset W_1'$ with $\Gamma' \cap \Gamma_{max} = \emptyset$. Take $\tilde{\Gamma}_{max} = \Gamma' \cup \Gamma_{max}$.

(b) If $\mathrm{int}(\mu_\infty) \neq \emptyset$, then $U = p(\mathrm{int}(\mu_\infty))$ is an open subset of W_1. Observe that $W_1 \setminus U$ is an open set that contains all possible points of discontinuity of μ_∞. If some regular open set $V \subset U$ is an element of the ultrafilter, then $\mu_\infty \restriction V$ is a limit section. If that is not the case, $V = \mathrm{int(X \setminus U)}$ is in the ultrafilter and $V \cap W_1$ is an open regular set where μ_∞ is discontinuous at every point. Proceed as in case 1a.

2. If there exists N such that for all $m > N$ $W_N = W_m$, we rephrase the arguments used in 1a, this time defining σ_∞ in a subset of W_N. Also, if $\bigcap_{k\in\omega} W_k$ is open and nonempty, we follow the same arguments as in 1b.
3. If for all k $W_{k+1} \neq W_k$ and $\mathrm{int}(\bigcap_{\mathrm{k}\in\omega} \mathrm{W_k}) \neq \emptyset$, let $W_1' = \mathrm{int}(\bigcap_{\mathrm{k}\in\omega} \mathrm{W_k})$ and use the same line of argument as in cases 1a and 1b.
4. If $\mathrm{int}(\bigcap_{\mathrm{k}\in\omega} \mathrm{W_k}) = \emptyset$, for all k such that $W_k \setminus W_{k+1} \neq \emptyset$ define σ_∞ on some open regular set $\overline{V}_k \subset \mathrm{int(W_k \setminus W_{k+1})}$ so that $\sigma_\infty \restriction V_k = \mu^k \restriction V_k$. Then, σ_∞ is defined in $\bigcup_{k\in\omega} V_k$, and repeat the line of argument used in case 1a.

Note that for all μ^k

- If $\sigma_\infty(x) = \mu^{k+n}(x)$, then $d_x(\sigma_\infty(x), \mu^k(x)) < k^{-1}$ for x in the common domain.
- If $\sigma_\infty(x) = \lim_{n\in\omega} \mu^n(x)$, then there exists N such that for $m > N$ $d_x(\sigma_\infty(x), \mu^m(x)) < k^{-1}$ and taking $m > k$, by the triangle inequality $d_x(\sigma_\infty(x), \mu^k(x)) < 2k^{-1}$.

This shows that

$$\sup_{x\in W_k \cap \bigcup V_n} d_x(\sigma_\infty(x), \mu^k(x)) < 2(k-1)^{-1} \tag{6}$$

and then σ_∞ is a limit section. Finally, check that $p \circ \sigma^\infty = Id_{\mathrm{dom}(\sigma_\infty)}$.

Before studying the semantics of the quotient space of a generic filter, we define the relation $\Vdash_U$ of local forcing in an open set U for a sheaf of metric structures. The definition is intended to make the following statements about local and point forcing valid

$$\begin{aligned} \mathfrak{A} \Vdash_U \phi(\sigma) < \varepsilon &\iff \forall x \in U\ \mathfrak{A} \Vdash_x \phi(\sigma) < \delta \text{ and} \\ \mathfrak{A} \Vdash_U \phi(\sigma) > \delta &\iff \forall x \in U\ \mathfrak{A} \Vdash_x \phi(\sigma) > \varepsilon, \end{aligned} \tag{7}$$

for some $\delta < \varepsilon$. This is possible as a consequence of the truth continuity lemma.

Definition 3 (Local forcing for Metric Structures). *Let $\mathfrak{A}$ be a Sheaf of metric structures defined in X, ε a positive real number, U an open set in X, and $\sigma_1, \ldots, \sigma_n$ sections defined in U. If ϕ is an $\mathcal{F}$- restricted formula the relations $\mathfrak{A} \Vdash_U \phi(\sigma) < \varepsilon$ and $\mathfrak{A} \Vdash_U \phi(\sigma) > \varepsilon$ are defined by the following statements*

Atomic formulas

- $\mathfrak{A} \Vdash_U d(\sigma_1, \sigma_2) < \varepsilon \iff \sup_{x\in U} d_x(\sigma_1(x), \sigma_2(x)) < \varepsilon$
- $\mathfrak{A} \Vdash_U R(\sigma_1, \ldots, \sigma_n) < \varepsilon \iff \sup_{x\in U} R^{\mathfrak{A}_x}(\sigma_1(x), \ldots, \sigma_n(x)) < \varepsilon$
- *Similar to the previous two, with $>$ instead of $<$ and* sup *replaced by* inf

Logical connectives

- $\mathfrak{A} \Vdash_U \max(\phi, \psi) < \varepsilon \iff \mathfrak{A} \Vdash_V \phi < \varepsilon$ *and* $\mathfrak{A} \Vdash_W \psi < \varepsilon$
- $\mathfrak{A} \Vdash_U \max(\phi, \psi) > \varepsilon \iff$ *There exist open sets* V *and* W *such that* $V \cup W = U$ *and* $\mathfrak{A} \Vdash_V \phi > \varepsilon$ *and* $\mathfrak{A} \Vdash_W \psi > \varepsilon$
- $\mathfrak{A} \Vdash_U \min(\phi, \psi) < \varepsilon \iff$ *There exist open sets* V *and* W *such that* $V \cup W = U$ *and* $\mathfrak{A} \Vdash_V \phi < \varepsilon$ *and* $\mathfrak{A} \Vdash_W \psi < \varepsilon$
- $\mathfrak{A} \Vdash_U \min(\phi, \psi) < \varepsilon \iff \mathfrak{A} \Vdash_U \phi < \varepsilon$ *and* $\mathfrak{A} \Vdash_U \psi < \varepsilon$
- $\mathfrak{A} \Vdash_U 1 \dot{-} \psi < \varepsilon \iff \mathfrak{A} \Vdash_U \psi > 1 \dot{-} \varepsilon$
- $\mathfrak{A} \Vdash_U 1 \dot{-} \psi > \varepsilon \iff \mathfrak{A} \Vdash_U \psi < 1 \dot{-} \varepsilon$
- $\mathfrak{A} \Vdash_U \phi \dot{-} \psi < \varepsilon \iff$ *One of the following holds*
  (i) *There exists* $r \in (0,1)$ *such that* $\mathfrak{A} \Vdash_U \phi < r$ *and* $\mathfrak{A} \Vdash_U \psi > r$
  (ii) *For all* $r \in (0,1)$, $\mathfrak{A} \Vdash_U \phi < r$ *if and only if* $\mathfrak{A} \Vdash_U \psi < r$
  (iii) $\mathfrak{A} \Vdash_U \phi < \varepsilon$
  (iv) *There exists* $r, q \in (0,1)$ *such that*
  $\mathfrak{A} \Vdash_U \phi > r$ *and* $\mathfrak{A} \Vdash_U \psi < r$
  $\mathfrak{A} \Vdash_U \phi < q + \varepsilon$ *and* $\mathfrak{A} \Vdash_U \psi > q$
  and for all $\delta < \varepsilon$ *and* $\mathfrak{A} \Vdash_U \phi > \delta$
- $\mathfrak{A} \Vdash_U \phi \dot{-} \psi > \varepsilon \iff$ *There exists* $q > 0$ *such that* $\mathfrak{A} \Vdash_U \psi < q$ *and* $\mathfrak{A} \Vdash \phi > q + \varepsilon$

Quantifiers

- $\mathfrak{A} \Vdash_U \inf_\sigma \phi(\sigma) < \varepsilon \iff$ *there exist an open covering* $\{U_i\}$ *of* U *and a family of section* μ_i *each one defined in* U_i *such that* $\mathfrak{A} \Vdash_{U_i} \phi(\mu_i) < \varepsilon$ *for all* i
- $\mathfrak{A} \Vdash_U \inf_\sigma \phi(\sigma) > \epsilon \iff$ *there exist* ε' *such that* $0 < \varepsilon < \varepsilon'$ *and an open covering* $\{U_i\}$ *of* U *such that for every section* μ_i *defined in* U_i $\mathfrak{A} \Vdash_{U_i} \phi(\mu_i) > \varepsilon'$
- $\mathfrak{A} \Vdash_U \sup_\sigma \phi(\sigma) < \varepsilon \iff$ *there exist* ε' *such that* $0 < \varepsilon' < \varepsilon$ *and an open covering* $\{U_i\}$ *of* U *such that for every section* μ_i *defined in* U_i $\mathfrak{A} \Vdash_{U_i} \phi(\mu_i) < \varepsilon'$
- $\mathfrak{A} \Vdash_U \sup_\sigma \phi(\sigma) > \varepsilon \iff$ *there exist an open covering* $\{U_i\}$ *of* U *and a family of section* μ_i *each one defined in* U_i *such that* $\mathfrak{A} \Vdash_{U_i} \phi(\mu_i) > \varepsilon$ *for all* i

Observe that the definition of local forcing leads to the equivalences

$$\begin{aligned} \mathfrak{A} \Vdash_U \inf_\sigma (1 \dot{-} \phi(\sigma)) > 1 \dot{-} \varepsilon &\iff \mathfrak{A} \Vdash_U \sup_\sigma \phi(\sigma) < \varepsilon, \\ \mathfrak{A} \Vdash_U \inf_\sigma (\phi(\sigma)) < \varepsilon &\iff \mathfrak{A} \Vdash_U \sup_\sigma (1 \dot{-} \phi(\sigma)) > 1 \dot{-} \varepsilon. \end{aligned} \tag{8}$$

The fact that we can obtain a similar statement to the Maximum Principle of [3] is even more important.

Theorem 2 (The Maximum Principle for Metric structures). *If* $\mathfrak{A} \Vdash_U \inf_\sigma \phi(\sigma) < \varepsilon$ *then there exists a section* μ *defined in an open set* W *dense in* U *such that* $\mathfrak{A} \Vdash_U \phi(\mu) < \varepsilon'$, *for some* $\varepsilon' < \varepsilon$.

Proof. That $\mathfrak{A} \Vdash_U \inf_\sigma \phi(\sigma) < \varepsilon$ is equivalent to the existence of an open covering $\{U_i\}$ and a family of sections $\{\mu_i\}$ such that $\mathfrak{A} \Vdash_{U_i} \phi(\mu_i) < \varepsilon'$ for some $\varepsilon' < \varepsilon$. The family of sections $\mathcal{S} = \{\mu | \operatorname{dom}(\mu) \subset \mathrm{U}$ and $\mathfrak{A} \Vdash_{\operatorname{dom}(\mu)} \phi(\mu) < \varepsilon'\}$ is nonempty and is partially ordered by inclusion. Consider the maximal element $\mu*$ of a chain of sections in $\mathcal{S}$. Then $\operatorname{dom}(\mu*)$ is dense in U and $\mathfrak{A} \Vdash_{\operatorname{dom}(\mu*)} \phi(\mu*) < \varepsilon'$.

3 The Metric Generic Model and its theory

In certain cases, the quotient space of the metric sheaf can be the universe of a metric structure in the same language as each of the fibers. We examine in this section one such case: sheaves of metric structures over *regular* topological spaces, and a generic for a filter of *regular* open sets. We do not claim that this is the optimal situation - however, we provide a proof of a version of a Generic Model Theorem for these Metric Generic models.

Definition 4 (Metric Generic Model). *Let $\mathfrak{A} = (X, p, E)$ be a sheaf of metric structures defined on a regular topological space X and $\mathbb{F}$ an ultrafilter of regular open sets in the topology of X. We define the Metric Generic Model $\mathfrak{A}[\mathbb{F}]$ by*

$$\mathfrak{A}[\mathbb{F}] = \{[\sigma]/_{\sim_\mathbb{F}} | \operatorname{dom}(\sigma) \in \mathbb{F}\}, \tag{9}$$

provided with the metric $d_\mathbb{F}$ defined above (see Lemma 3 and subsequent discussion), and with

–
$$f^{\mathfrak{A}[\mathbb{F}]}([\sigma_1]/_{\sim_\mathbb{F}}, \ldots, [\sigma_n]/_{\sim_\mathbb{F}}) = [f^{\mathfrak{A}}(\sigma_1, \ldots, \sigma_n)]/_{\sim_\mathbb{F}} \tag{10}$$

with modulus of uniform continuity $\Delta_f^{\mathfrak{A}[\mathbb{F}]} = \inf_{x \in X} \Delta_f^{\mathfrak{A}_x}$.

–
$$R^{\mathfrak{A}[\mathbb{F}]}([\sigma_1]/_{\sim_\mathbb{F}}, \ldots, [\sigma_n]/_{\sim_\mathbb{F}}) = \inf_{U \in \mathbb{F}_{\sigma_1 \cdots \sigma_n}} \sup_{x \in U} R_x(\sigma_1(x), \ldots, \sigma_n(x)) \tag{11}$$

with modulus of uniform continuity $\Delta_R^{\mathfrak{A}[\mathbb{F}]} = \inf_{x \in X} \Delta_R^{\mathfrak{A}_x}$.

–
$$c^{\mathfrak{A}[\mathbb{F}]} = [c]/_{\sim_\mathbb{F}} \tag{12}$$

Observe that the properties of $d_\mathbb{F}$ and the fact that $R^{\mathfrak{A}}$ is continuous ensure that the Metric Generic Model is well defined as a metric structure. Special attention should be paid to the uniform continuity of $R^{\mathfrak{A}[\mathbb{F}]}$. We prove this next:

Proof. It is enough to show this for an unary relation. First, suppose $d_\mathbb{F}([\sigma], [\mu]) < \inf_{x \in X} \Delta_R^{\mathfrak{A}_x}(\varepsilon)$, then

$$\inf_{U \in \mathbb{F}_{\sigma\mu}} \sup_{x \in U} d_x(\sigma(x), \mu(x)) < \inf_{x \in X} \Delta_R^{\mathfrak{A}_x}(\varepsilon) \tag{13}$$

which implies that there exists $V \in \mathbb{F}_{\sigma\mu}$ such that

$$\sup_{x \in V} d_x(\sigma(x), \mu(x)) < \inf_{x \in X} \Delta_R^{\mathfrak{A}_x}(\varepsilon),$$

and by the uniform continuity of each $R^{\mathfrak{A}_x}$,

$$\sup_{x \in V} |R(\sigma(x)) - R(\mu(x))| \leq \varepsilon$$

We now state that

$$\left| \inf_{U \in \mathbb{F}_\sigma} \sup_{x \in U} R(\sigma(x)) - \inf_{U \in \mathbb{F}_\mu} \sup_{x \in U} R(\sigma(x)) \right| \leq \sup_{x \in V} |R(\sigma(x)) - R(\mu(x))| \, .$$

First consider $R^{\mathfrak{A}[\mathbb{F}]}([\sigma]/_{\sim\mathbb{F}}) \geq R^{\mathfrak{A}[\mathbb{F}]}([\mu]/_{\sim\mathbb{F}})$,

$$\begin{aligned} |R^{\mathfrak{A}[\mathbb{F}]}([\sigma]/_{\sim\mathbb{F}}) - R^{\mathfrak{A}[\mathbb{F}]}([\mu]/_{\sim\mathbb{F}})| &= R^{\mathfrak{A}[\mathbb{F}]}([\sigma]/_{\sim\mathbb{F}}) - R^{\mathfrak{A}[\mathbb{F}]}([\mu]/_{\sim\mathbb{F}}) \\ &\leq \sup_{x \in V} R(\sigma(x)) - R^{\mathfrak{A}[\mathbb{F}]}([\mu]/_{\sim\mathbb{F}}). \end{aligned}$$

Now, for all $\delta > 0$ there exists $W \in \mathbb{F}_\mu$ such that

$$\sup_{x \in W} R(\mu(x)) < \inf_{U \in \mathbb{F}_\mu} \sup_{x \in U} R(\mu(x)) + \delta$$

and indeed the same is true for $V' = V \cap W \in \mathbb{F}_{\sigma\mu}$. Therefore

$$|R^{\mathfrak{A}[\mathbb{F}]}([\sigma]/_{\sim\mathbb{F}}) - R^{\mathfrak{A}[\mathbb{F}]}([\mu]/_{\sim\mathbb{F}})| \leq \sup_{x \in V'} R(\sigma(x)) - \sup_{x \in V'} R(\mu(x)) + \delta,$$

where we have substituted V by V' in the first term since $V' \subset V$, and we can apply the same arguments to it. Also, since δ is arbitrary

$$\begin{aligned} |R^{\mathfrak{A}[\mathbb{F}]}([\sigma]/_{\sim\mathbb{F}}) - R^{\mathfrak{A}[\mathbb{F}]}([\mu]/_{\sim\mathbb{F}})| &\leq \sup_{x \in V'} R(\sigma(x)) - \sup_{x \in V'} R(\mu(x)) \\ &\leq \sup_{x \in V'} \left(R(\sigma(x)) - R(\mu(x)) \right) \\ &\leq \left| \sup_{x \in V'} \left(R(\sigma(x)) - R(\mu(x)) \right) \right| \\ &\leq \sup_{x \in V'} |R(\sigma(x)) - R(\mu(x))| \leq \varepsilon \qquad (14) \end{aligned}$$

In the case of $R^{\mathfrak{A}[\mathbb{F}]}([\sigma]/_{\sim\mathbb{F}}) \leq R^{\mathfrak{A}[\mathbb{F}]}([\mu]/_{\sim\mathbb{F}})$ similar arguments hold.

It is worth mentioning that part of the "generality" of the so called generic model is lost. This is indeed true and it is a consequence of the additional conditions that we have imposed on the topology of the base space (regularity) and on the ultrafilter to obtain a Cauchy complete metric space.

We can now present the Generic Model Theorem (GMT) for metric structures. This provides a nice way to describe the theory of the metric generic model by means of the forcing relation and topological properties of the sheaf of metric structures.

Theorem 3 (Metric Generic Model Theorem). *Let $\mathbb{F}$ be an ultrafilter of regular open sets on a regular topological space X and $\mathfrak{A}$ a sheaf of metric structures on X. Then*

–
$$\mathfrak{A}[\mathbb{F}] \models \phi([\sigma]/_{\sim\mathbb{F}}) < \varepsilon \iff \exists U \in \mathbb{F} \text{ such that } \mathfrak{A} \Vdash_U \phi(\sigma) < \varepsilon \tag{15}$$

–
$$\mathfrak{A}[\mathbb{F}] \models \phi([\sigma]/_{\sim\mathbb{F}}) > \varepsilon \iff \exists U \in \mathbb{F} \text{ such that } \mathfrak{A} \Vdash_U \phi(\sigma) > \varepsilon \tag{16}$$

Proof. Atomic formulas:

– $\mathfrak{A}[\mathbb{F}] \models d_{\mathbb{F}}([\sigma_1]/_{\sim\mathbb{F}}, [\sigma_2]/_{\sim\mathbb{F}}) < \varepsilon$ iff $\inf_{U\in\mathbb{F}_{\sigma_1\sigma_2}} \sup_{x\in U} d_x(\sigma_1(x), \sigma_2(x)) < \varepsilon$. This is equivalent to saying that there exists $U \in \mathbb{F}_{\sigma_1\sigma_2}$ such that

$$\sup_{x\in U} d_x(\sigma_1(x), \sigma_2(x)) < \varepsilon.$$

Further, by definition, that it is equivalent to $\mathfrak{A} \Vdash_U d(\sigma_1, \sigma_2) < \varepsilon$.
– For $\mathfrak{A}[\mathbb{F}] \models R([\sigma_1]/_{\sim\mathbb{F}}, \dots, [\sigma_n]/_{\sim\mathbb{F}})$ the same arguments as before apply.
– $\mathfrak{A}[\mathbb{F}] \models d_{\mathbb{F}}([\sigma_1]/_{\sim\mathbb{F}}, [\sigma_2]/_{\sim\mathbb{F}}) > \varepsilon$ iff $\inf_{U\in\mathbb{F}_{\sigma_1\sigma_2}} \sup_{x\in U} d_x(\sigma_1(x), \sigma_2(x)) > \varepsilon$.
  - ($\Rightarrow$) Let $\inf_{U\in\mathbb{F}_{\sigma_1\sigma_2}} \sup_{x\in U} d_x(\sigma_1(x), \sigma_2(x)) = r$ and $\varepsilon' = (r+\varepsilon)/2$. Then, the set $V = p(\langle\sigma_1, \sigma_2\rangle \cap d^{\mathfrak{A}\,-1}(\varepsilon', 1])$ is nonempty and intersects every open set in $\mathbb{F}$. If $V \notin \mathbb{F}$, consider $X \setminus \overline{V}$. That set is not an element of $\mathbb{F}$ since $\mathrm{dom}(\sigma_1) \cap \mathrm{dom}(\sigma_2) \cap \mathrm{X} \setminus \overline{\mathrm{V}}$ would also be in $\mathbb{F}$ with $d_x(\sigma_1(x), \sigma_2(x)) \le \varepsilon'$ for all x in it. Therefore $\mathrm{int}(\overline{\mathrm{V}}) \cap \mathrm{dom}(\sigma_1) \cap \mathrm{dom}(\sigma_2) \in \mathbb{F}$ and for every element in this set $d_x(\sigma_1(x), \sigma_2(x)) \ge \varepsilon'$, which implies that there exists $U' \in \mathbb{F}$ such that $\inf_{x\in U'} d_x(\sigma_1(x), \sigma_2(x)) \ge \varepsilon' > \varepsilon$.
  - ($\Leftarrow$) If $\mathfrak{A} \Vdash_V d(\sigma_1, \sigma_2) > \varepsilon$ for some $V \in \mathbb{F}_{\sigma_1\sigma_2}$, then V intersects any open set in the generic filter and for any element in V, $d_x(\sigma_1(x), \sigma_2(x)) \ge r$ where $r = \inf_{x\in V} d_x(\sigma_1(x), \sigma_2(x))$. Thus, for all $U \in \mathbb{F}$,

$$\sup_{x\in U} d_x(\sigma_1(x), \sigma_2(x)) \ge r$$

  and therefore

$$\inf_{U\in\mathbb{F}_{\sigma_1\sigma_2}} \sup_{x\in U} d_x(\sigma_1(x), \sigma_2(x)) \ge r > \varepsilon.$$

– Statements similar to those claimed above show the case of

$$\mathfrak{A}[\mathbb{F}] \models R([\sigma_1]/_{\sim\mathbb{F}}, \dots, [\sigma_n]/_{\sim\mathbb{F}}) > \varepsilon.$$

Logical connectives:

– For the connectives $1 \dot{-}$, min and max, the result follows by a simple induction in each case. We only include the proof for one of these connectives.

$$\mathfrak{A}[\mathbb{F}] \models \min(\phi([\sigma_1]/_{\sim\mathbb{F}}), \psi([\sigma_2]/_{\sim\mathbb{F}})) < \varepsilon$$
$$\iff \mathfrak{A}[\mathbb{F}] \models \phi([\sigma_1]/_{\sim\mathbb{F}}) < \varepsilon \text{ or } \mathfrak{A}[\mathbb{F}] \models \psi([\sigma_2]/_{\sim\mathbb{F}}) \tag{17}$$

$$\iff \exists U_1 \in \mathbb{F}\ \mathfrak{A} \Vdash_{U_1} \phi(\sigma_1) < \varepsilon \text{ or } \exists U_2 \in \mathbb{F}\ \mathfrak{A} \Vdash_{U_2} \psi(\sigma_2) < \varepsilon$$
$$\iff \exists U \in \mathbb{F} \text{ such that } \mathfrak{A} \Vdash_U \min(\phi(\sigma_1), \psi(\sigma_2)) < \varepsilon. \tag{18}$$

– The step $\mathfrak{A}[\mathbb{F}] \models \phi([\sigma_1]/_{\sim\mathbb{F}}) \dot{-} \psi([\sigma_2]/_{\sim\mathbb{F}}) < \varepsilon$ needs a rather lengthy case by case analysis. The cases are
  - $\mathfrak{A}[\mathbb{F}] \models \phi([\sigma_1]/_{\sim\mathbb{F}}) < \psi([\sigma_2]/_{\sim\mathbb{F}})$,
  - $\mathfrak{A}[\mathbb{F}] \not\models \phi([\sigma_1]/_{\sim\mathbb{F}}) < \psi([\sigma_2]/_{\sim\mathbb{F}})$ and $\mathfrak{A}[\mathbb{F}] \not\models \phi([\sigma_1]/_{\sim\mathbb{F}}) > \psi([\sigma_2]/_{\sim\mathbb{F}})$,
  - $\mathfrak{A}[\mathbb{F}] \models \phi([\sigma_1]/_{\sim\mathbb{F}}) < \varepsilon$,
  - $\mathfrak{A} \not\models \phi([\sigma_1]/_{\sim\mathbb{F}}) < \varepsilon$, $\mathfrak{A} \models \phi([\sigma_1]/_{\sim\mathbb{F}}) > \psi([\sigma_2]/_{\sim\mathbb{F}})$ and $\mathfrak{A} \models \phi([\sigma_1]/_{\sim\mathbb{F}}) < \psi([\sigma_2]/_{\sim\mathbb{F}}) + \varepsilon$

  The proofs here are verifications of each case.

<u>Quantifiers:</u>

–

$$\begin{aligned} &\mathfrak{A}[\mathbb{F}] \models \inf_{[\sigma_i]/_{\sim\mathbb{F}}} \phi([\sigma_i]/_{\sim\mathbb{F}}) < \varepsilon \\ \iff\ &\exists[\sigma_1]/_{\sim\mathbb{F}} \text{ such that } \mathfrak{A}[\mathbb{F}] \models \phi([\sigma_1]/_{\sim\mathbb{F}}) < \varepsilon \\ \iff\ &\exists U_1 \in \mathbb{F}\ \exists\sigma_1 \text{ such that } \mathfrak{A} \Vdash_{U_1} \phi(\sigma_1) < \varepsilon \\ \Rightarrow\ &\exists U \in \mathbb{F} \text{ such that } \mathfrak{A} \Vdash_U \inf_\sigma \phi(\sigma) < \varepsilon \end{aligned} \tag{19}$$

For the other direction suppose that there exists $U \in \mathbb{F}$ such that $\mathfrak{A} \Vdash_U \inf_\sigma \phi(\sigma) < \varepsilon$. Then the family $\mathcal{I}_\varepsilon = \{U \in \mathbb{F} | \mathfrak{A} \Vdash_U \inf_\sigma \phi(\sigma) < \varepsilon\}$ is nonempty and can be partially ordered by the binary relation $\prec$ defined by: $U \prec V$ if and only if $U \supset V$. Consider the maximal element U' of a chain defined in $\mathcal{I}_\varepsilon$. Then there exists a covering $\{V_i\}$ of U' all whose elements are basic open sets of the above class, and a family of sections $\{\mu_i\}$, such that $\mathfrak{A} \Vdash_{V_i} \phi(\mu_i) < \varepsilon$. If any $V_i \in \mathbb{F}$ then $V_i = U'$. Otherwise it will contradict the maximality of U'. Also, if $\mathrm{int(X \setminus V_i)} \in \mathbb{F}$ then $\mathfrak{A} \Vdash_{\mathrm{int(X\setminus V_i)\cap U'}} \inf_\sigma \phi(\sigma) < \varepsilon$ in contradiction to the maximality of U'. We conclude that there exists μ such that $\mathfrak{A} \Vdash_{U'} \phi(\mu) < \varepsilon$.

–

$$\begin{aligned} &\mathfrak{A}[\mathbb{F}] \models \sup_{[\sigma_i]/_{\sim\mathbb{F}}} \phi([\sigma_i]/_{\sim\mathbb{F}}) < \varepsilon \\ \iff\ &\forall[\sigma_i]/_{\sim\mathbb{F}}\ \mathfrak{A}[\mathbb{F}] \models \phi([\sigma_i]/_{\sim\mathbb{F}}) < \varepsilon \\ \iff\ &\forall\sigma_i \exists U_i \in \mathbb{F} \ \text{ such that } \mathfrak{A} \Vdash_{U_i} \phi(\sigma_i) < \varepsilon \end{aligned} \tag{20}$$

($\Rightarrow$) We prove this by contradiction. Suppose there exists V in $\mathbb{F}$ such that for some σ_i $\mathfrak{A} \Vdash_V \phi(\sigma_i) \geq \varepsilon$, then $V \cap U_i$ is also in $\mathbb{F}$ and in this set $\phi(\sigma_i) < \varepsilon$ and $\phi(\sigma_i) \geq \varepsilon$ are forced simultaneously.
($\Leftarrow$) Suppose that there exists $U \in \mathbb{F}$ such that $\mathfrak{A} \Vdash_U \sup_\sigma \phi(\sigma) < \varepsilon$. Then the family $\mathcal{S}_\varepsilon = \{U \in \mathbb{F} | \mathfrak{A} \Vdash_U \sup_\sigma \phi(\sigma) < \varepsilon\}$ is nonempty. The proof follows by similar arguments to those used in the case of $\inf_{[\sigma_i]/_{\sim\mathbb{F}}} \phi([\sigma_i]/_{\sim\mathbb{F}}) < \varepsilon$ above.

We now stress that the Metric Generic Model Theorem (GMT) has distinct but strong connections with the Classical Theorem (see [3,8]). In the case of the Metric GMT, we can observe similarities in the forcing definitions if we consider the parallelism between the minimum function and the disjunction, the maximum function and the conjunction, the infimum and the existential quantifier.

On the other hand, differences are evident if we compare the supremum with the universal quantifier. The reason for this is that in this case the sentence $1\dot{-}(1\dot{-}\phi)$, which is our analog for the double negation in continuous logic, is equivalent to the sentence ϕ. Note that the point and local forcing definitions are consistent with this fact - i.e.,

$$\begin{aligned} \mathfrak{A} \Vdash_U 1\dot{-}(1\dot{-}\phi) < \varepsilon &\iff \mathfrak{A} \Vdash_U \phi < \varepsilon, \\ \mathfrak{A} \Vdash_U 1\dot{-}(1\dot{-}\phi) > \varepsilon &\iff \mathfrak{A} \Vdash_U \phi > \varepsilon. \end{aligned} \tag{21}$$

As another consequence, the metric version of the GMT does not require an analog definition to the Gödel translation.

We close this section by introducing a simple example that illustrates some of the elements just described. We study the metric sheaf for the continuous cyclic flow in a torus.

Let $X = S^1$, $E = S^1 \times S^1$ and $p = \pi_1$, be the projection function onto the first component. Then, we have $E_q = S^1$. Given a set of local coordinates x_i in S_i and a smooth vector field V on E, such that

$$\begin{aligned} V &= V_1 \frac{\partial}{\partial x_1} + V_2 \frac{\partial}{\partial x_2} \\ V_1(p) &\neq 0 \;\; \forall p \in S^1, \end{aligned} \tag{22}$$

we can take as the set of sections the family of integrable curves of V. The open sets of the sheaf can be described as local streams through E. Complex multiplication in every fiber is continuously extended to a function between integral curves. Every section can be extended to a global section.

Let us study the metric generic model of this sheaf. Note that X is a topological regular space and that it admits an ultrafilter $\mathbb{F}$ of regular open sets. First, observe that $\mathfrak{A}[\mathbb{F}]$ is a proper subset of the set of local integrable curves. In fact, every element in $\mathfrak{A}[\mathbb{F}]$ can be described as the equivalence class of a global section in E: For any element $[\sigma] \in \mathfrak{A}[\mathbb{F}]$, $U = dom(\sigma) \in \mathbb{F}$, and there exists a global integral curve μ in E such that $\rho_{\mathbb{F}}(\sigma, \mu) = 0$. This result leads to the conclusion that every ultrafilter of open sets in S^1 generates the same universe for $\mathfrak{A}[\mathbb{F}]$. Observe that every fiber can be made into a metric structure with a metric given by the length of the shortest path joining two points. This, of course, is a Cauchy complete and bounded metric space. Dividing the distance function by π, we may redefine this to make $d(x, y) \leq 1$, for x and y in S^1. Therefore, this manifold is also a metric sheaf. In addition, observe that complex multiplication in S^1 extends to the sheaf as a uniformly continuous function in the set of sections. For any element $[\sigma] \in \mathfrak{A}[\mathbb{F}]$, let $U = dom(\sigma) \in \mathbb{F}$ and μ be the global integral curve that extends σ. Thus, for arbitrary $\varepsilon > 0$

$$\mathfrak{A} \Vdash_U d^{\mathfrak{A}}(\sigma, \mu) < \varepsilon \tag{23}$$

and as a consequence

$$\mathfrak{A}[\mathbb{F}] \models d^{\mathfrak{A}[\mathbb{F}]}([\sigma], [\mu]) = 0.$$

In addition, the metric generic model satisfies the condition that *multiplication* between sections is left continuous. Let η and μ be sections whose domain is an element of the ultrafilter. For any $\varepsilon < 1/2$, if

$$\mathfrak{A} \Vdash_{\mathrm{dom}(\eta)\cap\mathrm{dom}(\mu)} d(\eta,\mu) < \varepsilon$$

then for any other section σ defined in an element of $\mathbb{F}$, it is true that in $V = \mathrm{dom}(\eta) \cap \mathrm{dom}(\mu) \cap \mathrm{dom}(\sigma)$

$$\mathfrak{A} \Vdash_V d(\eta\sigma, \mu\sigma) < \varepsilon \tag{24}$$

and also

$$\mathfrak{A} \Vdash_V 1 \dot{-} \max(d(\eta,\mu), 1 \dot{-} d(\eta\sigma, \mu\sigma)) < \varepsilon. \tag{25}$$

By the metric GMT, we can conclude that

$$\mathfrak{A}[\mathbb{F}] \models 1 \dot{-} \max(d^{\mathfrak{A}[\mathbb{F}]}([\eta], [\mu]), 1 \dot{-} d([\eta][\sigma], [\mu][\sigma])) < \varepsilon$$

and since σ, η and μ were chosen arbitrarily.

$$\mathfrak{A}[\mathbb{F}] \models \sup_\sigma \sup_\eta \sup_\mu \big[1 \dot{-} \max(d^{\mathfrak{A}[\mathbb{F}]}([\eta], [\mu]), 1 \dot{-} d([\eta][\sigma], [\mu][\sigma]))\big] < \varepsilon.$$

Right continuity, left invariance and right invariance of this metric can be expressed in the same fashion.

4 Perspectives

Our setting, main theorem, and constructions open up various new lines of research, both within model theory itself and applications.

4.1 Model Theory

The Model Theoretic analysis of the new objects constructed here (the metric generic model in Sect. 3 and the sheaf of metric structures itself) has so far been analyzed from a model theoretic perspective only up to the consequences of the Generic Model Theorem (this is also true of the work done in the case of sheaves with discrete First Order fibers). This reflects the current situation with the model theory of sheaves in general.

Additionally, extensions of this work to topological structures (fibers supporting themselves topological spaces, built in the framework of Flum and Ziegler [7]) have been explored; they yield similar results. Further lines of research (sheaves with fibers that are more general than metric or different from topologic in the style of Flum and Ziegler - for instance, fibers that are measure algebras, etc.) are yet to be explored. They will probably require a more purely category-theoretic perspective.

Finally, the connections to the Model Theory of Metric Abstract Elementary Classes (see [10,17]) and in particular to topological dynamical perspectives on typespaces, are a burgeoning field. Work of Abramsky, Mansfield and Soares [15] has opened the line of cohomology attached to non-locality in quantum mechanics. The cohomology is built on a version of sheaves different from ours. Metric sheaves provide further contexts for the study of these phenomena, as evidenced by the works cited in this paragraph (forthcoming work).

4.2 Applications

Besides our example, metric sheaves are natural places for applications to further **dynamical systems** obtained as sheaves of metric structures, where the dynamics is provided by the behavior of sections, naturally carefully chosen. More specifically, we expect that our results will be useful for various constructions of sheaves over topological spaces naturally associated to actions of compact groups over certain varieties - among other variations. In forthcoming work with Padilla, the second author has worked on generic cohomology for sheaves of *discrete* structures possibly endowed with an group action coherent with the sheaves. In future work, we hope to merge that new line of research with our own metric sheaves. Additionally, the first author is currently working in setting up applications to classical mechanics and possibly quantization. Other possible applications include Zilber's Structural Approximation (see [18]).

References

1. Ben Yaacov, I., Berenstein, A., Henson, C.W., Usvyatsov, A.: Model theory of metric structures. In: Chatzidakis, Z., Macpherson, D., Pillay, A., Wilkie, A. (eds.) Model Theory with Applications to Algebra and Analysis. Lecture Notes Series of the London Mathematical Society, vol. 2. Cambridge University Press, Cambridge (2008)
2. Caicedo, X.: Conectivos sobre espacios topológicos. Rev. Acad. Colomb. Cienc. **21**(81), 521–534 (1997)
3. Caicedo, X.: Lógica de los haces de estructuras. Rev. Acad. Colomb. Cienc. **19**(74), 569–586 (1995)
4. Carson, A.B.: The model-completion of theory of commutative regular rings. J. Algebra **27**, 136–146 (1973)
5. Comer, S.: Elementary properties of structures of sections. Bol. Soc. Mat. Mexicana. **19**, 78–85 (1974)
6. Ellerman, D.: Sheaves of structures and generalized ultraproducts. Ann. Math. Logic **7**, 163–195 (1974)
7. Flum, J., Ziegler, M.: Topological Model Theory. Springer, New York (1980)
8. Forero, A.: Una demostración alternativa del teorema de ultralímites. Revista Colombiana de Matemáticas **43**(2), 165–174 (2009)
9. Grothendieck, A.: A general theory of fibre spaces with structure sheaf. Technical report, University of Kansas (1958)
10. Hirvonen, Å., Hyttinen, T.: Categoricity in homogeneous complete metric spaces. Arch. Math. Logic **48**, 269–322 (2009)
11. Lopes, V.C.: Reduced products and sheaves of metric structures. Math. Log. Quart. **59**(3), 219–229 (2013)

12. Macintyre, A.: Model-completeness for sheaves of structures. Fund. Math. **81**, 73–89 (1973)
13. Macintyre, A.: Nonstandard analysis and cohomology. In: Nonstandard Methods and Applications in Mathematics. Lecture Notes in Logic, vol. 25. Association for Symbolic Logic (2006)
14. Montoya, A.: Contribuciones a la teoría de modelos de haces. Lecturas Matemáticas **28**(1), 5–37 (2007)
15. Soares, R., Abramsky, S., Mansfield, S.: The cohomology of non-locality and contextuality. In: 8th International Workshop on QPL. EPTCS, vol. 95(74) (2012)
16. Villaveces, A.: Modelos fibrados y modelos haces para la teoría de conjuntos. Master's thesis, Universidad de los Andes (1991)
17. Villaveces, A., Zambrano, P.: Around independence and domination in metric abstract elementary classes: assuming uniqueness of limit models. Math. Logic Q. **60**(3), 211–227 (2014)
18. Zilber, B.: Non-commutative Zariski geometries and their classical limit. Confl. Math. **2**, 265–291 (2010)

A Curry–Howard View of Basic Justification Logic

Konstantinos Pouliasis(✉)

Graduate Center, CUNY, New York, USA
kpouliasis@gradcenter.cuny.edu

Abstract. In this paper we suggest reading a constructive necessity of a formula ($\Box A$) as internalizing a notion of constructive truth of A (a proof within a deductive system I) and validity of A (a proof under an interpretation $[\![A]\!]_J$ within some system J). An example of such a relation is provided by the simply typed lambda calculus (as I) and its implementation in SK combinators (as J). We utilize justification logic to axiomatize the notion of validity-under-interpretation and, hence, treat a "semantical" notion in a purely proof-theoretic manner. We present the system in Gentzen-style natural deduction formulation and provide reduction and expansion rules for the $\Box$ connective. Finally, we add proof-terms and proof-term equalities to obtain a corresponding calculus (Jcalc^-) that can be viewed as an extension of the Curry–Howard isomorphism with justifications. We provide standard metatheoretic results and suggest a programming language interpretation in languages with foreign function interfaces (FFIs).

1 Introduction: Necessity and Constructive Semantics

In his seminal "Explicit Provability and Constructive Semantics" [1] Artemov developed a constructive, proof-theoretic semantics for *Brouwer-Heyting-Kolmogorov* proofs [15] in what turned out to be the first development of a family of logics that we now call justification logic. The general idea, upon which we build our calculus, is that semantics of a deductive system I can be viewed in a solely proof-theoretic manner as mappings of proof constructs of I into another proof system J (which we call justifications). As an example one could think I being Heyting arithmetic and J some "stronger" system (e.g. a classical axiomatization of Peano arithmetic, a classical or intuitionistic set theory etc.). What's more, such a semantic relation can be treated logically giving rise to a modality of explicit necessity. Different sorts of necessity (K, D, $S4$, $S5$) have been offered an explicit counterpart under the umbrella of justification logic. Some of them have been studied within a Curry–Howard setting [2]. Our paper focuses on K modality and should be viewed as the counterpart of [4] with justifications as we explain in Sect. 5.

J. Väänänen et al. (Eds.): WoLLIC 2016, LNCS 9803, pp. 316–337, 2016.
DOI: 10.1007/978-3-662-52921-8_20

1.1 Deductive Systems, Validity and Necessity

Following a framework championed by Lambek [10,11], let us assume two deductive systems I (with propositional universe U_I, a possibly non-empty signature of axioms Σ_I and an entailment relation $\Sigma_I; \Gamma \vdash_I A$) and J (resp. with U_J, Σ_J and $\Sigma_J; \Delta \vdash_J \phi$). We will be using Latin letters for the formulae of I and Greek letters for the formulae of J. We will be omitting the Σ signatures when they are not relevant.

For the entailment relations of the two systems we require the following elementary principles:

1. *Reflexivity.* In both relations Γ and Δ are multisets of formulas (contexts) that enjoy reflexivity:
$$A \in \Gamma \Longrightarrow \Gamma \vdash_I A$$
$$\phi \in \Delta \Longrightarrow \Delta \vdash_J \phi$$
2. *Compositionality.* Both relations are closed under deduction composition:
$$\Gamma \vdash_I A \text{ and } \Gamma^{'}, A \vdash_I B \Longrightarrow \Gamma, \Gamma^{'} \vdash_I B$$
$$\Delta \vdash_J \phi \text{ and } \Delta^{'}, \phi \vdash_J \psi \Longrightarrow \Delta, \Delta^{'} \vdash_J \psi$$
3. *Top.* Both systems have a distinguished top formula $\top$ for which under any Γ, Δ:
$$\Gamma \vdash_I \top_I \text{ and } \Delta \vdash_J \top_J$$

Now we can define:

Definition 1. *Given a deductive system I, an interpretation for I, noted by $[\![\bullet]\!]_J$, is a pair $(J, [\![\bullet]\!])$ of a deductive system J together with a (functional) mapping $[\![\bullet]\!] : U_I \to U_J$ on propositions of I into propositions of J extended to multisets of formulae of U_I with the following properties:*

1. *Top preservation.* $[\![\top_I]\!] = \top_J$
2. *Structural interpretation of contexts. For Γ contexts of the form $A_1, \ldots, A_n$:*
$$[\![\Gamma]\!] = [\![A_1]\!], \ldots, [\![A_n]\!]$$
(trivially empty contexts map to empty contexts. As in [10] they can be treated as the $\top$ element).

Definition 2. *Given a deductive system I and an interpretation $[\![\bullet]\!]_J$ for I we define a corresponding validation of a deduction $\Sigma_I; \Gamma \vdash_I A$ as a deduction $\Sigma_J; \Delta \vdash_J \phi$ in J such that $[\![A]\!] = \phi$ and $\Delta = [\![\Gamma]\!]$. We will be writing $[\![\Sigma_I; \Gamma \vdash_I A]\!]_J$ to denote such a validation.*

Definition 3. *Given a deductive system I, we say that an interpretation $[\![\bullet]\!]_J$ is logically complete when for all purely logical deductions $\mathcal{D}$ (i.e. deductions that make no use of Σ_I) in I there exists a corresponding (purely logical) validation $[\![\mathcal{D}]\!]$ in J. i.e*
$$\forall \mathcal{D}.\ \mathcal{D} : \Gamma \vdash_I A \Longrightarrow \exists [\![\mathcal{D}]\!] : [\![\Gamma \vdash A]\!]_J$$

Note, that we require existence but not uniqueness. Nevertheless, if we treat deductive systems in a proof irrelevant manner as preorders the above definition gives uniqueness vacuously. In a more refined approach where I and J are viewed as categories of proofs the above "logical completeness" translates to the requirement that if the set of (purely logical) arrows $Hom_I(\Gamma, A)$ is non empty then $Hom_J([\![\Gamma]\!], [\![A]\!]_J)$ cannot be empty (i.e. that $[\![\bullet]\!]_J$ can be extended to a functor). We leave a complete categorical semantics of our logic for future work but we expect a generalization of the endofunctorial interpretations of K modality appearing in [4,9].

Examples of triplets $(I,\ J,\ [\![\bullet]\!]_J)$ of logical systems that fall under the definition above are: any intuitionistic system mapped to a classical one under the embedding $[\![A \supset B]\!] = \tilde{\neg} A \tilde{\vee} B$ where $\tilde{\neg}$ and $\tilde{\vee}$ are classical connectives, the opposite direction under double negation translation, an intuitionistic system mapped to another intuitionistic system (i.e. a mapping of atomic formulas of I to atomic formulas of J extended naturally to the intuitionistic connectives or, simply, the identity mapping) etc. A vacuous validation (when $[\![\bullet]\!]_J$ maps everything to $\top$) gives another example.

We will focus on the case where I (the propositional part of our logic) is based on the implicative fragment of intuitionistic logic and show how justification logic provides for an axiomatization of such logically complete interpretations $[\![\bullet]\!]_J$ of implicative intuitionistic logic. In what follows we provide a natural deduction for an intuitionistic system I (truth), an axiomatization/specification of $[\![\bullet]\!]_J$ (treated abstractly as a function symbol on types) and a treatment of basic necessity that relates the two deductions by internalizing a notion of "double truth" (proof in I and existence of corresponding validation in J).

2 Judgments of Jcalc$^-$

We aim for a reading of necessity that internalizes a notion of "double proof" in two deductive systems. Motivated by the discussion and definitions in the previous section we will treat the notion of interpretation abstractly – as a function symbol on types – and axiomatize in accordance. Schematically we want:

$$\Box A \ \mathsf{true} := A \ \mathsf{true}\ \&\ A \ \mathsf{valid} = A \ \mathsf{true\ in\ I}\ \&\ [\![A]\!] \ \mathsf{true\ in\ J}$$

We will be dropping indexes I, J since they can be inferred by the different kinds of assumption contexts. In addition, we omit signatures Σ since they do not offer anything from a logical perspective.

Logical entailment for the proposed $\Box$ connective can be summarized easily given our previous discussion. Given a deduction $\mathcal{D} : A \vdash B$ and the existence of validation $[\![\mathcal{D}]\!] : [\![A]\!] \vdash [\![B]\!]$ then given $\Box A$ (i.e. a proof of a $\vdash A$ and a validation $\vdash [\![A]\!]$) we obtain a double proof of B (and hence, $\Box B$) by *compositionality* of the underlying systems. Using standard, proof tree notation with labeled assumptions we formulate our rule of the connective in natural deduction:

$$\dfrac{\begin{matrix} \\ \vdots \\ \Box A \end{matrix} \qquad \begin{matrix} \overline{A}^{\,x} \\ \vdots \\ B \end{matrix} \qquad \begin{matrix} \overline{[\![A]\!]}^{\,s} \\ \vdots \\ [\![B]\!] \end{matrix}}{\Box B}\ I_{\Box B} E^{x,s}_{\Box A}$$

We can, easily, generalize to $\Box$ed contexts (of the form $\Box A_1, \ldots, \Box A_i$) of arbitrary length:

$$\dfrac{\begin{matrix} \\ \vdots \\ \Box A_1 \end{matrix} \quad \ldots \quad \begin{matrix} \\ \vdots \\ \Box A_i \end{matrix} \qquad \begin{matrix} \overline{\Gamma' : A_1, \ldots A_i}^{\,\boldsymbol{x}} \\ \vdots \\ B \end{matrix} \qquad \begin{matrix} \overline{[\![\Gamma']\!] : [\![A_1]\!], \ldots [\![A_i]\!]}^{\,\boldsymbol{s}} \\ \vdots \\ [\![B]\!] \end{matrix}}{\Box B}\ I_{\Box B} E^{\boldsymbol{x},\boldsymbol{s}}_{\Box A_1 \ldots \Box A_i}$$

We read as "Introducing $\Box B$ after eliminating $\Box A_1 \ldots \Box A_i$ crossing out (vectors of) labels $\boldsymbol{x}, \boldsymbol{s}$". Interestingly, the same rule eliminates boxes and introduces new ones. This is not surprising for K modality (it is a left-right rule as we will see Sect. 2.4. See also discussion in [4,5]). We will be referring to this rule as "$\Box$ Intro–After–Elim" or, simply $\Box_{IE}$, from now on.

Note that we define the $\Box$ connective negatively, yet (pure) introduction rules for the $\Box$ connective are derivable. Such are instances of the previous Intro–After–Elim rule when Γ' is empty which conforms exactly with the idea of necessity internalizing double theoremhood.

$$\dfrac{\vdash B \qquad \vdash [\![B]\!]}{\Box B}\ I_{\Box B}$$

In the next section, we provide the whole calculus in natural deduction format. As expected we will extend the implicational fragment of intuitionistic logic with

- Judgments about validity (justification logic).
- Judgments that relate truth and validity (modal judgments).

2.1 Natural Deduction for Jcalc$^-$

Following type theory conventions, we first provide rules underlying type construction, then rules for well-formedness of (labeled) assumption contexts and rules introducing and eliminating connectives. The rules below should be obvious except for small caveat. On the one hand, the type universe of U_I and the proof trees of I are inductively defined as usual; on the other hand, the host theory J (its corresponding universe, connectives and proof trees) is "black boxed". What we actually axiomatize are the properties that all (logic preserving) interpretations of I should conform to, independently of the specifics of the host theory. Validity judgments should thus be read as specifications of provability (existence of proofs) of any candidate J.

We use Prop_0 to denote the type universe of I and $[\![\mathsf{Prop}_0]\!]$ to denote its image under an interpretation, Prop_1 denotes modal ("boxed") types and Prop

the union of $\mathsf{Prop}_0, \mathsf{Prop}_1$. We write P_k with k ranging in some subset of natural numbers to denote atomic propositions in I.

Judgments on Type Universe(s)

$$\frac{}{P_k \in \mathsf{Prop}_0}\ \textsc{Atom} \qquad \frac{}{\top \in \mathsf{Prop}_0}\ \textsc{Top} \qquad \frac{A \in \mathsf{Prop}_0}{\Box A \in \mathsf{Prop}_1}\ \textsc{Box} \qquad \frac{A \in \mathsf{Prop}_0 \qquad B \in \mathsf{Prop}_0}{A \supset B \in \mathsf{Prop}_0}\ \textsc{Arr}_0$$

$$\frac{A \in \mathsf{Prop}_1 \qquad B \in \mathsf{Prop}_1}{A \supset B \in \mathsf{Prop}_1}\ \textsc{Arr}_1 \qquad \frac{A \in \mathsf{Prop}_0}{[\![A]\!] \in [\![\mathsf{Prop}_0]\!]}\ \textsc{Brc}$$

For labeled contexts of assumptions we require standard wellformedness conditions (i.e. uniqueness of labels). We use letters x_i, or simply x, for labels of contexts with assumptions in Prop_0, x'_i or simply x' for contexts with assumptions in Prop_1 and s_i, or simply s, for $[\![\mathsf{Prop}_0]\!]$ contexts. We use $\circ$ for the empty context of Prop_0 and Prop_1 and $\dagger$ for the empty context of $[\![\mathsf{Prop}_0]\!]$. We abuse notation and write $x : A \in \Gamma$ (or, similarly, $s : [\![A]\!] \in \Delta$) to denote that the label x is assigned type A in Γ; or $\Gamma \in \mathsf{Prop}_0$ (resp. $\Gamma \in \mathsf{Prop}_1$, $\Delta \in [\![\mathsf{Prop}_0]\!]$) to denote that Γ is a wellformed context with co–domain of elements in Prop_0 (resp. in Prop_1, $[\![\mathsf{Prop}_0]\!]$). For $\Gamma \in \mathsf{Prop}_0$ we define $[\![\Gamma]\!]$ as the lifting of the context Γ through the $[\![\bullet]\!]$ symbol (with appropriate renaming of variables – e.g. $x_i \rightsquigarrow s_i$). For the vacuous case when Γ is empty we require $[\![\circ]\!] = \dagger$ to be well formed.

In the following entry we define proof trees (in turnstile representation) of the intuitionistic source theory I. For all following rules we assume $\Gamma, A, B \in \mathsf{Prop}_0$:

Judgments on Truth $\Gamma, A, B \in \mathsf{Prop}_0$

$$\frac{x : A \in \Gamma}{\Gamma \vdash A}\ \Gamma_0\text{-}\textsc{Refl} \qquad \frac{}{\Gamma \vdash \top}\ \top_0\mathrm{I} \qquad \frac{\Gamma, x : A \vdash B}{\Gamma \vdash A \supset B}\ {\supset_0}\mathrm{I} \qquad \frac{\Gamma \vdash A \supset B \qquad \Gamma \vdash A}{\Gamma \vdash B}\ {\supset_0}\mathrm{E}$$

For the calculus of interpretation (validity) we demand context reflexivity, compositionality and logical completeness with respect to intuitionistic implication. Logical completeness is specified axiomatically, since the host theory is "black boxed". Following justification logic, we use an axiomatic characterization of combinatory logic (for $\supset$) together with the requirement that the interpretation preserves modus ponens:

Judgments on Validity with $\Delta \in [\![\mathsf{Prop}_0]\!]$

$$\frac{s : [\![A]\!] \in \Delta}{\Delta \vdash [\![A]\!]}\ \Delta\text{-}\textsc{Refl} \qquad \frac{}{\Delta \vdash [\![\top]\!]}\ \textsc{Ax}_1 \qquad \frac{A, B \in \mathsf{Prop}_0}{\Delta \vdash [\![A \supset (B \supset A)]\!]}\ \textsc{Ax}_2$$

$$\frac{A, B, C \in \mathsf{Prop}_0}{\Delta \vdash [\![A \supset (B \supset C) \supset ((A \supset B) \supset (A \supset C))]\!]}\ \textsc{Ax}_3 \qquad \frac{\Delta \vdash [\![A \supset B]\!] \qquad \Delta \vdash [\![A]\!]}{\Delta \vdash [\![B]\!]}\ \mathrm{MP}$$

Finally, we have judgments in the $\Box$ed universe (Prop_1). These are context reflection, the $\Box$ Intro-After-Elim rule, and the rules for intuitionistic implication between $\Box$ed types[1].

Judgments on Necessity with $\Gamma \in \mathsf{Prop}_1$, $\mathsf{length}(\Gamma) = i$, $1 \leq k \leq i$ **and,** $\Gamma', A, A_k, B \in \mathsf{Prop}_0$

$$\frac{x' : \Box A \in \Gamma}{\Gamma \vdash \Box A}\ \Gamma_1\text{-}\textsc{Refl} \qquad \frac{(\forall A_i \in \Gamma'.\ \Gamma \vdash \Box A_i) \qquad \Gamma' \vdash B \qquad [\![\Gamma']\!] \vdash [\![B]\!]}{\Gamma \vdash \Box B}\ I_{\Box B} E^{\boldsymbol{x},\boldsymbol{s}}_{\Box A_1 \ldots \Box A_i}$$

$$\frac{\Gamma, x' : \Box A \vdash \Box B}{\Gamma \vdash \Box A \supset \Box B}\ \supset_1\text{I} \qquad \frac{\Gamma \vdash \Box A \supset \Box B \qquad \Gamma \vdash \Box A}{\Gamma \vdash \Box B}\ \supset_1\text{E}$$

(Pure) $\Box I$ **as Derivable Rule.** We stress here that $\Box$ can be introduced positively with the previous rule with $\Gamma' = \circ$. The first premise reduces to a simple requirement that $\Gamma \in \mathsf{Prop}_1$.

$$\frac{\circ \vdash A \qquad \dagger \vdash [\![A]\!]}{\Gamma \vdash \Box A}\ I_{\Box A}$$

A Simple Derivation. We show here that the K axiom of modal logic is a theorem (omitting some obvious steps). In the following

$$\Gamma := x_1' : \Box(A \supset B), x_2' : \Box A,\ \Gamma' = x_1 : A \supset B, x_2 : A,\ [\![\Gamma']\!] = s_1 : [\![A \supset B]\!], s_2 : [\![A]\!]$$

$$\cfrac{\cfrac{\cfrac{\Gamma \vdash \Box(A \supset B) \qquad \Gamma \vdash \Box A \qquad \Gamma' \vdash B \qquad [\![\Gamma']\!] \vdash [\![B]\!]}{\Box(A \supset B), \Box A \vdash \Box B}\ I_{\Box A} E^{x_1,x_2,s_1,s_2}_{\Box A \supset B, \Box A}}{\Box(A \supset B) \vdash \Box A \supset \Box B}\ \supset_1\text{I}}{\circ \vdash \Box(A \supset B) \supset \Box A \supset \Box B}\ \supset_1\text{I}$$

2.2 Logical Completeness, Admissibility of Necessitation and Completeness with Respect to Hilbert Axiomatization

Here we give a Hilbert axiomatization of the $\supset$ fragment of intuitionistic K logic in order to compare it with our system. Here $\vdash^{\mathcal{H}}$ captures the textbook (metatheoretic) notion of "deduction from assumptions" in a Hilbert style axiomatization. We assume the restriction of the system to formulas up to modal degree 1.

[1] The implication and elimination rules in Prop_1 actually coincide with the ones in Prop_0 since we are focusing on the case where I is intuitionistic. This need not necessarily be the case as we have explained. Intuitionistic implication among $\Box$ types should be read as "double proof of A implies double proof of B" and would still be defined even if we did not observe any kind of implication in I. Similarly, one could provide intuitionistic conjunction or disjunction between $\Box$ types independently of I and, vice versa, one could add connectives in I that are not observed between $\Box$ed types.

Hilbert Style Formulation

$$\textsc{ax}1.\ A \supset (B \supset A) \qquad \textsc{ax}2.\ (A \supset (B \supset C)) \supset ((A \supset B) \supset (A \supset C))$$

$$\text{K. } \Box(A \supset B) \supset \Box A \supset \Box B \qquad \text{MP}\ \frac{A \supset B \qquad A}{B} \qquad \textsc{Nec}\ \frac{\vdash^{\mathcal{H}} A}{\Box A}$$

It is easy to verify that axioms 1, 2 are derived theorems of Jcalc^- in Prop_0. The rule Modus Ponens is also admissible trivially, whereas axiom K was shown to be a theorem in the previous Sect. 2.1. The rule of Necessitation is not obviously admissible though. In our reading of necessity the admissibility of this rule is directly related to the requirement of "logical completeness of the interpretation" i.e. preservation of logical theoremhood. In general, adding more connectives in I would require additional specifications for the host theory to obtain necessitation.

The steps of the proof are given in the Appendix, but this is essentially the "lifting lemma" in justification logic [1]. The proof fully depends on the provability requirements imposed in the $[\![\mathsf{Prop}_0]\!]$ fragment.

Lemma 1 ($\Box$Lifting Lemma). *In Jcalc^-, for every $\Gamma, A \in \mathsf{Prop}_0$ if $\Gamma \vdash A$ then $[\![\Gamma]\!] \vdash [\![A]\!]$ and, hence, $\Box\Gamma \vdash \Box A$.*

We get admissibility of necessitation as a lemma for Γ empty:

Lemma 2 (Admissibility of Necessitation). *For $A \in \mathsf{Prop}_0$, if $\circ \vdash A$ then $\circ \vdash \Box A$.*

As a result:

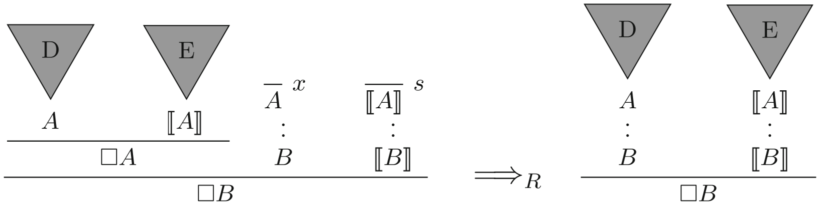
Theorem 1 (Completeness). *Jcalc^- is complete with respect to the Hilbert style formulation of degree-1 intuitionistic K modal logic.*

2.3 Harmony: Local Soundness and Local Completeness

Before we move on to show (Global) Soundness we provide evidence for the so called "local soundness" and "local completeness" of the $\Box$ connective following Gentzen's dictum. The local soundness and completeness for the $\supset$ connective is given elsewhere (e.g. [14]) and in Gentzen's original [6]. Gentzen's program can be described with the following two slogans:

a. Elim is left-inverse to Intro
b. Intro is right-inverse to Elim

Applied to the $\Box$ connective, the first principle says that introducing a $\Box A$ (resp. many $\Box A_1, \ldots, \Box A_i$) only to eliminate it (resp. them) directly is redundant. In other words, the elimination rule cannot give you more data than what were inserted in the introduction rule(s) ("elimination rules are not *too* strong"). We show here the "Elim-After-Singleton-Intro" sub-case. The exact same principle applies in the "Elim-after-Intro" of multiple $\Box$s shown in the Appendix A.3.



Dually, the second principle says eliminating a $\Box A$, should give enough information to directly reintroduce it (“elimination rules are not *too weak*”). This is an expansion principle.

$$\begin{array}{c}\mathcal{D}\\ \Box A\end{array} \quad \Longrightarrow_E \quad \dfrac{\begin{array}{c}\mathcal{D}\\ \Box A\end{array} \qquad \overline{A}^{\,x} \qquad \overline{[\![A]\!]}^{\,s}}{\Box A}\; I_{\Box A} E^{x,s}_{\Box A}$$

2.4 (Global) Soundness

Soundness is shown by proof theoretic techniques. Standardly, we add the bottom type ($\bot$) to Jcalc^- together with its elimination rule and show that the system is consistent ($\nvdash \bot$) by devising a sequent calculus and showing admissibility of cut. We only present the calculus here and collect the theorems towards consistency in the Appendix.

In the following we use $\Gamma \Rightarrow A$ (where $\Gamma, A \in \mathsf{Prop_0} \cup \mathsf{Prop_1}$) to denote sequents modulo Γ permutations where Γ is a multiset of Prop (no labels) and $\Delta \Rightarrow [\![\mathsf{A}]\!]$ for sequents corresponding to $[\![\mathsf{judgments}]\!]$ of the calculus modulo Δ permutations (with Δ (unlabeled) multiset of $[\![\mathsf{Prop_0}]\!]$). The multiset/modulo permutation approach is instructed by standard structural properties. All properties are stated formally and proved in the Appendix.

The $[\![\Gamma]\!] \Rightarrow [\![A]\!]$ relation is defined directly from $\vdash$:

Sequent Calculus ($[\![\mathsf{Prop_0}]\!]$)

$$[\![\Gamma]\!] \Rightarrow [\![A]\!] := \exists \Gamma' \in \pi([\![\Gamma]\!]) \text{ s.t } \Gamma' \vdash [\![A]\!]$$

where $\pi([\![\Gamma]\!])$ is the collection of permutations of $[\![\Gamma]\!]$.

Sequent Calculus (Prop)

$$\dfrac{}{\Gamma, A \Rightarrow A}\, Id \qquad \dfrac{\Gamma, A \supset B, B \Rightarrow C \qquad \Gamma, A \supset B \Rightarrow A}{\Gamma, A \supset B \Rightarrow C}\, \supset_L \qquad \dfrac{\Gamma, A \Rightarrow B}{\Gamma \Rightarrow A \supset B}\, \supset_R \qquad \dfrac{}{\Gamma, \bot \Rightarrow A}\, \bot_L$$

$$\dfrac{\Box\Gamma, \Gamma \Rightarrow A \qquad [\![\Gamma]\!] \Rightarrow [\![A]\!]}{\Box\Gamma \Rightarrow \Box A}\, \Box_{LR}$$

Standardly, we extend the system with the Cut rule and we obtain the extended system $\Gamma \Rightarrow^+ A := \Gamma \Rightarrow A + \mathsf{Cut}$. We show Completeness of $\Rightarrow^+$ with respect to Natural Deduction and Admissibility of Cut that leads to the consistency result

Theorem 2 (Consistency of Jcalc^-). $\nvdash \perp$

3 The Computational Side of Jcalc$^-$

In this section we add proof terms to represent natural deduction constructions. The meaning of these terms emerges naturally from Gentzen's principles that give reduction (computational β-rules) and expansion (i.e. extensionality η-rules) equalities for the each construct. We focus on the new constructs of the calculus that emerge from the judgmental interpretation of the $\Box$ connective as explained in Sect. 2.

There will be no computational (reduction) rules on provability terms. This conforms with our reading of these terms as *references* to proof constructs of an *abstracted* theory J that can be realized differently for a concrete J.

3.1 Proof Term Assignment

The following rules and their correspondence with natural deduction constructs Sect. 2.1 should be obvious to the reader familiar with the simply typed λ-calculus and basic justification logic. We do not repeat here the corresponding β, η equality rules since they are standard.

Judgments on Truth $\Gamma, A, B \in \mathsf{Prop_0}$ **and** $M := x_i \mid\ <>\ \mid \lambda x : A.\ M \mid (MM)$

$$\frac{x : A \in \Gamma}{\Gamma \vdash x : A}\ \Gamma_0\text{-}\textsc{Refl} \qquad \frac{}{\Gamma \vdash <>: \top}\ \top_0\mathrm{I} \qquad \frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x : A.\ M : A \supset B}\ \supset_0\mathrm{I}$$

$$\frac{\Gamma \vdash M : A \supset B \qquad \Gamma \vdash M' : A}{\Gamma \vdash (MM') : B}\ \supset_0\mathrm{E} \qquad + \beta\eta \text{ equalities for } \top, \supset$$

For judgments of $[\![\mathsf{Prop_0}]\!]$, we assume a countable set of constant names and demand that every combinatorial axiom of intuitionistic logic has a witness under the interpretation $[\![\bullet]\!]$. This is what justification logicians call "axiomatically appropriate constant specification". As usual we demand reflection of contexts in J and preservation of modus ponens – closedness under some notion of application (which we denote as $*$).

Judgments on Validity $\Delta \in [\![\mathsf{Prop_0}]\!]$ **and** $\mathsf{J} := s_i \mid C_i \mid \mathsf{J} * \mathsf{J}$

$$\frac{s : [\![A]\!] \in \Delta}{\Delta \vdash s : [\![A]\!]}\ \Delta\text{-}\textsc{Refl} \qquad \frac{}{\Delta \vdash C_\top : [\![\top]\!]}\ \textsc{Ax}_1 \qquad \frac{A, B \in \mathsf{Prop_0}}{\Delta \vdash C_{K^{A,B}} : [\![A \supset (B \supset A)]\!]}\ \textsc{Ax}_2$$

$$\frac{A, B, C \in \mathsf{Prop_0}}{\Delta \vdash C_{S^{A,B,C}} : [\![A \supset (B \supset C) \supset ((A \supset B) \supset (A \supset C))]\!]}\ \textsc{Ax}_3$$

$$\frac{\Delta \vdash \mathsf{J} : [\![A \supset B]\!] \qquad \Delta \vdash \mathsf{J}' : [\![A]\!]}{\Delta \vdash \mathsf{J} * \mathsf{J}' : [\![B]\!]}\ \textsc{App}$$

If J is a proof calculus and $[\![\bullet]\!]_J$ is an interpretation such that the specifications above are realized, then J can witness intuitionistic provability. This can be shown by the proof relevant version of the lifting lemma that states:

Lemma 3 ($[\![\bullet]\!]$Lifting Lemma). *Given $\Gamma, A \in \mathsf{Prop}_0$ s.t. and a term M s.t. $\Gamma \vdash M : A$ then there exists J s.t $[\![\Gamma]\!] \vdash \mathsf{J} : [\![A]\!]$.*

Proof Term Assignment and Gentzen Equalities for $\Box$ Judgments. Before we proceed, we will give a small primer of *let*-bindings as used in modern programming languages to provide for some intuition on how such terms work. Let us assume a rudimentary programming language that supports some basic types, say integers (int), as well as pairs of such types. Moreover, let us define a datatype Point as a pair of int i.e. as $(\mathsf{int}, \mathsf{int})$ In a language with *let*-bindings one can define a simple function that takes a Point and "shifts" it by adding 1 to each of its x and y coordinates as follows:

```
def shift (p:Point) =
    let (x,y) be p
    in
    (x+1,y+1)
```

If we call this function on the point `(2,3)`, then the computation `let (x,y) be (2,3) in (x+1,y+1)` is invoked. This expression reduces following the *let* reduction rule (i.e. pattern matching and substitution) to `(2+1,3+1)`; and as a result we obtain the value `(3,4)`. As we will see, *let* bindings – with appropriate typing restrictions for our system – are used in the assignment of proof terms for the $\Box_{IE}$ rule. Moreover, the reduction principle for such terms (β-rule) – obtained following Gentzen's equalities for the $\Box$ connective – is exactly the one that we just informally described.

We can now move forward with the proof term assignment for the $\Box_{IE}$ rule. We show first the sub-cases for Γ' empty (pure $\Box_I$) and Γ' singleton and explain the computational significance utilizing Gentzen's principles appropriated for the $\Box$ connective. We are directly translating proof tree equalities from Sect. 2.3 to proof term equalities. We generalize for arbitrary Γ' in the following subsection. We have, respectively, the following instances:

$$\frac{\Gamma \in \mathsf{Prop}_1 \qquad \circ \vdash M : B \qquad \dagger \vdash \mathsf{J} : [\![B]\!]}{\Gamma \vdash M \& \mathsf{J} : \Box B} \qquad \frac{\Gamma \vdash N : \Box A \qquad x : A \vdash M : B \qquad s : [\![A]\!] \vdash \mathsf{J} : [\![B]\!]}{\Gamma \vdash \mathsf{let}\ (x \& s\ \ \mathsf{be}\ N)\ \mathsf{in}\ (M \& \mathsf{J}) : \Box B}$$

Gentzen's Equalities for ($\Box$ Terms). Gentzen's reduction and expansion principles give computational meaning (dynamics) and an extensionality principle for linking terms. We omit naming the empty contexts for economy.

$$\dfrac{\square_I\,\dfrac{\Gamma \in \mathsf{Prop}_1 \qquad \vdash M : A \qquad \vdash \mathsf{j} : [\![A]\!]}{\Gamma \vdash M\&\mathsf{j} : \square A} \qquad x : A \vdash M' : B \qquad s : [\![A]\!] \vdash \mathsf{j}' : [\![B]\!]}{\Gamma \vdash \mathsf{let}\ \ (x\&s)\ \ \mathsf{be}\ \ (M\&\mathsf{J})\ \ \mathsf{in}\ \ (M'\&\mathsf{J}') : \square B}\ I_{\square B}E^{x,s}_{\square A}$$

$$\Longrightarrow_R$$

$$\dfrac{\Gamma \in \mathsf{Prop}_1 \qquad \vdash M'[M/x] : B \qquad \vdash \mathsf{J}'[\mathsf{J}/s] : [\![B]\!]}{\Gamma \vdash M'[M/x]\&\mathsf{J}'[\mathsf{j}/s] : \square B}\ I_{\square B}$$

Where the expressions $M'[M/x]$ and $\mathsf{J}'[\mathsf{J/s}]$ denote capture avoiding substitution, reflecting proof compositionality of the two calculi.

Following the expansion principle we obtain:

$$\Gamma \vdash M : \square A \ \ \Longrightarrow_E$$

$$\dfrac{\Gamma \vdash M : \square A \qquad x : A \vdash x : A \qquad s : [\![A]\!] \vdash s : [\![A]\!]}{\Gamma \vdash \mathsf{let}\ (x\&s\ \mathsf{be}\ M)\ \mathsf{in}\ (x\&s) : \square A}\ I_{\square A}E^{x,s}_{\square A}$$

That gives an η-equality as follows:

$$M : \square A =_\eta \ \ \mathsf{let}\ \ (x\&s\ \mathsf{be}\ M)\ \ \mathsf{in}\ \ (x\&s) : \square A$$

The η equality demands that every $M : \square A$ should be reducible to a form $M'\&\mathsf{J}'$.

Proof Term Assignment for the $\square$ Rule (Generically). After understanding the computational meaning of let expressions in the $\square_{IE}$ rule we can now give proof term assignment for the rule in the general case(i.e. for Γ' of arbitrary length). We define a helper syntactic construct –$\mathsf{let}^* \ldots\ \mathsf{in}$– as syntactic sugar for iterative let bindings based on the structure of contexts. The let^* macro takes four arguments: a context $\Gamma \in \mathsf{Prop}_0$, a context $\Delta \in [\![\mathsf{Prop}_1]\!]$, a possibly empty ([ ]) list of terms $Ns := N_1, \ldots, N_i$ - all three of the same length - and a term M. It is defined as follows for the empty and non-empty cases:

$\mathsf{let}^*\ (\circ;\ \dagger;\ [\,])\ \mathsf{in}\ M := M$
$\mathsf{let}^*\ (x_1 : A_1, \ldots, x_i : A_i\ ;\ s_1 : \phi_1, \ldots, s_i : \phi_i;\ N_1, \ldots, N_i)\ \mathsf{in}\ M :=$
$\mathsf{let}\ \{(x_1\&s_1)\ \mathsf{be}\ N_1, \ldots, (x_i\&s_i)\ \mathsf{be}\ N_i\}\ \mathsf{in}\ M$

Using this syntactic definition the rule $\square_{IE}$ rule can be written compactly:

$\square_{IE}$ **With** $\Gamma \in \mathsf{Prop}_1$, $\Gamma' \in \mathsf{Prop}_0$, $\mathsf{length}(\Gamma) = i$, $Ns := N_1 \ldots\ N_i$, $1 \leq k \leq i$

$$\dfrac{\forall A_k \in \Gamma'.\ \Gamma \vdash N_k : \square A_k \qquad \Gamma' \vdash M : B \qquad [\![\Gamma']\!] \vdash \mathsf{J} : [\![B]\!]}{\Gamma \vdash \mathsf{let}^*\ (\Gamma', [\![\Gamma']\!], Ns)\ \mathsf{in}\ (M\&\mathsf{J}) : \square B}\ I_{\square B}E^{\boldsymbol{x},\boldsymbol{s}}_{\square A_1 \ldots \square A_i}$$

It is obvious that all previously mentioned cases are captured with this formulation. The rule of β-equality can be given for multi-let bindings directly from Gentzen's reduction principle Sect. 2.3 generalized for the multiple intro case shown in the Appendix A.3.

$$\mathsf{let}\{(x_1\&s_1)\ \mathsf{be}\ (M_1\&\mathsf{J}_1),\ldots,(x_i\&s_i)\ \mathsf{be}\ (M_i\&\mathsf{J}_\mathsf{i})\}\ \mathsf{in}\ (M\&\mathsf{J}) =_\beta M[M_1/x_1,\ldots,M_i/x_i]\&\mathsf{J}[\mathsf{J}_1/s_1,\ldots,\mathsf{J}_\mathsf{i}/s_i]$$

3.2 Strong Normalization and Small-Step Semantics

In the Appendix A.4 we provide a proof of normalization for natural deduction (via cut elimination). This is "essentially" a strong normalization result for the proof term system also. In general we have shown the congruence obtained from $=_{\beta\eta}$ rules gives a consistent equational system. Nevertheless, we leave this for an extended version of this paper. Instead, we sketch briefly a weaker result: normalization under a deterministic,"call-by-value" reduction strategy for β-rules. This gives an idea of how the system computes and we can use it in the applications in the next section. As usual we characterize a subset of the closed terms as values and we provide rules for the reduction of the non-value closed terms. Note that for the constants of validity and their applicative closure we do not observe reduction properties but treat them as values – again conforming with the idea of J (and its reduction principles) being "black boxed".

Small step, call-by-value reduction $\rightarrow$

$$\frac{}{\lambda x.M\ \mathsf{value}} \qquad \frac{}{C_i\ \ \mathsf{value}} \qquad \frac{\mathsf{J}_1\ \mathsf{value} \qquad \mathsf{J}_2\ \mathsf{value}}{\mathsf{J}_1 * \mathsf{J}_2\ \mathsf{value}} \qquad \frac{M\ \mathsf{value} \qquad \mathsf{J}\ \mathsf{value}}{M\&\mathsf{J}\ \mathsf{value}}$$

$$\frac{M \rightarrow M'}{M\&\mathsf{J} \rightarrow M'\&\mathsf{J}} \qquad \frac{N_1\ \mathsf{value}\ \ldots\ N_{k-1}\ \mathsf{value} \qquad N_k \rightarrow N_k'}{\begin{array}{c}\mathsf{let}\{(x_1\&s_1)\ \mathsf{be}\ N_1,\ldots,\ (x_k\&s_k)\ \mathsf{be}\ N_k,\ldots\}\ \mathsf{in}\ M \rightarrow\\ \mathsf{let}\{(x_1\&s_1)\ \mathsf{be}\ N_1,\ldots,\ (x_k\&s_k)\ \mathsf{be}\ N_k',\ldots\}\ \mathsf{in}\ M\end{array}}$$

$$\frac{M_1\&\mathsf{J}_1\ \mathsf{value}\ \ldots\ M_i\&\mathsf{J}_\mathsf{i}\ \mathsf{value}}{\begin{array}{c}\mathsf{let}\{(x_1\&s_1)\ \mathsf{be}\ (M_1\&\mathsf{J}_1),\ldots,(x_i\&s_i)\ \mathsf{be}\ (M_i\&\mathsf{J}_\mathsf{i})\}\ \mathsf{in}\ (M\&\mathsf{J}) \rightarrow\\ M[M_1/x_1,\ldots,M_i/x_i]\&\mathsf{J}[\mathsf{J}_1/s_1,\ldots,\mathsf{J}_\mathsf{i}/s_i]\end{array}} \qquad \frac{M \rightarrow M'}{(MN) \rightarrow (M'N)}$$

$$\frac{N \rightarrow N'}{((\lambda x.M)N) \rightarrow ((\lambda x.M)N')} \qquad \frac{N\ \mathsf{value}}{((\lambda x.M)N) \rightarrow [N/x]M}$$

Using the reducibility candidates proof method [7]) we show:

Theorem 3 (Termination Under Small Step Reduction). *With $\rightarrow^*$ being the reflexive transitive closure of $\rightarrow$: for every closed term M and $A \in \mathsf{Prop}$ if $\vdash M : A$ then there exists N value s.t. $\vdash N : A$ and $M \rightarrow^* N$.*

4 A Programming Language View: Dynamic Linking and Separate Compilation

Our type system can be related to programming language design when considering *Foreign Function Interfaces*. This is a typical scenario in which a language

I interfaces another language J which is essentially "black boxed". For example, OCaml code might call C code to perform certain computations. In such cases I is a client and J is a host that provides implementations for an interface utilized by the client. Through software development, often the implementations of such an interface might change (i.e. a new version of the host language, or more dramatically, a complete switch of host language). We want a language design that satisfies two interconnected properties. First, *separate compilation* i.e. when implementations change we do not have to recompile client code and, yet, secondly, *dynamic linking* we want the client code to be linked dynamically to its new "meaning".

We will assume that both languages are functional and based on the lambda calculus. I.e. our interpretation function should have the property $[\![A \supset B]\!]_J = [\![A]\!]_J [\![\supset]\!]_J [\![B]\!]_J$ where $[\![\supset]\!]_J$ is the implication type constructor in J. The specifics of the host J and the concrete implementations are unknown to I but during the linker construction we assume that both languages share some basic types for otherwise typed "communication" of the two languages would be impossible. Simplifying, we consider that the only shared type is (int), i.e. the linker construction assumes $\bar{n} : [\![\mathsf{int}]\!]$ for every integer $n : \mathsf{int}$. Let us now assume source code in I that is interfacing a simple data structure, say an integer stack, with the following signature Σ:

```
using type intstack
  empty: intstack, push: int -> intstack -> intstack,
  pop: intstack -> int
```

And let us consider a simple program in I that is using the signature say,

`pop(push (1+1) empty):int`

This program involves two kinds of computations: a redex $(1 + 1)$ that can be reduced using the internal semantics of the language $1+1 \rightsquigarrow_I 2$ and the signature calls `pop (push 2 empty)` that are to be performed externally in whichever host language implements them. We treat dynamic linkers as "term re-writers" that map a computation to its meaning(s) based on different implementations. In the following we consider Σ to be the signature of the interface. Here are the steps towards the linker construction.

1. Reduce the source code based on the operational semantics of I until it doesn't have a redex:

   $$\Sigma; \bullet \vdash \mathtt{pop(push}\ (1+1)\ \mathtt{Empty)} \rightsquigarrow \mathtt{pop(push}\ 2\ \mathtt{Empty)} : \mathtt{int}$$

2. Contextualize the use of the signature at the final term in step 1:

   $$\Sigma; \mathtt{x}_1 : \mathtt{intstack}, \mathtt{x}_2 : \mathtt{int} \rightarrow \mathtt{intstack} \rightarrow \mathtt{intstack}, \mathtt{x}_3 : \mathtt{intstack} \rightarrow \mathtt{int} \vdash \mathtt{x}_3(\mathtt{x}_2\ 2\ \mathtt{x}_1) : \mathtt{int}$$

3. Rewrite the previous judgment assuming (abstract) implementations for the corresponding missing elements using the "known" specification for the shared elements.

   $$\mathtt{s}_1 : [\![\mathtt{instack}]\!], \mathtt{s}_2 : [\![\mathtt{int} \rightarrow \mathtt{intstack} \rightarrow \mathtt{intstack}]\!], \mathtt{s}_3 : [\![\mathtt{intstack} \rightarrow \mathtt{int}]\!] \vdash \mathtt{s}_3 * (\mathtt{s}_2 * \bar{2} * \mathtt{s}_1) : [\![\mathtt{int}]\!]$$

4. Combine the two previous judgments using the $\Box_{IE}$ rule.

$$\Sigma; \mathtt{x}_1' : \Box\mathtt{intstack}, \mathtt{x}_2' : \Box(\mathtt{int} \to \mathtt{intstack} \to \mathtt{intstack}), \mathtt{x}_3' : \Box(\mathtt{intstack} \to \mathtt{int}) \vdash$$
$$\mathtt{let}\{\mathtt{x}_1\&\mathtt{s}_1 \text{ be } \mathtt{x}_1',\ \mathtt{x}_2\&\mathtt{s}_2 \text{ be } \mathtt{x}_2',\ \mathtt{x}_3\&\mathtt{s}_3 \text{ be } \mathtt{x}_3'\} \text{ in } (\mathtt{x}_3(\mathtt{x}_2\ 2\ \mathtt{x}_1)\ \&\ \mathtt{s}_3 * (\mathtt{s}_2 * \bar{2} * \mathtt{s}_1)) : \Box\mathtt{int}$$

5. Using λ-abstraction three times we obtain the dynamic linker:

$$\Sigma; \circ \vdash$$
$$\mathtt{linker} = \lambda\mathtt{x}_1'.\ \lambda\mathtt{x}_2'.\lambda x_3'.$$
$$\mathtt{let}\{\mathtt{x}_1\&\mathtt{s}_1 \text{ be } \mathtt{x}_1',\ \mathtt{x}_2\&\mathtt{s}_2 \text{ be } \mathtt{x}_2',\ \mathtt{x}_3\&\mathtt{s}_3 \text{ be } \mathtt{x}_3'\} \text{ in}(\mathtt{x}_3(\mathtt{x}_2\ 2\ \mathtt{x}_1)\ \&\ \mathtt{s}_3 * (\mathtt{s}_2 * \bar{2} * \mathtt{s}_1))$$
$$: \Box(\mathtt{instack}) \to \Box(\mathtt{int} \to \mathtt{intstack} \to \mathtt{intstack}) \to \Box(\mathtt{intstack} \to \mathtt{int}) \to \Box\mathtt{int}$$

Let us see how it can be used in the presence of different implementations:

1. Suppose the developer responsible for the implementation of the interface is providing an array based implementation for the stack in some language J i.e. we get references to typechecked code fragments of J as follows[2]:

$$\mathtt{create}() : \mathtt{intarray},\ \mathsf{add\_array} : \mathtt{int}_\mathtt{J} \to_\mathtt{J} \mathtt{intarray} \to_\mathtt{J} \mathtt{intarray}$$
$$\mathsf{pop\_array} : \mathtt{intarray} \to_\mathtt{J} \mathtt{int}$$

2. A unification algorithm check is performed to verify the conformance of the implementations to the signature taking into account fixed type sharing equalities ($[\![\mathsf{int}]\!] = \mathsf{int}_\mathsf{J}$). In our case it produces:

$$[\![\to]\!] = \to_\mathsf{J}, [\![\mathsf{intstack}]\!] = \mathsf{intarray}$$

3. We thus obtain typechecked links using the $\Box_I$ rule. For example:

$$\frac{\Sigma; \circ \vdash \mathsf{push} : \mathtt{int} \to \mathtt{intstack} \to \mathtt{intstack} \quad \bullet \vdash \mathsf{add\_array} : [\![\mathtt{int} \to \mathtt{intstack} \to \mathtt{intstack}]\!]}{\Sigma; \circ \vdash \mathsf{push}\ \&\ \mathsf{add\_array} : \Box(\mathtt{int} \to \mathtt{intstack} \to \mathtt{intstack})}$$

And analogously:

$$\Sigma; \circ \vdash \mathsf{pop}\ \&\ \mathsf{pop\_array} : \Box(\mathtt{intstack} \to \mathtt{int}) \quad \Sigma; \circ \vdash \mathsf{empty}\ \&\mathsf{create}() : \Box\mathtt{intstack}$$

4. Finally we can compute the next step in the computation for the expression applying the linker to the obtained pairings:

$$\Sigma; \bullet\vdash (\mathtt{linker}\ (\mathsf{empty}\ \&\ \mathtt{create}())\ (\mathsf{push}\ \&\ \mathsf{add\_array})\ (\mathsf{pop}\ \&\ \mathsf{pop\_array})) : \Box\mathtt{int}$$

which reduces to:

$$\Sigma; \bullet \vdash \mathtt{let}\{(\mathtt{x}_1\&\mathtt{s}_1) \text{ be } (\mathsf{empty}\ \&\mathsf{create}()),\ (\mathtt{x}_2\&\mathtt{s}_2) \text{ be } (\mathsf{push}\ \&\ \mathsf{add\_array}),\ \ (\mathtt{x}_3\&\mathtt{s}_3) \text{ be } (\mathsf{pop}\ \&\ \mathsf{pop\_array})\}$$
$$\text{in } (\mathtt{x}_3(\mathtt{x}_2\ 2\ \mathtt{x}_1)\ \&\ \mathtt{s}_3 * (\mathtt{s}_2 * \bar{2} * \mathtt{s}_1)) : \Box\mathtt{int}$$

The last expression reduces to (β-reduction for let):

$$\Sigma; \bullet \vdash\ \mathsf{pop}(\mathsf{push}\ 2\ \mathsf{empty})\ \&\ \mathsf{pop\_array} * (\mathsf{add\_array} * \bar{2} * \mathsf{empty}) : \Box\mathtt{int}$$

[2] We have changed the return type of pop to avoid products. This is just for economy and products can easily be handled.

giving exactly the next step of the computation for the source expression. The good news is that the linker computes correctly the next step given any conforming set of implementations. It is easy to see that given a list implementation the very same process would produce a different computation step:

$$\Sigma; \bullet \vdash \texttt{pop(push 2 empty)} \;\&\; \texttt{pop\_list} * (\texttt{Cons} * \bar{2} * []) : \Box\texttt{int}$$

We conclude with some remarks that:

- The construction gives a mechanism of abstractions that works not only over different implementations in the same language but even for implementations in different (applicative) languages.
- We assumed in the example that the two languages are based on the lambda calculus and implement a curried, higher-order function space. It is easy to see that such host satisfies the requirements for the $[\![\bullet]\!]$ (with C_S, C_K being the S, K combinators in λ form and $*$ translating to λ application).
- Often, the host language of a foreign call is not a language that satisfies such specifications. This situation occurs when we have bindings from a functional language to a lower level language[3]. Such cases can be captured by adding conjunction (and pairs), tuning the specifications of J accordingly and loosening the assumption that $[\![\bullet]\!]$ is total on types.
- Introduction of modal types is clearly relative to the $[\![\bullet]\!]$ function on types. It would be interesting to consider examples where $[\![\bullet]\!]$ is realized by non-trivial mappings such as $[\![A \supset B]\!] = !A \multimap B$ from the embedding on intuitionistic logic to intuitionistic linear logic [8]. That will showcase an example of modality that works when lifting to a completely different logic or, correspondingly, to an essentially different computational model.
- Finally, it should be clear from the operational semantics and this example that we did not demand any equalities (or, reduction rules) for the proofs in J, but mere existence of specific terms. This is in accordance to justification logic. Analogously, we did not observe computation in the host language but only the construction of the linkers as program transformers. We were careful, to say that our calculus corresponds to the dynamic linking part of separate compilation. This, of course, does not tell the whole story of program execution in such cases. Foreign function calls, return the control to the client after the result gets calculated in the external language. For example, the execution of the program `pop (push 2 empty) + 2` should "escape" the client to compute the stack calls and then return for the last addition. Our modality is concerned only with passing the control from the client to the host dynamically and, as such, is a K (non-factive) modality. Capturing the continuation of the computation and the return of the control back to the source would require a factive modality and a notion of "reverse" of the mapping $[\![\bullet]\!]$. We would like to explore such an extension in future work.

[3] In this setting the type signature of push would be: int $\times$ intstack $\rightarrow$ instack.

5 Related and Future Work

Directly related work with our calculus, in the same fashion that justification logic and LP [1] are related to modal logic, is [4]. The work in [4] provides a calculus for explicit assignments (substitutions) which is actually a sub-case of Jcalc^- with $[\![\bullet]\!]$ identity. This sub-case captures dynamic linking where the host language is the very same one; such need appears in languages with module mechanisms (i.e. implementation hiding and separate compilation within the very same language). In general, the judgmental approach to modality is heavily influenced by [12]. In a sense, our treatment of validity-as-explicit-provability also generalizes the approach there without having to commit to a "factive" modality. Finally, important results on programming paradigms related to justification logic have been obtained in [2,3]. Immediate future developments would be to interpret modal formulas of higher degree under the same principles. This corresponds to dynamic linking in two or more steps (i.e., when the host becomes itself a client of another interface that is implemented dynamically in a third level and, so on). Some preliminary results towards this direction have been developed in [13].

A Appendix

A.1 Theorems

Theorem 4 (Deduction Theorem for Validity Judgments). *Given any* $\Gamma, A, B \in \mathsf{Prop}_0$ *then* $\Gamma, x : A \vdash B \Longrightarrow [\![\Gamma]\!] \vdash [\![A \supset B]\!]$.

Proof. The proof proceeds by induction on the derivations $\Gamma, A, B \in \mathsf{Prop}_0$. Note that the axiomatization of $[\![\mathsf{Prop}_0]\!]$ derives the sequents:$\Delta \vdash [\![A \supset A]\!]$ for any $\Delta \in [\![\mathsf{Prop}_0]\!]$ (as in combinatory logic the I combinator is derived from SK). This handles the reflection case. The rest of the cases are treated exactly as in the proof of completeness of combinatorial axiomatization with respect to the natural deduction in intuitionistic logic. Note that this theorem cannot be proven without the logical specification Ax_1, Ax_2. I.e. it is exactly the requirements of the logical specification that ensure that all interpretations should be complete with respect to intuitionistic implication.

Lemma 4 ($[\![\bullet]\!]$Lifting Lemma). *Given* $\Gamma, A \in \mathsf{Prop}_0$ *then* $\Gamma \vdash A \Longrightarrow [\![\Gamma]\!] \vdash [\![A]\!]$.

Proof. The proof goes by induction on the derivations trivially for all the cases($\supset_{E_0}$ is treated using the App rule that internalizes Modus ponens). For the $\supset_{I_0}$ the previous theorem has to be used.

Lemma 1 ($\Box$Lifting Lemma). *In* Jcalc^-, *for every* $\Gamma, A \in \mathsf{Prop}_0$ *if* $\Gamma \vdash A$ *then* $[\![\Gamma]\!] \vdash [\![A]\!]$ *and, hence,* $\Box\Gamma \vdash \Box A$.

Proof. Assuming a derivation $\mathcal{D} :: \Gamma \vdash A$ from Lemma 4 there exists corresponding validity derivation $\mathcal{E} :: [\![\Gamma]\!] \vdash [\![A]\!]$. Using the two as premises in the $\Box_{IE}$ with $\Gamma := \Box\Gamma$ we obtain $\Box\Gamma \vdash \Box A$.

From the previous we get:

Lemma 2 (Admissibility of Necessitation). *For $A \in \mathsf{Prop}_0$, if $\circ \vdash A$ then $\circ \vdash \Box A$.*

Theorem 5 (Collapse $\Box$ Lemma). *If $\Box\Gamma \vdash \Box A$ for $\Gamma, A \in \mathsf{Prop}_0$ then $\Gamma \vdash A$.*

Theorem 6 (Weakening). *For the N.D. system of Jcalc^-, with $\Gamma, \Gamma' \in \mathsf{Prop}_0$.*

1. *If $\Gamma \vdash A$ then $\Gamma, \Gamma' \vdash A$.*
2. *If $\Box\Gamma \vdash \Box A$ then $\Box\Gamma, \Box\Gamma' \vdash \Box A$.*

Proof. By induction on derivations for the first item. For the second item, given $\Box\Gamma \vdash \Box A$ by the collapse lemma we get $\Gamma \vdash A$ which by the previous item gives $\Gamma, \Gamma' \vdash A$. Using the lifting lemma we get $[\![\Gamma, \Gamma']\!] \vdash [\![A]\!]$. Using the last two items we and the $\Box$ rule gives the result.

Theorem 7 (Contraction). *For the N.D. system of Jcalc, with $\Gamma, x : A, B \in$ Prop_0*

1. *If $\Gamma, x : A, x' : A, \Gamma' \vdash B$ then $\Gamma, x : A, \Gamma' \vdash B$.*
2. *If $\Box\Gamma, x : \Box A, x' : \Box A, \Box\Gamma' \vdash \Box B$ then $\Box\Gamma, x : \Box A, \Box\Gamma' \vdash \Box B$*

Proof. Similarly with previous theorem.

Theorem 8 (Permutation). *For the N.D. system of Jcalc, with $\Gamma \in \mathsf{Prop}_0$ and $\pi\Gamma$ the collection of permutations of Γ.*

1. *If $\Gamma \vdash A$ and $\Gamma' \in \pi\Gamma$ then $\Gamma' \vdash A$.*
2. *If $\Box\Gamma \vdash \Box A$ then $\pi\Box\Gamma \vdash \Box A$.*

Proof. As in the previous item.

Theorem 9 (Substitution Principle). *The following hold for both kinds of judgments:*

1. *If $\Gamma, x : A \vdash M : B$ and $\Gamma \vdash N : A$ then $\Gamma \vdash M[N/x] : B$*
2. *If $[\![\Gamma]\!], s : [\![A]\!] \vdash \mathsf{J} : [\![B]\!]$ and $[\![\Gamma]\!] \vdash \mathsf{J}^{'} : [\![B]\!]$ then $[\![\Gamma]\!] \vdash \mathsf{J}[\mathsf{J}^{'}/\mathsf{s}][\![B]\!]$*

All previous theorems can actually be stated for proof terms too. We should discuss the following:

Theorem 10 (Deduction Theorem/Emulation of λ abstraction). *If Γ, $A \in \mathsf{Prop}_0$ and $\Gamma, x : A \vdash M : B$ then there exists J s.t. $[\![\Gamma]\!] \vdash \mathsf{J} : [\![A \supset B]\!]$.*

Lemma 5 ($[\![\bullet]\!]$Lifting Lemma for terms). *If $\Gamma, A \in \mathsf{Prop}_0$ and $\Gamma \vdash M : A$ then there exists J s.t. $[\![\Gamma]\!] \vdash \mathsf{J} : [\![A]\!]$.*

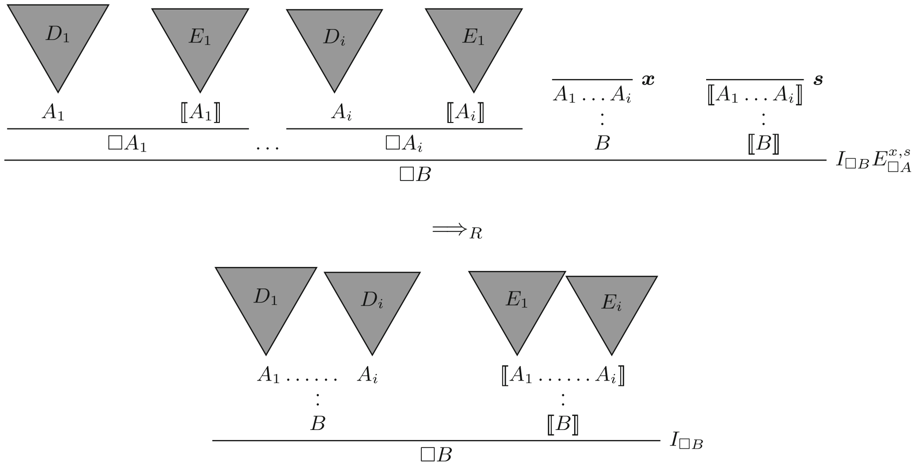
In both theorems the existence of this J, J' is algorithmic following the induction proof.

A.2 Linking on the Function Space

The above mentioned algorithms permit for translating λ abstractions to polynomials of S, K combinators which is a standard result in the literature. We do not give the details here but the translation is syntax driven as it can be seen by the inductive nature of the proofs.

Henceforth, we can generalize the construction in Sect. 4 so that it permits for dynamic linking of functions of the client (with missing implementations) such as λ`n : int.push n empty` dynamically given that the host actually implements a higher-order function space (that is it implements the combinators S, K in, say, own lambda calculus λ^J). Given implementations of `push_impl`, `empty_impl` the linker produces an application expression consisting of `push_impl`, `empty_impl`, S and K. The execution of the target expression will happen in the host after dereferencing $\mathsf{push\_impl}, \mathsf{empty\_impl}$ (dynamic part) and the combinators S, K (constant part) as, say, lambdas (e.g. $K = \lambda^J x.\lambda^J y.x$).

A.3 Gentzen's Reduction Principle for $\Box$(General)



A.4 Notes on the Cut Elimination Proof and Normalization of Natural Deduction

Standardly, we add the bottom type and elimination rule in the natural deduction and show that in Jcalc + $\bot$: $\nvdash \bot$. The addition goes as follows:

$$\frac{}{\bot \in \mathsf{Prop}_0}\ \textsc{Bot} \qquad\qquad \frac{\Gamma \vdash \bot \qquad A \in \mathsf{Prop}}{\Gamma \vdash A}\ E_\bot$$

Our proof strategy follows directly [16]. We construct an intercalation calculus [17] corresponding to the Prop fragment with the following two judgments:

$A \Uparrow$ for "Proposition A has normal deduction".
$A^{\downarrow}$ for "Proposition A is extracted from hypothesis".

This calculus is, essentially, restricting the natural deduction to canonical derivations. The $[\![\mathsf{judgments}]\!]$ are not annotated and are directly ported from the natural deduction since we observe consistency in Prop. The construction is identical to [16] (Chap. 3) for the $\mathsf{Hypotheses}, \mathsf{Coercion}, \supset, \bot$ cases, we add the $\Box$ case.

$$\frac{x : A \downarrow \in \Gamma^{\downarrow}}{\Gamma^{\downarrow} \vdash^{-} A \downarrow}\ \Gamma\text{-HYP} \qquad\qquad \frac{\Gamma^{\downarrow} \vdash^{-} A \downarrow}{\Gamma^{\downarrow} \vdash^{-} A \Uparrow}\ {\downarrow}{\Uparrow}$$

$$\frac{\Gamma^{\downarrow}, x : A \downarrow \vdash^{-} B \Uparrow}{\Gamma^{\downarrow} \vdash^{-} A \supset B \Uparrow} \supset\mathrm{I}^{x} \frac{\Gamma^{\downarrow} \vdash^{-} A \supset B \downarrow \qquad \Gamma^{\downarrow} \vdash^{-} A \Uparrow}{\Gamma^{\downarrow} \vdash^{-} B \downarrow} \supset\mathrm{E} \qquad \frac{\Gamma^{\downarrow} \vdash^{-} \bot \downarrow \qquad A \in \mathsf{Prop}}{\Gamma^{\downarrow} \vdash^{-} A \Uparrow}\ E_{\bot}$$

$$\frac{\Gamma^{\downarrow} \vdash \Box\Gamma' \downarrow \qquad \Gamma'^{\downarrow} \vdash A \Uparrow \qquad [\![\Gamma']\!] \vdash [\![A]\!]}{\Gamma^{\downarrow} \vdash \Box A \Uparrow}\ \Box_{IE}$$

Where $\Gamma^{\downarrow} \vdash \Box\Gamma'$ abbreviates $\forall A_i \in \Gamma'.\ \Gamma^{\downarrow} \vdash \Box A_i \downarrow$. We prove simultaneously by induction:

Theorem 11 (Soundness of Normal Deductions). *The following hold:*

1. *If $\Gamma^{\downarrow} \vdash^{-} A \Uparrow$ then $\Gamma \vdash A$, and*
2. *If $\Gamma^{\downarrow} \vdash^{-} A \downarrow$ then $\Gamma \vdash A$.*

Proof. Simultaneously by induction on derivations.

It is easy to see that this restricted proof system $\nvdash^{-} \bot \Uparrow$. It is hard to show its completeness to the non-restricted natural deduction ($\vdash + \bot_E$ of Jcalc) directly. For that reason we add a rule to make it complete ($\vdash^{+}$) preserving soundness and get a system of Annotated Deductions. We show the correspondence of the restricted system ($\vdash^{-}$) to a cut-free sequent calculus (JSeq), the correspondence of the extended system ($\vdash^{+}$) to $\mathsf{Jseq} + \mathsf{Cut}$ and show cut elimination.[4]

To obtain completeness we add the rule:

$$\frac{\Gamma^{\downarrow} \vdash A \Uparrow}{\Gamma^{\downarrow} \vdash A \downarrow}\ {\Uparrow}{\downarrow}$$

We define $\vdash^{+} := \ \vdash^{-}$ with ${\Uparrow}{\downarrow}\mathsf{Rule}$. We show:

Theorem 12 (Soundness of Annotated Deductions). *The following hold:*

1. *If $\Gamma^{\downarrow} \vdash^{+} A \Uparrow$ then $\Gamma \vdash A$, and*
2. *If $\Gamma^{\downarrow} \vdash^{+} A \downarrow$ then $\Gamma \vdash A$.*

Proof. As previous item.

[4] In reality, the sequent calculus formulation is built exactly upon intuitions on the intercalation calculus. We refer the reader to the references.

Theorem 13 (Completeness of Annotated Deductions). *The following hold:*

1. *If $\Gamma \vdash A$ then $\Gamma \downarrow \vdash^{+} A \Uparrow$, and*
2. *If $\Gamma \vdash A$ then $\Gamma \downarrow \vdash^{+} A \downarrow$.*

Proof. By induction over the structure of the $\Gamma \vdash A$ derivation.

Next we move with devising a sequent calculus formulation corresponding to normal proofs $\Gamma^{\downarrow} \vdash^{-} A \Uparrow$. The calculus that is given in the main body of this theorem. We repeat it here for completeness.

Sequent Calculus ($[\![\mathsf{Prop}_0]\!]$)

$$\Delta \Rightarrow [\![A]\!] := \exists \Delta' \in \pi(\Delta) \text{ s.t } \Delta' \vdash [\![A]\!]$$

where $\pi(\Delta)$ is the collection of wellformed $[\![\mathsf{Prop}_0]\!]$ contexts $\Delta' \vdash [\![\mathsf{wf}]\!]$ with some permutation of the multiset Δ as co–domain.

Sequent Calculus (Prop)

$$\frac{}{\Gamma, A \Rightarrow A}\, Id \qquad \frac{\Gamma, A \supset B, B \Rightarrow C \qquad \Gamma, A \supset B \Rightarrow A}{\Gamma, A \supset B \Rightarrow C} \supset_L \qquad \frac{\Gamma, A \Rightarrow B}{\Gamma \Rightarrow A \supset B} \supset_R \qquad \frac{}{\Gamma, \bot \Rightarrow A} \bot_L$$

$$\frac{\Box\Gamma, \Gamma \Rightarrow A \qquad [\![\Gamma]\!] \Rightarrow [\![A]\!]}{\Box\Gamma \Rightarrow \Box A} \Box_{LR}$$

We want to show correspondence of the sequent calculus w.r.t normal proofs ($\vdash^{-}$). Two lemmas are required to show soundness.

Lemma 6 (Substitution principle for extractions). *The following hold:*

1. *If $\Gamma_1^{\downarrow}, x : A^{\downarrow}, \Gamma_2^{\downarrow} \vdash^{-} B \Uparrow$ and*
   $\Gamma_1^{\downarrow} \vdash^{-} A \Uparrow$ then $\Gamma_1^{\downarrow}, \Gamma_2^{\downarrow} \vdash^{-} B \Uparrow$
2. *If $\Gamma_1^{\downarrow}, x : A^{\downarrow}, \Gamma_2^{\downarrow} \vdash^{-} B \downarrow$ and $\Gamma_1^{\downarrow} \vdash^{-} A \downarrow$ then $\Gamma_1^{\downarrow}, \Gamma_2^{\downarrow} \vdash^{-} B \Uparrow$*

Proof. Simultaneously by induction on the derivations $A \downarrow$ and $A \Uparrow$.

And making use of the previous we can show, with ($\lfloor A$ defined previously):

Lemma 7 (Collapse principle for normal deductions). *The following hold:*

1. *If $\Gamma^{\downarrow}, \vdash^{-} A \Uparrow$ then $\lfloor \Gamma^{\downarrow} \vdash^{-} \lfloor A \Uparrow$ and,*
2. *If $\Gamma^{\downarrow} \vdash^{-} A \downarrow$ then $\lfloor \Gamma^{\downarrow} \vdash^{-} \lfloor A \downarrow$*

Using the previous lemmas and by induction we can show:

Theorem 14 (Soundness of the Sequent Calculus). *If $\Gamma \Rightarrow B$ then $\Gamma^{\downarrow} \vdash^{-} B \Uparrow$.*

Theorem 15 (Soundness of the Sequent Calculus with Cut). *If $\Gamma \Rightarrow^{+} B$ then $\Gamma^{\downarrow} \vdash^{+} B \Uparrow$.*

Next we define the $\Gamma \Rightarrow^{+} A$ as $\Gamma \Rightarrow A$ plus the rule:

$$\frac{\Gamma \Rightarrow^{+} A \qquad \Gamma, A \Rightarrow^{+} B}{\Gamma \Rightarrow^{+} B}\ \textsc{Cut}$$

Proof. As before. The cut rule case is handled by the $\Uparrow\downarrow$ and substitution for extractions principle showcasing that the correspondence of the cut rule to the coercion from normal to extraction derivations.

Standard structural properties (*Weakening, Contraction*) to show completeness. We do not show these here but they hold.

Theorem 16 (Completeness of the Sequent Calculus). *The following hold:*

1. *If* $\Gamma^{\downarrow} \vdash^{-} B \Uparrow$ *then* $\Gamma \Rightarrow B$ *and,*
2. *If* $\Gamma^{\downarrow} \vdash^{-} A \downarrow$ *and* $\Gamma, A \Rightarrow B$ *then* $\Gamma \Rightarrow B$

Proof. Simultaneously by induction on the given derivations making use of the structural properties.
Similarly we show for the extended systems.

Theorem 17 (Completeness of the Sequent Calculus with Cut). *The following hold:*

1. *If* $\Gamma^{\downarrow} \vdash^{+} B \Uparrow$ *then* $\Gamma \Rightarrow^{+} B$ *and,*
2. *If* $\Gamma^{\downarrow} \vdash^{+} A \downarrow$ *and* $\Gamma, A \Rightarrow^{+} B$ *then* $\Gamma \Rightarrow^{+} B$.

Proof. As before. The extra case is handled by the Cut rule.

After establishing the correspondence of $\vdash^{-}$ with $\Rightarrow$ and of $\vdash^{+}$ with $\Rightarrow^{+}$ we move on with:

Theorem 18 (Admissibility of Cut). *If* $\Gamma \Rightarrow A$ *and* $\Gamma, A \Rightarrow B$ *then* $\Gamma \Rightarrow B$.

The proof is by triple induction on the structure of the formula, and the given derivations and we leave it for a technical report. This gives easily:

Theorem 19 (Cut Elimination). *If* $\Gamma \Rightarrow^{+} A$ *then* $\Gamma \Rightarrow A$.

Which in turn gives us:

Theorem 20 (Normalization for Natural Deduction). *If* $\Gamma \vdash A$ *then* $\Gamma^{\downarrow} \vdash^{-} A \Uparrow$

Proof. From assumption $\Gamma \vdash A$ which by Theorem 13 gives $\Gamma \vdash^{+} A \Uparrow$. By Theorem 17 and Cut Elimination we obtain $\Gamma \Rightarrow A$ which by Theorem 14 completes the proof.

As a result we obtain:

Theorem 2 (Consistency of Jcalc^{-}). $\nvdash \perp$

Proof. By contradiction, assume $\vdash \perp$ then $\Rightarrow \perp$ which is not possible.

References

1. Artemov, S.: Explicit provability and constructive semantics. Bull. Symbolic Logic **7**(1), 1–36 (2001)
2. Artemov, S., Bonelli, E.: The intensional lambda calculus. In: Artemov, S., Nerode, A. (eds.) LFCS 2007. LNCS, vol. 4514, pp. 12–25. Springer, Heidelberg (2007)
3. Bavera, F., Bonelli, E.: Justification logic and audited computation. J. Logic Comput., p. exv037 (2015)
4. Bellin, G., de Paiva, V.C., Ritter, E.: Extended curry-howard correspondence for a basic constructive modal logic (2001). (preprint; presented at M4M–2, ILLC, UvAmsterdam)
5. Bierman, G.M., de Paiva, V.C.V.: On an intuitionistic modal logic. Stud. Logica **65**(3), 383–416 (2000)
6. Gentzen, G.: Untersuchungen über das logische schließen I. Math. Z. **39**(1), 176–210 (1935)
7. Girard, J.Y., Taylor, P., Lafont, Y.: Proofs and Types. Cambridge University Press, New York (1989). ISBN 0-521-37181-3
8. Girard, J.-Y.: Linear logic. Theor. Comput. Sci. **50**(1), 1–101 (1987)
9. Kavvos, A.: System k: a simple modal λ-calculus. arXiv preprint arXiv:1602.04860 (2016)
10. Lambek, J.: Deductive systems and categories. Theory Comput. Syst. **2**(4), 287–318 (1968)
11. Lambek, J.: Deductive systems and categories II. standard constructions and closed categories. In: Category Theory, Homology Theory and Their Applications I, pp. 76–122. Springer, New York (1969)
12. Pfenning, F., Davies, R.: A judgmental reconstruction of modal logic. Math. Struct. Comput. Sci. **11**(04), 511–540 (2001). ISSN 0960–1295
13. Pouliasis, K., Primiero, G.: J-calc: a typed lambda calculus for intuitionistic justification logic. Electron. Notes Theor. Comput. Sci. **300**, 71–87 (2014). ISSN 1571–0661
14. Prawitz, D.: Natural Deduction: A Proof-Theoretical study. Almquist & Wiksell, Stockholm (1965)
15. Troelstra, A.S., Van Dalen, D.: Constructivism in Mathematics: An Introduction. North-Holland, Amsterdam (1988)
16. Pfenning, F.: Automated theorem proving (2004). http://www.cs.cmu.edu/~fp/courses/atp/handouts/atp.pdf
17. Sieg, W., Byrnes, J.: Normal natural deduction proofs (in classical logic). Stud. Logica **60**(1), 67–106 (1998)

On the Formalization of Some Results of Context-Free Language Theory

Marcus Vinícius Midena Ramos[1,2](✉), Ruy J.G.B. de Queiroz[2], Nelma Moreira[3], and José Carlos Bacelar Almeida[4,5]

[1] Universidade Federal do Vale do São Francisco, Juazeiro, Bahia, Brazil
marcus.ramos@univasf.edu.br,mvmr@cin.ufpe.br
[2] Universidade Federal de Pernambuco, Recife, Pernambuco, Brazil
ruy@cin.ufpe.br
[3] Universidade do Porto, Porto, Portugal
nam@dcc.fc.up.pt
[4] Universidade do Minho, Braga, Portugal
jba@di.uminho.pt
[5] HASLab-INESC TEC, Porto, Portugal

Abstract. This work describes a formalization effort, using the Coq proof assistant, of fundamental results related to the classical theory of context-free grammars and languages. These include closure properties (union, concatenation and Kleene star), grammar simplification (elimination of useless symbols, inaccessible symbols, empty rules and unit rules), the existence of a Chomsky Normal Form for context-free grammars and the Pumping Lemma for context-free languages. The result is an important set of libraries covering the main results of context-free language theory, with more than 500 lemmas and theorems fully proved and checked. This is probably the most comprehensive formalization of the classical context-free language theory in the Coq proof assistant done to the present date, and includes the important result that is the formalization of the Pumping Lemma for context-free languages.

Keywords: Context-free language theory · Language closure · Grammar simplification · Chomsky Normal Form · Pumping Lemma · Formalization · Coq

1 Introduction

This work is about the mathematical formalization of an important subset of the context-free language theory, including some of its most important results such as the Chomsky Normal Form and the Pumping Lemma.

The formalization has been done in the Coq proof assistant. This represents a novel approach towards formal language theory, specially context-free language theory, as virtually all textbooks, general literature and classes on the subject rely on an informal (traditional) mathematical approach. The objective of this work, thus, is to elevate the status of this theory to new levels in accordance

J. Väänänen et al. (Eds.): WoLLIC 2016, LNCS 9803, pp. 338–357, 2016.
DOI: 10.1007/978-3-662-52921-8_21

with the state-of-the-art in mathematical accuracy, which is accomplished with the use of interactive proof assistants.

The choice of using Coq comes from its maturity, its widespread use and the possibility of extracting certified code from proofs. HOL4 and Agda have also been used in the formalization of context-free language theory (see Sect. 7), however they do not comply to at least one of these criteria.

The formalization is discussed in Sects. 2 (method used in most of the formalization), 3 (closure properties), 4 (grammar simplification), 5 (CNF - Chomsky Normal Form) and 6 (Pumping Lemma). The main definitions used in the formalization are presented in Appendix A. The library on binary trees and their relation to CNF derivations is briefly discussed in Appendix B.

Formal language and automata theory formalization is not a completely new area of research. In Sect. 7, a summary of these accomplishments is presented. Most of the formalization effort on general formal language theory up to date has been dedicated to the regular language theory, and not so much to context-free language theory. Thus, this constitutes the motivation for the present work. Final conclusions are presented in Sect. 8.

In order to follow this paper, the reader is required to have basic knowledge of Coq and context-free language theory. The recommended starting point for Coq is the book by Bertot and Castéran [1]. Background on context-free language theory can be found in [2] or [3], among others. A more detailed and complete discussion of the results of this work can be found in [4]. The source files of the formalization are available for download from [5].

2 Method

Except for the Pumping Lemma, the present formalization is essentially about context-free grammar manipulation, that is, about the definition of a new grammar from a previous one (or two), such that it satisfies some very specific properties. This is exactly the case when we define new grammars that generate the union, concatenation, closure (Kleene star) of given input grammar(s). Also, when we create new grammars that exclude empty rules, unit rules, useless symbols and inaccessible symbols from the original ones. Finally, it is also the case when we construct a new grammar that preserves the language of the original grammar and still observes the Chomsky Normal Form.

In the general case, the mapping of grammar $g_1 = (V_1, \Sigma, P_1, S_1)$ into grammar $g_2 = (V_2, \Sigma, P_2, S_2)$ requires the definition of a new set of non-terminal symbols N_2, a new set of rules P_2 and a new start symbol S_2. Similarly, the mapping of grammar $g_1 = (V_1, \Sigma, P_1, S_1)$ and grammar $g_2 = (V_2, \Sigma, P_2, S_2)$ into grammar $g_3 = (V_3, \Sigma, P_3, S_3)$ requires the definition of a new set of non-terminal symbols N_3, a new set of rules P_3 and a new start symbol S_3.

For all cases of grammar manipulation, we consider that the original and final sets of terminal symbols are the same. Also, we have devised the following common approach to constructing the desired grammars:

1. Depending on the case, inductively define the type of the new non-terminal symbols; this will be important, for example, when we want to guarantee that the start symbol of the grammar does not appear in the right-hand side of any rule or when we have to construct new non-terminals from the existing ones; the new type may use some (or all) symbols of the previous type (via mapping), and also add new symbols;
2. Inductively define the rules of the new grammar, in a way that it allows the construction of the proofs that the resulting grammar has the required properties; these new rules will likely make use of the new non-terminal symbols described above; the new definition may exclude some of the original rules, keep others (via mapping) and still add new ones;
3. Define the new grammar by using the new type of non-terminal symbols and the new rules; define the new start symbol (which might be a new symbol or an existing one) and build a proof of the finiteness of the set of rules for this new grammar;
4. State and prove all the lemmas and theorems that will assert that the newly defined grammar has the desired properties;
5. Consolidate the results within the same scope and finally with the previously obtained results.

In the following sections, this approach will be explored with further detail for each main result. The definitions of Appendix A are used throughout.

3 Closure Properties

The basic operations of union, concatenation and closure for context-free grammars are described in a rather straightforward way. These operations provide new context-free grammars that generate, respectively, the union, concatenation and the Kleene star closure of the language(s) generated by the input grammar(s)[1].

For the union, given two arbitrary context-free grammars g_1 and g_2, we want to construct g_3 such that $L(g_3) = L(g_1) \cup L(g_2)$ (that is, the language generated by g_3 is the union of the languages generated by g_1 and g_2).

The classical informal proof constructs $g_3 = (V_3, \Sigma, P_3, S_3)$ from g_1 and g_2 such that $N_3 = N_1 \cup N_2 \cup \{S_3\}$ and $P_3 = P_1 \cup P_2 \cup \{S_3 \rightarrow S_1, S_3 \rightarrow S_2\}$. With the appropriate definitions for the new set of non-terminal symbols, the new set of rules and the new start symbol, we are able to construct a new grammar `g_uni` such that `g3 = g_uni g1 g2`.

For the concatenation, given two arbitrary context-free grammars g_1 and g_2, we want to construct g_3 such that $L(g_3) = L(g_1) \cdot L(g_2)$ (that is, the language generated by g_3 is the concatenation of the languages generated by g_1 and g_2).

The classical informal proof constructs $g_3 = (V_3, \Sigma, P_3, S_3)$ from g_1 and g_2 such that $N_3 = N_1 \cup N_2 \cup \{S_3\}$ and $P_3 = P_1 \cup P_2 \cup \{S_3 \rightarrow S_1 S_2\}$. With the appropriate definitions for the new set of non-terminal symbols, the new set of

[1] The results of this section are available in libraries `union.v`, `concatenation.v` and `closure.v`.

rules and the new start symbol, we are able to construct a new grammar `g_cat` such that `g3 = g_cat g1 g2`.

For the Kleene star, given an arbitrary context-free grammar g_1, we want to construct g_2 such that $L(g_2) = (L(g_1))^*$ (that is, the language generated by g_2 is the reflexive and transitive concatenation (Kleene star) of the language generated by g_1).

The classical informal proof constructs $g_2 = (V_2, \Sigma, P_2, S_2)$ from g_1 such that $N_2 = N_1 \cup N_2 \cup \{S_2\}$ and $P_2 = P_1 \cup P_2 \cup \{S_2 \to S_2 S_1, S_2 \to S_1\}$. With the appropriate definitions for the new set of non-terminal symbols, the new set of rules and the new start symbol, we are able to construct a new grammar `g_uni` such that `g2 = g_clo g1`.

Although simple in their structure, it must be proved that the definitions `g_uni`, `g_cat` and `g_clo` always produce the correct result. In other words, these definitions must be "certified", which is one of the main goals of formalization. In order to accomplish this, we must first state the theorems that capture the expected semantics of these definitions. Finally, we have to derive proofs of the correctness of these theorems.

This can be done with a pair of theorems for each grammar definition: the first relates the output to the inputs, and the other one does the converse, providing assumptions about the inputs once an output is generated. This is necessary in order to guarantee that the definitions do only what one would expect, and no more.

For union, we prove (considering that g_3 is the union of g_1 and g_2 and S_3, S_1 and S_2 are, respectively, the start symbols of g_3, g_1 and g_2): $\forall g_1,\, g_2,\, s_1,\, s_2, (S_1 \Rightarrow^*_{g_1} s_1 \to S_3 \Rightarrow^*_{g_3} s_1) \wedge (S_2 \Rightarrow^*_{g_2} s_2 \to S_3 \Rightarrow^*_{g_3} s_2)$. For the converse of union we prove: $\forall s_3, (S_3 \Rightarrow^*_{g_3} s_3) \to (S_1 \Rightarrow^*_{g_1} s_3) \vee (S_2 \Rightarrow^*_{g_2} s_3)$. Together, the two theorems represent the semantics of the context-free grammar union operation.

For concatenation, we prove (considering that g_3 is the concatenation of g_1 and g_2 and S_3, S_1 and S_2 are, respectively, the start symbols of g_3, g_1 and g_2): $\forall g_1,\, g_2,\, s_1,\, s_2, (S_1 \Rightarrow^*_{g_1} s_1) \wedge (S_2 \Rightarrow^*_{g_2} s_2) \to (S_3 \Rightarrow^*_{g_3} s_1 \cdot s_2)$. For the converse of concatenation, we prove: $\forall g_3,\, s_3, (S_3 \Rightarrow^*_{g_3} s_3) \to \exists s_1,\, s_2, (S_1 \Rightarrow^*_{g_1} s_1) \wedge (S_2 \Rightarrow^*_{g_2} s_2) \wedge (s_3 = s_1 \cdot s_2)$.

For closure, we prove (considering that g_2 is the Kleene star of g_1 and S_2 and S_1 are, respectively, the start symbols of g_2 and g_1): $\forall g_1,\, s_1,\, s_2, (S_2 \Rightarrow^*_{g_2} \epsilon) \wedge ((S_2 \Rightarrow^*_{g_2} s_2) \wedge (S_1 \Rightarrow^*_{g_1} s_1) \to S_2 \Rightarrow^*_{g_2} s_2 \cdot s_1)$. Finally: $\forall s_2, (S_2 \Rightarrow^*_{g_2} s_2) \to (s_2 = \epsilon) \vee (\exists\, s_1,\, s_2' \,|\, (s_2 = s_2' \cdot s_1) \wedge (S_2 \Rightarrow^*_{g_2} s_2') \wedge (S_1 \Rightarrow^*_{g_1} s_1))$.

In all three cases, the correctness proofs are straightforward and follow closely the informal proofs available in most textbooks. The formalization consists of a set of short and readable lemmas, except for the details related to mappings involving sentential forms. Since every grammar is defined with a different set of non-terminal symbols (i.e. uses a different type for these symbols), sentential forms from one grammar have to "mapped" to sentential forms of another grammar in order to be usable and not break the typing rules of Coq. This required a lot of effort in order to provide and use the correct mapping functions, and also to cope with it during proof construction. This is something that we don't see in informal proofs, and is definitely a burden when doing the formalization.

The completeness proofs, on the other hand, resulted in single lemmas with reasonably long scripts (~280 lines) in each case. Intermediate lemmas were not easily identifiable as in the correctness cases and, besides the initial induction of predicate `derives`, the long list of various types of case analysis increased the complexity of the scripts, which are thus more difficult to read.

It should be added that the closure operations considered here can be explained in a very intuitive way (either with grammars or automata), and for this reason many textbooks don't even bother going into the details with mathematical reasoning. Because of this, our formalization was a nice exercise in revealing how simple and intuitive proofs can grow in complexity with many details not considered before.

4 Simplification

The definition of a context-free grammar, and also the operations specified in the previous section, allow for the inclusion of symbols and rules that may not contribute to the language being generated. Besides that, context-free grammars may also contain rules that can be substituted by equivalent smaller and simpler ones. Unit rules, for example, do not expand sentential forms (instead, they just rename the symbols in them) and empty rules can cause them to contract. Although the appropriate use of these features can be important for human communication in some situations, this is not the general case, since it leads to grammars that have more symbols and rules than necessary, making difficult its comprehension and manipulation. Thus, simplification is an important operation on context-free grammars.

Let G be a context-free grammar, $L(G)$ the language generated by this grammar and ϵ the empty string. Different authors use different terminology when presenting simplification results for context-free grammars. In what follows, we adopt the terminology and definitions of [2].

Context-free grammar simplification comprises the manipulation of rules and symbols, as described below:

1. An *empty rule* $r \in P$ is a rule whose right-hand side β is empty (e.g. $X \rightarrow \epsilon$). We prove that for all G there exists G' such that $L(G) = L(G')$ and G' has no empty rules, except for a single rule $S \rightarrow \epsilon$ if $\epsilon \in L(G)$; in this case, S (the initial symbol of G') does not appear on the right-hand side of any rule of G';
2. A *unit rule* $r \in P$ is a rule whose right-hand side β contains a single non-terminal symbol (e.g. $X \rightarrow Y$). We prove that for all G there exists G' such that $L(G) = L(G')$ and G' has no unit rules;
3. A symbol $s \in V$ is *useful* ([2], p. 116) if it is possible to derive a string of terminal symbols from it using the rules of the grammar. Otherwise, s is called an *useless symbol*. A useful symbol s is one such that $s \Rightarrow^* \omega$, with $\omega \in \Sigma^*$.

Naturally, this definition concerns mainly non-terminals, as terminals are trivially useful. We prove that for all G such that $L(G) \neq \emptyset$, there exists G' such that $L(G) = L(G')$ and G' has no useless symbols;

4. A symbol $s \in V$ is *accessible* ([2], p. 119) if it is part of at least one string generated from the root symbol of the grammar. Otherwise, it is called an *inaccessible symbol*. An accessible symbol s is one such that $S \Rightarrow^* \alpha s \beta$, with $\alpha, \beta \in V^*$. We prove that for all G there exists G' such that $L(G) = L(G')$ and G' has no inaccessible symbols.

Finally, we prove a unification result: that for all G, if G is non-empty, then there exists G' such that $L(G) = L(G')$ and G' has no empty rules (except for one, if G generates the empty string), no unit rules, no useless symbols, no inaccessible symbols and the start symbol of G' does not appear on the right-hand side of any other rule of G'.[2]

In all these four cases and the five grammars that are discussed next (namely `g_emp`, `g_emp'`, `g_unit`, `g_use` and `g_acc`), the proof of `rules_finite` is based on the proof of the corresponding predicate for the argument grammar. Thus, all new grammars satisfy the `cfg` specification and are finite as well.

Result (1) is achieved in two steps. In the first step, we map grammar g_1 into an equivalent grammar g_2 (except for the empty string), which is free of empty rules and whose start symbol does not appear on the right-hand side of any rule. This is done by eliminating empty rules and substituting rules that have nullable symbols in the right-hand side by a set of equivalent rules. Next, we use g_2 to map g_1 into g_3 which is fully equivalent to g_1 (including the empty string if this is the case).

Observe that resulting grammar (`g_emp g` or g_2) does not generate the empty string, even if `g` (or g_1) does so. The second step, thus, consists of constructing g_3 such that it generates all the strings of g_2 plus the empty string if g_1 does so. This is done by conditionally adding a rule that maps the start symbol to the empty string.

We define `g_emp' g` (or g_3) such that `g_emp' g` generates the empty string if `g` generates the empty string. This is done by stating that every rule from `g_emp g` is also a rule of `g_emp' g` and also by adding a new rule that allow `g_emp' g` to generate the empty string directly if necessary.

The proof of the correctness of the previous definitions is achieved through the following Coq theorem:

```
Theorem g_emp'_correct: ∀ g: cfg non_terminal terminal,
g_equiv (g_emp' g) g ∧ (produces_empty g → has_one_empty_rule (g_emp' g)) ∧
(∼ produces_empty g → has_no_empty_rules (g_emp' g)) ∧
start_symbol_not_in_rhs (g_emp' g).
```

New predicates are used in this statement: `produces_empty`, for a grammar that produces the empty string, `has_one_empty_rule`, to describe a grammar that has a single empty rule among its set of rules (one whose left-hand side

[2] The results of this section are available in libraries `emptyrules.v`, `unitrules.v`, `useless.v`, `inaccessible.v` and `simplification.v`.

is the initial symbol), `has_no_empty_rules` for a grammar that has no empty rules at all and `start_symbol_not_in_rhs` to state that the start symbol does not appear in the right-hand side of any rule of the argument grammar.

The proof of `g_emp'_correct` is reduced to the proof of the equivalence of grammars `g` and `g_emp g`. The most complex part of this formalization, by far, is to prove this equivalence, as expressed by lemmas `derives_g_g_emp` and `derives_g_emp_g`. These lemmas state, respectively, that every sentential form of `g` (except for the empty string) is also generated by `g_emp` and that every sentential form of `g_emp` is also generated by `g`. While the second case was relatively straightforward, the first proved much more difficult. This happens because the application of a rule of `g` can cause a non-terminal symbol to be eliminated from the sentential form (if it is an empty rule), and for this reason we have to introduce a new structure and do many case analysis in the sentential form of `g` in order to determine the corresponding new rule of `g_emp g` that has to be used in the derivation. We believe that the root of this difficulty was the desire to follow strictly the informal proof of [2] (Theorem 5.1.5), which depends on an intuitive lemma (Lemma 3.1.5), however not easily formalizable. Probably for this reason, the solution constructed in our formalization is definitely not easy or readable, and this motivates the continued search for a simpler and more elegant one.

Result (2) is achieved in only one step. We first define the relation `unit` such that, for any two non-terminal symbols X and Y, `unit X Y` is true when $X \Rightarrow^{+} Y$ ([2], p. 114). This means that Y can be derived from X by the use of one or more unit rules.

The mapping of grammar g_1 into an equivalent grammar g_2 such that g_2 is free of unit rules consists basically of keeping all non-unit rules of g_1 and creating new rules that reproduce the effect of the unit rules that were left behind. No new non-terminal symbols are necessary. The correctness of `g_unit` comes from the following theorem:

```
Theorem g_unit_correct: ∀ g: cfg non_terminal terminal,
g_equiv (g_unit g) g ∧ has_no_unit_rules (g_unit g).
```

The predicate `has_no_unit_rules` states that the argument grammar has no unit rules at all.

We find important similarities in the proofs of grammar equivalence for the elimination of empty rules (lemma `g_emp'_correct`) and the elimination of unit rules (lemma `g_unit_correct`). In both cases, going backwards (from the new to the original grammar) was relatively straightforward and required no special machinery. On the other hand, going forward (from the original to the new grammar) proved much more complex and required new definitions, functions and lemmas in order to complete the corresponding proofs.

The proof that every sentence generated by the original grammar is also generated by the transformed grammar (without unit rules) requires the introduction of the `derives3` predicate specially for this purpose. Because this definition represents the derivation of sentences directly from a non-terminal symbol, it is possible to abstract over the use of unit rules. Since `derives3` is a mutual inductive definition, we had to create a specialized induction principle (derives3_ind_2) and use it explicitly, which resulted in more complex proofs.

Result (3) is obtained in a single and simple step, which consists of inspecting all rules of grammar g_1 and eliminating the ones that contain useless symbols in either the left or right-hand side. The other rules are kept in the new grammar g_2. Thus, $P_2 \subseteq P_1$. No new non-terminals are required.

The g_use definition, of course, can only be used if the language generated by the original grammar is not empty, that is, if the start symbol of the original grammar is useful. If it were useless then it would be impossible to assign a root to the grammar and the language would be empty. The correctness of the useless symbol elimination operation is certified by proving theorem g_use_correct, which states that every context-free grammar whose start symbol is useful generates a language that can also be generated by an equivalent context-free grammar whose symbols are all useful.

```
Theorem g_use_correct: ∀ g: cfg non_terminal terminal,
non_empty g → g_equiv (g_use g) g ∧ has_no_useless_symbols (g_use g).
```

The predicates non_empty, and has_no_useless_symbols used above assert, respectively, that grammar g generates a language that contains at least one string (which in turn may or may not be empty) and the grammar has no useless symbols at all.

Result (4) is similar to the previous case: the rules of the original grammar g_1 are kept in the new grammar g_2 as long as their left-hand consist of accessible non-terminal symbols (by definition, if the left-hand side is accessible then all the symbols in the right-hand side of the same rule are also accessible). If this is not the case, then the rules are left behind. Thus, $P_2 \subseteq P_1$.

The correctness of the inaccessible symbol elimination operation is certified by proving theorem g_acc_correct, which states that every context-free grammar generates a language that can also be generated by an equivalent context-free grammar whose symbols are all accessible.

```
Theorem g_acc_correct: ∀ g: cfg non_terminal terminal,
g_equiv (g_acc g) g ∧ has_no_inaccessible_symbols (g_acc g).
```

In a way similar to has_no_useless_symbols, the absence of inaccessible symbols in a grammar is expressed by predicate has_no_inaccessible_symbols used above.

The proof of g_acc_correct is also natural when compared to the arguments of the informal proof. It has only 384 lines on Coq script and, despite the similarities between it and the proof of g_use_correct, it is still ~40% shorter than that. This is partially due to a difference in the definitions of g_use_rules and g_acc_rules: in the first case, in order to be eligible as a rule of g_use, a rule of g must provably consist only of useful symbols in both the left and right-hand sides; in the second, it is enough to prove that only the left-hand side is accessible (the rest is consequence of the definition). Since we have a few uses of the constructors of these definitions, the simpler definition of g_acc_rules resulted in simpler and shorter proofs. As a matter of fact, it should be possible to do something similar to the definition of g_use_rules, since the left-hand side of a rule is automatically useful once all the symbols in the right-hand side are proved useful (a consequence of the definition). This will be considered in a future review of the formalization.

So far we have only considered each simplification strategy independently of the others. If one wants to obtain a new grammar that is simultaneously free of empty and unit rules, and of useless and inaccessible symbols, it is not enough to consider the previous independent results: it is necessary to establish a suitable order to apply these simplifications, in order to guarantee that the final result satisfies all desired conditions. Then, it is necessary to prove that the claims do hold.

For the order, we should start with (i) the elimination of empty rules, followed by (ii) the elimination of unit rules. The reason for this is that (i) might introduce new unit rules in the grammar, and (ii) will surely not introduce empty rules, as long as the original grammar is free of them (except for $S \to \epsilon$, in which case S, the initial symbol of the grammar, must not appear on the right-hand side of any rule). Then, elimination of useless and inaccessible symbols (in either order) is the right thing to do, since they only remove rules from the original grammar (which is specially important because they do not introduce new empty or unit rules). The formalization of this result is captured in the following theorem:

```
Theorem g_simpl_ex_v1: ∀ g: cfg non_terminal terminal, non_empty g →
 ∃ g': cfg (non_terminal' non_terminal) terminal, g_equiv g' g ∧
 has_no_inaccessible_symbols g' ∧ has_no_useless_symbols g' ∧
(produces_empty g → has_one_empty_rule g') ∧
(∼ produces_empty g → has_no_empty_rules g') ∧
 has_no_unit_rules g' ∧ start_symbol_not_in_rhs g'.
```

The proof of g_simpl_ex_v1 demands auxiliary lemmas to prove that the characteristics of the initial transformations are preserved by the following ones. For example, that all of the unit rules elimination, useless symbol elimination and inaccessible symbol elimination operations preserve the characteristics of the empty rules elimination operation.

5 Chomsky Normal Form

The Chomsky Normal Form (CNF) theorem, proposed and proved by Chomsky in [6], asserts that $\forall\ G = (V, \Sigma, P, S),\ \exists\ G' = (V', \Sigma, P', S') \mid L(G) = L(G') \wedge \forall\ (\alpha \to_{G'} \beta) \in P', (\beta \in \Sigma) \vee (\beta \in N \cdot N)$.

That is, every context-free grammar can be converted to an equivalent one whose rules have only one terminal symbol or two non-terminal symbols in the right-hand side. Naturally, this is valid only if G does not generate the empty string. If this is the case, then the grammar that has this format, plus a single rule $S' \to_G \epsilon$, is also considered to be in the Chomsky Normal Form, and generates the original language, including the empty string. It can also be assured that in either case the start symbol of G' does not appear on the right-hand side of any rule of G'.

The existence of a CNF can be used for a variety of purposes, including to prove that there is an algorithm to decide whether an arbitrary context-free language accepts an arbitrary string, and to test if a language is not context-free (using the Pumping Lemma for context-free languages, which can be proved with the help of CNF grammars).

The idea of mapping G into G' consists of creating a finite number of new non-terminal symbols and new rules, in the following way:

1. For every terminal symbol σ that appears in the right-hand side of a rule $r = \alpha \rightarrow_G \beta_1 \cdot \sigma \cdot \beta_2$ of G, create a new non-terminal symbol $[\sigma]$, a new rule $[\sigma] \rightarrow_{G'} \sigma$ and substitute σ for $[\sigma]$ in r;
2. For every rule $r = \alpha \rightarrow_G N_1 N_2 \cdots N_k$ of G, where N_i are all non-terminals, create a new set of non-terminals and a new set of rules such that $\alpha \rightarrow_{G'} N_1[N_2 \cdots N_k], [N_2 \cdots N_k] \rightarrow_{G'} N_2[N_3 \cdots N_k], \cdots, [N_{k-2}N_{k-1}N_k] \rightarrow_{G'} N_{k-2}$ $[\ N_{k-1}N_k\ ], [N_{k-1}N_k] \rightarrow_{G'} N_{k-1}N_k$.

Case (1) substitutes all terminal symbols of the grammar for newly created non-terminal symbols. Case (2) splits rules that have three or more non-terminal symbols on the right-hand side by a set of rules that have only two non-terminal symbols in the right-and side. Both changes preserve the language of the original grammar.

It is clear from above that the original grammar must be free of empty and unit rules in order to be converted to a CNF equivalent. Also, it is desirable that the original grammar contains no useless and no inaccessible symbols, besides assuring that the start symbol does not appear on the right-hand side of any rule. Thus, it will be required that the original grammar be first simplified according to the results of Sect. 4.

Given the original grammar g_1, we construct two new grammars g_2 and g_3. The first generates the same set of sentences of g_1, except for the empty string, and the second includes the empty string:

1. Construct g_2 such that $L(g_2) = L(g_1) - \epsilon$;
2. Construct g_3 (using g_2) such that $L(g_3) = L(g_2) \cup \{\epsilon\}$.

Then, either g_2 or g_3 will be used to prove the existence of a CNF grammar equivalent to g_1.

For step 1, the construction of g_2 (that is, `g_cnf g`) is more complex, as we need to substitute terminals for new non-terminals, introduce new rules for these non-terminals and also split the rules with three or more symbols on the right-hand side.

Next, we prove that g_2 is equivalent to `g` (or g_1). It should be noted, however, that the set of rules defined above do not generate the empty string. If this is the case, then we construct g_3 (that is, `g_cnf'`) with a new empty rule.

The statement of the CNF theorem can then be presented as:[3]

```
Theorem g_cnf_ex: ∀ g: cfg non_terminal terminal,
(produces_empty g ∨ ~produces_empty g) ∧
(produces_non_empty g ∨ ~produces_non_empty g) →
∃ g': cfg (non_terminal' (emptyrules.non_terminal' non_terminal) terminal)
terminal, g_equiv g' g ∧ (is_cnf g' ∨ is_cnf_with_empty_rule g') ∧
start_symbol_not_in_rhs g'.
```

The new predicates used above assert, respectively, that the argument grammar (i) produces at least one non-empty string (`produces_non_empty`), (ii) is in the Chomsky Normal Form (`is_cnf`) and (iii) is in the Chomsky Normal Form and has a single empty rule with the start symbol in the left-hand side (`is_cnf_with_empty_rule`).

It should be observed that the statement of `g_cnf_ex` is not entirely constructive, as we require, for any context-free grammar `g`, a proof that either `g` produces the empty string or `g` does not produce the empty string, and also that `g` produces a non-empty string or `g` does not produce a non-empty string. Since we have not yet included a proof of the decidability of these predicates in our formalization (something that we plan to do in the future), the statement of the lemma has to require such proofs explicitly. They are demanded, respectively, by the elimination of empty rules and elimination of useless symbols phases of grammar simplification.

The formalization of this section required a lot of insights not directly available from the informal proofs, the most important being the definition of the predicate `g_cnf_rules` (for the rules of the `g_cnf` grammar). In a first attempt, this inductive definition resulted with 14 constructors. Although correct, it was refined many times until it was simplified to only 4 after the definition of the type of the new non-terminals was adjusted properly, with a single constructor. This effort resulted in elegant definitions which allowed the simplification of the corresponding proofs, thus leading to a natural and readable formalization. In particular, the strategy used in the proof of lemma `derives_g_cnf_g` (which states that every sentence produced by `g_cnf g` is also produced by `g`) is a very simple and elegant one, which uses the information already available in the definition of `g_cnf_rules`.

6 Pumping Lemma

The Pumping Lemma is a property that is verified for all context-free languages (CFLs) and was stated and proved for the first time by Bar-Hillel, Perles and Shamir in 1961 ([7]). It does not characterize the CFLs, however, since it is also verified by some languages that are not context-free. It states that, for every context-free language and for every sentence of such a language that has a certain minimum length, it is possible to obtain an infinite number of new sentences that must also belong to the language. This minimum length depends only on the language defined. In other words (let $\mathcal{L}$ be defined over alphabet

[3] The results of this section are available in library `chomsky.v`.

Σ): $\forall\ \mathcal{L}, (\text{cfl}\ \mathcal{L}) \rightarrow \exists\ n \,|\, \forall\ \alpha, (\alpha \in \mathcal{L}) \wedge (|\alpha| \geq n) \rightarrow \exists\ u, v, w, x, y \in \Sigma^* \,|\, (\alpha = uvwxy) \wedge (|vx| \geq 1) \wedge (|vwx| \leq n)\ \wedge \forall\ i, uv^iwx^iy \in \mathcal{L}$.

The Pumping Lemma is stated in Coq as follows:[4], [5], [6], [7]

```
Lemma pumping_lemma: ∀ l: lang terminal,
(contains_empty l ∨ ∼ contains_empty l) ∧
(contains_non_empty l ∨ ∼ contains_non_empty l) → cfl l →
∃ n: nat, ∀ s: sentence, l s → length s ≥ n →
∃ u v w x y: sentence, s = u ++v ++w ++x ++y ∧
length (v ++x) ≥ 1 ∧ length (u ++y) ≥ 1 ∧ length (v ++w ++x) ≤ n ∧
∀ i: nat, l (u ++(iter v i) ++w ++(iter x i) ++y).
```

A typical use of the Pumping Lemma is to show that a certain language is not context-free by using the contrapositive of the statement of the lemma. The proof proceeds by assuming that the language is context-free, and this leads to a contradiction from which one concludes that the language in question can not be context-free.

The Pumping Lemma derives from the fact that the number of non-terminal symbols in any context-free grammar G that generates $\mathcal{L}$ is finite. There are different strategies that can be used to prove that the lemma can be derived from this fact. We searched through 13 proofs published in different textbooks and articles by different authors, and concluded that in 6 cases ([2,7–11]) the strategy uses CNF grammars and binary trees for representing derivations. Other 5 cases ([12–16]) present tree-based proofs that however do not require the grammar to be in CNF. Finally, Harrison ([17]) proves the Pumping Lemma as a corollary to the more general Ogden's Lemma and Amarilli and Jeanmougin ([18]) use a strategy with pushdown automata instead of context-free grammars.

The difference between the proofs that use binary trees and those that use general trees is that the former uses $n = 2^k$ (where k is the number of non-terminal symbols the grammar) and the latter uses $n = m^k$ (where m is the length of the longest right-hand side among all rules of the grammar and k is the number of non-terminal symbols in the grammar). In both cases, the idea is the same: to show that sufficiently long sentences have parse trees for which a maximal path contains at least two instances of the same non-terminal symbol.

Since 11 out of 13 proofs considered use grammars and generic trees and, of these, 6 use CNF grammars and binary trees (including the authors of the original proof), this strategy was considered as the choice for the present work. Besides that, binary trees can be easily represented in Coq as simple inductive types, where generic trees require mutually inductive types, which increases the complexity of related proofs. Thus, for all these reasons we have adopted the proof strategy that uses CNF grammars and binary trees in what follows.

[4] This statement contains the extra clause `length (u ++ y) >= 1`, corresponding to $|uy| \geq 1$, which is normally not mentioned in textbooks.

[5] Predicates `contains_empty` and `contains_non_empty` are indeed decidable and thus it would not be necessary to explicitly state that they satisfy the Law of the Excluded Middle. However, this property has not been addressed in the formalization yet, which justifies the statement of the lemma as it is.

[6] Application `iter l i` on a list `l` and a natural `i` yields list $\mathtt{l}^{\mathtt{i}}$.

[7] The results of this section are available in library `pumping.v`.

The classical proof considers that G is in the Chomsky Normal Form, which means that derivation trees have the simpler form of binary trees. Then, if the sentence has a certain minimum length, the frontier of the derivation tree should have two or more instances of the same non-terminal symbol in some path that starts in the root of this tree. Finally, the context-free character of G guarantees that the subtrees related to these duplicated non-terminal symbols can be cut and pasted in such a way that an infinite number of new derivation trees are obtained, each of which is related to a new sentence of the language. The formal proof presented here is based in the informal proof available in [3].

The proof of the Pumping Lemma starts by finding a grammar G that generates the input language L (this is a direct consequence of the predicate `cfl`, which states that the language is context-free). Next, we obtain a CNF grammar G' that is equivalent to G, using previous results. Then, G is substituted for G' and the value for n is defined as 2^k, where k is the length of the list of non-terminals of G' (which in turn is obtained from the predicate `rules_finite`). An arbitrary sentence α of $L(G')$ that satisfies the required minimum length n is considered. Lemma `derives_g_cnf_equiv_btree` is then applied in order to obtain a `btree` t that represents the derivation of α in G'. Naturally we have to ensure that $\alpha \neq \epsilon$, which is true since by assumption $|\alpha| \geq 2^k$.

The next step is to obtain a path (a sequence of non-terminal symbols ended by a terminal symbol) that has maximum length, that is, whose length is equal to the height of t plus 1. This is accomplished by means of the definition `bpath` and the lemma `btree_ex_bpath`. The length of this path (which is $\geq k + 2$) allows one to infer that it must contain at least one non-terminal symbol that appears at least twice in it. This result comes from the application of the lemma `pigeon` which represents a list version of the well-known pigeonhole principle:

```
Lemma pigeon: ∀ A: Type, ∀ x y: list A, (∀ e: A, In e x → In e y) →
length x = length y + 1→ ∃ d: A, ∃ x1 x2 x3: list A, x = x1 ++[d] ++x2 ++[d] ++x3.
```

This lemma (and other auxiliary lemmas) is included in library pigeon.v, and its proof requires the use of classical reasoning (and thus library `Classical_Prop` of the Coq Standard Library). This is necessary in order to have a decidable equality on the type of the non-terminals of the grammar, and this is the only place in the whole formalization where this is required. Nevertheless, we plan to pursue in the future a constructive version of this proof.

Since a path is not unique in a tree, it is necessary to use some other representation that can describe this path uniquely, which is done by the predicate `bcode` and the lemma `bpath_ex_bcode`. A `bcode` is a sequence of boolean values that tell how to navigate in a `btree`. Lemma `bpath_ex_bcode` asserts that every path in a `btree` can be assigned a `bcode`.

Once the path has been identified with a repeated non-terminal symbol, and a corresponding `bcode` has been assigned to it, lemma `bcode_split` is applied twice in order to obtain the two subtrees t_1 and t_2 that are associated respectively to the first and second repeated non-terminals of t.

From this information it is then possible to extract most of the results needed to prove the goal, except for the pumping condition. This is obtained by an auxiliary lemma `pumping_aux`, which takes as hypothesis the fact that a tree t_1 (with frontier vwx) has a subtree t_2 (with frontier w), both with the same roots, and asserts the existence of an infinite number of new trees obtained by repeated substitution of t_2 by t_1 or simply t_1 by t_2, with respectively frontiers $v^i w x^i, i \geq 1$ and w, or simply $v^i w x^i, i \geq 0$.

The proof continues by showing that each of these new trees can be combined with tree t obtained before, thus representing strings $uv^i w x^i y, i \geq 0$ as necessary. Finally, we prove that each of these trees is related to a derivation in G', which is accomplished by lemma `btree_equiv_produces_g_cnf`.

The formalization of the Pumping Lemma is quite readable and easily modifiable to an alternative version that uses a smaller value of n (as in the original proof contained in [7]). It builds nicely on top of the previous results on grammar normalization, which in turn is a consequence of grammar simplification. It is however long (`pumping_lemma` has 436 lines of Coq script) and the key insights for its formalization were (i) the construction of the library trees.v, specially the lemmas that relate binary trees to CNF grammars; (ii) the identification and isolation of lemma `pumping_aux`, to show the pumping of subtrees in a binary tree and (iii) the proof of lemma `pigeon`. None of these aspects are clear from the informal proof, they showed up only while working in the formalization.

7 Related Work

Context-free language theory formalization is a relatively new area of research, when compared with the formalization of regular languages theory, with some results already obtained with the Coq, HOL4 and Agda proof assistants.

The pioneer work in context-free language theory formalization is probably the work by Filliâtre and Courant ([19]), which led to incomplete results (along with some important results in regular language theory) and includes closure properties (e.g. union), the partial equivalence between pushdown automata and context-free grammars and parts of a certified parser generator. No paper or documentation on this part of their work has been published however.

Most of the extensive effort started in 2010 and has been devoted to the certification and validation of parser generators. Examples of this are the works of Koprowski and Binsztok (using Coq, [20]), Ridge (using HOL4, [21]), Jourdan, Pottier and Leroy (using Coq, [22]) and, more recently, Firsov and Uustalu (in Agda, [23]). These works assure that the recognizer fully matches the language generated by the corresponding context-free grammar, and are important contributions in the construction of certified compilers.

On the more theoretical side, on which the present work should be considered, Norrish and Barthwal published on general context-free language theory formalization using the HOL4 proof assistant ([24–26]), including the existence of Chomsky and Greibach normal forms for grammars, the equivalence of pushdown automata and context-free grammars and closure properties. These results are from the PhD thesis of Barthwal ([27]), which includes also a proof of the Pumping Lemma for context-free languages. Thus, Barthwal extends our work with pushdown automata and Greibach Normal Form results, and for this reason it is the most complete formalization of context-free language theory up to date, in any proof assistant. Recently, Firsov and Uustalu proved the existence of a Chomsky Normal Form grammar for every general context-free grammar, using the Agda proof assistant ([28]). For a discussion of the similarities and differences of our work and those of Barthwal and Firsov, please refer to [4].

A special case of the Pumping Lemma for context-free languages, namely the Pumping Lemma for regular languages, is included in the comprehensive work of Doczkal, Kaiser and Smolka on the formalization of regular languages ([29]).

8 Conclusions

This is probably the most comprehensive formalization of the classical context-free language theory done to the present date in the Coq proof assistant, and includes the important result that is the second ever formalization of the Pumping Lemma for context-free languages (the first in the Coq proof assistant). It is also the first ever proof of the alternative statement of the Pumping Lemma that uses a smaller value of n (for more details, see [4,7]).

The whole formalization consists of 23,984 lines of Coq script spread in 18 libraries (each library corresponds to a different file), not including the example files. The libraries contain 533 lemmas and theorems, 99 constructors, 63 definitions (not including fixpoints), 40 inductive definitions and 20 fixpoints among 1,067 declared names.

The present work represents a relevant achievement in the areas of formal language theory and mathematical formalization. As explained before, there is no record that the author is aware of, of a project with a similar scope in the Coq proof assistant covering the formalization of context-free language theory. The results published so far are restricted to parser certification and theoretical results in proof assistants other than Coq. This is not the case, however, for regular language theory, and in a certain sense the present work can be considered as an initiative that complements and extends that work with the objective of offering a complete framework for reasoning with the two most popular and important language classes from the practical point of view. It is also relevant from the mathematical perspective, since there is a clear trend towards increased and widespread usage of interactive proof assistants and the construction of libraries for fundamental theories.

Plans for future development include the definition of new devices (e.g. pushdown automata) and results (e.g. equivalence of pushdown automata and context-free grammars), code extraction and general enhancements of the libraries, with migration of parts of development into SSReflect (to take advantage, for example, of finite type results).

A Definitions

We present next the main definitions used in the formalization.[8] Context-free grammars are represented in Coq very closely to the usual algebraic definition. Let $G = (V, \Sigma, P, S)$ be a context-free grammar. The sets $N = V \backslash \Sigma$ and Σ are represented as types (usually denoted by names such as, for example, `non_terminal` and `terminal`), separately from G. The idea is that these sets are represented by inductive type definitions whose constructors are its inhabitants. Thus, the number of constructors in an inductive type corresponds exactly to the number of (non-terminal or terminal) symbols in a grammar.

Once these types have been defined, we can create abbreviations for sentential forms (`sf`), sentences (`sentence`) and lists of non-terminals (`nlist`). The first corresponds to the list of the disjoint union of the types `non-terminal` and `terminal`, while the other two correspond to simple lists of, respectively, `non-terminal` and `terminal` symbols.

The record representation `cfg` has been used for G. The definition states that `cfg` is a new type and contains three components. The first component is the `start_symbol` of the grammar (a non-terminal symbol) and the second is `rules`, that represents the rules of the grammar. Rules are propositions (represented in Coq by `Prop`) that take as arguments a non-terminal symbol and a (possibly empty) list of non-terminal and terminal symbols (corresponding, respectively, to the left and right-hand side of a rule). Grammars are parametrized by types `non_terminal` and `terminal`.

```
Record cfg (non_terminal terminal : Type): Type:= {
start_symbol: non_terminal;
rules: non_terminal → sf → Prop;
rules_finite:
  ∃ n: nat,
  ∃ ntl: nlist,
  ∃ tl: tlist,
  rules_finite_def start_symbol rules n ntl tl }.
```

The predicate `rules_finite_def` assures that the set of rules of the grammar is finite by proving that the length of right-hand side of every rule is equal or less than a given value, and also that both left and right-hand side of the rules are built from finite sets of, respectively, non-terminal and terminal symbols (represented here by lists). This represents an overhead in the definition of a grammar, but it is necessary in order to allow for the definition of `non_terminal` and `terminal` as generic types in Coq.

Since generic types might have an infinite number of elements, one must make sure that this is not the case when defining the `non_terminal` and `terminal` sets. Also, even if these types contain a finite number of inhabitants (constructors), it is also necessary to prove that the set of rules is finite. All of these is captured by predicate `rules_finite_def`. Thus, for every `cfg` defined directly

[8] The results of this appendix are available in libraries `cfg.v` and `cfl.v`.

of constructed from previous grammars, it will be necessary to prove that the predicate `rules_finite_def` holds.

The other fundamental concept used in this formalization is the idea of *derivation*: a grammar `g` *derives* a string `s2` from a string `s1` if there exists a series of rules in `g` that, when applied to `s1`, eventually results in `s2`. A direct derivation (i.e. the application of a single rule) is represented by $s_1 \Rightarrow s_2$, and the reflexive and transitive closure of this relation (i.e. the application of zero or more rules) is represented by $s_1 \Rightarrow^* s_2$. An inductive predicate definition of this concept in Coq (`derives`) uses two constructors:

```
Inductive derives
  (non_terminal terminal : Type)
  (g : cfg non_terminal terminal)
  : sf → sf → Prop :=
  | derives_refl :
    ∀ s : sf,
    derives g s s
  | derives_step :
    ∀ (s1 s2 s3 : sf)
    ∀ (left : non_terminal)
    ∀ (right : sf),
    derives g s1 (s2 ++inl left :: s3) →
    rules g left right → derives g s1 (s2 ++right ++s3)
```

The constructors of this definition (`derives_refl` and `derives_step`) are the axioms of our theory. Constructor `derives_refl` asserts that every sentential form `s` can be derived from `s` itself. Constructor `derives_step` states that if a sentential form that contains the left-hand side of a rule is derived by a grammar, then the grammar derives the sentential form with the left-hand side replaced by the right-hand side of the same rule. This case corresponds to the application of a rule in a direct derivation step.

A grammar `generates` a string if this string can be derived from its start symbol. Finally, a grammar `produces` a sentence if it can be derived from its start symbol.

Two grammars g_1 (with start symbol S_1) and g_2 (with start symbol S_2) are *equivalent* (denoted $g_1 \equiv g_2$) if they generate the same language, that is, $\forall s, (S_1 \Rightarrow^*_{g_1} s) \leftrightarrow (S_2 \Rightarrow^*_{g_2} s)$. This is represented in our formalization in Coq by the predicate `g_equiv`.

With these and other definitions (see [4]), it is possible to prove various lemmas about grammars and derivations, and also operations on grammars, all of which are useful when proving the main theorems of this work.

Library `cfg.v` contains 4,393 lines of Coq script (~18.3% of the total) and 105 lemmas and theorems (~19.7% of the total).

B Generic Binary Trees Library

In order to support the formalization of the Pumping Lemma in Sect. 6, an extensive library of definitions and lemmas on binary trees and their relation to CNF grammars has been developed.[9] This library is based in the definition of a binary tree (`btree`) whose internal nodes are non-terminal symbols and leaves are terminal symbols. The type `btree` is defined with the objective of representing derivation trees for strings generated by context-free grammars in the Chomsky Normal Form:

```
Inductive btree (non_terminal terminal: Type): Type:=
| bnode_1: non_terminal → terminal → btree
| bnode_2: non_terminal → btree → btree → btree.
```

The constructors of `btree` relate to the two possible forms that the rules of a CNF grammar can assume (namely with one terminal symbol or two non-terminal symbols in the right-hand side). Naturally, the inhabitants of the type `btree` can only represent the derivation of non-empty strings.

Next, we have to relate binary trees to CNF grammars. This is done with the predicate `btree_cnf`, used to assert that a binary tree `bt` represents a derivation in CNF grammar `g`. Now we can show that binary trees and derivations in CNF grammars are equivalent. This is accomplished by two lemmas, one for each direction of the equivalence. Lemma `derives_g_cnf_equiv_btree` asserts that for every derivation in a CNF grammar exists a binary tree that represents this derivation. It is general enough in order to accept that the input grammar might either be a CNF grammar, or a CNF grammar with an empty rule. If this is the case, then we have to ensure that the derived sentence is not empty. Lemma `btree_equiv_derives_g_cnf` proves that every binary tree that satisfies `btree_cnf` corresponds to a derivation in the same (CNF) grammar.

Among other useful lemmas, the following one is of fundamental importance in the proof of the Pumping Lemma, as it relates the length of the frontier of a binary tree to its height:

```
Lemma length_bfrontier_ge:
∀ t: btree,
∀ i: nat,
length (bfrontier t) ≥ 2 ^ (i − 1) →
bheight t ≥ i.
```

The notion of subtree is also important, and is defined inductively as follows (note that a tree is not, in this definition, a subtree of itself):

```
Inductive subtree (t: btree): btree → Prop:=
| sub_br: ∀ tl tr: btree, ∀ n: non_terminal,
          t = bnode_2 n tl tr → subtree t tr
| sub_bl: ∀ tl tr: btree, ∀ n: non_terminal,
          t = bnode_2 n tl tr → subtree t tl
| sub_ir: ∀ tl tr t': btree, ∀ n: non_terminal,
          subtree tr t' → t = bnode_2 n tl tr → subtree t t'
| sub_il: ∀ tl tr t': btree, ∀ n: non_terminal,
          subtree tl t' → t = bnode_2 n tl tr → subtree t t'.
```

[9] The results of this appendix are available in library `trees.v`.

The following lemmas, related to subtrees, among many others, are also fundamental in the proof of the Pumping Lemma:

```
Lemma subtree_trans:
∀ t1 t2 t3: btree,
subtree t1 t2 → subtree t2 t3 → subtree t1 t3.

Lemma subtree_includes:
∀ t1 t2: btree,
subtree t1 t2 → ∃ l r : sentence,
bfrontier t1 = l ++bfrontier t2 ++r ∧ (l ≠ [] ∨ r ≠ []).
```

Library `trees.v` has 4,539 lines of Coq script (~18.9% of the total) and 84 lemmas (~15.7% of the total)).

References

1. Bertot, Y., Castéran, P.: Interactive Theorem Proving and Program Development. Springer, Heidelberg (2004)
2. Sudkamp, T.A.: Languages and Machines, 3rd edn. Addison-Wesley, Redwood City (2006)
3. Ramos, M.V.M., Neto, J.J., Vega, I.S.: Linguagens Formais: Teoria Modelagem e Implementação. Bookman, Brisbane (2009)
4. Ramos, M.V.M.: Formalization of Context-Free Language Theory. Ph.D. thesis, Centro de Informática-UFPE (2016). www.univasf.edu.br/~marcus.ramos/tese.pdf. Accessed 5 May 2016
5. Ramos, M.V.M.: Source files of [4] (2016). https://github.com/mvmramos/v1. Accessed 3 May 2016
6. Chomsky, A.N.: On certain formal properties of grammar. Inf. Control **2**, 137–167 (1959)
7. Bar-Hillel, Y.: Language and Information: Selected Essays on Their Theory and Application. Addison-Wesley Series in Logic. Addison-Wesley Publishing Co., Redwood City (1964)
8. Hopcroft, J.E., Ullman, J.D.: Formal Languages and Their Relation to Automata. Addison-Wesley Longman Publishing Co., Inc., Boston (1969)
9. Davis, M.D., Sigal, R., Weyuker, E.J.: Computability, Complexity, and Languages: Fundamentals of Theoretical Computer Science, 2nd edn. Academic Press Professional Inc., San Diego (1994)
10. Kozen, D.C.: Automata and Computability. Springer, Heidelberg (1997)
11. Hopcroft, J.E., Motwani, R., Rotwani, Ullman, J.D.: Introduction to Automata Theory, Languages and Computability, 2nd edn. Addison-Wesley Longman Publishing Co., Inc., Boston (2000)
12. Ginsburg, S.: The Mathematical Theory of Context-Free Languages. McGraw-Hill Inc., New York (1966)
13. Denning, P.J., Dennis, J.B., Qualitz, J.E.: Machines, Languages and Computation. Prentice-Hall, Upper Saddle River (1978)
14. Brookshear, J.G.: Theory of Computation: Formal Languages, Automata, and Complexity. Benjamin-Cummings Publishing Co., Inc., Redwood City (1989)
15. Lewis, H.R., Papadimitriou, C.H.: Elements of the Theory of Computation, 2nd edn. Prentice Hall PTR, Upper Saddle River (1998)

16. Sipser, M.: Introduction to the Theory of Computation, 2nd edn. International Thomson Publishing, Toronto (2005)
17. Harrison, M.A.: Introduction to Formal Language Theory, 1st edn. Addison-Wesley Longman Publishing Co., Inc., Boston (1978)
18. Amarilli, A., Jeanmougin, M.: A proof of the pumping lemma for context-free languages through pushdown automata. CoRR abs/1207.2819 (2012)
19. INRIA: Coq users' contributions (2015). http://www.lix.polytechnique.fr/coq/pylons/contribs/index. Accessed 26 Oct 2015
20. Koprowski, A., Binsztok, H.: TRX: a formally verified parser interpreter. In: Gordon, A.D. (ed.) ESOP 2010. LNCS, vol. 6012, pp. 345–365. Springer, Heidelberg (2010). http://dx.doi.org/10.1007/978-3-642-11957-6_19. Accessed 26 Oct 2015
21. Ridge, T.: Simple, functional, sound and complete parsing for all context-free grammars. In: Jouannaud, J.-P., Shao, Z. (eds.) CPP 2011. LNCS, vol. 7086, pp. 103–118. Springer, Heidelberg (2011)
22. Jourdan, J.-H., Pottier, F., Leroy, X.: Validating LR(1) parsers. In: Seidl, H. (ed.) Programming Languages and Systems. LNCS, vol. 7211, pp. 397–416. Springer, Heidelberg (2012)
23. Firsov, D., Uustalu, T.: Certified CYK parsing of context-free languages. J. Log. Algebraic Methods Program. **83**(56), 459–468 (2014). The 24th Nordic Workshop on Programming Theory (NWPT 2012)
24. Barthwal, A., Norrish, M.: A formalisation of the normal forms of context-free grammars in HOL4. In: Dawar, A., Veith, H. (eds.) CSL 2010. LNCS, vol. 6247, pp. 95–109. Springer, Heidelberg (2010)
25. Barthwal, A., Norrish, M.: Mechanisation of PDA and grammar equivalence for context-free languages. In: Dawar, A., de Queiroz, R. (eds.) WoLLIC 2010. LNCS, vol. 6188, pp. 125–135. Springer, Heidelberg (2010)
26. Barthwal, A., Norrish, M.: A mechanisation of some context-free language theory in HOL4. J. Comput. Syst. Sci. **80**(2), 346–362 (2014). WoLLIC 2010 Special Issue, Dawar, A., de Queiroz, R. (eds.)
27. Barthwal, A.: A formalisation of the theory of context-free languages in higher order logic. Ph.D. thesis, The Australian National Universityd (2010). https://digitalcollections.anu.edu.au/bitstream/1885/16399/1/Barthwal%20Thesis%202010.pdf. Accessed 27 Nov 2015
28. Firsov, D., Uustalu, T.: Certified normalization of context-free grammars. In: Proceedings of the 2015 Conference on Certified Programs and Proofs. CPP 2015, pp. 167–174. ACM, New York (2015)
29. Doczkal, C., Kaiser, J.-O., Smolka, G.: A constructive theory of regular languages in Coq. In: Gonthier, G., Norrish, M. (eds.) CPP 2013. LNCS, vol. 8307, pp. 82–97. Springer, Heidelberg (2013)

The Semantics of Corrections

Deniz Rudin(✉), Karl DeVries, Karen Duek, Kelsey Kraus, and Adrian Brasoveanu

University of California, Santa Cruz, USA
rudin@ucsc.edu

1 Introduction

Consider the sentences in (1):

(1) a. Andrew, *uh, sorry*, $[\text{Anders}]_F$ ate a taco. (full correction)
b. Anders made, *uh, sorry*, $[\text{ate}]_F$ a taco. (elliptical correction)
c. Anders made, *uh, sorry*, he $[\text{ate}]_F$ a taco. (anaphoric correction)

In each sentence, the speaker makes a mistake, signals that they've made a mistake (*uh, sorry*), and finally corrects their mistake.[1]

We will refer to the underlined material as the ANCHOR (a.k.a. *reparandum*; see Shriberg 1994), the italicized material as the TRIGGER (a.k.a. editing term), all subsequent material as the CORRECTION (a.k.a. alteration + continuation), and the anchor-correction pair as the (ERROR) CORRECTION STRUCTURE. We will abstain from explicitly annotating subsequent examples.

'Repair'/'revision' cases comparable to the above have been given significant attention in psychology (e.g. Levelt 1983), psycholinguistics (e.g. Clark and Tree 2002; Ferreira et al. 2004), conversation analysis (e.g. Schegloff et al. 1977) and computational linguistics (e.g. Heeman and Allen 1999; Hough and Purver 2012) but these phenomena have not been given much attention in generative linguistics, with the recent exception of Ginzburg et al. (2014), who analyze error corrections as a special type of clarification requests (Purver 2004).

Ginzburg et al. (2014) analyze corrections within an incremental dialogue understanding framework, and seek to unify them with other forms of disfluency. We will pursue a distinct line of investigation focusing specifically on correction structures from a grammatical perspective, though what we unearth will be of interest to theories of incremental interpretation. We will be particularly concerned with interactions between correction structures and: (i) contrastive focus, building on Segmented Discourse Representation Theory (SDRT) and related approaches; see van Leusen (1994, 2004), Asher and Gillies (2003), Asher and Lascarides (2009), (ii) propositional anaphora, and (iii) anaphora to quantificational dependencies.

[1] We expect that many of the generalizations we propose about self-corrections will extend to cross-speaker corrections, but we will not be discussing such data here.

J. Väänänen et al. (Eds.): WoLLIC 2016, LNCS 9803, pp. 358–374, 2016.
DOI: 10.1007/978-3-662-52921-8_22

In Sect. 2, we begin by considering (and casting doubt on) the intuitive analysis that error correction structures are a form of revision that creates a single proposition out of (parts of) the anchor and correction. We then look at the data in closer detail in Sect. 3 and argue that the anchor and correction are parsed as separate clauses, based on facts involving contrastive focus, telescoping, and propositional anaphora. Section 4 follows up with a brief proposal for a formal semantics and formal pragmatics of corrections. The final Sect. 5 provides a summary and outlines potential directions for future work.

2 The Snip & Glue Approach

Previous analyses (notably Asher and Gillies 2003; Asher and Lascarides 2009; Ferreira et al. 2004; Heeman and Allen 1999; Ginzburg et al. 2014; van Leusen 1994, 2004), though couched in very different frameworks, all pursue versions of a 'snip & glue' approach: the interpretation of correction structures proceeds by removing mistaken material and replacing it with corrected material – the mistaken portion of the anchor is deleted (snip) and the correction is attached to what remains of the anchor (glue). The result of the interpretational process is a single meaning assigned to a single sentence.

We have three empirical arguments that any snip & glue treatment of corrections (on its own) is inadequate: (i) error correction structures are a kind of contrastive structure (see van Leusen 1994, 2004; Asher and Gillies 2003; Asher and Lascarides 2009 for similar observations); (ii) anaphora in error correction structures behaves like anaphora between sentences; and finally (iii) propositional anaphora to either half of the correction structure is possible. In the next section, we elaborate on each of these claims in turn.

3 The Empirical Ground

In this section, we overview the main empirical features of correction structures and indicate to what extent previous analyses account for these features.

3.1 Three Types of Corrections

We consider three types of corrections. First, we look at elliptical corrections: these are error correction structures in which the correction is missing otherwise obligatory syntactic material.

(2) Elliptical corrections:
  a. Anders made, uh, sorry, [ate]$_F$ a taco.
  b. Anders made a taco, uh, sorry, [ate]$_F$.
  c. Anders made a taco, uh, sorry, [a chalupa]$_F$.
  d. Andrew made a taco, uh, sorry, [Anders]$_F$.

[2]These structures are the only kind examined at length by previous theorists. It is probably partly for this reason that snip & glue approaches to correction structures seem to be intuitively satisfying.

The second type is what we call full corrections – error correction structures in which the correction does not rely on the anchor for its interpretation.

(3) Full Corrections
 a. Andrew, uh, sorry, [Anders]$_F$ ate a taco.
 b. Andrew ate, uh, sorry, [Anders]$_F$ ate a taco.
 c. Andrew ate a taco. Uh, sorry, [Anders]$_F$ ate a taco.

These structures are less obviously addressed by the snip & glue approach, but an intuitive approach might be to simply discard the anchor entirely.

The final type of corrections we consider is anaphoric corrections: the correction contains pronominal elements that rely on material from the anchor for their interpretation. These are the least studied type of corrections, and the most important for the account we will propose in this paper.

(4) Anaphoric corrections
 a. Anders made, uh, sorry, he [ate]$_F$ a taco.
 b. Anders made a taco, uh, sorry, he [ate]$_F$ it.
 c. Anders made a taco, uh, sorry, he [ate]$_F$ one.
 d. Anders made a taco, uh, sorry, [ate]$_F$ it.
 e. Anders made a taco, uh, sorry, [ate]$_F$ one.
 f. Every boy made, uh, sorry, he [ate]$_F$ a taco.
 g. Every boy made some tacos, uh, sorry, they [ate]$_F$ them.

These structures are problematic for snip & glue approaches: the anaphoric dependencies suggest that anchor and correction are not interpretationally merged, and the interpretation of the anchor (although incorrect) is not discarded.

We argue that all three types of corrections deserve a unified account, and that snip & glue approaches on their own cannot provide such an account.

[2] Note that a corresponding correction structure where the correction is a bare noun is infelicitous:
(1) # Anders made a taco, uh, sorry, [chalupa]$_F$.

This appears to be an idiosyncratic property of singular count nouns, as the following felicitous examples demonstrate:
(2) a. Anders made some tacos, uh, sorry, [chalupas]$_F$.
 b. Anders drank some water, uh, sorry, [soda]$_F$.

3.2 Corrections and Contrast

An important fact about corrections is that they must contain at least one focus-marked element. As the examples in (5) show, focus placement goes on the locus of correction. Furthermore, if there are multiple correction loci, the correction structure needs to have multiple foci, as shown in (6).

(5) a. Andrew, uh, sorry, $[\text{Anders}]_F$ ate a taco.
b. ? Andrew, uh, sorry, Anders ate a $[\text{taco}]_F$.

(6) a. Anders made a taco, uh, sorry, $[\text{ate}]_F$ a $[\text{chalupa}]_F$.
b. ? Anders made a taco, uh, sorry, $[\text{ate}]_F$ a chalupa.
c. ? Anders made a taco, uh, sorry, ate a $[\text{chalupa}]_F$.

All of these foci are contrastive: focus placement in the correction must correspond to the location of mistakes in the anchor, because those are the only places where the anchor and correction differ.[3] We assume Rooth's (1992) definition of contrast:

(7) CONTRASTING PHRASES (Rooth 1992):
Construe a phrase α as contrasting with a phrase β iff $[\![\beta]\!]^{\mathrm{o}} \in [\![\alpha]\!]^{\mathrm{f}}$.

For any phrase α, $[\![\alpha]\!]^{\mathrm{o}}$ is the ordinary semantic value of α, and $[\![\alpha]\!]^{\mathrm{f}}$ is the 'focus-semantic value' of α, or the set of all ordinary semantic values derivable from α via replacement of focus-marked elements in α with elements of the same semantic type.[4] For details on the notions of contrast and focus being assumed here, see Rooth (1992).

In order for the anchor and correction to be viewed as contrastive in the Roothian sense, each needs to have an independently calculable semantic value. A snip & glue account where the result is one semantic value built by combining the correction with cannibalized parts from the anchor will need to do something fairly complex to account for the focus facts.[5]

[3] Asher and Gillies (2003), Asher and Lascarides (2009), van Leusen (1994, 2004) already notice that the focus/background partition of the correction should be matched in the anchor. They ultimately propose a version of the snip & glue approach involving non-monotonic logics for Common Ground (CG) update.

[4] Contrastive focus can be applied to elements that differ only in terms of pronunciation (see Artstein 2004 for details), and, as expected if corrections are indeed contrast structures, such elements participate in correction structures as well:
(1) Anders ate a tomahto, uh, sorry, a $\text{to}[\text{may}]_F\text{to}$.

[5] For example, the SDRT approach in Asher and Gillies (2003) has multiple layers of representation and multiple logics associated with these layers. Focus/background information is represented in a 'lower' layer and CG update is performed in a 'higher'-level logic that non-monotonically reasons over and integrates the lower-level representations.

3.3 Corrections and Telescoping

The subtype of anaphoric corrections that we call telescoping corrections (see (4f)) are particularly relevant for understanding the semantics of corrections. The term telescoping refers to cross-sentential dependencies between singular pronouns and quantifiers. The set of quantifiers that participate in telescoping is quite small (examples from/based on Roberts 1987):

(8) a. {Every, Each} boy walked to the stage. He shook the President's hand and returned to his seat.
b. * {No, Most, Half of the, Twenty} boys walked to the stage. He shook the President's hand and returned to his seat.

[6]In contrast, the set of quantifiers that can be picked up cross-sententially by a plural pronoun is larger (see (4g) for a parallel correction structure):

(9) a. $\{Every, Each\}$ boy walked to the stage. They shook the President's hand and returned to their seats.
b. $\{Most, Halfofthe, Twenty\}$ boys walked to the stage. They shook the President's hand and returned to their seats.
c. * No boy(s) walked to the stage. They shook the President's hand and returned to their seats.

Strikingly, we see the exact same restrictions applying to relations between quantifiers and pronouns in error correction structures[7]:

(10) a. $\{Every, Each\}$ boy made, uh, sorry, he $[\text{ate}]_F$ three tacos.
b. * $\{No, Most, Halfofthe, Twenty\}$ boys made, uh, sorry, he $[\text{ate}]_F$ three tacos.

(11) a. $\{Every, Each\}$ boy made, uh, sorry, they $[\text{ate}]_F$ some tacos.
b. $\{Most, Halfofthe, Twenty\}$ boys made, uh, sorry, they $[\text{ate}]_F$ some tacos.
c. * No boy(s) made, uh, sorry, they $[\text{ate}]_F$ some tacos.

[8]These parallels in singular/plural anaphora behavior indicate that anaphora between anchors and corrections behaves like anaphora between *separate sentences*, not like within-sentence binding. Importantly, the telescoping facts are unexpected for snip & glue accounts, which merge anchor and correction into a single sentence.

[6] Generally a plural pronoun strategy is preferred to the telescoping strategy, but telescoping is at least marginally grammatical. We've found in our own experimental work (not reported here) that the same is true for telescoping in corrections.

[7] We were first made aware of examples of this kind by Milward and Cooper (1994), though those authors do not note their theoretical significance.

[8] Cases like this are better with polarity reversal:

(1) No boy made, uh, sorry, they $[\text{did}]_F$ make some tacos.

3.4 Corrections and Propositional Anaphora

Error correction structures allow propositional anaphora with *that* to either the interpretation of the anchor or the interpretation of the correction:

(12) a. **A:** Anders ate fifty, uh, sorry, he ate $[\text{five}]_F$ tacos.
B: That would've been crazy!

b. **A:** Anders ate fifty, uh, sorry, he ate $[\text{five}]_F$ tacos.
B: That's much easier to believe!

It is unclear how this would be explained from the perspective of a snip & glue account, in which the anchor is never assigned a full interpretation. SDRT-style accounts, for example, could capture this because they countenance two representational layers, one of which contains two discourse representation structures for the anchor and the trigger – assuming propositional anaphora resolution happens at the 'right' point and takes advantage of the 'right' representational layer. However, we believe that all these empirical characteristics of error correction structures can be accounted for in a simpler way, outlined in (14) below.

4 Proposal

In Sect. 3.2, we argued that error correction structures are contrastive structures. We discussed the contrastive nature of corrections in Roothian terms: we need to identify a suitable part of the anchor that can provide the antecedent for the focus anaphora contributed by the correction; this is closely related (but not identical) to the SDRT proposal that the focus-background partitions of the correction and anchor should match (van Leusen 1994, 2004; Asher and Gillies 2003; Asher and Lascarides 2009).

It is easy to see how a contrast relation can be established between the correction and the anchor if both of them are complete—as in (13a). However, establishing the contrast relation is trickier if the anchor or the correction or both are incomplete—as in (13b).

(13) a. Anders ate a taco. Uh, sorry, Anders ate a $[\text{chalupa}]_F$.

b. Anders ate, uh, sorry, $[\text{made}]_F$ a taco.

Given the need to establish a contrast rhetorical relation, we hypothesize the following semantics for corrections:

(14) a. Contrast-Driven Theory of Correction Interpretation (Broad Strokes):
Fill in missing material in the anchor and correction in whatever way will result in the ordinary semantic value of the anchor being a member of the focus semantic value of the correction.

b. Contrast-Driven Theory of Correction Interpretation (Thinner Strokes):
Formalization in Compositional DRT (CDRT; Muskens 1996) – see Sect. 4.1 below.

We also propose the following additional semantic/pragmatic component associated with the interpretation of error correction structures (closely following the proposal in Ginzburg et al. 2014):

(15) THE DISCOURSE EFFECT OF ERROR CORRECTION STRUCTURES: Upon calculation of the relation of contrast between the correction and the anchor:
- the speaker's commitment to the anchor is canceled
- the speaker's commitment to the correction is asserted
- the only commitment placed on the table (in the sense Farkas and Bruce 2010) as a Common Ground (CG) update proposal is the one contributed by the correction

In order to interface with standard models of the formal pragmatics of CG update that treat propositions as sets of worlds (see e.g. Stalnaker 1978), our formalization presented in Sect. 4.1 is couched in a possible-world-based compositional semantics.

4.1 Formalization in CDRT

In this section, we put forth a basic formalization of our proposal for the semantics of error correction structures in (14) above. We build on CDRT and add:

– discourse referents (drefs) for propositions
– logical forms of the kind needed for focus semantics (or parasitic scope)

Our system has the following basic types: e (entities), t (truth values), s (variable assignments) and $\mathbf{w}$ (possible worlds). For simplicity, and to explicitly indicate that the compositional aspects of the system largely follow classic Montagovian semantics, we introduce the following abbreviations:

(16) Type abbreviations:

a. $\mathbf{e} := se$; 'individuals' are drefs for individuals, basically individual concepts

b. $\mathbf{s} := s(\mathbf{w}t)$; intensionality: sentences are interpreted relative to the current assignment and the current proposition/set of worlds that are live candidates for the actual world

c. $\mathbf{t} := s(st)$; the interpretation of a sentence is a DRS, i.e., a binary relation between an input and an output assignment – see also DPL formulas, Groenendijk and Stokhof (1991)

A discourse referent (dref) for individuals $u_{\mathbf{e}}$ is of type $\mathbf{e} := se$. That is, a dref for individuals is basically an individual concept: it denotes an individual (type e) relative to a context of interpretation/variable assignment (type s). Similarly, a dref for propositions $p_{\mathbf{s}}$ is of type $\mathbf{s} := s(\mathbf{w}t)$. A propositional dref denotes a set of worlds (type $\mathbf{w}t$) relative to a variable assignment.

Given that we intensionalize our logic with plural propositional drefs rather than singular possible-world drefs, we have to decide how to interpret lexical intensional relations:

(17) **Lexical relations**. When an intensional n-ary static lexical relation R of type $\mathbf{w}(e(e(\ldots t)))$ is interpreted relative to a propositional dref $p_{\mathbf{s}}$, it is interpreted distributively relative to the worlds in p:
$R_p(u_1, \ldots, u_n) := \lambda i_s.\ \forall w_{\mathbf{w}} \in pi\ (R(w)(u_1 i) \ldots (u_n i))$

With lexical relations in place, we can introduce basic discourse representation structures (DRSs).

(18) **Basic DRSs**.

a. We abbreviate introducing drefs $\nu_1, \ldots, \nu_n$ as: $[\nu_1, \ldots, \nu_n]$

b. We abbreviate a DRS that contains only conditions $C_1, \ldots, C_n$ as: $[C_1, \ldots, C_n]$

c. Dynamic conjunction is symbolized as ';'; for two DRSs D, D' of type $\mathbf{t}$, we have that:
$D; D' := \lambda i_s.\lambda j_s.\ \exists k_s(Dik \wedge D'kj)$, where '$\wedge$' is classical static conjunction

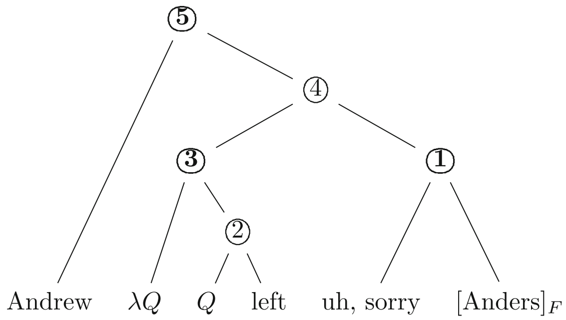
d. A DRS $[\nu_1, \ldots, \nu_n |\ C_1, \ldots, C_n]$ introducing some drefs and contributing some conditions is just the abbreviation of the dynamic conjunction $[\nu_1, \ldots, \nu_n];\ [C_1, \ldots, C_n]$.

A simple error correction structure like (19) is interpreted as in (20):

(19) Andrew left, uh, sorry, $[\text{Anders}]_F$.

(20) a. *uh, sorry* $\rightsquigarrow \lambda A_{\alpha}.\lambda B_{\alpha(\mathbf{st})}.\lambda A'_{\alpha}.$
$[p_1, p_2];\ B(A')(p_1);\ B(A)(p_2);\ CG \mathrel{+}= p_2$

b.



c. ① $\rightsquigarrow \lambda B_{((\mathbf{est})\mathbf{st})\mathbf{st}}.\lambda A'_{(\mathbf{est})\mathbf{st}}.\ [p_1, p_2];\ B(A')(p_1);$
$B(\lambda P_{\mathbf{est}}.\lambda p_{\mathbf{s}}.\ [u_2 |\ u_2 = \text{ANDERS}];\ P(u_2)(p))(p_2);\ CG \mathrel{+}= p_2$

③ $\rightsquigarrow \lambda Q_{(\mathbf{est})\mathbf{st}}.\lambda p_{\mathbf{s}}.\ Q(\lambda x_{\mathbf{e}}.\lambda p_{\mathbf{s}}.\ [\text{LEAVE}_p(x)])(p)$

⑤ $\rightsquigarrow [p_1, p_2, u_1, u_2 |\ u_1 = \text{ANDREW},\ \text{LEAVE}_{p_1}(u_1),$
$u_2 = \text{ANDERS},\ \text{LEAVE}_{p_2}(u_2)];\ CG \mathrel{+}= p_2$

In (20c), we assume a Lewis-style typing with the ‘intensionalization’ type **s** being innermost (closest to the type of sentences **t**). We also assume Montagovian type lifts for proper names, which are of type $(\mathbf{e}(\mathbf{st}))(\mathbf{st})$, e.g.,

$$Anders \rightsquigarrow \lambda P_{\mathbf{e(st)}}.\lambda p_{\mathbf{s}}.\ [u_2 |\ u_2 = \text{ANDERS}];\ P(u_2)(p)$$

Variables are subscripted with their types. We assume complex types associate to the right and we usually omit parentheses indicating association to the right, e.g., instead of $\mathbf{e}(\mathbf{st})$ and $(\mathbf{e}(\mathbf{st}))(\mathbf{st})$, we usually write **est** and (**est**)**st**.

As (20a) shows, the trigger contributes the crucial operator relating the correction to the anchor.[9] This operator takes three arguments:

- the correction A_α (the type α is underspecified and is dictated by the correction itself)—this is *Anders* in our case;
- the mistaken part of the anchor A'_α that must have the same type as the correction—this is *Andrew* in our case;
- the remaining part of the anchor $B_{\alpha(\mathbf{st})}$ that can be predicated of both A and A'—this is a type-lifted version of *left* in our case; this type lifting happens systematically as a consequence of (i) the mistake *Andrew* scoping out of the anchor and (ii) the trigger+correction *uh, sorry, Anders* taking (parasitic) scope immediately under the scoped-out mistake.

The logical form (LF) in (20) is the result of establishing anchor-correction contrast. That is, the trigger + correction constituent (*uh sorry, Anders* in this case) adjoins at a point that divides the anchor in two parts: (i) one part of the anchor is the mistake (*Andrew* in our case), and enters in a contrastive relation with the correction (that is, the ordinary semantic value of the mistake is a member of the focus semantic value of the correction); (ii) the second part of the anchor (*left*, or a type-lifted version thereof, in this case) can be predicated of both the mistake and the correction. That is, LFs for correction structures are derived via the following informal algorithm:

(21) Correction LF generation algorithm (first pass):

I. Adjoin the trigger (the correction operator) to the correction.

II. Adjoin the anchor to the resulting structure.

III. Identify that portion of the anchor that is a member of the focus semantic value of the correction, and move it to an adjoining position, leaving in place a variable and lambda-abstractor of the appropriate type.

[9] We’ve represented the trigger *uh, sorry* as a lexical item contributing the crucial operator relating the correction to the anchor. This is a convenient notational choice that indicates no deep assumption of our theory; we assume that the correction operator is available independently of the way a speaker indicates that they’re making a correction, which may in principle be non-verbal.

[10] In this case, the correction is *[Anders]$_F$*, and the portion of the anchor that is a member of the focus semantic value of *[Anders]$_F$* is *Andrew*. Therefore, *Andrew* is scoped over the correction structure, leaving a lambda abstractor over a variable of type (**est**)**st**.

Once the correction operator in (20a) takes its arguments, it introduces two propositional drefs p_1 and p_2 for the anchor and the correction respectively, and requires only the p_2 dref to be added to the CG.

In the simple example in (20), the partition of the anchor induced by the adjunction site of the trigger + correction constituent is fairly directly related to the SDRT idea that the partitioning of the anchor matches the focus-background partition of the correction. In general, however, our account does not require the focus-background of the correction and of the anchor to match. We simply require the trigger+correction adjunction site to partition the anchor in such a way that one part of it (the mistake) contrasts with the correction, and the remaining part can be predicated of both correction and mistake. The difference between our proposal and the SDRT focus/background matching proposal becomes clear when we consider multiple correction loci, which are associated with multiple foci. For example, according to our proposal, the LF of (6a) above would partition the anchor into the mistake *made a taco* and the remaining part of the anchor *Anders*. And we would require the ordinary value of the entire mistake *made a taco* to be a member of the focus value of the entire correction *[ate]$_F$ a [chalupa]$_F$*.[11]

In sum, error correction structures show that the clause-like semantic values of both the anchor and the correction become part of the interpretation context but in different ways: only the correction ends up being added to the CG, but the interpretation of the anchor is crucial for establishing anchor-correction contrast and also for providing suitable antecedents for anaphora in the anchor (see the anaphoric corrections discussed in Sect. 3.1).

In order for our proposal to generalize across all types of correction structures, the algorithm in (21) must be made somewhat more complex. The most complex cases are corrections in which the anchor is missing syntactically obligatory material, as in the elliptical correction in (2a). In (2a), the verb *made* in the anchor is missing its direct object. Because of this, we need to derive an LF for it like the

[10] This step of the algorithm enforces the contrast generalization from Sect. 3.2. Note, however, that it does not rule out superfluous focus placement, as in the following infelicitous example:

(1) # Anders made a taco, uh, sorry, [ate]$_F$ a [taco]$_F$.

In this case, the VP of the anchor is indeed a member of the focus semantic value of the correction, as *taco* is (trivially) of the same semantic category as itself. This problem could be solved by adding a constraint against triviality to the generation of focus alternatives, ruling out focus alternatives that include the ordinary semantic values of focus-marked elements.

[11] As we already indicated in Footnote 10, we assume that the multiple foci in the correction induce a suitable focus semantic value for the entire correction: assuming that 'contrastive' focus semantic values do not include ordinary values, we require that when multiple foci are present, any alternative that contains the ordinary value of any of the foci should be excluded from the focus value.

one in (23a), where that missing direct object slot is filled in with a variable Q, and the direct object of the correction, *a taco*, moves up to take scope over the entire anchor-correction structure so that it can bind both direct object variables. This type of LF is familiar from Right Node Raising constructions (e.g., *Jane likes and Bill hates this kind of sea salt caramels*), and the intonational contour associated with correction structures like (2a) seems to be very similar to such Right Node Raising constructions. To derive LFs like (23a), we need to make two additions to the informal LF generation algorithm above:

(22) Correction LF generation algorithm (final pass):

I. Adjoin the trigger (the correction operator) to the correction.

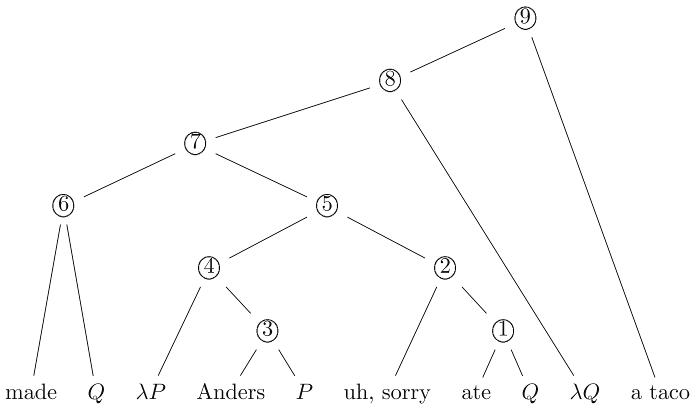
II. Adjoin the anchor to the resulting structure.

III. Insert a variable of the appropriate type to fill in missing syntactically obligatory structure.

IV. Identify that portion of the anchor that is a member of the focus semantic value of the correction, and move it to an adjoining position, leaving in place a variable and lambda-abstractor of the appropriate type.

V. Identify that portion of the correction that corresponds to an unbound variable in the anchor, and move it to an adjoining position so that it can take scope over that variable.

(23) a. LF for (2a):



b. a taco $\rightsquigarrow \lambda P_{\mathbf{e(st)}}.\lambda p_{\mathbf{s}}.\ [u_2 |\ \textsc{taco}_p(u_2)];\ P(u_2)(p)$

② $\rightsquigarrow \lambda Q'_{\mathbf{(est)st}}.\lambda P'_{\mathbf{est}}.\ [p_1, p_2];\ Q'(P')(p_1);$
$Q'(\lambda x_{\mathbf{e}}.\lambda p_{\mathbf{s}}.\ Q(\lambda x'_{\mathbf{e}}.\lambda p_{\mathbf{s}}.\ [\textsc{eat}_p(x, x')])(p))(p_2);\ CG \mathrel{+}= p_2$

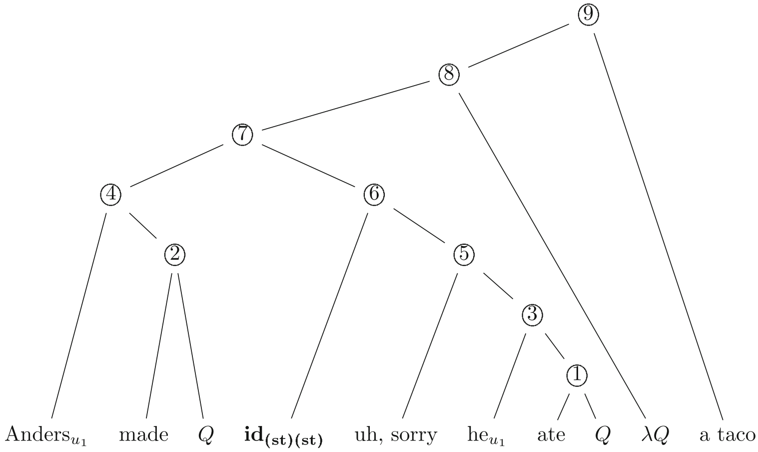
⑤ $\rightsquigarrow \lambda P'_{\mathbf{est}}.\ [p_1, p_2, u_1 |\ u_1 = \textsc{Anders}];\ P'(u_1)(p_1);$
$Q(\lambda x'_{\mathbf{e}}.\lambda p_{\mathbf{s}}.\ [\textsc{eat}_p(u_1, x')])(p_2);\ CG \mathrel{+}= p_2$

⑦ $\rightsquigarrow [p_1, p_2, u_1 |\ u_1 = \textsc{Anders}];\ Q(\lambda x'_{\mathbf{e}}.\lambda p_{\mathbf{s}}.[\textsc{make}_p(u_1, x')])(p_1);$
$Q(\lambda x'_{\mathbf{e}}.\lambda p_{\mathbf{s}}.[\textsc{eat}_p(u_1, x')])(p_2);\ CG \mathrel{+}= p_2$

⑨ $\rightsquigarrow [p_1, p_2, u_1 |\ u_1 = \textsc{Anders}];\ [u_2 |\ \textsc{taco}_{p_1}(u_2),\ \textsc{make}_{p_1}(u_1, u_2)];$
$[u_2 |\ \textsc{taco}_{p_2}(u_2),\ \textsc{eat}_{p_2}(u_1, u_2)];\ CG \mathrel{+}= p_2$

Anaphoric corrections like (4a) are analyzed as shown in (24a). To maintain the general format for the correction operator contributed by *uh, sorry*, we assume the covert insertion of a node denoting the identity function $\mathbf{id}_{\mathbf{(st)(st)}}$ over objects of type $\mathbf{st}$. This is for convenience only, we could also generalize the interpretation of the correction operator in a suitable way.

(24) a. LF for (4a):



b. $\mathbf{id}_{\mathbf{(st)(st)}} \rightsquigarrow \lambda \mathcal{D}_{\mathbf{st}}.\, \mathcal{D}$

he$_{u_1}$ $\rightsquigarrow \lambda P_{\mathbf{e(st)}}.\lambda p_{\mathbf{s}}.\ P(u_1)(p)$

⑤ $\rightsquigarrow \lambda f_{\mathbf{(st)(st)}}.\lambda \mathcal{D}_{\mathbf{st}}.\ [p_1, p_2];\ f(\mathcal{D})(p_1);$
$f(Q(\lambda x'_{\mathbf{e}}.\lambda p_{\mathbf{s}}.\,[\textsc{eat}_p(u_1, x')]))(p_2);\ CG \mathrel{+}= p_2$

⑥ $\rightsquigarrow \lambda \mathcal{D}_{\mathbf{st}}.\,[p_1, p_2]; \mathcal{D}(p_1); Q(\lambda x'_{\mathbf{e}}.\lambda p_{\mathbf{s}}.[\textsc{eat}_p(u_1, x')])(p_2); CG \mathrel{+}= p_2$

⑦ $\rightsquigarrow [p_1, p_2, u_1 |\ u_1 = \textsc{Anders}];\ Q(\lambda x'_{\mathbf{e}}.\lambda p_{\mathbf{s}}.[\textsc{make}_p(u_1, x')])(p_1);$
$Q(\lambda x'_{\mathbf{e}}.\lambda p_{\mathbf{s}}.[\textsc{eat}_p(u_1, x')])(p_2);\ CG \mathrel{+}= p_2$

⑨ $\rightsquigarrow [p_1, p_2, u_1 |\ u_1 = \textsc{Anders}];\ [u_2 |\ \textsc{taco}_{p_1}(u_2),\ \textsc{make}_{p_1}(u_1, u_2)];$
$[u_2 |\ \textsc{taco}_{p_2}(u_2),\ \textsc{eat}_{p_2}(u_1, u_2)];\ CG \mathrel{+}= p_2$

We present an alternative formulation of the interpretation of corrections couched in Categorial Grammar in Appendix A.

4.2 Telescoping Corrections

In this section we show how our account generalizes to telescoping error correction structures like (4f), or their plural counterparts (4g). We build on Dynamic Plural Logic (DPlL) (van den Berg 1996; Nouwen 2003) and Plural Compositional DRT (PCDRT) (Brasoveanu 2007), which recasts DPlL in classical type logic and incorporates discourse reference to possible worlds. DPlL/PCDRT enables us to treat updates with universal quantifiers in much the same way as updates with proper names or indefinites, so our CDRT account of anaphoric corrections like (4a)/(4b) can be straightforwardly generalized to (4f) and (4g).

The main difference between CDRT and DPlL/PCDRT is that updates are binary relations over sets of assignments of type $(st)((st)t)$, rather than binary relations over single assignments of type $s(st)$. Our type $\mathbf{t}$ therefore becomes

$\mathbf{t} := (st)((st)t)$. Since we work with sets of assignments, our 'intensionalization' type can simply be $\mathbf{s} := s\mathbf{w}$, i.e., the type of drefs for possible worlds. The reason is that given a set of assignments I_{st} and a dref $p_{s\mathbf{w}}$, we retrieve a set of worlds (i.e., a proposition) as shown in (25). Introducing new drefs relative to a set of assignments (26) is just the cumulative-style generalization of introducing drefs relative to single assignments. Lexical relations are still interpreted distributively (27), but relative to a set of assignments rather than a propositional dref. Similarly, dynamic conjunction is still interpreted as relation composition (28). To handle quantifiers, we introduce a maximization operator $\mathbf{M}_u(D)$ that extracts the set of entities that satisfies the update D and stores it in dref u (29).

(25) $p_{s\mathbf{w}}I_{st} = \{pi : i_s \in I\}$ (pI is the image of I under function p)

(26) $[\nu_1, \ldots, \nu_n] := \lambda I_{st}.\lambda J_{st}.\ \forall i_s \in I \exists j_s \in J(i[\nu_1, \ldots, \nu_n]j) \wedge \forall j_s \in J \exists i_s \in I(i[\nu_1, \ldots, \nu_n]j)$

(27) $R_p(u_1, \ldots, u_n) := \lambda I_{st}.\ I \neq \emptyset \wedge \forall i_s \in I\ (R(pi)(u_1 i) \ldots (u_n i))$

(28) $D; D' := \lambda I_{st}.\lambda J_s.\ \exists K_s(DIK \wedge D'KJ)$

(29) $\mathbf{M}_u(D) := \lambda I_{st}.\lambda J_{st}.\ ([u]; D)IJ \wedge \neg\exists K_{st}(([u]; D)IK \wedge uJ \subsetneq uK)$

Universal quantification contributes a maximization operator over the restrictor, and the nuclear scope further elaborates on the maximal restrictor-satisfying dref (30). Singular or plural anaphora in subsequent sentences can pick up the maximal dref introduced by the universal *every*, in much the same way that the nuclear scope of an *every* quantification can pick up that dref and further elaborate on it. To properly distinguish between singular anaphora (telescoping) and plural anaphora, we need to extend the system with a notion of distributivity and a notion of discourse plurality/singularity. But the basic system outlined here is enough to show that we can now capture telescoping corrections in the same way we capture regular anaphoric corrections, as shown in (31) (cf. (24)).[12]

(30) $\text{every}_u \rightsquigarrow \lambda P_{\mathbf{est}}.\lambda P'_{\mathbf{est}}.\lambda p_{\mathbf{s}}.\ \mathbf{M}_u(P(u)(p));\ P'(u)(p)$

(31) Every$_{u_1}$ boy made, uh sorry, he$_{u_1}$ / they$_{u_1}$ ate a taco. $\rightsquigarrow$
$[p_1, p_2]; \mathbf{M}_{u_1}([\text{BOY}_{p_1}(u_1)]); [p'_1, u_2 | p'_1 \sqsubseteq p_1, \text{TACO}_{p'_1}(u_2), \text{MAKE}_{p'_1}(u_1, u_2)];$
$[u_2 | p_2 \sqsubseteq p_1, \text{TACO}_{p_2}(u_2),\ \text{EAT}_{p_2}(u_1, u_2)];\ CG \mathrel{+}= p_2$

5 Conclusion

We have argued that in error correction structures, the anchor and the correction are given separate interpretations, in opposition to standard accounts in which

[12] To derive the correct truth conditions, we need to introduce an additional propositional dref and suitable subset relations between propositional drefs to capture the fact that anaphora from the correction to a quantifier in the anchor builds on part of the content contributed by the anchor. The subset relations $p'_1 \sqsubseteq p_1$ and $p_2 \sqsubseteq p_1$ need to preserve the full dependency structure associated with the worlds in p_1. That is, for any p_1-world that we retain in the subsets p'_1 or p_2, we need to retain the full range of u_1-entities associated with that world.

the output of an error correction structure is a single unified interpretation for the entire structure. On the basis of focus placement facts we have argued that error correction structures are a species of contrast structure. On the basis of telescoping facts, we have argued that the anchor and correction are treated as separate sentences. And finally on the basis of propositional anaphora facts, we have argued that the interpretation of the anchor is still accessible after the correction has been completed. In light of these facts, we conclude that snip & glue accounts of error correction are inadequate on their own.

One way to think about the present account of error corrections relative to the SDRT one or the one in Ginzburg et al. (2014) is that it tries to see how far we can get in a relatively unstructured version of dynamic semantics in which (i) we have only Dynamic Predicate Logic (DPL) + propositional drefs (+the tech needed for subclausal compositionality) and (ii) we assume a monotonic version of incremental interpretation (no non-monotonic glue logic). An important point that emerges is that simply adding propositional drefs and incorporating a separate CG update that involves only some of these propositional drefs is enough to capture the basic interpretation of corrections. This enables us to incorporate telescoping corrections fairly easily because the basic DPL system can be generalized to a dynamic plural logic.

We will close by mentioning two broad follow-up questions. First, what is the fine-grained structure of elliptical corrections? Must corrections be constituents? What is the relation between error correction structures, fragment answers and better-studied forms of ellipsis, like gapping, stripping and sluicing? It is, to the best of our knowledge, a novel observation that error correction structures involve syntax/semantics 'in the silence' as Merchant (2001) puts it. Studying error correction structures as a new addition to the typology of elliptical constructions could significantly increase our understanding of the nature of structured silences in natural language.

Second, what new types of psycholinguistic evidence can correction structures provide about the fine details of incremental processing? How do listeners recognize that they're hearing an error correction structure? What is the time course of correction interpretation and how does this vary between the three different types of corrections we studied? Are there processing costs associated with 'filling in' missing material? Finally, what happens when the target of the correction is ambiguous, e.g., *John recognized Mary, uh, sorry, Bill* (where *Bill* could correct either *John* or *Mary*)? What factors affect disambiguation for one resolution or another, e.g., identifying *John* or *Mary* as the target of correction in the example we just mentioned?

Acknowledgments. We want to thank Pranav Anand, Donka Farkas, Andreas Walker, Paul Willis, Erik Zyman and the audiences at the UCSC S-Circle (October 2015), CUSP (Stanford, November 2015), the UCSC SPLAP Workshop (February 2016), CoCoLab (Stanford, March 2016) and three WoLLIC reviewers for discussion and comments on this work. The usual disclaimers apply.

A Categorial Grammar Formulation

Here we present an alternative syntactic account of error correction structures in categorical grammar that preserves our semantic account. For reasons of space we suppress non-propositional drefs, and work through non-quantified cases only. For full sentence corrections, the correction denotes a binary relation between sentences that updates the common ground only with the proposition associated with the correction:

$$\dfrac{\dfrac{\text{John left}}{\mathrm{S} : \text{LEAVE}(j)} \qquad \dfrac{\dfrac{\text{uh, sorry}}{(\mathrm{S}\backslash\mathrm{S})/\mathrm{S} : \lambda A.\lambda A'.[p_1, p_2|A'(p_1), A(p_2)]\ ;\ CG{+}{=}\ p_2} \qquad \dfrac{\text{John arrived}}{\mathrm{S} : \text{ARRIVE}(j)}}{\mathrm{S}\backslash\mathrm{S} : \lambda A'.[p_1, p_2|A'(p_1), \text{ARRIVE}_{p_2}(j)]\ ;\ CG{+}{=}\ p_2}}{\mathrm{S}\text{: } [p_1, p_2|\text{LEAVE}_{p_1}(j), \text{ARRIVE}_{p_2}(j)]\ ;\ CG{+}{=}\ p_2}$$

To handle partial corrections we generalize the type of the correction structure to denote a relation between verb phrases. The correction essentially builds a conjunction in which only the conjunct associated with the correction is added to the common ground.

$$\dfrac{\dfrac{\text{left}}{\mathrm{NP}\backslash\mathrm{S} : \text{LEAVE}} \qquad \dfrac{\dfrac{\text{uh sorry}}{\begin{array}{l}((\mathrm{NP}\backslash\mathrm{S})\backslash(\mathrm{NP}\backslash\mathrm{S}))/(\mathrm{NP}\backslash\mathrm{S}) :\\ \lambda A.\lambda A'.\lambda x.[p_1, p_2|A'(x)(p_1), A(x)(p_2)]\ ;\ CG{+}{=}\ p_2\end{array}} \qquad \dfrac{\text{arrived}}{\mathrm{NP}\backslash\mathrm{S} : \text{ARRIVE}}}{(\mathrm{NP}\backslash\mathrm{S})\backslash(\mathrm{NP}\backslash\mathrm{S}) : \lambda A'.\lambda x.[p_1, p_2|A'(x)(p_1), \text{ARRIVE}_{p_2}(x)]\ ;\ CG{+}{=}\ p_2}}{\mathrm{NP}\backslash\mathrm{S} : \lambda x.[p_1, p_2|\text{LEAVE}_{p_1}(x), \text{ARRIVE}_{p_2}(x)]\ ;\ CG{+}{=}\ p_2}$$

The correction then takes the subject as its final argument resulting in the desired update:

$$\dfrac{\dfrac{\text{John}}{\mathrm{NP} : j} \qquad \dfrac{\text{left uh sorry arrived}}{\mathrm{NP}\backslash\mathrm{S} : \lambda x.[p_1, p_2|\text{LEAVE}_{p_1}(x), \text{ARRIVE}_{p_2}(x)]\ ;\ CG{+}{=}\ p_2}}{\mathrm{S} : [p_1, p_2|\text{LEAVE}_{p_1}(j), \text{ARRIVE}_{p_2}(j)]\ ;\ CG{+}{=}\ p_2}$$

We also need to handle error correction structures which contain material between the correction and the constituent that needs to be replaced. This material needs to be made available both to the anchor and the correction. We utilize a pair forming operator $\circ$ that creates pairs of semantic values:

$$\dfrac{\mathrm{X} : \alpha \qquad \mathrm{Y} : \beta}{\mathrm{X} \circ \mathrm{Y} : \langle \alpha, \beta \rangle}$$

We now analyze error correction structures with intervening material in terms of pair formation. The correction, taking a verb to its right as its first argument, expects a verb-object pair to its left. It then feeds the object to both verbs:

$$\dfrac{\dfrac{\dfrac{\text{met}}{(\mathrm{NP}\backslash\mathrm{S})/\mathrm{NP} : \text{MEET}} \qquad \dfrac{\text{Bill}}{\mathrm{NP} : b}}{((\mathrm{NP}\backslash\mathrm{S})/\mathrm{NP}) \circ \mathrm{NP} : \langle \text{MEET}, b \rangle} \qquad \dfrac{\text{uh sorry saw}}{\begin{array}{l}(((\mathrm{NP}\backslash\mathrm{S})/\mathrm{NP}) \circ \mathrm{NP})\backslash(\mathrm{NP}\backslash\mathrm{S}) :\\ \lambda A'.\lambda x.[p_1, p_2|A'(1)(A'(2))(x)(p_1), \text{SEE}_{p_2}(A'(2))(x)]\ ;\ CG{+}{=}\ p_2\end{array}}}{\mathrm{NP}\backslash\mathrm{S} : \lambda x.[p_1, p_2|\text{MEET}_{p_1}(b)(x), \text{SEE}_{p_2}(b)(x)]\ ;\ CG{+}{=}\ p_2}$$

The correction then takes the subject as its final argument, and generates the desired update:

$$\dfrac{\dfrac{\text{John}}{\mathrm{NP} : j} \qquad \dfrac{\text{met Bill uh sorry saw}}{\mathrm{NP}\backslash\mathrm{S} : \lambda x.[p_1, p_2|\text{MEET}_{p_1}(b)(x), \text{SEE}_{p_2}(b)(x)]\ ;\ CG{+}{=}\ p_2}}{\mathrm{S} : [p_1, p_2|\text{MEET}_{p_1}(b)(j), \text{SEE}_{p_2}(b)(j)]\ ;\ CG{+}{=}\ p_2}$$

This account avoids movement of the intervening material at the cost of introducing a pair-forming operator. This operator allows us to store the semantic value associated with the object so that it can be used to saturate the verb in both the anchor and the correction.

References

Artstein, R.: Focus below the word level. Nat. Lang. Semant. **12**, 1–22 (2004)

Asher, N., Lascarides, A.: Agreement, disputes and commitments in dialogue. J. Semant. **26**, 109–158 (2009)

Asher, N., Gillies, A.: Common ground, corrections, and coordination. Argumentation **17**, 481–512 (2003)

van den Berg, M.: The dynamics of nominal anaphora. Dissertation, University of Amsterdam, Amsterdam (1996)

Brasoveanu, A.: Structured nominal and modal reference. Dissertation, Rutgers University (2007)

Clark, H.H., Tree, J.E.F.: Using uh and um in spontaneous speaking. Cognition **84**(1), 73–111 (2002). doi:10.1016/S0010-0277(02)00017-3

Farkas, D.F., Bruce, K.B.: On reacting to assertions and polar questions. J. Semant. **27**, 81–118 (2010)

Ferreira, F., Lau, E., Bailey, K.: Disfluencies, language comprehension, and tree adjoining grammars. Cogn. Sci. **28**, 721–749 (2004)

Ginzburg, J., Fernández, R., David, S.: Disfluencies as intra-utterance dialogue moves. Semant. Pragmat. **7**, 64 (2014)

Groenendijk, J., Stokhof, M.: Dynamic predicate logic. Linguist. Philos. **14**(1), 39–100 (1991)

Heeman, P., Allen, J.: Speech repairs, intonational phrases and discourse markers: modeling speaker' utterances in spoken dialogue. Comput. Linguist. **25**(4), 527–571 (1999)

Hough, J., Purver, M.: Processing self-repairs in an incremental type-theoretic dialogue system. In: Proceedings of the 16th SemDial Workshop on the Semantics and Pragmatics of Dialogue (SeineDial), Paris, pp. 136–144 (2012). http://www.eecs.qmul.ac.uk/~mpurver/papers/hough-purver12semdial.pdf

van Leusen, N.: The interpretation of corrections. In: Bosch, P., van der Sandt, R. (eds.) Proceedings of the Conference on Focus and Natural Language Processing, vol. 3, pp. 1–13. IBM Working paper 7, TR-80.94-007. IBM Deutschland GmhB (1994)

van Leusen, N.: Compatibility in context: a diagnosis of correction. J. Semant. **21**, 415–441 (2004)

Levelt, W.: Monitoring and self-repair in speech. Cognition **14**, 41–104 (1983)

Merchant, J.: Sluicing, Islands, and the Theory of Ellipsis. Oxford University Press, Oxford (2001)

Milward, D., Cooper, R.: Applications, theory, and relationship to dynamic semantics. In: The 15th International Conference on Computational Linguistics (COLING 1994), pp. 748–754. COLING 1994 Organizing Comm., Kyoto Japan (1994)

Muskens, R.: Combining Montague semantics and discourse representation. Linguist. Philos. **19**(2), 143–186 (1996)

Nouwen, R.: Dynamic aspects of quantification. Dissertation, UIL-OTS, Utrecht University (2003)

Purver, M.: The theory and use of clarification requests in dialogue. Dissertation, King's College, University of London (2004). http://www.dcs.qmul.ac.uk/mpurver/papers/purver04thesis.pdf

Roberts, C.: Modal subordination, anaphora, and distributivity. Dissertation, University of Massachusetts Amherst (1987)

Rooth, M.: A theory of focus interpretation. Nat. Lang. Semant. **1**, 75–116 (1992)

Schegloff, E., Jefferson, G., Sacks, H.: The preference for self-correction in the organization of repair in conversation. Language **53**(2), 361–382 (1977)

Shriberg, E.: Preliminaries to a theory of speech disfluencies. Dissertation, University of California at Berkeley (1994)

Stalnaker, R.: Assertion. Syntax Semant. **9**, 315–332 (1978)

The Expressive Power of k-ary Exclusion Logic

Raine Rönnholm(✉)

University of Tampere, 33014 Tampere, Finland
raine.ronnholm@uta.fi

Abstract. In this paper we study the expressive power of k-ary exclusion logic, EXC[k], that is obtained by extending first order logic with k-ary exclusion atoms. It is known that without arity bounds exclusion logic is equivalent with dependence logic. From the translations between them we see that the expressive power of EXC[k] lies in between k-ary and $(k+1)$-ary dependence logics. We will show that, at least in the case of unary exclusion logic, the both of these inclusions are proper.

In a recent work by the author it was shown that k-ary inclusion-exclusion logic is equivalent with k-ary existential second order logic, ESO[k]. We will show that, on the level of sentences, it is possible to simulate inclusion atoms with exclusion atoms, and this way express ESO[k]-sentences by using only k-ary exclusion atoms. For this translation we also need to introduce a novel method for "unifying" the values of certain variables in a team. As a consequence, EXC[k] captures ESO[k] on the level of sentences, and thus we get a strict arity hierarchy for exclusion logic. It also follows that k-ary inclusion logic is strictly weaker than EXC[k].

1 Introduction

Exclusion logic is an extension of first order logic with *team semantics*. In team semantics the truth of formulas is interpreted by using sets of assignments which are called *teams*. This approach was introduced by Hodges [13] to define compositional semantics for the *IF-logic* by Hintikka and Sandu [11]. The truth for the IF-logic was originally defined by using semantic games of imperfect information [12], and in thus teams can be seen as parallel positions in a semantic game. Teams can also be seen as databases [19], and thus the study of logics with team semantics has natural connections with the study of database dependencies.

For first order logic team semantics is just a generalization of Tarski semantics and has the same expressive power. But if we extend first order logic with new atomic formulas we get higher expressive power and can define more complex properties of teams. The first new atoms for this framework were *dependence atoms* introduced by Väänänen [19]. In *dependence logic* the semantics for these atoms are defined by functional dependencies of the values of variables in a team. Several new atoms have been presented for this framework with the motivation from simple database dependencies – such as *independence atoms* by Grädel and Väänänen [8] and *inclusion and exclusion atoms* by Galliani [5]. Lately there

J. Väänänen et al. (Eds.): WoLLIC 2016, LNCS 9803, pp. 375–391, 2016.
DOI: 10.1007/978-3-662-52921-8_23

has been research on these atoms with an attempt to formalize the dependency phenomena in different fields of science, such as database theory [15], belief presentation [4] and quantum mechanics [14].

If we extend first order logic with inclusion/exclusion atoms we obtain *inclusion and exclusion logics*. The team semantics for these atoms are very simple: Suppose that $\boldsymbol{t}_1, \boldsymbol{t}_2$ are k-tuples of terms and X is a team. The k-ary inclusion atom $\boldsymbol{t}_1 \subseteq \boldsymbol{t}_2$ says that the values of $\boldsymbol{t}_1$ are included in the values of $\boldsymbol{t}_2$ in the team X. The k-ary exclusion atom $\boldsymbol{t}_1 \mid \boldsymbol{t}_2$ dually says that $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$ get distinct values in X, i.e. for all assignments $s, s' \in X$ we have $s(\boldsymbol{t}_1) \neq s'(\boldsymbol{t}_2)$.

Galliani [5] has shown that without arity bounds exclusion logic is equivalent with dependence logic. Thus, on the level of sentences, it captures *existential second order logic*, ESO [19]. Inclusion logic is not comparable with dependence logic in general [5], but captures *positive greatest fixed point logic* on the level of sentences, as shown by Galliani and Hella [7]. Hence exclusion logic captures NP and inclusion logic captures PTIME over finite structures with linear order.

In order to understand the nature of these atoms, there has been research on the bounded arity fragments of the corresponding logics. Durand and Kontinen [3] have shown that, on the level of sentences, k-ary dependence logic captures the fragment of ESO in which at most $(k-1)$-ary functions can be quantified[1]. From this it follows that dependence logic has a strict arity hierarchy over sentences, since the arity hierarchy of ESO (over arbitrary vocabulary) is known to be strict, as shown by Ajtai [1]. These earlier results, however, do not tell much about the expressive power of k-ary exclusion logic, EXC[k], since the existing translation from it to dependence logic does not respect the arities of atoms.

There has not been much research on exclusion logic after Galliani proved its equivalence with dependence logic. In this paper we will show that the relationship between these two logics becomes nontrivial when we consider their bounded arity fragments. This also leads to results on the relation between inclusion and exclusion logics, which is interesting because they can be seen as duals to each other, as we have argued in [17].

By inspecting Galliani's translations [5] between exclusion and dependence logics more closely, we observe that EXC[k] is stronger than k-ary dependence logic but weaker than $(k+1)$-ary dependence logic. Thus it is natural to ask whether the expressive power of EXC[k] is strictly in between k-ary and $(k+1)$-ary dependence logics. We will show that this holds at least when $k = 1$.

In an earlier work by the author [17] it was shown that both INC[k]- and EXC[k]-formulas could be translated into k-ary ESO, ESO[k], which gives us an upper bound for the expressive power of EXC[k]. In [17] it was also shown that conversely ESO[k]-formulas with at most k-ary free relation variables can be expressed in k-ary inclusion-exclusion logic, INEX[k], and consequently INEX[k] captures ESO[k] on the level of sentences.

Since exclusion logic is closed downwards, unlike inclusion-exclusion logic, we know that EXC[k] is strictly weaker than INEX[k]. However, in certain cases we can simulate the use of inclusion atoms with exclusion atoms: Suppose that x, w,

[1] See [6,9] for similar arity hierarchy results on independence and inclusion logics.

w^c are variables such that the sets values of w and w^c in X are complements of each other. Now we have $\mathcal{M} \vDash_X x \subseteq w$ iff $\mathcal{M} \vDash_X x \mid w^c$. This can be generalized for k-ary atoms if the values of k-tuples $\boldsymbol{w}$ and $\boldsymbol{w}^c$ are complementary.

We will use the observation above to modify our translation (in [17]) from ESO[k] to INEX[k]. If we only consider *sentences* of exclusion logic, we can quantify the needed complementary values, and then replace inclusion atoms in the translation with the corresponding exclusion atoms. The remaining problem is that in our translation we also needed a new connective called *term value preserving disjunction* [17] to avoid the loss of information on the values of certain variables when evaluating disjunctions. This operator can be defined by using both inclusion and exclusion atoms [17], but it is undefinable in exclusion logic since it is not closed downwards.

In [17] we introduced new operators called *inclusion and exclusion quantifiers* and defined them in inclusion-exclusion logic. Furthermore, we showed that *universal inclusion quantifier* $(\forall\, \boldsymbol{x} \subseteq \boldsymbol{t})$ could be defined also in exclusion logic. We will then consider the use of this quantifier in somewhat trivial looking form $(\forall\, \boldsymbol{x} \subseteq \boldsymbol{x})$. This operator turns out to be useful as it "unifies" the values of variables in a team. We will use it to define new operators called *unifier*, *unified existential quantifier* and *unifying disjunction*.

This unifying disjunction will give us an alternative method to avoid the loss of information in the translation from ESO[k]. This completes our translation, and proves the equivalence between EXC[k] and ESO[k] on the level of sentences. Hence we also get a strict arity hierarchy for exclusion logic since the arity hierarchy for ESO is known to be strict. We also get the interesting consequence that k-ary inclusion logic is *strictly* weaker than EXC[k] on the level of sentences.

See the extended version of this paper [18] for more details and examples. Certain proofs, that have been omitted here, are also presented in [18].

2 Preliminaries

2.1 Syntax and Team Semantics for First Order Logic

A vocabulary L is a set of *relation symbols* R, *function symbols* f and *constant symbols* c. The set of *L-terms*, $\mathrm{T_L}$, is defined in the standard way. The set of variables occurring in a tuple $\boldsymbol{t}$ of L-terms is denoted by $\mathrm{Vr}(\boldsymbol{t})$.

Definition 1. FO_L*-formulas are defined as follows:*

$$\varphi ::= t_1 = t_2 \mid \neg t_1 = t_2 \mid R\,\boldsymbol{t} \mid \neg R\,\boldsymbol{t} \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid \exists\, x\, \varphi \mid \forall\, x\, \varphi$$

FO_L*-formulas of the form* $t_1 = t_2$, $\neg t_1 = t_2$, $R\,\boldsymbol{t}$ *and* $\neg R\,\boldsymbol{t}$ *are called* literals.

Let $\varphi \in \mathrm{FO}_L$. We denote the set of *subformulas* of φ by $\mathrm{Sf}(\varphi)$, the set of variables occurring in φ by $\mathrm{Vr}(\varphi)$ and the set of *free variables* of φ by $\mathrm{Fr}(\varphi)$.

An *L-model* $\mathcal{M} = (M, \mathcal{I})$, where the *universe* M is any nonempty set and the *interpretation* $\mathcal{I}$ is a function whose domain is the vocabulary L. The interpretation $\mathcal{I}$ maps constant symbols to elements in M, k-ary relation symbols to

k-ary relations in M and k-ary function symbols to functions $M^k \to M$. For all $k \in L$ we write $k^{\mathcal{M}} := \mathcal{I}(k)$. An *assignment* s for M is a function that is defined in some set of variables, $\mathrm{dom}(s)$, and ranges over M. A *team* X for M is any set of assignments for M with a common domain, denoted by $\mathrm{dom}(X)$.

Let s be an assignment and a be any element in M. The assignment $s[a/x]$ is defined in $\mathrm{dom}(s) \cup \{x\}$, and it maps the variable x to a and all other variables as s. If $\boldsymbol{x} := x_1 \dots x_k$ is a tuple of variables and $\boldsymbol{a} := (a_1, \dots, a_k) \in M^k$, we write $s[\boldsymbol{a}/\boldsymbol{x}] := s[a_1/x_1, \dots, a_k/x_k]$. For a team X, a set $A \subseteq M^k$ and for a function $\mathcal{F} : X \to \mathcal{P}(M^k)\backslash\{\emptyset\}$ we use the following notations:

$$\begin{cases} X[A/\boldsymbol{x}] := \big\{ s[\boldsymbol{a}/\boldsymbol{x}] \mid s \in X,\ \boldsymbol{a} \in A \big\} \\ X[\mathcal{F}/\boldsymbol{x}] := \big\{ s[\boldsymbol{a}/\boldsymbol{x}] \mid s \in X,\ \boldsymbol{a} \in \mathcal{F}(s) \big\}. \end{cases}$$

Let $\mathcal{M}$ be an L-model, s an assignment and t an L-term s.t. $\mathrm{Vr}(t) \subseteq \mathrm{dom}(s)$. The *interpretation of t with respect to $\mathcal{M}$ and s*, is denoted simply by $s(t)$. For a team X and a tuple $\boldsymbol{t} := t_1 \dots t_k$ of L-terms we write

$$s(\boldsymbol{t}) := (s(t_1), \dots, s(t_k)) \ \text{ and } \ X(\boldsymbol{t}) := \{s(\boldsymbol{t}) \mid s \in X\}.$$

If $A \subseteq M$, we write $\overline{A} := M\backslash A$. We are now ready to define *team semantics* for first order logic.

Definition 2. *Let $\mathcal{M}$ be a model, $\varphi \in \mathrm{FO}_L$ and X a team s.t.* $\mathrm{Fr}(\varphi) \subseteq \mathrm{dom}(X)$. *We define the* truth of φ in the model $\mathcal{M}$ and the team X, $\mathcal{M} \vDash_X \varphi$, *as:*

- $\mathcal{M} \vDash_X t_1 = t_2$ *iff* $s(t_1) = s(t_2)$ *for all* $s \in X$.
- $\mathcal{M} \vDash_X \neg t_1 = t_2$ *iff* $s(t_1) \neq s(t_2)$ *for all* $s \in X$.
- $\mathcal{M} \vDash_X R\,\boldsymbol{t}$ *iff* $X(\boldsymbol{t}) \subseteq R^{\mathcal{M}}$.
- $\mathcal{M} \vDash_X \neg R\,\boldsymbol{t}$ *iff* $X(\boldsymbol{t}) \subseteq \overline{R^{\mathcal{M}}}$.
- $\mathcal{M} \vDash_X \psi \wedge \theta$ *iff* $\mathcal{M} \vDash_X \psi$ *and* $\mathcal{M} \vDash_X \theta$.
- $\mathcal{M} \vDash_X \psi \vee \theta$ *iff there are* $Y, Y' \subseteq X$ *s.t.* $Y \cup Y' = X$, $\mathcal{M} \vDash_Y \psi$ *and* $\mathcal{M} \vDash_{Y'} \theta$.
- $\mathcal{M} \vDash_X \exists x\, \psi$ *iff there exists* $F : X \to \mathcal{P}(M)\backslash\{\emptyset\}$ *s.t.* $\mathcal{M} \vDash_{X[F/x]} \psi$.
- $\mathcal{M} \vDash_X \forall x\, \psi$ *iff* $\mathcal{M} \vDash_{X[M/x]} \psi$.

Remark 1. The semantics for existential quantifier above allows to select several witnesses for x. In FO it is equivalent to pick only a single witness, and thus the truth condition can be written in an equivalent form (so-called *strict semantics*): $\mathcal{M} \vDash_X \exists x\, \psi$ iff there is $F : X \to M$ s.t. $\mathcal{M} \vDash_{X[F/x]} \psi$, where $X[F/x]$ is the team $\{x[F(s)/x] \mid s \in X\}$. Since this truth condition is equivalent also for exclusion logic [5], we will use it in the proofs for exclusion logic to simplify them.

For $\varphi \in \mathrm{FO}_L$ and tuple $\boldsymbol{x} := x_1 \dots x_k$ we write: $\exists\, \boldsymbol{x}\, \varphi := \exists x_1 \dots \exists x_k \varphi$ and $\forall\, \boldsymbol{x}\, \varphi := \forall x_1 \dots \forall x_k \varphi$. It is easy to show that

- $\mathcal{M} \vDash_X \exists\, \boldsymbol{x}\, \varphi$ iff there exists $\mathcal{F} : X \to \mathcal{P}(M^k)\backslash\{\emptyset\}$ s.t. $\mathcal{M} \vDash_{X[\mathcal{F}/\boldsymbol{x}]} \varphi$.
- $\mathcal{M} \vDash_X \forall\, \boldsymbol{x}\, \varphi$ iff $\mathcal{M} \vDash_{X[M^k/\boldsymbol{x}]} \varphi$.

In strict semantics the first condition turns into the form: $\mathcal{M} \vDash_X \exists\, \boldsymbol{x}\, \varphi$ iff there exists $\mathcal{F} : X \to M^k$ s.t. $\mathcal{M} \vDash_{X[\mathcal{F}/\boldsymbol{x}]} \varphi$, where $X[\mathcal{F}/\boldsymbol{x}] := \big\{ s[\mathcal{F}(s)/\boldsymbol{x}] \mid s \in X \big\}$.

First order logic with team semantics has so-called *flatness*-property:

Proposition 1 ([19], Flatness). *Let X be a team and $\varphi \in \mathrm{FO}_L$. The following equivalence holds: $\mathcal{M} \vDash_X \varphi$ iff $\mathcal{M} \vDash_{\{s\}} \varphi$ for all $s \in X$.*

We use notations $\vDash_s^{\mathrm{T}}$ and $\vDash^{\mathrm{T}}$ for truth in a model with standard Tarski semantics. Team semantics can be seen just as a generalization of Tarski semantics:

Proposition 2 ([19]). *The following equivalences hold:*

$$\mathcal{M} \vDash_s^{\mathrm{T}} \varphi \text{ iff } \mathcal{M} \vDash_{\{s\}} \varphi \quad \text{for all } \mathrm{FO}_L\text{-formulas } \varphi \text{ and assignments } s.$$
$$\mathcal{M} \vDash^{\mathrm{T}} \varphi \text{ iff } \mathcal{M} \vDash_{\{\emptyset\}} \varphi \quad \text{for all} \mathrm{FO}_L\text{-sentences } \varphi.$$

Note that, by flatness, $\mathcal{M} \vDash_X \varphi$ if and only if $\mathcal{M} \vDash_s^{\mathrm{T}} \varphi$ for all $s \in X$. By Proposition 2 it is natural to write $\mathcal{M} \vDash \varphi$, when we mean that $\mathcal{M} \vDash_{\{\emptyset\}} \varphi$. Note that $\mathcal{M} \vDash_\emptyset \varphi$ holds trivially for all FO_L-formulas φ by Definition 2. In general we say that any logic $\mathcal{L}$ with team semantics has the *empty team property* if $\mathcal{M} \vDash_\emptyset \varphi$ holds for all $\mathcal{L}$-formulas φ. We define two more important properties for any logic $\mathcal{L}$ with team semantics.

Definition 3. *Let $\mathcal{L}$ be any logic with team semantics. We say that*

- *$\mathcal{L}$ is* local, *if the truth of formulas is determined only by the values of their free variables in a team, i.e. we have: $\mathcal{M} \vDash_X \varphi$ iff $\mathcal{M} \vDash_{X \upharpoonright \mathrm{Fr}(\varphi)} \varphi$.*
- *$\mathcal{L}$ is* closed downwards *if we have: If $\mathcal{M} \vDash_X \varphi$ and $Y \subseteq X$, then $\mathcal{M} \vDash_Y \varphi$.*

By flatness it is easy to see that FO is local and closed downwards.

2.2 Inclusion and Exclusion Logics

Inclusion logic (INC) and exclusion logic (EXC) are obtained by adding inclusion and exclusion atoms, respectively, to first order logic with team semantics:

Definition 4. *If $\boldsymbol{t}_1, \boldsymbol{t}_2$ are k-tuples of L-terms, $\boldsymbol{t}_1 \subseteq \boldsymbol{t}_2$ is a k-*ary inclusion atom. *INC_L-formulas are formed like FO_L-formulas by allowing the use of (non-negated) inclusion atoms like literals. Let $\mathcal{M}$ be a model and X a team s.t. $\mathrm{Vr}(\boldsymbol{t}_1 \boldsymbol{t}_2) \subseteq \mathrm{dom}(X)$. We define the truth of $\boldsymbol{t}_1 \subseteq \boldsymbol{t}_2$ in $\mathcal{M}$ and X as:*

$$\mathcal{M} \vDash_X \boldsymbol{t}_1 \subseteq \boldsymbol{t}_2 \quad \text{iff for all } s \in X \text{ there exists } s' \in X \text{ s.t. } s(\boldsymbol{t}_1) = s'(\boldsymbol{t}_2).$$

Equivalently we have $\mathcal{M} \vDash_X \boldsymbol{t}_1 \subseteq \boldsymbol{t}_2$ iff $X(\boldsymbol{t}_1) \subseteq X(\boldsymbol{t}_2)$.

*If $\boldsymbol{t}_1, \boldsymbol{t}_2$ are k-tuples of L-terms, $\boldsymbol{t}_1 \mid \boldsymbol{t}_2$ is a k-*ary exclusion atom. *EXC_L-formulas are formed as FO_L-formulas, but (non-negated) exclusion atoms may be used as literals are used in FO. Let $\mathcal{M}$ be a model and X a team s.t. $\mathrm{Vr}(\boldsymbol{t}_1 \boldsymbol{t}_2) \subseteq \mathrm{dom}(X)$. We define the truth of $\boldsymbol{t}_1 \mid \boldsymbol{t}_2$ in $\mathcal{M}$ and X as:*

$$\mathcal{M} \vDash_X \boldsymbol{t}_1 \mid \boldsymbol{t}_2 \quad \text{iff for all } s, s' \in X : s(\boldsymbol{t}_1) \neq s'(\boldsymbol{t}_2).$$

Equivalently we have $\mathcal{M} \vDash_X \boldsymbol{t}_1 \mid \boldsymbol{t}_2$ iff $X(\boldsymbol{t}_1) \cap X(\boldsymbol{t}_2) = \emptyset$ (iff $X(\boldsymbol{t}_1) \subseteq \overline{X(\boldsymbol{t}_2)}$).

Inclusion-exclusion logic (INEX) is defined simply by combining inclusion and exclusion logics. If $\varphi \in \mathrm{EXC}_L$ contains at most k-ary exclusion atoms, we say that φ is a formula of *k-ary exclusion logic*, EXC[k]. *k-ary inclusion logic* (INC[k]) and *k-ary inclusion-exclusion logic* (INEX[k]) are defined analogously.

The following properties have all been shown by Galliani [5]: EXC, INC and INEX are all local and satisfy empty team property. EXC is also closed downwards, unlike INC which is *closed under unions*. If we would use strict semantics for existential quantifier in INC, it would not be local. This is one of the reasons why the semantics given in Definition 2 is usually considered more natural.

3 Expressing Useful Operators for Exclusion Logic

3.1 *k*-ary Dependence Atoms and Intuitionistic Disjunction

Let us review the semantics for *dependence atoms* of *dependence logic* [19]. Let $t_1 \dots t_k$ be L-terms. The k-ary dependence atom $=\!(t_1 \dots t_{k-1}, t_k)$ has the following truth condition: $\mathcal{M} \vDash_X =\!(t_1 \dots t_{k-1}, t_k)$ if and only if we have:

$$\text{for all } s, s' \in X \text{ for which } s(t_1 \dots t_{k-1}) = s'(t_1 \dots t_{k-1}) \text{ also } s(t_k) = s'(t_k),$$

for all L-models $\mathcal{M}$ and teams X for which $\mathrm{Vr}(t_1 \dots t_k) \subseteq \mathrm{dom}(X)$. This truth condition can be read as "the value of t_k is (functionally) dependent on the values of $t_1, \dots, t_{k-1}$". By using Galliani's translation between dependence logic and exclusion logic, we can express k-ary dependence atoms in EXC[k]:

Proposition 3 ([5]). *Let $\boldsymbol{t} = t_1 \dots t_k$ be a tuple of L-terms. The k-ary dependence atom $=\!(t_1 \dots t_{k-1}, t_k)$ is equivalent with the* $\mathrm{EXC}_L[k]$*-formula φ:*

$$\varphi := \forall x\, (x = t_k \vee t_1 \dots t_{k-1} x \mid \boldsymbol{t}), \text{ where } x \text{ is a fresh variable.}$$

In particular, we can express *constancy atom*[2] $=\!(t)$ in EXC[k] for any $k \geq 1$.

The semantics of *intuitionistic disjunction* $\sqcup$ is obtained by lifting the Tarski semantics of classical disjunction from single assignments to teams. That is, $\mathcal{M} \vDash_X \varphi \sqcup \psi$ iff $\mathcal{M} \vDash_X \varphi$ or $\mathcal{M} \vDash_X \psi$. Galliani [4] has shown that this operator can be expressed by using constancy atoms. Hence we can define it as an abbreviation in EXC[k] for any $k \geq 1$.

3.2 Universal Inclusion Quantifier and the Unification of Values

In [17] we have considered inclusion and exclusion dependencies from a new perspective by introducing inclusion and exclusion quantifiers. Let $\boldsymbol{x}$ be a k-tuple of variables, $\boldsymbol{t}$ a k-tuple of L-terms and $\varphi \in \mathrm{INEX}_L$. We review the semantics for *universal* inclusion and exclusion quantifiers $(\forall \boldsymbol{x} \subseteq \boldsymbol{t})$ and $(\forall \boldsymbol{x} \mid \boldsymbol{t})$:

$$\mathcal{M} \vDash_X (\forall \boldsymbol{x} \subseteq \boldsymbol{t})\, \varphi \;\text{ iff }\; \mathcal{M} \vDash_{X[A/\boldsymbol{x}]} \varphi, \text{ where } A = X(\boldsymbol{t}).$$

$$\mathcal{M} \vDash_X (\forall \boldsymbol{x} \mid \boldsymbol{t})\, \varphi \;\text{ iff }\; \mathcal{M} \vDash_{X[A/\boldsymbol{x}]} \varphi, \text{ where } A = \overline{X(\boldsymbol{t})}.$$

[2] $=\!(t)$ is true in a nonempty team X iff t has a constant value in X, i.e. $|X(t)| = 1$.

To define these quantifiers as abbreviations in INEX we needed to use *both* k-ary inclusion and exclusion atoms. However, we can alternatively define quantifier $(\forall\, \boldsymbol{x} \subseteq^{\mathrm{e}} \boldsymbol{t})$ as an abbreviation by using only k-ary exclusion atoms (see [17]). This quantifier has the same truth condition as $(\forall\, \boldsymbol{x} \subseteq \boldsymbol{t})$ above, when φ is a formula of *exclusion logic*.

Hence the universal inclusion quantifier for k-tuples of variables can be defined for both INEX$[k]$ and EXC$[k]$, although these definitions have to be given differently. From now on we will always use the plain notation $(\forall\, \boldsymbol{x} \subseteq \boldsymbol{t})$ and assume it be defined in the right way depending on whether we use it with INEX or EXC.

When defining quantifier $(\forall\, \boldsymbol{x} \subseteq \boldsymbol{t})$, we allowed the variables in the tuple $\boldsymbol{x}$ to occur in $\mathrm{Vr}(\boldsymbol{t})$. In particular, we accept the quantifiers of the form $(\forall\, \boldsymbol{x} \subseteq \boldsymbol{x})$. Quantifiers of this form may seem trivial, but they turn out to be rather useful operators. Let us analyze their truth condition:

$$\mathcal{M} \vDash_X (\forall\, \boldsymbol{x} \subseteq \boldsymbol{x})\, \varphi \;\text{ iff }\; \mathcal{M} \vDash_{X'} \varphi, \text{ where } X' = X[X(\boldsymbol{x})/\boldsymbol{x}].$$

Note that the team X' is not necessarily the same team as X, although we have $\mathrm{dom}(X') = \mathrm{dom}(X)$ and even $X'(\boldsymbol{x}) = X(\boldsymbol{x})$. Consider the following example.

Example 1. Let $X = \{s_1, s_2\}$ where $s_1(v_1) = a$, $s_2(v_1) = b$ and $a \neq b$. Now

$$\begin{aligned} X[X(v_1)/v_1] = X[\{a,b\}/v_1] &= \{s_1[a/v_1], s_1[b/v_1], s_2[a/v_1], s_2[b/v_1]\} \\ &= \{s_1, s_2, s_1[b/v_1], s_2[a/v_1]\} \neq X. \end{aligned}$$

We say that the quantifier $(\forall\, \boldsymbol{x} \subseteq \boldsymbol{x})$ *unifies* the values of the tuple $\boldsymbol{x}$ in a team. After executing this operation for a team X, each $s \in X \upharpoonright (\mathrm{dom}(X) \setminus \mathrm{Vr}(\boldsymbol{x}))$ "carries" the information on the whole relation $X(\boldsymbol{x})$. This also makes the values of the tuple $\boldsymbol{x}$ *independent* of all the other variables in $\mathrm{dom}(X)$. To simplify our notation we introduce the following operator.

Definition 5. *Let $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ be tuples of variables that are not necessarily of the same length.* The unifier of the values of $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ *is defined as:*

$$\mathbf{U}(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)\, \varphi := \; (\forall\, \boldsymbol{x}_1 \subseteq \boldsymbol{x}_1) \dots (\forall\, \boldsymbol{x}_n \subseteq \boldsymbol{x}_n)\, \varphi.$$

Note that if the longest of the tuples $\boldsymbol{x}_i$ is a k-tuple, then this operator can be defined in EXC$[k]$ (and in INEX$[k]$). If we require the variables in the tuples $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ to be *disjoint* (i.e. no variable occurs in more than one tuple), their order does not matter, and we obtain the following truth condition for the unifier:

$$\mathcal{M} \vDash_X \mathbf{U}(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)\, \varphi \text{ iff } \mathcal{M} \vDash_{X[X(\boldsymbol{x}_1)/\boldsymbol{x}_1, \dots, X(\boldsymbol{x}_n)/\boldsymbol{x}_n]} \varphi,$$

We have $\mathbf{U}(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)\varphi \equiv \mathbf{U}(\boldsymbol{x}_1) \dots \mathbf{U}(\boldsymbol{x}_n)\varphi$, but one should note that usually $\mathbf{U}(\boldsymbol{x}_1 \dots \boldsymbol{x}_n)\varphi \not\equiv \mathbf{U}(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)\varphi$. To see this, consider X s.t. $v_1, v_2 \in \mathrm{dom}(X)$ and let $X_1 := X[X(v_1 v_2)/v_1 v_2]$ and $X_2 := X[X(v_1)/v_1, X(v_2)/v_2]$. Now we have $X_1(v_1 v_2) = X(v_1 v_2)$ but $X_2(v_1 v_2) = X(v_1) \times X(v_2)$. It is easy to see that X_1 and X_2 are identical only if $X(v_1 v_2) = X(v_1) \times X(v_2)$.

Remark 2. It holds that $\mathbf{U}(v_1v_2,\, v_2v_3)\,\varphi \equiv \mathbf{U}(v_1,v_2,v_3)\,\varphi \equiv \mathbf{U}(v_2v_3,\, v_1v_2)\,\varphi$. We can generalize this property of the unification to show that the order of the unified tuples is irrelevant, even if they are not disjoint. We omit the proofs for these claims, since we only use the unifier for disjoint tuples in this paper.

This new operator can be used in combination with other logical operators to form new useful tools for the framework of team semantics. We will introduce here two such operators. The definitions are given more generally for INEX, but they can be defined in the same way for EXC as well.

Definition 6. *Let $\boldsymbol{x}$ be a k-tuple of variables and $\varphi \in \mathrm{INEX}_L$.* Unified existential quantifier $\exists^{\mathbf{U}}$ *is defined as:* $\exists^{\mathbf{U}}\, \boldsymbol{x}\varphi := \exists\, \boldsymbol{x}\, \mathbf{U}(\boldsymbol{x})\, \varphi$.

Proposition 4 (See [18] for a Proof). *Let $\boldsymbol{x}$ be a k-tuple and $\varphi \in \mathrm{INEX}_L$. Now $\mathcal{M} \vDash_X \exists^{\mathbf{U}}\, \boldsymbol{x}\varphi$ iff there exists a nonempty set $A \subseteq M^k$ s.t. $\mathcal{M} \vDash_{X[A/\boldsymbol{x}]} \varphi$.*

If we use this quantifier in EXC (or any other downwards closed logic), the following equivalence holds: $\mathcal{M} \vDash_X \exists^{\mathbf{U}}\, \boldsymbol{x}\, \varphi$ iff there exists $\boldsymbol{a} \in M^k$ s.t. $\mathcal{M} \vDash_{X[\{\boldsymbol{a}\}/\boldsymbol{x}]} \varphi$. For single variables this truth condition is equivalent with the semantics of the quantifier $\exists^1$ that was introduced in [16].

Definition 7. *Let $\varphi, \psi \in \mathrm{INEX}_L$ and let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ be k-tuples of disjoint variables.* Unifying disjunction for tuples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ *is defined as:*

$$\varphi \underset{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n}{\vee^{\mathbf{U}}} \psi := \exists\, y_1\, \exists\, y_2\, \mathbf{U}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)\big((y_1 = y_2 \wedge \varphi) \vee (y_1 \neq y_2 \wedge \psi)\big),$$

where y_1, y_2 are fresh variables.

Proposition 5 (See [18] for a Proof). *Let φ, ψ be INEX_L-formulas and let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ be k-tuples of disjoint variables. Now for all L-models $\mathcal{M}$ with at least two elements we have*

$$\mathcal{M} \vDash_X \varphi \underset{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n}{\vee^{\boldsymbol{U}}} \psi \text{ iff there exist } Y, Y' \subseteq X \text{ s.t. } Y \cup Y' = X,$$
$$\mathcal{M} \vDash_{Y[X(\boldsymbol{x}_1)/\boldsymbol{x}_1,\ldots,X(\boldsymbol{x}_n)/\boldsymbol{x}_n]} \varphi \text{ and } \mathcal{M} \vDash_{Y'[X(\boldsymbol{x}_1)/\boldsymbol{x}_1,\ldots,X(\boldsymbol{x}_n)/\boldsymbol{x}_n]} \psi.$$

This operator will play a very important role in the next section.

4 The Expressive Power of EXC[$\boldsymbol{k}$]

4.1 Relationship Between EXC and Dependence Logic

Galliani [5] has shown that, without arity bounds, EXC is equivalent with dependence logic. However, if we consider the bounded arity fragments, this relationship becomes nontrivial. We first review Galliani's translation from exclusion logic to dependence logic (the translation is slightly simplified here).

Proposition 6 **([5]).** *Let $\boldsymbol{t}_1, \boldsymbol{t}_2$ be k-tuples of L-terms. The k-ary exclusion atom $\boldsymbol{t}_1 \mid \boldsymbol{t}_2$ is logically equivalent with the depencende logic formula φ:*

$$\varphi := \forall\, \boldsymbol{y} \exists\, w_1 \exists\, w_2 \big(=\!(w_1) \wedge =\!(\boldsymbol{y}, w_2) \wedge ((w_1 = w_2 \wedge \boldsymbol{y} \neq \boldsymbol{t}_1) \vee (w_1 \neq w_2 \wedge \boldsymbol{y} \neq \boldsymbol{t}_2))\big),$$

where $\boldsymbol{y}$ is a k-tuple of fresh variables and w_1, w_2 are fresh variables.

If we inspect Galliani's translations more closely, we obtain the following:

Corollary 1. *The expressive power of* EXC$[k]$ *is in between k-ary dependence logic and $(k+1)$-ary dependence logic on the level of formulas.*

Proof. By using translation in Proposition 3 we can express k-ary dependence atoms with k-ary exclusion atoms. And by using translation in Proposition 6 we can express k-ary exclusion atoms with $(k+1)$-ary dependence atoms.

By this result it is natural to ask whether these inclusions are proper, or does EXC$[k+1]$ collapse to some fragment of dependence logic. Let us inspect the special case $k = 1$ with the following example.

Example 2 (Compare with a Similar Example for INEX *in* [17]*).* Let $\mathcal{G} = (V, E)$ be an undirected graph. Now we have

(a) $\mathcal{G}$ is disconnected if and only if

$$\mathcal{G} \vDash \forall\, z \,\exists\, x_1 \,\exists\, x_2 \,\big((x_1 = z \vee x_2 = z) \wedge x_1 \,|\, x_2 \wedge (\forall\, y_1 \subseteq x_1)(\forall\, y_2 \subseteq x_2) \neg E y_1 y_2\big).$$

(b) $\mathcal{G}$ is k-colorable if and only if

$$\mathcal{G} \vDash \gamma_{\leq k} \;\sqcup\; \forall\, z \,\exists\, x_1 \ldots \exists\, x_k \Big(\bigvee_{i \leq k} x_i = z \;\wedge\; \bigwedge_{i \neq j} x_i \,|\, x_j \\ \wedge \bigwedge_{i \leq k} (\forall\, y_1 \subseteq x_i)(\forall\, y_2 \subseteq x_i) \neg E y_1 y_2 \Big),$$

where $\gamma_{\leq k} := \exists\, x_1 \ldots \exists\, x_k \,\forall\, y \,\big(\bigvee_{i \leq k} y = x_i\big)$.

Corollary 2. *The expressive power of* EXC[1] *is properly in between* 1*-ary and* 2*-ary dependence logics, on the level of both sentences and formulas.*

Proof. By Corollary 1, the expressive power of EXC[1] is in between 1-ary and 2-ary dependence logics. By Galliani [5], 1-ary dependence logic is not stronger than FO on the level of sentences. However, by Example 2, there are sentences of EXC[1] that cannot be expressed in FO. Thus EXC[1] is strictly stronger than 1-ary dependence logic on the level of sentences.

On the other hand, there are properties that are definable 2-ary dependence logic, but which cannot be expressed in *existential monadic second order logic*, EMSO, such as infinity of a model and even cardinality [19]. But since INEX[1] is equivalent with EMSO on the level of sentences [17], EXC[1] must be strictly weaker than 2-ary dependence logic on the level of sentences.

4.2 Capturing the Arity Fragments of ESO with EXC

In this subsection we will compare the expressive power of EXC with *existential second order logic*, ESO. We denote the k-ary fragment of ESO (where at most k-ary relation symbols can be quantified) by ESO[k]. We will formulate a translation from ESO[k] to EXC[k] on the level of sentences by using the idea from the following observation: Suppose that X is a team and $\boldsymbol{x}$, $\boldsymbol{w}$, $\boldsymbol{w}^c$ are tuples variables s.t. $X(\boldsymbol{w}^c) = \overline{X(\boldsymbol{w})}$. Now we have: $\mathcal{M} \vDash_X \boldsymbol{x} \subseteq \boldsymbol{w}$ iff $\mathcal{M} \vDash_X \boldsymbol{x} \mid \boldsymbol{w}^c$.

In our translation from ESO[k] to INEX[k] [17] the quantified k-ary relation symbols P_i of a ESO$_L$-formula were simply replaced with k-tuples $\boldsymbol{w}_i$ of quantified *first order* variables. Then the formulas of the form $P_i\boldsymbol{t}$ were replaced with the inclusion atoms $\boldsymbol{t} \subseteq \boldsymbol{w}_i$ and the formulas of the form $\neg P_i\boldsymbol{t}$ with the exclusion atoms $\boldsymbol{t} \mid \boldsymbol{w}_i$. To eliminate inclusion atoms from this translation we also need to quantify a tuple $\boldsymbol{w}_i^c$ of variables for each P_i and set a requirement that $\boldsymbol{w}_i^c$ must be given complementary values to $\boldsymbol{w}_i$. This requirement is possible to set in exclusion logic if we are restricted to sentences. Then we simply replace inclusion atoms $\boldsymbol{t} \subseteq \boldsymbol{w}_i$ with the corresponding exclusion atoms $\boldsymbol{t} \mid \boldsymbol{w}_i^c$.

We also need to consider the quantification of the empty set and the full relation M^k as special cases. This is because tuples $\boldsymbol{w}_i$ and also their "complements" $\boldsymbol{w}_i^c$ must always be given a nonempty set of values. For this we use special "label variables" w_i° and $w_i^\bullet$ for each relation symbol P_i. We first quantify some constant value for a variable u and then we can give this value for w_i° to "announce" the quantification of the empty set or analogously we can give it for $w_i^\bullet$ to announce the quantification of the full relation. In order to give these label values, there must be at least two elements in the model. For handling the special case of single element models we will use the following easy lemma:

Lemma 1. *Let φ be* ESO$_L$*-sentence. Now there exists a* FO$_L$*-sentence χ, such that we have $\mathcal{M} \vDash \varphi$ iff $\mathcal{M} \vDash \chi$, for all L-models $\mathcal{M} = (M, \mathcal{I})$ for which $|M| = 1$.*

The remaining problem is that in the translation from ESO to INEX we also needed a new connective called *term value preserving disjunction* [17] to avoid the loss of information on the values of variables $\boldsymbol{w}_i$ when evaluating disjunctions. This time we can use unifying disjunction instead to avoid the loss of information on the values of both the tuples $\boldsymbol{w}_i$ and the tuples $\boldsymbol{w}_i^c$. We are now ready to formulate our main theorem.

Theorem 1. *For every* ESO$_L[k]$*-sentence Φ there exists an* EXC$_L[k]$*-sentence φ such that*

$$\mathcal{M} \vDash \varphi \quad \textit{iff} \quad \mathcal{M} \vDash \Phi.$$

Proof. Since Φ is an ESO$_L[k]$-sentence, there exists a FO$_L$-sentence δ and relation symbols $P_1, \dots, P_n$ so that $\Phi = \exists P_1 \dots \exists P_n \delta$. Without losing generality, we may assume that $P_1, \dots, P_n$ are all k-ary. Let $\boldsymbol{w}_1, \dots, \boldsymbol{w}_n$ and $\boldsymbol{w}_1^c, \dots, \boldsymbol{w}_n^c$ be k-tuples of variables and $w_1^\circ, \dots, w_n^\circ, w_1^\bullet, \dots, w_n^\bullet$ and u be variables such that all of these variables are distinct and do not occur in the formula δ.

Let $\psi \in \mathrm{Sf}(\delta)$. The formula ψ' is defined recursively:

$$
\begin{aligned}
\psi' &= \psi \quad \text{if}\, \psi \,\text{is a literal and}\, P_i \,\text{does not occur in}\ \psi\ \text{for any}\ i \le n \\
(P_i\boldsymbol{t})' &= (\boldsymbol{t} \mid \boldsymbol{w}_i^c \vee w_i^{\bullet} = u) \wedge w_i^{\circ} \neq u \quad \text{for all}\ i \in \{1, \dots, n\} \\
(\neg P_i\boldsymbol{t})' &= (\boldsymbol{t} \mid \boldsymbol{w}_i \vee w_i^{\circ} = u) \wedge w_i^{\bullet} \neq u \quad \text{for all}\ i \in \{1, \dots, n\} \\
(\psi \wedge \theta)' &= \psi' \wedge \theta' \\
(\psi \vee \theta)' &= \psi' \veebar^{\mathbf{U}} \theta', \qquad \text{where}\ \veebar^{\mathbf{U}} := \vee^{\mathbf{U}}_{\boldsymbol{w}_1, \dots, \boldsymbol{w}_n, \boldsymbol{w}_1^c, \dots, \boldsymbol{w}_n^c} \\
(\exists\, x\, \psi)' &= \exists\, x\, \psi',\ (\forall\, x\, \psi)' = \forall\, x\, \psi'.
\end{aligned}
$$

Let χ be a FO_L-sentence determined by the Lemma 1 for the sentence Φ and let $\boldsymbol{z}$ be a k-tuple of fresh variables. Let $\gamma_{=1}$ be a shorthand for the sentence $\forall\, z_1 \forall\, z_2\, (z_1 = z_2)$. Now we can define the sentence φ in the following way:

$$
\begin{aligned}
\varphi \,:=\, & (\gamma_{=1} \wedge \chi) \sqcup \exists\, u \,\exists\, w_1^{\circ} \dots \exists\, w_n^{\circ} \,\exists\, w_1^{\bullet} \dots \exists\, w_n^{\bullet} \\
& \forall\, \boldsymbol{z}\ \exists\, \boldsymbol{w}_1 \dots \exists\, \boldsymbol{w}_n \,\exists\, \boldsymbol{w}_1^c \dots \exists\, \boldsymbol{w}_n^c \Big(\bigwedge_{i \le n} (\boldsymbol{z} = \boldsymbol{w}_i \vee \boldsymbol{z} = \boldsymbol{w}_i^c) \wedge \delta' \Big).
\end{aligned}
$$

Clearly φ is an $\mathrm{EXC}_L[k]$-sentence.

We write $V^* := \mathrm{Vr}(u w_1^{\circ} \dots w_n^{\circ} w_1^{\bullet} \dots w_n^{\bullet} \boldsymbol{w}_1 \dots \boldsymbol{w}_n \boldsymbol{w}_1^c \dots \boldsymbol{w}_n^c)$.

Before proving the claim of this theorem, we prove the following two lemmas:

Lemma 2. *Let $\mathcal{M}$ be an L-model with at least two elements. Let $\mu \in \mathrm{Sf}(\delta)$ and let X a team for which $V^* \subseteq \mathrm{dom}(X)$ and the following assumptions hold:*

$$
\begin{cases}
X(\boldsymbol{w}_i) \cup X(\boldsymbol{w}_i^c) = M^k \ \ \textit{for each}\ i \le n. \\
\textit{The values of}\ w_i^{\circ}, w_i^{\bullet}\ (i \le n)\ \textit{and}\ u\ \textit{are constants in}\ X.
\end{cases}
$$

Let $\mathcal{M}' := \mathcal{M}[\boldsymbol{A}/\boldsymbol{P}]$ $(= \mathcal{M}[A_1/P_1, \dots, A_n/P_n])$, where

$$
A_i = \begin{cases}
\emptyset & \textit{if}\ X(w_i^{\circ}) = X(u)\ \textit{and}\ X(w_i^{\bullet}) \neq X(u) \\
M^k & \textit{if}\ X(w_i^{\bullet}) = X(u)\ \textit{and}\ X(w_i^{\circ}) \neq X(u) \\
X(\boldsymbol{w}_i) & \textit{else.}
\end{cases}
$$

Now the following implication holds: *If $\mathcal{M} \vDash_X \mu'$, then $\mathcal{M}' \vDash_X \mu$.*

We prove this claim by structural induction on μ:

– If μ is a literal and P_i does not occur in μ for any $i \le n$, then the claim holds trivially since $\mu' = \mu$.

– Let $\mu = P_j\boldsymbol{t}$ for some $j \le n$.
Suppose that $\mathcal{M} \vDash_X (P_j\boldsymbol{t})'$, i.e. $\mathcal{M} \vDash_X (\boldsymbol{t} \mid \boldsymbol{w}_j^c \vee w_j^{\bullet} = u) \wedge w_j^{\circ} \neq u$. Because the values of u, w_j° are constants in X and $\mathcal{M} \vDash_X w_j^{\circ} \neq u$, we have $X(w_j^{\circ}) \neq X(u)$. If $X(w_j^{\bullet}) = X(u)$, then $A_j = M^k$ and thus trivially $\mathcal{M}' \vDash_X P_j\boldsymbol{t}$.
Suppose then that $X(w_j^{\bullet}) \neq X(u)$ whence $A_j = X(\boldsymbol{w}_j)$. Because the values of u, $w_j^{\bullet}$ are constants in X and $\mathcal{M} \vDash_X (\boldsymbol{t} \mid \boldsymbol{w}_j^c \vee w_j^{\bullet} = u)$, it must hold that $\mathcal{M} \vDash_X \boldsymbol{t} \mid \boldsymbol{w}_j^c$. Now $X(\boldsymbol{t}) \cap X(\boldsymbol{w}_j^c) = \emptyset$ and $X(\boldsymbol{w}_j) \cup X(\boldsymbol{w}_j^c) = M^k$. Thus $X(\boldsymbol{t}) \subseteq \overline{X(\boldsymbol{w}_j^c)} \subseteq X(\boldsymbol{w}_j) = A_j$ and therefore $\mathcal{M}' \vDash_X P_j\boldsymbol{t}$.

– Let $\mu = \neg P_j \boldsymbol{t}$ for some $j \leq n$.
  Suppose first that $\mathcal{M} \vDash_X (\neg P_j \boldsymbol{t})'$, i.e. $\mathcal{M} \vDash_X (\boldsymbol{t} \mid \boldsymbol{w}_j \vee w_j^\circ = u) \wedge w_j^\bullet \neq u$. Because the values of u, $w_j^\bullet$ are constants and $\mathcal{M} \vDash_X w_j^\bullet \neq u$, we have $X(w_j^\bullet) \neq X(u)$. If $X(w_j^\circ) = X(u)$, then $A_j = \emptyset$ and thus trivially $\mathcal{M}' \vDash_X \neg P_j \boldsymbol{t}$.
  Suppose then that $X(w_i^\circ) \neq X(u)$ whence $A_j = X(\boldsymbol{w}_j)$. Because the values of u, w_j° are constants in X and $\mathcal{M} \vDash_X \boldsymbol{t} \mid \boldsymbol{w}_j \vee w_j^\circ = u$, we have $\mathcal{M} \vDash_X \boldsymbol{t} \mid \boldsymbol{w}_j$. Now $X(\boldsymbol{t}) \subseteq \overline{X(\boldsymbol{w}_j)} = \overline{A_j}$ and thus $\mathcal{M}' \vDash_X \neg P_j \boldsymbol{t}$.
– The case $\mu = \psi \wedge \theta$ is straightforward to prove.
– Let $\mu = \psi \vee \theta$.
  Suppose that $\mathcal{M} \vDash_X (\psi \vee \theta)'$, i.e. $\mathcal{M} \vDash_X \psi' \veebar^{\mathbf{U}} \theta'$. By Proposition 5 there exist $Y_1, Y_2 \subseteq X$ s.t. $Y_1 \cup Y_2 = X$, $\mathcal{M} \vDash_{Y_1^*} \psi'$ and $\mathcal{M} \vDash_{Y_2^*} \theta'$, where

$$\begin{cases} Y_1^* := Y_1[X(\boldsymbol{w}_1)/\boldsymbol{w}_1, \ldots, X(\boldsymbol{w}_n)/\boldsymbol{w}_n, X(\boldsymbol{w}_1^c)/\boldsymbol{w}_1^c, \ldots, X(\boldsymbol{w}_n^c)/\boldsymbol{w}_n^c] \\ Y_2^* := Y_2[X(\boldsymbol{w}_1)/\boldsymbol{w}_1, \ldots, X(\boldsymbol{w}_n)/\boldsymbol{w}_n, X(\boldsymbol{w}_1^c)/\boldsymbol{w}_1^c, \ldots, X(\boldsymbol{w}_n^c)/\boldsymbol{w}_n^c]. \end{cases}$$

  Now the sets of values for $\boldsymbol{w}_i$ and $\boldsymbol{w}_i^c$ are the same in Y_1^* and Y_2^* as in X. Because the values of u and w_i°, $w_i^\bullet$ are constants in X they have (the same) constant values in Y_1^* and Y_2^*. Hence, by the induction hypothesis, we have $\mathcal{M}' \vDash_{Y_1^*} \psi$ and $\mathcal{M}' \vDash_{Y_2^*} \theta$. Since none of the variables in V^* occurs in $\psi \vee \theta$, by locality $\mathcal{M}' \vDash_{Y_1} \psi$ and $\mathcal{M}' \vDash_{Y_2} \theta$. Therefore $\mathcal{M}' \vDash_X \psi \vee \theta$.
– The cases $\mu = \exists x\, \psi$ and $\mu = \forall x\, \psi$ are straightforward to prove.
  (Note here that, since $x \notin V^*$, the assumptions of Lemma 2 hold in the resulting team also after the quantification of x.)

Lemma 3. *Let $\mathcal{M}$ be an L-model with at least two elements. Let $\mu \in \mathrm{Sf}(\delta)$ and X be a team such that $\mathrm{dom}(X) = \mathrm{Fr}(\mu)$. Assume that $A_1, \ldots, A_n \subseteq M^k$, $\mathcal{M}' := \mathcal{M}[\boldsymbol{A}/\boldsymbol{P}]$ and $a, b \in M$ s.t. $a \neq b$. Let*

$$X' := X\big[\{a\}/u, B_1^\circ/w_1^\circ, \ldots, B_n^\circ/w_n^\circ, B_1^\bullet/w_1^\bullet, \ldots, B_n^\bullet/w_n^\bullet,$$
$$B_1/\boldsymbol{w}_1, \ldots, B_n/\boldsymbol{w}_n, B_1^c/\boldsymbol{w}_1^c, \ldots, B_n^c/\boldsymbol{w}_n^c\big],$$
$$\textit{where} \begin{cases} B_i^\circ = \{a\},\ B_i^\bullet = \{b\} \textit{ and } B_i = B_i^c = M & \textit{if } A_i = \emptyset \\ B_i^\circ = \{b\},\ B_i^\bullet = \{a\} \textit{ and } B_i = B_i^c = M & \textit{if } A_i = M^k \\ B_i^\circ = \{b\}, B_i^\bullet = \{b\}, B_i = A_i \textit{ and } B_i^c = \overline{A_i} & \textit{else.} \end{cases}$$

Now the following implication holds: If $\mathcal{M}' \vDash_X \mu$, then $\mathcal{M} \vDash_{X'} \mu'$.

We prove this claim by structural induction on μ. Note that if $X = \emptyset$, then also $X' = \emptyset$ and thus the claim holds by empty team property. Hence we may assume that $X \neq \emptyset$.

– If μ is a literal and P_i does not occur in μ for any $i \leq n$, then the claim holds by locality since $\mu' = \mu$.
– Let $\mu = P_j \boldsymbol{t}$ for some $j \leq n$.
  Suppose first that $\mathcal{M}' \vDash_X P_j \boldsymbol{t}$, i.e. $X(\boldsymbol{t}) \subseteq P_j^{\mathcal{M}'} = A_j$. Since $X \neq \emptyset$, also $X(\boldsymbol{t}) \neq \emptyset$ and thus $A_j \neq \emptyset$. Hence $X'(w_j^\circ) = \{b\}$, and thus $\mathcal{M} \vDash_{X'} w_i^\circ \neq u$

since $X'(u) = \{a\}$. If $A_j = M^k$, then $X'(w_i^\bullet) = \{a\}$ and thus $\mathcal{M} \vDash_{X'} w_j^\bullet = u$, whence $\mathcal{M} \vDash_{X'} (\boldsymbol{t} \mid \boldsymbol{w}_i^c \vee w_i^\bullet = u) \wedge w_i^\circ \neq u$, i.e. $\mathcal{M} \vDash_{X'} (P_j \boldsymbol{t})'$.

Suppose then that $A_j \neq M^k$. Now we have $X'(\boldsymbol{w}_j^c) = \overline{A_j}$, i.e. $\overline{X'(w_j^c)} = A_j$, and thus $X'(\boldsymbol{t}) = X(\boldsymbol{t}) \subseteq A_j = \overline{X'(\boldsymbol{w}_j^c)}$. Hence $\mathcal{M} \vDash_{X'} \boldsymbol{t} \mid \boldsymbol{w}_j^c$ and therefore $\mathcal{M} \vDash_{X'} (\boldsymbol{t} \mid \boldsymbol{w}_i^c \vee w_i^\bullet = u) \wedge w_i^\circ \neq u$, i.e. $\mathcal{M} \vDash_{X'} (P_j \boldsymbol{t})'$.

– Let $\mu = \neg P_j \boldsymbol{t}$ for some $j \leq n$.

Suppose that $\mathcal{M}' \vDash_X \neg P_j \boldsymbol{t}$, i.e. $X(\boldsymbol{t}) \subseteq \overline{P_j^{\mathcal{M}'}} = \overline{A_j}$. Since $X \neq \emptyset$, $X(\boldsymbol{t}) \neq \emptyset$ and thus $\overline{A_j} \neq \emptyset$, i.e. $A_j \neq M^k$. Hence $X'(w_j^\bullet) = \{b\}$, and thus $\mathcal{M} \vDash_{X'} w_i^\bullet \neq u$ since $X'(u) = \{a\}$. If $A_j = \emptyset$, then $X'(w_i^\circ) = \{a\}$ and thus $\mathcal{M} \vDash_{X'} w_j^\circ = u$, whence $\mathcal{M} \vDash_{X'} (\boldsymbol{t} \mid \boldsymbol{w}_i \vee w_i^\circ = u) \wedge w_i^\bullet \neq u$, i.e. $\mathcal{M} \vDash_{X'} (\neg P_j \boldsymbol{t})'$.

Suppose then that we have $A_j \neq \emptyset$. Then $X'(\boldsymbol{w}_j) = A_j$ and thus it holds that $X'(\boldsymbol{t}) = X(\boldsymbol{t}) \subseteq \overline{A_j} = \overline{X'(w_j)}$. Hence we have $\mathcal{M} \vDash_{X'} \boldsymbol{t} \mid \boldsymbol{w}_j$ and therefore $\mathcal{M} \vDash_{X'} (\boldsymbol{t} \mid \boldsymbol{w}_i \vee w_i^\circ = u) \wedge w_i^\bullet \neq u$, i.e. $\mathcal{M} \vDash_{X'} (\neg P_j \boldsymbol{t})'$.

– The case $\mu = \psi \wedge \theta$ is straightforward to prove.

– Let $\mu = \psi \vee \theta$.

Suppose that $\mathcal{M}' \vDash_X \psi \vee \theta$, i.e. there exist $Y_1, Y_2 \subseteq X$ s.t. $Y_1 \cup Y_2 = X$, $\mathcal{M}' \vDash_{Y_1} \psi$ and $\mathcal{M}' \vDash_{Y_2} \theta$. Let Y_1', Y_2' be the teams obtained by extending the teams Y_1, Y_2 as X' is obtained by extending X. Then, by the induction hypothesis, we have $\mathcal{M} \vDash_{Y_1'} \psi'$ and $\mathcal{M} \vDash_{Y_2'} \theta'$. Now the following holds:

$$\begin{cases} Y_1' = Y_1'[X(\boldsymbol{w}_1)/\boldsymbol{w}_1, \ldots, X(\boldsymbol{w}_n)/\boldsymbol{w}_n, X(\boldsymbol{w}_1^c)/\boldsymbol{w}_1^c, \ldots, X(\boldsymbol{w}_n^c)/\boldsymbol{w}_n^c] \\ Y_2' = Y_2'[X(\boldsymbol{w}_1)/\boldsymbol{w}_1, \ldots, X(\boldsymbol{w}_n)/\boldsymbol{w}_n, X(\boldsymbol{w}_1^c)/\boldsymbol{w}_1^c, \ldots, X(\boldsymbol{w}_n^c)/\boldsymbol{w}_n^c]. \end{cases}$$

Note that also $Y_1', Y_2' \subseteq X'$ and $Y_1' \cup Y_2' = X'$. Thus by Proposition 5 $\mathcal{M} \vDash_{X'} \psi' \veebar^{\mathbf{U}} \theta'$, i.e. $\mathcal{M} \vDash_{X'} (\psi \vee \theta)'$.

– Let $\mu = \exists x\, \psi$ (the case $\mu = \forall x\, \psi$ is proven similarly).

Suppose that $\mathcal{M}' \vDash_X \exists x\, \psi$, i.e. there exists $F : X \to M$ s.t. $\mathcal{M}' \vDash_{X[F/x]} \psi$. Let $F' : X' \to M$ such that $s \mapsto F(s \upharpoonright \mathrm{Fr}(\mu))$ for each $s \in X'$. Note that F' is well defined since $\mathrm{dom}(X) = \mathrm{Fr}(\mu)$ by the assumption.

Let $(X[F/x])'$ be a team that is obtained by extending the team $X[F/x]$ analogously as X' is obtained by extending X. Now by induction hypothesis we have $\mathcal{M} \vDash_{(X[F/x])'} \psi'$. By the definition of F' it is easy to see that $(X[F/x])' = X'[F'/x]$ and thus $\mathcal{M} \vDash_{X'[F'/x]} \psi'$. Hence we have $\mathcal{M} \vDash_{X'} \exists x\, \psi'$, i.e. $\mathcal{M} \vDash_{X'} (\exists x\, \psi)'$.

We are ready prove: $\mathcal{M} \vDash \varphi$ iff $\mathcal{M} \vDash \Phi$. Suppose first $\mathcal{M} \vDash \varphi$, i.e. $\mathcal{M} \vDash \gamma_{=1} \wedge \chi$ or

$$\mathcal{M} \vDash \exists u \exists w_1^\circ \ldots \exists w_n^\circ \exists w_1^\bullet \ldots \exists w_n^\bullet \\ \forall \boldsymbol{z}\, \exists \boldsymbol{w}_1 \ldots \exists \boldsymbol{w}_n \exists \boldsymbol{w}_1^c \ldots \exists \boldsymbol{w}_n^c \big(\bigwedge_{i \leq n} (\boldsymbol{z} = \boldsymbol{w}_i \vee \boldsymbol{z} = \boldsymbol{w}_i^c) \wedge \delta' \big). \qquad (\star)$$

If $\mathcal{M} \vDash \gamma_{=1} \wedge \chi$, the claim holds by Lemma 1. Suppose then($\star$), whence by the (strict) semantics of existential quantifier there are $a, b_1 \ldots b_n, b_1', \ldots, b_n' \in M$ s.t.

$$\mathcal{M} \vDash_{X_1} \forall \boldsymbol{z}\, \exists \boldsymbol{w}_1 \ldots \exists \boldsymbol{w}_n \exists \boldsymbol{w}_1^c \ldots \exists \boldsymbol{w}_n^c \big(\bigwedge_{i \leq n} (\boldsymbol{z} = \boldsymbol{w}_i \vee \boldsymbol{z} = \boldsymbol{w}_i^c) \wedge \delta' \big),$$

where $X_1 := \{\emptyset[a/u, b_1/w_1^\circ, \dots, b_n/w_n^\circ, b_1'/w_1^\bullet, \dots, b_n'/w_n^\bullet]\}$. Note that since X_1 consists only of a single assignment, the values of u, w_i° and $w_i^\bullet$ $(i \leq n)$ are trivially constants in the team X_1. Let $X_2 := X_1[M^k/\boldsymbol{z}]$. Now there exist functions $\mathcal{F}_i : X_2[\mathcal{F}_1/\boldsymbol{w}_1, \dots, \mathcal{F}_{i-1}/\boldsymbol{w}_{i-1}] \to M^k$ such that

$$\mathcal{M} \vDash_{X_3} \exists\, \boldsymbol{w}_1^c \dots \exists\, \boldsymbol{w}_n^c \Big(\bigwedge_{i \leq n} (\boldsymbol{z} = \boldsymbol{w}_i \vee \boldsymbol{z} = \boldsymbol{w}_i^c) \wedge \delta' \Big),$$

where $X_3 := X_2[\mathcal{F}_1/\boldsymbol{w}_1, \dots, \mathcal{F}_n/\boldsymbol{w}_n]$.

Hence there exist functions $\mathcal{F}_i' : X_3[\mathcal{F}_1'/\boldsymbol{w}_1^c, \dots, \mathcal{F}_{i-1}'/\boldsymbol{w}_{i-1}^c] \to M^k$ such that $\mathcal{M} \vDash_{X_4} \bigwedge_{i \leq n} (\boldsymbol{z} = \boldsymbol{w}_i \vee \boldsymbol{z} = \boldsymbol{w}_i^c) \wedge \delta'$, where $X_4 := X_3[\mathcal{F}_1'/\boldsymbol{w}_1^c, \dots, \mathcal{F}_n'/\boldsymbol{w}_n^c]$. Since $X_4(\boldsymbol{z}) = M^k$ and $\mathcal{M} \vDash_{X_4} \bigwedge_{i \leq n} (\boldsymbol{z} = \boldsymbol{w}_i \vee \boldsymbol{z} = \boldsymbol{w}_i^c)$, it is easy to see that $X(\boldsymbol{w}_i) \cup X(\boldsymbol{w}_i^c) = M^k$ for each $i \leq n$. Now all the assumptions of Lemma 2 hold for the team X_4. Let $\mathcal{M}' := \mathcal{M}[\boldsymbol{A}/\boldsymbol{P}]$, where

$$A_i = \begin{cases} \emptyset & \text{if } X_4(w_i^\circ) = X_4(u) \text{ and } X_4(w_i^\bullet) \neq X_4(u) \\ M^k & \text{if } X_4(w_i^\bullet) = X_4(u) \text{ and } X_4(w_i^\circ) \neq X_4(u) \\ X_4(\boldsymbol{w}_i) & \text{else.} \end{cases}$$

Since $\mathcal{M} \vDash_{X_4} \delta'$, by Lemma 2 we have $\mathcal{M}' \vDash_{X_4} \delta$. By locality $\mathcal{M}' \vDash \delta$, and therefore $\mathcal{M} \vDash \varPhi$.

Suppose then that $\mathcal{M} \vDash \varPhi$. If $|M| = 1$, then by Lemma 1 we have $\mathcal{M} \vDash \gamma_{=1} \wedge \chi$ and thus $\mathcal{M} \vDash \varphi$. Hence we may assume that $|M| \geq 2$, whence there exist $a, b \in M$ s.t. $a \neq b$. Since $\mathcal{M} \vDash \varPhi$, there exist $A_1, \dots, A_n \subseteq M^k$ s.t. $\mathcal{M}[\boldsymbol{A}/\boldsymbol{P}] \vDash \delta$. Let

$$\begin{aligned} X' := \{\emptyset\}\big[\{a\}/u, B_1^\circ/w_1^\circ, \dots, B_n^\circ/w_n^\circ, B_1^\bullet/w_1^\bullet, \dots, B_n^\bullet/w_n^\bullet, \\ B_1/\boldsymbol{w}_1, \dots, B_n/\boldsymbol{w}_n, B_1^c/\boldsymbol{w}_1^c, \dots, B_n^c/\boldsymbol{w}_n^c\big], \end{aligned}$$

where $B_i^\circ, B_i^\bullet, B_i, B_i^c$ $(i \leq n)$ are defined as in the assumptions of Lemma 3. Since $\mathcal{M}[\boldsymbol{A}/\boldsymbol{P}] \vDash \delta$, by Lemma 3 we have $\mathcal{M} \vDash_{X'} \delta'$. Let

$$\mathcal{F} : \{\emptyset\} \to M^{2n+1}, \ \emptyset \mapsto ab_1 \dots b_n b_1' \dots b_n',$$
$$\text{where} \quad \begin{cases} b_i = a & \text{if } A_i = \emptyset \\ b_i = b & \text{else} \end{cases} \quad \text{and} \quad \begin{cases} b_i' = a & \text{if } A_i = M^k \\ b_i' = b & \text{else.} \end{cases}$$

Let $X_1 := \{\emptyset\}[\mathcal{F}/uw_1^\circ \dots w_n^\circ w_1^\bullet \dots w_n^\bullet]$ and let $X_2 := X_1[M^k/\boldsymbol{z}]$. We fix some $\boldsymbol{b}_i \in A_i$ for each $i \leq n$ for which $A_i \neq \emptyset$ and define the functions

$$\mathcal{F}_i : X_2[\mathcal{F}_1/\boldsymbol{w}_1, \dots, \mathcal{F}_{i-1}/\boldsymbol{w}_{i-1}] \to M^k, \quad \begin{cases} s \mapsto s(\boldsymbol{z}) & \text{if } s(\boldsymbol{z}) \in A_i \text{ or } A_i = \emptyset \\ s \mapsto \boldsymbol{b}_i & \text{else.} \end{cases}$$

Let $X_3 := X_2[\mathcal{F}_1/\boldsymbol{w}_1, \dots, \mathcal{F}_n/\boldsymbol{w}_n]$. We fix some $\boldsymbol{b}_i' \in \overline{A_i}$ for each $i \leq n$ for which $A_i \neq M^k$ and define

$$\mathcal{F}_i' : X_3[\mathcal{F}_1'/\boldsymbol{w}_1^c, \dots, \mathcal{F}_{i-1}'/\boldsymbol{w}_{i-1}^c] \to M^k, \begin{cases} s \mapsto s(\boldsymbol{z}) & \text{if } s(\boldsymbol{z}) \in \overline{A_i} \text{ or } A_i = M^k \\ s \mapsto \boldsymbol{b}_i' & \text{else.} \end{cases}$$

Let $X_4 := X_3[\mathcal{F}_1'/\boldsymbol{w}_1^c, \dots, \mathcal{F}_n'/\boldsymbol{w}_n^c]$. By the definitions of the functions $\mathcal{F}_i, \mathcal{F}_i'$ it is easy to see that $\mathcal{M} \vDash_{X_4} \bigwedge_{i \leq n}(\boldsymbol{z} = \boldsymbol{w}_i \vee \boldsymbol{z} = \boldsymbol{w}_i^c)$. Also, clearly for each $s \in X_4$ and $i \leq n$ it holds that $s(w_i^\circ) \in B_i^\circ$, $s(w_i^\bullet) \in B_i^\bullet$, $s(\boldsymbol{w}_i) \in B_i$ and $s(\boldsymbol{w}_i^c) \in B_i^c$. By the definitions of the choice functions for the variables in V^* we have:

$$X_4[X_4(\boldsymbol{w}_1)/\boldsymbol{w}_1, \dots, X_4(\boldsymbol{w}_n)/\boldsymbol{w}_n, X_4(\boldsymbol{w}_1)/\boldsymbol{w}_1^c, \dots, X_4(\boldsymbol{w}_n)/\boldsymbol{w}_n^c] = X'[M^k/\boldsymbol{z}].$$

Hence $X_4 \subseteq X'[M^k/\boldsymbol{z}]$. Since $\mathcal{M} \vDash_{X'} \delta'$, by locality $\mathcal{M} \vDash_{X'[M^k/\boldsymbol{z}]} \delta'$. Since EXC is closed downwards, we have $\mathcal{M} \vDash_{X_4} \delta'$. Hence $\mathcal{M} \vDash_{X_4} \bigwedge_{i \leq n}(\boldsymbol{z} = \boldsymbol{w}_i \vee \boldsymbol{z} = \boldsymbol{w}_i^c) \wedge \delta'$ and furthermore $\mathcal{M} \vDash \varphi$.

Corollary 3. *On the level of sentences* EXC$[k] \equiv$ ESO$[k]$.

Proof. In [17] we presented a translation from EXC$[k]$ to ESO$[k]$. By Theorem 1, on the level of sentences, there is also a translation from ESO$[k]$ to EXC$[k]$.

4.3 Relationship Between INC[k] and EXC[k]

Since by [17] INEX$[k]$ captures ESO$[k]$, by Corollary 3 we can deduce that INEX$[k] \equiv$ EXC$[k]$ on the level of sentences. Hence on the level of sentences k-ary inclusion atoms do not increase the expressive power of EXC$[k]$.

By Dawar [2], 3-colorability of a graph cannot be expressed in fixed point logic. Since by [7] INC is equivalent with positive greatest fixed point logic, this property is not expressible in INC. However, since it can be expressed in EXC[1] (Example 2), INC$[k]$ is *strictly* weaker than EXC$[k]$ on the level of sentences.

This consequence is somewhat surprising since inclusion and exclusion atoms can be seen as duals of each other [17]. As a matter of fact, exclusion atoms can also be simulated with inclusion atoms in an analogous way as we simulated inclusion atoms with exclusion atoms. To see this, suppose that X is a team and $\boldsymbol{x}$, $\boldsymbol{w}$, $\boldsymbol{w}^c$ are tuples variables s.t. $X(\boldsymbol{w}^c) = \overline{X(\boldsymbol{w})}$. Now we have: $\mathcal{M} \vDash_X \boldsymbol{x} \mid \boldsymbol{w}$ iff $\mathcal{M} \vDash_X \boldsymbol{x} \subseteq \boldsymbol{w}^c$ (compare with our observation in the beginning of Sect. 4.2).

By this observation, it would be natural to assume that ESO$_L[k]$-sentences could be expressed with INC$[k]$-sentences similarly as we did with EXC$[k]$-sentences. But this is impossible as we deduced above. The problem is that in INC there is no way to "force" the tuples $\boldsymbol{w}$ and $\boldsymbol{w}^c$ to be quantified in such a way that their values would be complements of each other. However, there is a possibility this could be done in inclusion logic with *strict semantics*, since Hannula and Kontinen [10] have shown that this logic is equivalent with ESO. We will study this question in a future work.

5 Conclusion

In this paper we analyzed the expressive power of k-ary exclusion atoms. We first observed that the expressive power of EXC$[k]$ is between k-ary and $(k+1)$-ary dependence logics, and that when $k = 1$, these inclusions are proper. By simulating the use of inclusion atoms with exclusion atoms and by using the complementary values, we were able to translate ESO[k]-sentences into EXC[k]. By combining this with our earlier translation we managed to capture the k-ary fragment of ESO by using only k-ary exclusion atoms, which resolves the expressive power of EXC[k] on the level of sentences. However, on the level of formulas our results are not yet conclusive.

As mentioned in the introduction, by [3], on the level of sentences k-ary dependence logic captures the fragment of ESO where $(k-1)$-ary functions can be quantified. Thus 1-ary dependence logic is not more expressive than FO, but 2-ary dependence logic is strictly stronger than EMSO – which can be captured with EXC[1]. Also, the question whether EXC[k] is properly in between k- and $(k+1)$-ary dependence logic for all $k \geq 2$, amounts to showing whether k-ary relational fragment of ESO is properly between $(k-1)$-ary and k-ary functional fragments of ESO for any $k \geq 2$. To our best knowledge this is still an open problem, even though, by the result of Ajtai [1], both relational and functional fragments of ESO have a strict arity hierarchy (over arbitrary vocabulary).

In order to formulate the translation in our main theorem, we needed use a new operator to called *unifier* which is expressible in exclusion logic. This is a very simple but interesting operator for the framework of team semantics by its own right, and its properties deserve to be studied further – either independently or by adding it to some other logics with team semantics.

References

1. Ajtai, M.: Σ_1^1-formulae on finite structures. Ann. Pure Appl. Logic **724**(1), 1–48 (1983)
2. Dawar, A.: A restricted second order logic for finite structures. Inf. Comput. **143**(2), 154–174 (1998)
3. Durand, A., Kontinen, J.: Hierarchies in dependence logic. ACM Trans. Comput. Log. **13**(4), 31 (2012)
4. Galliani, P.: The Dynamics of Imperfect Information. Institute for Logic Language and Computation, Amsterdam (2012)
5. Galliani, P.: Inclusion and exclusion dependencies in team semantics - on some logics of imperfect information. Ann. Pure Appl. Logic **163**(1), 68–84 (2012)
6. Galliani, P., Hannula, M., Kontinen, J.: Hierarchies in independence logic. In: CSL 2013, pp. 263–280 (2013)
7. Galliani, P., Hella, L.: Inclusion logic and fixed point logic. In: CSL 2013, pp. 281–295 (2013)
8. Grädel, E., Väänänen, J.A.: Dependence and independence. Studia Logica **101**(2), 399–410 (2013)
9. Hannula, M.: Hierarchies in inclusion logic with lax semantics. In: Banerjee, M., Krishna, S.N. (eds.) ICLA. LNCS, vol. 8923, pp. 100–118. Springer, Heidelberg (2015)
10. Hannula, M., Kontinen, J.: Hierarchies in independence and inclusion logic with strict semantics. J. Log. Comput. **25**(3), 879–897 (2015)
11. Hintikka, J., Sandu, G.: Informational independence as a semantical phenomenon VIII. In: Fenstad, J.E. (ed.) Logic, Methodology and Philosophy of Science, pp. 571–589. Elsevier, Amsterdam (1989)

12. Hintikka, J., Sandu, G.: Game-theoretical semantics. In: van Benthem, J., ter Meulen, A. (eds.) Handbook of Logic and Language, pp. 361–410. Elsevier, Amsterdam (1997)
13. Hodges, W.: Compositional semantics for a language of imperfect information. Log. J. IGPL **5**(4), 539–563 (1997)
14. Hyttinen, T., Paolini, G., Väänänen, J.: Quantum team logic and Bell's inequalities. Rev. Symb. Log. **8**, 722–742 (2015)
15. Kontinen, J., Link, S., Väänänen, J.: Independence in database relations. In: Libkin, L., Kohlenbach, U., de Queiroz, R. (eds.) WoLLIC 2013. LNCS, vol. 8071, pp. 179–193. Springer, Heidelberg (2013)
16. Kontinen, J., Väänänen, J.A.: On definability in dependence logic. J. Log. Lang. Inf. **18**(3), 317–332 (2009)
17. Rönnholm, R.: Capturing k-ary inclusion-exclusion logic with k-ary existential second order logic (2015). arXiv:1502.05632 [math.LO]
18. Rönnholm, R.: The expressive power of k-ary exclusion logic (2016). arXiv:1605.01686 [math.LO]
19. Väänänen, J.A.: Dependence logic - a new approach to independence friendly logic. London Mathematical Society Student Texts, vol. 70. Cambridge University Press (2007)

Characterizing Relative Frame Definability in Team Semantics via the Universal Modality

Katsuhiko Sano[1] and Jonni Virtema[2,3(✉)]

[1] Japan Advanced Institute of Science and Technology, Nomi, Japan
katsuhiko.sano@gmail.com
[2] University of Helsinki, Helsinki, Finland
jonni.virtema@gmail.com
[3] Leibniz Universität Hannover, Hanover, Germany

Abstract. Let $\mathcal{ML}(\boxed{u}^+)$ denote the fragment of modal logic extended with the universal modality in which the universal modality occurs only positively. We characterise the relative definability of $\mathcal{ML}(\boxed{u}^+)$ relative to finite transitive frames in the spirit of the well-known Goldblatt–Thomason theorem. We show that a class $\mathbb{F}$ of finite transitive frames is definable in $\mathcal{ML}(\boxed{u}^+)$ relative to finite transitive frames if and only if $\mathbb{F}$ is closed under taking generated subframes and bounded morphic images. In addition, we study modal definability in team-based logics. We study (extended) modal dependence logic, (extended) modal inclusion logic, and modal team logic. With respect to global model definability we obtain a trichotomy and with respect to frame definability a dichotomy. As a corollary we obtain relative Goldblatt–Thomason -style theorems for each of the logics listed above.

1 Introduction

Team semantics was introduced by Hodges [15] in the context of the so-called *independence-friendly logic* of Hintikka and Sandu [14]. The fundamental idea behind team semantics is crisp. The idea is to shift from single assignments to sets of assignments as the satisfying elements of formulas. Väänänen [19] adopted team semantics as the core notion for his *dependence logic*. The syntax of first-order dependence logic extends the syntax of first-order logic by novel atomic formulas called dependence atoms. The intuitive meaning of the dependence atom $=(x_1, \ldots, x_n, y)$ is that inside a team the value of y is functionally determined by the values of $x_1, \ldots, x_n$. After the introduction of dependence logic in 2007 the study of related logics with team semantics has boomed. One of the most important developments in the area of team semantics was the introduction of *independence logic* [10] in which dependence atoms of dependence logic

The work of the first author was partially supported by JSPS KAKENHI Grant-in-Aid for Young Scientists (B) Grant Number 15K21025 and JSPS Core-to-Core Program (A. Advanced Research Networks). The work of the second author was supported by grant 292767 of the Academy of Finland, and by Jenny and Antti Wihuri Foundation.

J. Väänänen et al. (Eds.): WoLLIC 2016, LNCS 9803, pp. 392–409, 2016.
DOI: 10.1007/978-3-662-52921-8_24

are replaced by *independence atoms*. Soon after, Galliani [5] showed that independence atoms can be further analysed, and alternatively expressed, in terms of inclusion and exclusion atoms.

Concurrently a vibrant research on modal and propositional logics with team semantics has emerged. In the context of modal logic, any subset of the domain of a Kripke model is called a team. In modal team semantics, formulas are evaluated with respect to team-pointed Kripke models. The study of *modal dependence logic* was initiated by Väänänen [20] in 2008. Shortly after, *extended modal dependence logic* ($\mathcal{EMDL}$) was introduced by Ebbing et al. [4] and *modal independence logic* by Kontinen et al. [16]. The focus of the research has been in the computational complexity and expressive power. Hella et al. [11] established that exactly the properties of teams that have the so-called *empty team property*, are *downward closed* and closed under the so-called team *k-bisimulation*, for some finite k, are definable in $\mathcal{EMDL}$. Kontinen et al. [17] have shown that exactly the properties of teams that are closed under the team k-bisimulation are definable in the so-called *modal team logic*, whereas Hella and Stumf established [12] that the so-called *extended modal inclusion logic* is characterised by the empty team property, union closure, and closure under team k-bisimulation. See the survey [3] for a detailed exposition on the expressive power and computational complexity of related logics.

The study of frame definability in the team semantics context was initiated by Sano and Virtema [18]. Let $\mathcal{ML}(\boxed{\mathsf{u}}^+)$ denote the syntactic fragment of modal logic with universal modality in which the universal modality occurs only positively. Sano and Virtema established a surprising connection between $\mathcal{ML}(\boxed{\mathsf{u}}^+)$ and particular team-based modal logics and gave a Goldblatt–Thomason -style theorem for the logics in question. They showed that with respect to frame definability $\mathcal{ML}(\boxed{\mathsf{u}}^+)$, $\mathcal{MDL}$ and $\mathcal{EMDL}$ coincide. Moreover, they established that an elementary class of Kripke frames is definable in $\mathcal{ML}(\boxed{\mathsf{u}}^+)$ (and thus in $\mathcal{MDL}$ and $\mathcal{EMDL}$) if and only if it is closed under taking generated subframes and bounded morphic images, and reflects ultrafilter extensions and finitely generated subframes.

Since most familiar modal logics enjoy the finite model property, one may wonder if we can restrict our attention to classes of finite frames for characterizing modal definability. For basic modal logic this was done in [1]. It is immediate to see that the reflection of ultrafilter extensions should be redundant under such restriction because ultrafilter extensions of finite frames are just those frames themselves. Interestingly, a modally undefinable property sometimes becomes definable within a suitable class of finite frames. A first-order condition of irreflexivity (for any w, wRw fails) of the accessibility relation is known to be undefinable by a set of modal formulas, since the condition violates the closure of a modally definable class under surjective bounded morphisms (consider a bounded morphism sending a frame of two symmetric points to a frame of a single reflexive point). It is, however, also known that irreflexivity becomes definable within the class of finite transitive frames by the Loeb axiom

$\Box(\Box p \to p) \to \Box p$. Such phenomena motivate us to study relative definability also in the context of team-based modal logics.

In this paper, we provide Goldblatt–Thomason -style theorem for the relative definability of $\mathcal{ML}(\boxed{\mathrm{u}}^{+})$ relative to finite transitive frames in the spirit of [1] with the help of Jankov-Fine formulas (cf. [2, Theorem 3.21]). We show that a class $\mathbb{F}$ of finite transitive frames is definable in $\mathcal{ML}(\boxed{\mathrm{u}}^{+})$ relative to finite transitive frames if and only if $\mathbb{F}$ is closed under taking generated subframes and bounded morphic images. In addition, we study modal definability in team-based logics. We study (extended) modal dependence logic, (extended) modal inclusion logic, and modal team logic. We obtain strict hierarchies with respect to both global model definability and frame definability.

2 Modal Logic with Universal Modality

In this section, we introduce modal logic with universal modality and give some basic definitions and results concerning frame definability. In team-based logics it is customary to define the syntax in negation normal form, that is to assume that negations occur only in front of proposition symbols. This is due to the fact that the team semantics negation, that corresponds to the negation used in Kripke semantics, is not the contradictory negation of team semantics. Since in this article we consider extensions of modal logic in the framework of team semantics, we define the syntax of modal logic also in negation normal form.

Let Φ be a set of atomic propositions. The set of formulas for *modal logic* $\mathcal{ML}(\Phi)$ is generated by the following grammar

$$\varphi ::= p \mid \neg p \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid \Diamond\varphi \mid \Box\varphi, \quad \text{where } p \in \Phi.$$

The syntax of *modal logic with universal modality* $\mathcal{ML}(\boxed{\mathrm{u}})(\Phi)$ is obtained by extending the syntax of $\mathcal{ML}(\Phi)$ by the grammar rules

$$\varphi ::= \boxed{\mathrm{u}}\,\varphi \mid \Diamond_{\mathrm{u}}\,\varphi.$$

The syntax of *modal logic with positive universal modality* $\mathcal{ML}(\boxed{\mathrm{u}}^{+})(\Phi)$ is obtained by extending the syntax of $\mathcal{ML}(\Phi)$ by the grammar rule $\varphi ::= \boxed{\mathrm{u}}\,\varphi$. As usual, if the underlying set Φ of atomic propositions is clear from the context, we drop "(Φ)" and just write $\mathcal{ML}$, $\mathcal{ML}(\boxed{\mathrm{u}})$, etc. We also use the shorthands $\neg\varphi$, $\varphi \to \psi$, and $\varphi \leftrightarrow \psi$. By $\neg\varphi$ we denote the formula that can be obtained from $\neg\varphi$ by pushing all negations to the atomic level, and by $\varphi \to \psi$ and $\varphi \leftrightarrow \psi$, we denote $\neg\varphi \vee \psi$ and $(\varphi \to \psi) \wedge (\psi \to \varphi)$, respectively.

A (Kripke) *frame* is a pair $\mathfrak{F} = (W, R)$ where W, called the *domain* of $\mathfrak{F}$, is a non-empty set and $R \subseteq W \times W$ is a binary relation on W. By $\mathbb{F}_{\mathrm{all}}$, we denote the class of all frames. We use $|\mathfrak{F}|$ to denote the domain of the frame $\mathfrak{F}$. A (Kripke) *Φ-model* is a tuple $\mathfrak{M} = (W, R, V)$, where (W, R) is a frame and $V : \Phi \to \mathcal{P}(W)$ is a valuation of the proposition symbols. By $\mathbb{M}_{\mathrm{all}}(\Phi)$, we denote the class of all Φ-models. The semantics of modal logic, i.e., the *satisfaction relation* $\mathfrak{M}, w \Vdash \varphi$, is defined via pointed Φ-*models* as usual. For the universal modality $\boxed{u}$ and its dual $\Diamond_u$, we define

$$\begin{aligned} \mathfrak{M}, w \Vdash \boxed{u}\,\varphi &\quad \Leftrightarrow \mathfrak{M}, v \Vdash \varphi, \text{ for every } v \in W, \\ \mathfrak{M}, w \Vdash \Diamond_u\,\varphi &\quad \Leftrightarrow \mathfrak{M}, v \Vdash \varphi, \text{ for some } v \in W. \end{aligned}$$

A formula set Γ is *valid in a model* $\mathfrak{M} = (W, R, V)$ (notation: $\mathfrak{M} \Vdash \Gamma$), if $\mathfrak{M}, w \Vdash \varphi$ holds for every $w \in W$ and every $\varphi \in \Gamma$. When Γ is a singleton $\{\varphi\}$, we simply write $\mathfrak{M} \Vdash \varphi$.

Below we assume only that the logics $\mathcal{L}(\Phi)$ and $\mathcal{L}'(\Phi)$ are such that the global satisfaction relation for Kripke models (i.e., $\mathfrak{M} \Vdash \varphi$) is defined. A set Γ of $\mathcal{L}(\Phi)$-formulas is *valid in a frame* $\mathfrak{F}$ (written: $\mathfrak{F} \Vdash \Gamma$) if $(\mathfrak{F}, V) \Vdash \varphi$ for every valuation $V : \Phi \to \mathcal{P}(W)$ and every $\varphi \in \Gamma$. A set Γ of $\mathcal{L}(\Phi)$-formulas is *valid in a class* $\mathbb{F}$ *of frames* (written: $\mathbb{F} \Vdash \Gamma$) if $\mathfrak{F} \Vdash \Gamma$ for every $\mathfrak{F} \in \mathbb{F}$. Given a set Γ of $\mathcal{L}(\Phi)$-formulas, $\mathbb{FR}(\Gamma) := \{\mathfrak{F} \in \mathbb{F}_{\text{all}} \mid \mathfrak{F} \Vdash \Gamma\}$ and $\text{Mod}(\Gamma) := \{\mathfrak{M} \in \mathbb{M}_{\text{all}}(\Phi) \mid \mathfrak{M} \Vdash \Gamma\}$. We say that Γ *defines* the class $\mathbb{F}$ of frames and the class $\mathbb{C}$ of models, if $\mathbb{F} = \mathbb{FR}(\Gamma)$ and $\mathbb{C} = \text{Mod}(\Gamma)$, respectively. When Γ is a singleton $\{\varphi\}$, we simply say that φ defines the class $\mathbb{F}$ (or $\mathbb{C}$). A class $\mathbb{F}$ of frames (models) is $\mathcal{L}(\Phi)$-*definable* if there exists a set Γ of $\mathcal{L}(\Phi)$-formulas such that $\mathbb{FR}(\Gamma) = \mathbb{F}$ ($\text{Mod}(\Gamma) = \mathbb{F}$).

It was shown in [18] that with respect to frame definability, we have that $\mathcal{ML} < \mathcal{ML}(\boxed{u}^+) < \mathcal{ML}(\boxed{u})$. Moreover the frame definability of each of the mentioned logics have been characterised with respect to first-order definable frame classes. For the characterisations the notions of *disjoint unions*, *generated subframes*, *bounded morphisms*, and *ultrafilter extensions* are required. Definitions for these constructions can be found, e.g., in [2], and in Appendix B.

The following results were proved for $\mathcal{ML}$ by Goldblatt and Thomason [7], for $\mathcal{ML}(\boxed{u}^+)$ by Sano and Virtema [18], and for $\mathcal{ML}(\boxed{u})$ by Goranko and Passy [9]. A frame class $\mathbb{F}$ *reflects* finitely generated subframes whenever it is the case for all frames $\mathfrak{F}$ that, if every finitely generated subframe of $\mathfrak{F}$ is in $\mathbb{F}$, then $\mathfrak{F} \in \mathbb{F}$.

Theorem 1 (Goldblatt–Thomason theorems for $\mathcal{ML}$, $\mathcal{ML}(\boxed{u}^+)$ *and* $\mathcal{ML}(\boxed{u})$**).** *(i) An elementary frame class is* $\mathcal{ML}$*-definable if and only if it is closed under taking bounded morphic images, generated subframes, disjoint unions and reflects ultrafilter extensions.*
(ii) An elementary frame class is $\mathcal{ML}(\boxed{u}^+)$*-definable if and only if it is closed under taking generated subframes and bounded morphic images, and reflects ultrafilter extensions and finitely generated subframes.*
(iii) An elementary frame class is $\mathcal{ML}(\boxed{u})$*-definable if and only if it is closed under taking bounded morphic images and reflects ultrafilter extensions.*

3 Finite Goldblatt-Thomason-Style Theorem for Relative Modal Definability with Positive Universal Modality

Given a class $\mathbb{G}$ of frames, we say that a set of formulas *defines* a class $\mathbb{F}$ of frames within $\mathbb{G}$ if, for all frames $\mathfrak{F} \in \mathbb{G}$, the equivalence: $\mathfrak{F} \Vdash \varphi \Leftrightarrow \mathfrak{F} \in \mathbb{F}$ holds. A frame $\mathbb{F} = (W, R)$ is called *finite* whenever W is a finite set and *transitive* whenever R is a transitive relation. In what follows, let $\mathbb{F}_{\text{fintra}}$ be the class of all finite transitive frames and $\mathbb{F}_{\text{fin}}$ the class of all finite frames.

With the help of frame constructions such as bounded morphic images, disjoint unions, generated subframes, we first review the existing characterisations of relative $\mathcal{ML}$- and $\mathcal{ML}(\boxed{u})$-definability within the class of finite transitive frames. We then give a novel characterisation of relative $\mathcal{ML}(\boxed{u}^+)$-definability again within the class of finite transitive frames.

Theorem 2 (Finite Goldblatt–Thomason Theorems for $\mathcal{ML}$ [1] *and* $\mathcal{ML}(\boxed{u})$ [6]).

1. *A class of finite transitive frames is $\mathcal{ML}$-definable within the class $\mathbb{F}_{\text{fintra}}$ of all finite transitive frames if and only if it is closed under taking bounded morphic images, generated subframes, and disjoint unions.*
2. *A class of finite frames is $\mathcal{ML}(\boxed{u})$-definable within the class $\mathbb{F}_{\text{fin}}$ of all finite frames if and only if it is closed under taking bounded morphic images.*

In order to show the corresponding characterisation of relative definability in $\mathcal{ML}(\boxed{u}^+)$, a variant of the Jankov-Fine formula is defined.

Definition 1. *Let $\mathfrak{F} = (W, R)$ be a finite transitive frame. Put $W := \{ w_0, \dots, w_n \}$. Associate a new proposition variable p_{w_i} with each w_i and define $\Box^+\varphi := \Box\varphi \land \varphi$. The Jankov-Fine formula $\varphi_{\mathfrak{F},w_i}$ at w_i is defined as the conjunction of all the following formulas:*

1. p_{w_i}
2. $\Box(p_{w_0} \lor \dots \lor p_{w_n})$.
3. $\bigwedge \{ \Box^+(p_{w_i} \to \neg p_{w_j}) \mid w_i \neq w_j \}$.
4. $\bigwedge \{ \Box^+(p_{w_i} \to \Diamond p_{w_j}) \mid (w_i, w_j) \in R \}$.
5. $\bigwedge \{ \Box^+(p_{w_i} \to \neg\Diamond p_{w_j}) \mid (w_i, w_j) \notin R \}$.

The Jankov-Fine formula $\varphi_{\mathfrak{F}}$ is defined as $\bigvee_{w \in W} \boxed{u} \neg\varphi_{\mathfrak{F},w}$.

We note that the Jankov-Fine formula $\varphi_{\mathfrak{F},w_i}$ at w_i is an $\mathcal{ML}$-formula and thus the Jankov-Fine formula $\varphi_{\mathfrak{F}}$ is an $\mathcal{ML}(\boxed{u}^+)$-formula.

Lemma 1 (For a proof, see Appendix A). *Let $\mathfrak{F} = (W, R)$ be a finite transitive frame. For any transitive frame $\mathfrak{G}$, the following are equivalent:*

(i) *the Jankov-Fine formula $\varphi_{\mathfrak{F}}$ is not valid in $\mathfrak{G}$,*
(ii) *there is a finite set $Y \subseteq |\mathfrak{G}|$ such that $\mathfrak{F}$ is a bounded morphic image of $\mathfrak{G}_Y$, where $\mathfrak{G}_Y$ is the subframe of $\mathfrak{G}$ generated by Y.*

Theorem 3. *For every class $\mathbb{F}$ of finite transitive frames, the following are equivalent:*

(i) $\mathbb{F}$ *is $\mathcal{ML}(\boxed{u}^+)$-definable within $\mathbb{F}_{\text{fintra}}$.*
(ii) $\mathbb{F}$ *is closed under taking generated subframes and bounded morphic images.*

Proof. The direction from (i) to (ii) is easy to establish, so we focus on the converse direction. Assume (ii). Define $\mathrm{Log}(\mathbb{F}) = \{ \varphi \in \mathcal{ML}(\boxdot^+) \mid \mathbb{F} \Vdash \varphi \}$. We show that $\mathrm{Log}(\mathbb{F})$ defines $\mathbb{F}$ within $\mathbb{F}_{\mathrm{fintra}}$. Fix any finite and transitive frame $\mathfrak{F} \in \mathbb{F}_{\mathrm{fintra}}$. In what follows, we show the following equivalence:

$$\mathfrak{F} \in \mathbb{F} \iff \mathfrak{F} \Vdash \mathrm{Log}(\mathbb{F}).$$

The left-to-right direction is immediate, so we concentrate on the converse direction. Assume $\mathfrak{F} \Vdash \mathrm{Log}(\mathbb{F})$. Since $\mathfrak{F}$ is finite and transitive, let us take the Jankov-Fine formula $\varphi_{\mathfrak{F}}$. Since $\varphi_{\mathfrak{F}}$ is not valid in $\mathfrak{F}$, $\varphi_{\mathfrak{F}} \notin \mathrm{Log}(\mathbb{F})$. Thus there is a *transitive* frame $\mathfrak{G} \in \mathbb{F}$ (recall that $\mathbb{F}$ is a class of transitive frames) such that $\varphi_{\mathfrak{F}}$ is not valid in $\mathfrak{G}$. By Lemma 1, there is a finite set $Y \subseteq |\mathfrak{G}|$ such that $\mathfrak{F}$ is a bounded morphic image of $\mathfrak{G}_Y$. Since $\mathfrak{G} \in \mathbb{F}$, $\mathfrak{G}_Y \in \mathbb{F}$ by $\mathbb{F}$'s closure under generated subframes. It follows from $\mathbb{F}$'s closure under bounded morphic images that $\mathfrak{F} \in \mathbb{F}$, as desired. □

4 Modal Logics with Team Semantics

In this section we define the team-based modal logics that are relevant for this paper. We survey basic properties and known result concerning expressive power.

4.1 Basic Notions of Team Semantics

A subset T of the domain of a Kripke model $\mathfrak{M}$ is called *a team of* $\mathfrak{M}$. Before we define the so-called *team semantics* for $\mathcal{ML}$, let us first introduce some notation that makes defining the semantics simpler.

Definition 2. *Let $\mathfrak{M} = (W, R, V)$ be a model and T and S teams of $\mathfrak{M}$. Define*

$$R[T] := \{w \in W \mid \exists v \in T(vRw)\} \text{ and } R^{-1}[T] := \{w \in W \mid \exists v \in T(wRv)\}.$$

For teams T and S of $\mathfrak{M}$, we write $T[R]S$ if $S \subseteq R[T]$ and $T \subseteq R^{-1}[S]$.

Thus, $T[R]S$ holds if and only if for every $w \in T$ there exists some $v \in S$ such that wRv, and for every $v \in S$ there exists some $w \in T$ such that wRv. The team semantics for $\mathcal{ML}$ is defined as follows. We use the symbol "$\models$" for team semantics instead of the symbol "$\Vdash$" which was used for Kripke semantics.

Definition 3. *Let $\mathfrak{M}$ be a Kripke model and T a team of $\mathfrak{M}$. The satisfaction relation $\mathfrak{M}, T \models \varphi$ for $\mathcal{ML}(\Phi)$ is defined as follows.*

$$\begin{aligned}
\mathfrak{M}, T \models p &\quad\Leftrightarrow\quad w \in V(p) \text{ for every } w \in T.\\
\mathfrak{M}, T \models \neg p &\quad\Leftrightarrow\quad w \notin V(p) \text{ for every } w \in T.\\
\mathfrak{M}, T \models (\varphi \wedge \psi) &\quad\Leftrightarrow\quad \mathfrak{M}, T \models \varphi \text{ and } \mathfrak{M}, T \models \psi.\\
\mathfrak{M}, T \models (\varphi \vee \psi) &\quad\Leftrightarrow\quad \mathfrak{M}, T_1 \models \varphi \text{ and } \mathfrak{M}, T_2 \models \psi \text{ for some } T_1 \text{ and } T_2\\
&\qquad\qquad \text{such that } T_1 \cup T_2 = T.\\
\mathfrak{M}, T \models \Diamond\varphi &\quad\Leftrightarrow\quad \mathfrak{M}, T' \models \varphi \text{ for some } T' \text{ such that } T[R]T'.\\
\mathfrak{M}, T \models \Box\varphi &\quad\Leftrightarrow\quad \mathfrak{M}, T' \models \varphi, \text{ where } T' = R[T].
\end{aligned}$$

A set Γ of formulas is *valid in a model* $\mathfrak{M} = (W, R, V)$ (in team semantics), in symbols $\mathfrak{M} \models \Gamma$, if $\mathfrak{M}, T \models \varphi$ holds for every team T of $\mathfrak{M}$ and every $\varphi \in \Gamma$. Likewise, we say that Γ is *valid in a Kripke frame* $\mathfrak{F}$ and write $\mathfrak{F} \models \Gamma$, if $(\mathfrak{F}, V) \models \Gamma$ hold for every valuation V. When Γ is a singleton $\{\varphi\}$, we simply write $\mathfrak{M} \models \varphi$ and $\mathfrak{F} \models \varphi$.

The formulas of $\mathcal{ML}$ have the following flatness property.

Proposition 1 (Flatness). *Let $\mathfrak{M}$ be a Kripke model and T be a team of $\mathfrak{M}$. Then, for every formula φ of $\mathcal{ML}(\Phi)$*

$$\mathfrak{M}, T \models \varphi \Leftrightarrow \forall w \in T : \mathfrak{M}, w \Vdash \varphi.$$

From flatness if follows that for every model $\mathfrak{M}$, frame $\mathfrak{F}$, and formula φ of $\mathcal{ML}$, $\mathfrak{M} \Vdash \varphi$ iff $\mathfrak{M} \models \varphi$ and $\mathfrak{F} \Vdash \varphi$ iff $\mathfrak{F} \models \varphi$.

Recall from Sect. 2 what it means that a set of modal formulas defines a class of frames and models. All the related definitions can be adapted for logics with team semantics by simply substituting $\Vdash$ by $\models$.

Definition 4. *We write $\mathcal{L} \leq_M \mathcal{L}'$ if every $\mathcal{L}$-definable class of models is also $\mathcal{L}'$-definable. We write $\mathcal{L} =_M \mathcal{L}'$ if both $\mathcal{L} \leq_M \mathcal{L}'$ and $\mathcal{L}' \leq_M \mathcal{L}$ hold and write $\mathcal{L} <_M \mathcal{L}'$ if $\mathcal{L} \leq_M \mathcal{L}'$ but $\mathcal{L}' \not\leq_M \mathcal{L}$.*

Definition 5. *We write $\mathcal{L} \leq_F \mathcal{L}'$ if every $\mathcal{L}$-definable class of frames is also $\mathcal{L}'$-definable. We write $\mathcal{L} =_F \mathcal{L}'$ if both $\mathcal{L} \leq_F \mathcal{L}'$ and $\mathcal{L}' \leq_F \mathcal{L}$ hold and write $\mathcal{L} <_F \mathcal{L}'$ if $\mathcal{L} \leq_F \mathcal{L}'$ but $\mathcal{L}' \not\leq_F \mathcal{L}$.*

The most important closure properties in the study of team-based logics are downward closure, union closure, and the concept of team bisimulation.

Definition 6. *Let $\mathcal{L}$ be some team-based modal logic, $\mathfrak{M}$ a Kripke model, and T, S teams of $\mathfrak{M}$. We say that a formula $\varphi \in \mathcal{L}$ is*

1. downward closed *if $\mathfrak{M}, T \models \varphi$, whenever $\mathfrak{M}, S \models \varphi$ and $T \subseteq S$.*
2. union closed if *$\mathfrak{M}, T \cup S \models \varphi$, whenever $\mathfrak{M}, T \models \varphi$ and $\mathfrak{M}, S \models \varphi$.*

A logic $\mathcal{L}$ is called downward closed *(*union closed*) if every formula $\varphi \in \mathcal{L}$ is downward closed (union closed). We say that $\mathcal{L}$ has the* empty team *property, if $\mathfrak{M}, \emptyset \models \varphi$ holds for every model $\mathfrak{M}$ and every formula $\varphi \in \mathcal{L}$.*

Team bisimulation and its finite approximation team k-bisimulation can be defined via the corresponding concepts of ordinary modal logic. In the definition below, we denote by $\rightleftarrows$ and $\rightleftarrows_k$ the notions of bisimulation and k-bisimulation of ordinary modal logic (see, e.g., [2]), respectively.

Definition 7. *Let $\mathfrak{M}, T$ and $\mathfrak{M}', T'$ be team pointed Kripke models. We say that $\mathfrak{M}, T$ and $\mathfrak{M}', T'$ are team bisimilar, and write $\mathfrak{M}, T \; [\rightleftarrows] \; \mathfrak{M}', T'$ if*

1. *for every $w \in T$ there exist some $w' \in T'$ such that $\mathfrak{M}, w \rightleftarrows \mathfrak{M}', w'$, and*
2. *for every $w' \in T'$ there exist some $w \in T$ such that $\mathfrak{M}, w \rightleftarrows \mathfrak{M}', w'$.*

*The team k-*bisimulation *relation $[\rightleftarrows_k]$ is defined analogously with $\rightleftarrows$ replaced by $\rightleftarrows_k$.*

4.2 Extensions of Modal Logic via Connectives

We first introduce two expressive extensions of modal logic: an extension by the so-called intuitionistic disjunction and an extension by the so-called contradictory negation. These two logics are of great interest, since with respect to expressive power the logics subsume all most studied team-based modal logics, in particular all of those defined in Sect. 4.3.

Modal logic with intuitionistic disjunction $\mathcal{ML}(\varovee)(\Phi)$ is obtained by extending the syntax of $\mathcal{ML}(\Phi)$ by the grammar rule $\varphi ::= (\varphi \varovee \varphi)$ with the following semantics:

$$\mathfrak{M}, T \models (\varphi \varovee \psi) \quad \Leftrightarrow \quad \mathfrak{M}, T \models \varphi \text{ or } \mathfrak{M}, T \models \psi.$$

Modal team logic $\mathcal{MTL}(\Phi)$ is obtained by extending the syntax of $\mathcal{ML}(\Phi)$ by the contradictory negation, i.e., the grammar rule $\varphi ::= \sim\varphi$ with the following semantics:

$$\mathfrak{M}, T \models \sim\varphi \quad \Leftrightarrow \quad \mathfrak{M}, T \not\models \varphi.$$

The following theorem for $\mathcal{ML}(\varovee)$ was proven by Hella et al. [11] and for $\mathcal{MTL}$ by Kontinen et al. [17].

Theorem 4. *A class $\mathbb{C}$ of team pointed Kripke models is definable by a single formula of*

1. *$\mathcal{ML}(\varovee)$ iff $\mathbb{C}$ is downward closed, closed under team k-bisimulation, for some $k \in \mathbb{N}$, and admits the empty team property.*
2. *$\mathcal{MTL}$ iff $\mathbb{C}$ is closed under team k-bisimulation, for some $k \in \mathbb{N}$.*

4.3 Extensions of Modal Logic with Atomic Dependency Notions

The syntax of *modal dependence logic* $\mathcal{MDL}(\Phi)$ and *extended modal dependence logic* $\mathcal{EMDL}(\Phi)$ is obtained by extending the syntax of $\mathcal{ML}(\Phi)$ by the following grammar rule for each $n \in \omega$:

$$\varphi ::= \mathrm{dep}(\varphi_1, \ldots, \varphi_n, \psi)\,, \text{ where } \varphi_1, \ldots, \varphi_n, \psi \in \mathcal{ML}(\Phi).$$

In the additional grammar rules above for $\mathcal{MDL}$, we require that $\varphi_1, \ldots, \varphi_n, \psi$ are proposition symbols in Φ. The intuitive meaning of the (modal) dependence atom $\mathrm{dep}(\varphi_1, \ldots, \varphi_n, \psi)$ is that the truth value of the formula ψ is completely determined by the truth values of $\varphi_1, \ldots, \varphi_n$. The formal definition is given below:

$$\begin{aligned}\mathfrak{M}, T \models \mathrm{dep}(\varphi_1, \ldots, \varphi_n, \psi) \quad \Leftrightarrow \quad & \forall w, v \in T : \bigwedge_{1 \le i \le n} (\mathfrak{M}, \{w\} \models \varphi_i \Leftrightarrow \mathfrak{M}, \{v\} \models \varphi_i) \\ & \text{implies } (\mathfrak{M}, \{w\} \models \psi \Leftrightarrow \mathfrak{M}, \{v\} \models \psi).\end{aligned}$$

The syntax of *modal inclusion logic* $\mathcal{MINC}(\Phi)$ and *extended modal inclusion logic* $\mathcal{EMINC}(\Phi)$ is obtained by extending the syntax of $\mathcal{ML}(\Phi)$ by the following grammar rule for each $n \in \omega$:

$$\varphi ::= \varphi_1, \ldots, \varphi_n \subseteq \psi_1, \ldots, \psi_n, \text{ where } \varphi_1, \psi_1, \ldots, \varphi_n, \psi_n \in \mathcal{ML}(\Phi).$$

In the additional grammar rules above for $\mathcal{MINC}$, we require that the formulas $\varphi_1, \psi_1, \ldots, \varphi_n, \psi_n$ are proposition symbols in Φ. The meaning of the (modal) inclusion atom $\varphi_1, \ldots, \varphi_n \subseteq \psi_1, \ldots, \psi_n$ is that the truth values that occur in a given team for the tuple $\varphi_1, \ldots, \varphi_n$ occur also as truth values for the tuple $\psi_1, \ldots \psi_n$. The formal definition is given below:

$$\begin{aligned}\mathfrak{M}, T \models\ & \varphi_1, \ldots, \varphi_n \subseteq \psi_1, \ldots, \psi_n \\ & \Leftrightarrow \forall w \in T \exists v \in T : \bigwedge_{1 \leq i \leq n} (\mathfrak{M}, \{w\} \models \varphi_i \Leftrightarrow \mathfrak{M}, \{v\} \models \psi_i).\end{aligned}$$

With respect to expressive power the following are known, see e.g., [3,12]:

$$\mathcal{ML} < \mathcal{MDL} < \mathcal{EMDL} = \mathcal{ML}(\varovee) < \mathcal{MTL}$$

$$\mathcal{ML} < \mathcal{MINC} < \mathcal{EMINC} < \mathcal{MTL}.$$

The fact that $\mathcal{MINC} < \mathcal{EMINC}$ holds is known but no published proof is known by the authors. The proof is an easy exercise, see Appendix C.

Proposition 2 (Closure properties). *The logics weaker or equal to $\mathcal{ML}(\varovee)$ with respect to expressive power are downward closed. The logics weaker or equal to $\mathcal{EMINC}$ with respect to expressive power are union closed.*

Note that the $\mathcal{MTL}$ is neither downward nor union closed. The modal depth of φ, denoted by $\mathrm{md}(\varphi)$, is defined in the obvious way (for basic modal logic, see e.g., [2]); intuitionistic disjunction and contradictory negation are handled in the same manner as Boolean connectives. For dependence atoms and inclusion atoms, we define that

$$\begin{aligned}\mathrm{md}(\mathrm{dep}(\varphi_1, \ldots, \varphi_n, \psi)) &= \max\{\mathrm{md}(\varphi_1), \ldots, \mathrm{md}(\varphi_n), \mathrm{md}(\psi)\}, \\ \mathrm{md}(\varphi_1, \ldots, \varphi_n \subseteq \psi_1, \ldots, \psi_n) &= \max\{\mathrm{md}(\varphi_1), \mathrm{md}(\psi_1), \ldots, \mathrm{md}(\varphi_n), \mathrm{md}(\psi_n)\}.\end{aligned}$$

If $\mathcal{L}$ is a logic and $k \in \mathbb{N}$, we write $\mathfrak{M}, T \equiv_k^{\mathcal{L}} \mathfrak{M}', T'$, if $\mathfrak{M}, T$ and $\mathfrak{M}', T'$ agree on all $\mathcal{L}$-formulas φ with $\mathrm{md}(\varphi) \leq k$.

Theorem 5 [17]**.** *Let $\mathcal{L}$ be a team-based logic that is weaker or equal to $\mathcal{MTL}$ with respect to expressive power. Then $\mathfrak{M}, T\ [\rightleftarrows_k]\ \mathfrak{M}', T' \Rightarrow \mathfrak{M}, T \equiv_k^{\mathcal{L}} \mathfrak{M}', T'$.*

5 Modal Definability in Team Semantics

The expressive power of the most studied team-based modal logics is quite well understood. See Table 1 for the known characterisations. However the related topics of definability with respect to models and with respect to frames has received less attention. In [18] a Goldbaltt-Thomason -style characterisation is given for modal dependence logic. Moreover it was shown that with respect to frame definability $\mathcal{MDL}$ and $\mathcal{EMDL}$ coincide. In this section we study definability of $\mathcal{MINC}$ and $\mathcal{MTL}$, see Tables 2 and 3 for a summary of known results together with the results of this sections on definability.

Table 1. Characterisation of expressive powers of different team-based logics. E.g., a class $\mathbb{C}$ of team pointed Kripke models is definable by a single $\mathcal{EMDL}$-formula if and only if $\mathfrak{M}, \emptyset \in \mathbb{C}$, for every $\mathfrak{M}$, $\mathbb{C}$ is closed under the so-called team k-bisimulation, for some finite k, and $\mathbb{C}$ is downward closed.

Logic	Closure properties				References
	empty team property	team k-bisimulation	downward closure	union closure	
$\mathcal{ML}$	×	×	×	×	[13]
$\mathcal{ML}(\ovee)$	×	×	×		[11, Corollary 3.6]
$\mathcal{EMDL}$	×	×	×		[11, Corollary 4.5]
$\mathcal{EMINC}$	×	×		×	[12, Theorem 3.10]
$\mathcal{MTL}$		×			[17, Theorem 3.4]

Table 2. Characterisation of frame definability of different modal logics with respect to first-order definable frame classes. E.g., an elementary class $\mathbb{F}$ of Kripke frames is definable in $\mathcal{EMDL}$ if and only if $\mathbb{F}$ is closed under taking generated subframes and bounded morphic images, and reflects ultrafilter extensions and finitely generated subframes.

<table>
<tr><th rowspan="2">Logic</th><th colspan="3">Closure under</th><th colspan="2">Reflects</th><th rowspan="2">References</th></tr>
<tr><th>disjoint unions</th><th>bounded morphic images</th><th>generated subframes</th><th>ultrafilter extensions</th><th>finitely generated subframes</th></tr>
<tr><td>$\mathcal{ML}$</td><td rowspan="3">×</td><td rowspan="3">×</td><td rowspan="3">×</td><td rowspan="3">×</td><td rowspan="3">× [a]</td><td>[7]</td></tr>
<tr><td>$\mathcal{MINC}$</td><td>Theorem 8</td></tr>
<tr><td>$\mathcal{EMINC}$</td><td>Theorem 8</td></tr>
<tr><td>$\mathcal{ML}(\boxed{u}^+)$</td><td rowspan="5"></td><td rowspan="5">×</td><td rowspan="5">×</td><td rowspan="5">×</td><td rowspan="5">×</td><td>[18, Theorem 3]</td></tr>
<tr><td>$\mathcal{ML}(\ovee)$</td><td>[18, Corollary 1]</td></tr>
<tr><td>$\mathcal{MDL}$</td><td>[18, Corollary 1]</td></tr>
<tr><td>$\mathcal{EMDL}$</td><td>[18, Corollary 1]</td></tr>
<tr><td>$\mathcal{MTL}$</td><td>Theorem 11</td></tr>
<tr><td>$\mathcal{ML}(\boxed{u})$</td><td></td><td>×</td><td></td><td>×</td><td></td><td>[9, Corollary 3.9]</td></tr>
</table>

[a] If a class of frames is closed under disjoint unions and bounded morphic images then it reflects finitely generated subframes.

Table 3. Hierarchy of definability of different modal logics with team semantics.

Model definability	$\{\mathcal{ML}, \mathcal{MINC}, \mathcal{EMINC}\} <_M \mathcal{MDL} <_M \{\mathcal{EMDL}, \mathcal{ML}(\ovee), \mathcal{MTL}\}$
Frame definability	$\{\mathcal{ML}, \mathcal{MINC}, \mathcal{EMINC}\} <_F \{\mathcal{MDL}, \mathcal{EMDL}, \mathcal{ML}(\ovee), \mathcal{MTL}\}$

5.1 Hintikka Formulas and Types

It is well known that for any finite set of proposition symbols Φ, any finite $k \in \mathbb{N}$, and any pointed Φ-model (K, w), there exists a modal formula of modal depth k that characterises (K, w) completely up to k-equivalence (i.e. equivalence up to modal depth k). These *Hintikka formulas* (or *characteristic formulas*) are defined as follows (see e.g. [8]):

Definition 8. *Assume that Φ is a finite set of proposition symbols. Let $k \in \mathbb{N}$ and let $(\mathfrak{M}, w)$ be a pointed Φ-model. The k-th* Hintikka formula $\chi^k_{\mathfrak{M},w}$ *of $(\mathfrak{M}, w)$ is defined recursively as follows:*

– $\chi^0_{\mathfrak{M},w} := \bigwedge\{p \mid p \in \Phi, w \in V(p)\} \wedge \bigwedge\{\neg p \mid p \in \Phi, w \notin V(p)\}$.
– $\chi^{k+1}_{\mathfrak{M},w} := \chi^k_{\mathfrak{M},w} \wedge \bigwedge_{v\in R[w]} \Diamond\chi^k_{\mathfrak{M},v} \wedge \Box \bigvee_{v\in R[w]} \chi^k_{\mathfrak{M},v}$.

It is easy to see that $\mathrm{md}\left(\chi^k_{\mathfrak{M},w}\right) = k$, and $\mathfrak{M}, w \models \chi^k_{\mathfrak{M},w}$ for every pointed Φ-model $(\mathfrak{M}, w)$. By a straightforward inductive argument, it can be shown that, for each finite Φ and k, there are only finitely many non-equivalent k-th Hintikka formulas. Thus $\chi^k_{\mathfrak{M},w}$ is essentially finite (the possibly infinite conjunction $\bigwedge_{v\in R[w]}$ and disjunction $\bigvee_{v\in R[w]}$ can be replaced by finite ones while preserving equivalence).

Proposition 3 (see, e.g., [8]). *Let Φ be a finite set of proposition symbols, $k \in \mathbb{N}$, and $(\mathfrak{M}, w)$ and $(\mathfrak{M}', w')$ pointed Φ-models. Then*

$$\mathfrak{M}, w \equiv^{\mathcal{ML}}_k \mathfrak{M}', w' \quad\iff\quad \mathfrak{M}', w' \models \chi^k_{\mathfrak{M},w}.$$

Definition 9. *Let $\mathfrak{M}$ be a Kripke Φ-model and $\mathbb{C}$ a class of Kripke Φ-models. We define that*

$$\begin{aligned}\mathrm{tp}^{\Phi}_k(\mathfrak{M}) &:= \{\chi^k_{\mathfrak{M},w} \mid w \text{ is a point of } \mathfrak{M}\},\\ \mathrm{tp}^{\Phi}_k(\mathfrak{M}, T) &:= \{\chi^k_{\mathfrak{M},w} \mid w \in T\},\\ \mathrm{tp}^{\Phi}_k(\mathbb{C}) &:= \{\mathrm{tp}^{\Phi}_k(\mathfrak{M}) \mid \mathfrak{M} \in \mathbb{C}\}.\end{aligned}$$

Proposition 4. *Let $\mathcal{L}$ be any team-based logic weaker than or equal to $\mathcal{MTL}$ w.r.t. expressive power. Then $\mathrm{tp}^{\Phi}_k(\mathfrak{M}, T) = \mathrm{tp}^{\Phi}_k(\mathfrak{M}', T') \Rightarrow \mathfrak{M}, T \equiv^{\mathcal{L}}_k \mathfrak{M}', T'$.*

Proof. Assume that $\mathrm{tp}^{\Phi}_k(\mathfrak{M}, T) = \mathrm{tp}^{\Phi}_k(\mathfrak{M}', T')$. By Proposition 3 and the definition of team bisimulation, it follows that $\mathfrak{M}, T\ [\rightleftarrows_k]\ \mathfrak{M}', T'$. The claim now follows by Theorem 5. ☐

5.2 Global Modal & Frame Definability in $\mathcal{MINC}$ and $\mathcal{ML}$ Coincide

Lemma 2. *Let Φ be a finite set of proposition symbols, $\varphi \in \mathcal{EMINC}(\Phi)$, and $k = \mathrm{md}(\varphi)$. Then $\mathfrak{M} \in \mathrm{Mod}(\varphi)$ iff $\mathrm{tp}^{\Phi}_k(\mathfrak{M}) \subseteq \bigcup\{\mathrm{tp}^{\Phi}_k(\mathfrak{M}') \mid \mathfrak{M}' \in \mathrm{Mod}(\varphi)\}$.*

Proof. The direction from left to right is trivial. Assume then that

$$\mathrm{tp}^{\Phi}_k(\mathfrak{M}) \subseteq \bigcup\{\mathrm{tp}^{\Phi}_k(\mathfrak{M}') \mid \mathfrak{M}' \in \mathrm{Mod}(\varphi)\} \tag{1}$$

holds, and let T be an arbitrary team of $\mathfrak{M}$. It suffices to establish that $\mathfrak{M}, T \models \varphi$. From (1) it follows that there exists some $n \in \mathbb{N}$, models $\mathfrak{M}_i \in \mathrm{Mod}(\varphi)$, teams S_i of $\mathfrak{M}_i$ and T_i of $\mathfrak{M}$, $i \leq n$, such that

$$T_1 \cup \cdots \cup T_n = T \text{ and } \mathrm{tp}^{\Phi}_k(\mathfrak{M}_i, S_i) = \mathrm{tp}^{\Phi}_k(\mathfrak{M}, T_i), \quad \text{for each } i \leq n.$$

Note that such finite n exists, since $\mathrm{tp}_k^{\Phi}(\mathfrak{M})$ is essentially finite. Since each $\mathfrak{M}_i \in \mathrm{Mod}(\varphi)$, it follows that $\mathfrak{M}_i, S_i \models \varphi$, for $i \leq n$. Thus from Proposition 4 and the fact that $\mathrm{tp}_k^{\Phi}(\mathfrak{M}_i, S_i) = \mathrm{tp}_k^{\Phi}(\mathfrak{M}, T_i)$, for $i \leq n$, it follows that $\mathfrak{M}, T_i \models \varphi$, for $i \leq n$. Now, by union closure (Proposition 2), we conclude that $\mathfrak{M}, T \models \varphi$. □

Theorem 6. *A class $\mathbb{C}$ of Kripke models is definable by a single $\mathcal{EMINC}$-formula if and only if the class if definable by a single $\mathcal{ML}$-formula.*

Proof. The if direction is trivial. For the other direction, let $\mathbb{C}$ be a class of Kripke models that is definable by a single $\mathcal{EMINC}$ formula and let φ be an $\mathcal{EMINC}(\Phi)$-formula that defines $\mathbb{C}$. Without lose of generality, we may assume that Φ is finite. Let k denote the modal depth of φ. We will show that the $\mathcal{ML}(\Phi)$ formula

$$\varphi^* := \bigvee \{\chi^k_{\mathfrak{M},w} \mid \mathfrak{M} \in \mathrm{Mod}(\varphi), w \in \mathfrak{M}\}$$

defines $\mathbb{C}$. Since over a finite set of proposition symbols there exists only finitely many essentially different k-Hintikka-formulas, φ^* is essentially a finite $\mathcal{ML}(\Phi)$ -formula. By assumption $\mathbb{C} = \mathrm{Mod}(\varphi)$. Thus by Lemma 2

$$\mathfrak{M} \in \mathbb{C} \text{ iff } \mathrm{tp}_k^{\Phi}(\mathfrak{M}) \subseteq \bigcup \{\mathrm{tp}_k^{\Phi}(\mathfrak{M}') \mid \mathfrak{M}' \in \mathrm{Mod}(\varphi)\}. \tag{2}$$

Observe that by flatness (Proposition 1)

$$\mathfrak{M}, T \models \varphi^* \text{ iff } \mathrm{tp}_k^{\Phi}(\mathfrak{M}, T) \subseteq \bigcup \{\mathrm{tp}_k^{\Phi}(\mathfrak{M}') \mid \mathfrak{M}' \in \mathrm{Mod}(\varphi)\},$$

and thus it follows that

$$\mathfrak{M} \models \varphi^* \text{ iff } \mathrm{tp}_k^{\Phi}(\mathfrak{M}) \subseteq \bigcup \{\mathrm{tp}_k^{\Phi}(\mathfrak{M}') \mid \mathfrak{M}' \in \mathrm{Mod}(\varphi)\}. \tag{3}$$

From (2) and (3) the claim follows. □

The following theorems directly follow.

Theorem 7. *$\mathcal{EMINC} =_M \mathcal{ML}$.*

Theorem 8. *$\mathcal{EMINC} =_F \mathcal{ML}$.*

Proof. Clearly any $\mathcal{ML}$-definable class of Kripke frames is also definable in $\mathcal{EMINC}$. The converse follows directly from Theorem 6.

Let $\mathfrak{F}$ be a Kripke frame, φ an $\mathcal{EMINC}$-formula and φ^* the related $\mathcal{ML}$-formula given by Theorem 6 such that $\mathrm{Mod}(\varphi) = \mathrm{Mod}(\varphi^*)$. Now, by definition, $\mathfrak{F} \models \varphi$ if and only if $(\mathfrak{F}, V) \models \varphi$ for every valuation V. Since $\mathrm{Mod}(\varphi) = \mathrm{Mod}(\varphi^*)$, this holds if and only if $(\mathfrak{F}, V) \models \varphi^*$ for every valuation V, which by definition holds if and only if $\mathfrak{F} \models \varphi^*$. Now let $\mathbb{F}$ be some $\mathcal{EMINC}$-definable class of Kripke frames and let Γ be a set of $\mathcal{EMINC}$-formulas that defines $\mathbb{F}$. Define $\Gamma^* := \{\varphi^* \mid \varphi \in \Gamma\}$. Clearly Γ^* is a set of $\mathcal{ML}$-formulas that defines $\mathbb{F}$. □

5.3 Global Modal & Frame Definability in $\mathcal{MTL}$ & $\mathcal{ML}(\varovee)$ Coincide

Lemma 3. *Let φ be and $\mathcal{MTL}$-formula and $k = \mathrm{md}(\varphi)$. Then*

$$\mathfrak{M} \in \mathrm{Mod}(\varphi) \text{ iff } \mathrm{tp}_k^{\Phi}(\mathfrak{M}) \subseteq \Gamma \in \mathrm{tp}_k^{\Phi}\big(\mathrm{Mod}(\varphi)\big), \quad \text{for some } \Gamma.$$

Proof. The direction from left to right is trivial. Assume then that $\mathrm{tp}_k^{\Phi}(\mathfrak{M}) \subseteq \Gamma \in \mathrm{tp}_k^{\Phi}\big(\mathrm{Mod}(\varphi)\big)$ holds for some Γ. Thus there exists a Kripke model $\mathfrak{M}'$ such that $\mathfrak{M}' \in \mathrm{Mod}(\varphi)$ and $\mathrm{tp}_k^{\Phi}(\mathfrak{M}') = \Gamma$. For the sake of a contradiction, assume that $\mathfrak{M} \notin \mathrm{Mod}(\varphi)$. Thus there exists a team T of $\mathfrak{M}$ such that $\mathfrak{M}, T \not\models \varphi$. Since $\mathrm{tp}_k^{\Phi}(\mathfrak{M}) \subseteq \mathrm{tp}_k^{\Phi}(\mathfrak{M}')$ it follows that there exists a team T' of $\mathfrak{M}'$ such that $\mathrm{tp}_k^{\Phi}(\mathfrak{M}, T) = \mathrm{tp}_k^{\Phi}(\mathfrak{M}', T')$. Thus by Proposition 4, we conclude that $\mathfrak{M}', T' \not\models \varphi$. This is a contradiction and thus $\mathfrak{M} \in \mathrm{Mod}(\varphi)$ holds. □

Theorem 9. *A class $\mathbb{C}$ of Kripke models is definable in $\mathcal{MTL}$ by a single formula if and only if it is definable in $\mathcal{ML}(\varovee)$ by a single formula.*

Proof. The fact that every class of Kripke models that is definable by a single $\mathcal{ML}(\varovee)$-formula is also definable by a single $\mathcal{MTL}$-formula follows directly by Theorem 4.

Let $\mathbb{C}$ be an arbitrary single formula $\mathcal{MTL}$-definable class of Kripke models and let φ be an $\mathcal{MTL}$-formula that defines $\mathbb{C}$. Let k denote the modal depth of φ. We will show that the $\mathcal{ML}(\varovee)$-formula

$$\varphi^* := \bigovee_{\Gamma \in \mathrm{tp}_k^{\Phi}(\mathbb{C})} \big(\bigvee \Gamma\big)$$

defines $\mathbb{C}$. Note that since $\mathrm{tp}_k^{\Phi}(\mathbb{C})$ is a family of sets of k-Hintikka formulas the outer disjunction is essentially finite. Likewise, since each Γ is a collection of k-Hintikka formulas, it follows by flatness (remember that Hintikka formulas are $\mathcal{ML}$-formulas) that the inner disjunctions are essentially finite. Thus φ^* is essentially a finite $\mathcal{ML}(\varovee)$-formula.

Assume first that $\mathfrak{M} \in \mathbb{C}$. By definition $\mathrm{tp}_k^{\Phi}(\mathfrak{M}) \in \mathrm{tp}_k^{\Phi}(\mathbb{C})$. Clearly, for each team T of $\mathfrak{M}$, it holds that $\mathfrak{M}, T \models \bigvee \mathrm{tp}_k^{\Phi}(\mathfrak{M})$, and thus that $\mathfrak{M}, T \models \varphi^*$. Therefore $\mathfrak{M} \models \varphi^*$. Assume then that $\mathfrak{M} \models \varphi^*$. Thus $\mathfrak{M}, W \models \varphi^*$, where W is the domain of $\mathfrak{M}$. Therefore there exists a set $\Gamma \in \mathrm{tp}_k^{\Phi}(\mathbb{C})$ such that $\mathfrak{M}, W \models \bigvee \Gamma$. Thus $\mathrm{tp}_k^{\Phi}(\mathfrak{M}) = \mathrm{tp}_k^{\Phi}(\mathfrak{M}, W) \subseteq \Gamma$. Recall that $\mathbb{C} = \mathrm{Mod}(\varphi)$. Now since $\Gamma \in \mathrm{tp}_k^{\Phi}(\mathbb{C}) = \mathrm{tp}_k^{\Phi}\big(\mathrm{Mod}(\varphi)\big)$, it follows from Lemma 3 that $\mathfrak{M} \in \mathrm{Mod}(\varphi) = \mathbb{C}$. □

The following theorems directly follow.

Theorem 10. $\mathcal{MTL} =_M \mathcal{ML}(\varovee)$.

Theorem 11. $\mathcal{MTL} =_F \mathcal{ML}(\varovee)$.

Proof. The fact that every $\mathcal{ML}(\varovee)$ definable class of Kripke frames is also definable in $\mathcal{MTL}$ follows directly by Theorem 4.

Let $\mathbb{F}$ be an arbitrary $\mathcal{MTL}$-definable class of Kripke frames and let Γ a set of $\mathcal{MTL}$-formulas that defines $\mathbb{F}$. By Theorem 9, for each $\varphi \in \mathcal{MTL}$ there exists a formula $\varphi^* \in \mathcal{ML}(\otimes)$ such that $\mathrm{Mod}(\varphi^*) = \mathrm{Mod}(\varphi)$. Recall that φ defines the class $\mathrm{Mod}(\varphi)$ of Kripke models. Now clearly $\mathfrak{F} \models \varphi$ iff $(\mathfrak{F}, V) \in \mathrm{Mod}(\varphi)$ for every valuation V iff $(\mathfrak{F}, V) \in \mathrm{Mod}(\varphi^*)$ for every valuation V iff $\mathfrak{F} \models \varphi^*$. Define $\Gamma := \{\varphi^* \mid \varphi \in \Gamma\}$. Clearly for each frame $\mathfrak{F}$ it holds that $\mathfrak{F} \models \Gamma$ iff $\mathfrak{F} \models \Gamma^*$. □

It was established in [18] that $\mathcal{ML} <_F \mathcal{MDL} =_F \mathcal{EMDL} =_F \mathcal{ML}(\otimes)$. When combined with Theorems 8 and 11 the following hierarchy is obtained.

Theorem 12. $\{\mathcal{ML}, \mathcal{MINC}, \mathcal{EMINC}\} <_F \{\mathcal{MDL}, \mathcal{EMDL}, \mathcal{ML}(\otimes), \mathcal{MTL}\}$.

It is an easy exercise to show that $\mathcal{ML} <_M \mathcal{MDL}$ and $\mathcal{MDL} <_M \mathcal{EMDL}$, see Appendix C. Moreover it follows from the work of Hella et al. [11] that $\mathcal{EMDL} = \mathcal{ML}(\otimes)$. Thus by Theorems 7 and 10 we obtain the following trichotomy.

Theorem 13

$$\{\mathcal{ML}, \mathcal{MINC}, \mathcal{EMINC}\} <_M \mathcal{MDL} <_M \{\mathcal{EMDL}, \mathcal{ML}(\otimes), \mathcal{MTL}\}.$$

6 Conclusion

In this paper, we studied relative frame definability of a fragment of modal logic with universal modality in which the universal modality occurs only positively. Moreover we studied definability of particular modal logics with team semantics. We showed that a class $\mathbb{F}$ of finite transitive frames is definable in $\mathcal{ML}(\boxed{\mathsf{u}}^+)$ relative to finite transitive frames if and only if $\mathbb{F}$ is closed under taking generated subframes and bounded morphic images. In addition, we established the following trichotomy with respect to model definability

$$\{\mathcal{ML}, \mathcal{MINC}, \mathcal{EMINC}\} <_M \mathcal{MDL} <_M \{\mathcal{EMDL}, \mathcal{ML}(\otimes), \mathcal{MTL}\}$$

and the following dichotomy with respect to frame definability

$$\{\mathcal{ML}, \mathcal{MINC}, \mathcal{EMINC}\} <_F \{\mathcal{MDL}, \mathcal{EMDL}, \mathcal{ML}(\otimes), \mathcal{MTL}\}.$$

Since it is known that $\mathcal{MDL} =_F \mathcal{ML}(\boxed{\mathsf{u}}^+)$, we obtained as a corollary relative Goldblatt–Thomason -style theorems for each of the logics listed above.

Note that our results imply that the model (frame) definability of every logic between $\mathcal{EMDL}$ ($\mathcal{MDL}$) and $\mathcal{MTL}$ coincides. In particular, we obtain results concerning modal independence logic $\mathcal{MIL}$ and extended modal independence logic $\mathcal{EMIL}$, since with respect to expressive power $\mathcal{MDL} \leq \mathcal{MIL} \leq \mathcal{MTL}$ and $\mathcal{EMDL} \leq \mathcal{EMIL} \leq \mathcal{MTL}$.

We conclude with some open questions:

1. Where does $\mathcal{MIL}$ lie with respect to modal definability?
2. Is there some natural fragment of $\mathcal{ML}(\boxdot^+)$ that coincided with $\mathcal{MDL}$ or $\mathcal{MIL}$ with respect to model definability?
3. Can we use the notion of local bounded morphism (cf. [1]) to drop the requirement of transitivity from Theorem 3.
4. Can we characterize model definability of team-based logics in terms of semantic constructions?

A Proof of Lemma 1

Lemma 1. *Let $\mathfrak{F} = (W, R)$ be a finite transitive frame. For any transitive frame $\mathfrak{G}$, the following are equivalent:*

(i) *the Jankov-Fine formula $\varphi_{\mathfrak{F}}$ is not valid in $\mathfrak{G}$,*
(ii) *there is a finite set $Y \subseteq |\mathfrak{G}|$ such that $\mathfrak{F}$ is a bounded morphic image of $\mathfrak{G}_Y$, where $\mathfrak{G}_Y$ is the subframe of $\mathfrak{G}$ generated by Y.*

Proof. The direction from (ii) to (i) is immediate from the fact that $\varphi_{\mathfrak{F}}$ is not valid in $\mathfrak{F}$ under the natural valuation sending p_{w_i} to $\{ w_i \}$ (Note: validity of $\mathcal{ML}(\boxdot^+)$-formulas is closed under taking under bounded morphic images and generated subframes, see [18]). So, we focus on the converse direction.

Assume (i). It follows from $\mathfrak{G} \nVdash \varphi_{\mathfrak{F}}$ that $(\mathfrak{G}, V) \nVdash \varphi_{\mathfrak{F}}$, for some assignment V. Thus, for each $i \leq n$, there exists a point v_i of $\mathfrak{G}$ such that $(\mathfrak{G}, V), v_i \Vdash \varphi_{\mathfrak{F},w_i}$. Put $Y := \{ v_i \mid 0 \leq i \leq n \}$, let $\mathfrak{G}_Y$ denote the subframe of $\mathfrak{G}$ generated by Y, and let U be the reduction of V into the frame $\mathfrak{G}_Y$. Since satisfaction of $\mathcal{ML}$-formulas is closed under taking generated submodels (see, e.g., [2, Proposition 2.6]), it follows that $(\mathfrak{G}_Y, U), v_i \Vdash \varphi_{\mathfrak{F},w_i}$, for each $i \leq n$. Let us put $\mathfrak{G}_Y = (G_Y, S)$. The first clause of the Jankov-Fine formula $\varphi_{\mathfrak{F},w_i}$ implies that, for each $i \leq n$, $U(p_{w_i}) \neq \emptyset$. By the second and the third clause, we obtain $\bigcup_{w \in W} U(p_w) = G_Y$ and $U(p_{w_i}) \cap U(p_{w_j}) = \emptyset$ for any distinct indices i and j. This enables us to define a surjective mapping $f : G_Y \to W$. Define $f(v) := w_i$ if $v \in U(p_{w_i})$. Clearly f is a well defined surjection.

In what follows, we show that f is a bounded morphism. The condition (Forth) is established as follows. Assume that xSy and let i, j be such that $f(x) = w_i$ and $f(y) = w_j$. Thus $x \in U(p_{w_i})$ and $y \in U(p_{w_j})$. Since $\mathfrak{G}_Y$ is Y-generated, x is reachable from some $v_k \in Y$. Suppose for a contradiction that $w_i R w_j$ fails in $\mathfrak{F}$. Then $\Box^+(p_{w_i} \to \neg\Diamond p_{w_j})$ is a conjunct in the Jankov-Fine formula $\varphi_{\mathfrak{F},w_k}$. Recall that $(\mathfrak{G}_Y, U), v_k \Vdash \varphi_{\mathfrak{F},w_k}$. It now follows from $(\mathfrak{G}_Y, U), v_k \Vdash \Box^+(p_{w_i} \to \neg\Diamond p_{w_j})$ that xSy fails. A contradiction. Therefore, $w_i R w_j$ holds in $\mathfrak{F}$.
The condition (Back) is shown as follows. Assume that $f(x) R w_j$ and let i be such that $f(x) = w_i$. From the definition of f, it follows that $x \in U(p_{w_i})$. Since $\mathfrak{G}_Y$ is Y-generated, x is reachable from some $v_k \in Y$. Since $w_i R w_j$, we have that $\Box^+(p_{w_i} \to \Diamond p_{w_j})$ is a conjunct in the Jankov-Fine formula $\varphi_{\mathfrak{F},w_k}$. Recall again that $(\mathfrak{G}_Y, U), v_k \Vdash \varphi_{\mathfrak{F},w_k}$. It follows from $(\mathfrak{G}_Y, U), v_k \Vdash \Box^+(p_{w_i} \to \Diamond p_{w_j})$ and $x \in U(p_{w_i})$ that there is some y such that $f(y) = w_j$ and xSy holds, as desired. $\square$

B Frame Constructions

Definition 10 (Disjoint Unions). *Let* $\{\,\mathfrak{F}_i \mid i \in I\,\}$ *be a pairwise disjoint family of frames, where* $\mathfrak{F}_i = (W_i, R_i)$*. The* disjoint union $\biguplus_{i\in I}\mathfrak{F}_i = (W, R)$ *of* $\{\,\mathfrak{F}_i \mid i \in I\,\}$ *is defined by* $W = \bigcup_{i\in I} W_i$ *and* $R = \bigcup_{i\in I} R_i$*.*

Definition 11 (Generated Subframes). *Given any two frames* $\mathfrak{F} = (W, R)$ *and* $\mathfrak{F} = (W', R')$*,* $\mathfrak{F}'$ *is a* generated subframe *of* $\mathfrak{F}$ *if* (i) $W' \subseteq W$*,* (ii) $R' = R \cap (W')^2$*,* (iii) $w'Rv'$ *implies* $v' \in W'$*, for every* $w' \in W'$*. We say that* $\mathfrak{F}'$ *is the* generated subframe *of* $\mathfrak{F}$ *by* $X \subseteq |\mathfrak{F}|$ *(notation:* $\mathfrak{F}_X$*) if* $\mathfrak{F}'$ *is the smallest generated subframe of* $\mathfrak{F}$ *whose domain contains* X*.* $\mathfrak{F}'$ *is a* finitely generated subframe *of* $\mathfrak{F}$ *if there is a finite set* $X \subseteq |\mathfrak{F}|$ *such that* $\mathfrak{F}'$ *is* $\mathfrak{F}_X$*.*

A frame class $\mathbb{F}$ *reflects* finitely generated subframes whenever it is the case for all frames $\mathfrak{F}$ that, if every finitely generated subframe of $\mathfrak{F}$ is in $\mathbb{F}$, then $\mathfrak{F} \in \mathbb{F}$.

Definition 12 (Bounded Morphism). *Given any two frames* $\mathfrak{F} = (W, R)$ *and* $\mathfrak{F}' = (W', R')$*, a function* $f : W \to W'$ *is a* bounded morphism *if it satisfies the following two conditions:*

(**Forth**) *If* wRv*, then* $f(w)R'f(v)$*.*
(**Back**) *If* $f(w)R'v'$*, then* wRv *and* $f(v) = v'$ *for some* $v \in W$*.*

If f *is surjective, we say that* $\mathfrak{F}'$ *is a* bounded morphic image *of* $\mathfrak{F}$*.*

Definition 13 (Ultrafilter Extensions). *Let* $\mathfrak{F} = (W, R)$ *be a Kripke frame, and* $\mathrm{Uf}(W)$ *denote the set of all ultrafilters on* W*. Define the binary relation* $R^{\mathfrak{ue}}$ *on the set* $\mathrm{Uf}(W)$ *as follows:* $\mathcal{U}R^{\mathfrak{ue}}\mathcal{U}'$ *iff* $X \in \mathcal{U}'$ *implies* $m_R(X) \in \mathcal{U}$*, for every* $X \subseteq W$*, where* $m_R(X) := \{\, w \in W \mid wRw' for some w' \in X\,\}$*. The frame* $\mathfrak{ue}\mathfrak{F} = (\mathrm{Uf}(W), R^{\mathfrak{ue}})$ *is called the* ultrafilter extension *of* $\mathfrak{F}$*.*

A frame class $\mathbb{F}$ *reflects* ultrafilter extensions if $\mathfrak{ue}\mathfrak{F} \in \mathbb{F}$ implies $\mathfrak{F} \in \mathbb{F}$ for every frame $\mathfrak{F}$.

C Separations in Definability

Proposition 5. *With respect to expressive power* $\mathcal{MINC} < \mathcal{EMINC}$*.*

Proof. For $\varphi \in \mathcal{MINC}(\{p\})$, let φ^* denote the $\mathcal{ML}(\{p\})$-formula obtained from φ by substituting each inclusion atom in φ by the formula $(p \vee \neg p)$. Since $p \subseteq p$ is essentially the only inclusion atom in $\mathcal{MINC}(\{p\})$, it is easy to see that, for every $\varphi \in \mathcal{MINC}(\{p\})$, φ and φ^* are equivalent.

Let $\mathfrak{M} = (W, R, V)$ be a Kripke $\{p\}$-model such that $W = \{1, 2, 3\}$, R={(1,2)}, and $V(p) = \{1, 2, 3\}$. We claim that there does not exists a $\mathcal{MINC}$-formula that is equivalent with $p \subseteq \Diamond p$. For the sake of a contradiction, assume that $\psi \in \mathcal{MINC}$ is such a formula. Clearly $\mathfrak{M}, \{1, 3\} \models p \subseteq \Diamond p$ and thus, by assumption, $\mathfrak{M}, \{1, 3\} \models \psi$. By our observation above, $\mathfrak{M}, \{1, 3\} \models \psi^*$ follows. Now since ψ^* is an $\mathcal{ML}$-formula, it follows by Proposition 1 that $\mathfrak{M}, \{3\} \models \psi^*$. Thus $\mathfrak{M}, \{3\} \models \psi$ and therefore $\mathfrak{M}, \{3\} \models p \subseteq \Diamond p$. However clearly $\mathfrak{M}, \{3\} \not\models p \subseteq \Diamond p$, a contradiction.

Proposition 6. *$\mathcal{ML} <_M \mathcal{MDL}$.*

Proof. Let $\mathfrak{M}_i = (W_i, R_i, V_i)$, $i \leq 2$, be Φ-models such that $W_0 = \{1, 2\}$, $W_1 = \{1\}$, $W_2 = \{2\}$, $R_0 = R_1 = R_2 = \emptyset$, and, for each $p \in \Phi$, $V_0(p) = V_1(p) = \{1\}$, and $V_2(p) = \emptyset$. It is easy to conclude by flatness of $\mathcal{ML}$ that

$$\mathfrak{M}_0 \in \mathrm{Mod}(\varphi) \text{ iff } \mathfrak{M}_1, \mathfrak{M}_2 \in \mathrm{Mod}(\varphi)$$

holds for every $\varphi \in \mathcal{ML}$. Thus

$$\mathfrak{M}_0 \in \mathrm{Mod}(\Gamma) \text{ iff } \mathfrak{M}_1, \mathfrak{M}_2 \in \mathrm{Mod}(\Gamma)$$

holds for every $\Gamma \subseteq \mathcal{ML}$. However $\mathfrak{M}_1, \mathfrak{M}_2 \in \mathrm{Mod}(\mathrm{dep}(p))$ but $\mathfrak{M}_0 \notin \mathrm{Mod}(\mathrm{dep}(p))$. Thus we conclude that $\mathrm{Mod}(\mathrm{dep}(p))$ is not definable in $\mathcal{ML}$.

Proposition 7. *$\mathcal{MDL} <_M \mathcal{EMDL}$.*

Proof. Let $\mathfrak{M}_i = (W_i, R_i, V_i)$, $i \leq 2$, be Φ-models such that $W_0 = \{1, 2\}$, $W_1 = \{1\}$, $W_2 = \{2\}$, $R_0 = \{(1, 1)\}$, $R_1 = \{(1, 1)\}$, $R_2 = \emptyset$, and, for each $p \in \Phi$, $V_0(p) = \{1, 2\}$, $V_1(p) = \{1\}$, and $V_2(p) = \{2\}$. It is easy to conclude (see [4, Theorem 1] for details) that

$$\mathfrak{M}_0 \in \mathrm{Mod}(\varphi) \text{ iff } \mathfrak{M}_1, \mathfrak{M}_2 \in \mathrm{Mod}(\varphi)$$

holds for every $\varphi \in \mathcal{MDL}$. Thus

$$\mathfrak{M}_0 \in \mathrm{Mod}(\Gamma) \text{ iff } \mathfrak{M}_1, \mathfrak{M}_2 \in \mathrm{Mod}(\Gamma)$$

holds for every $\Gamma \subseteq \mathcal{MDL}$. However $\mathfrak{M}_1, \mathfrak{M}_2 \in \mathrm{Mod}(\mathrm{dep}(\Diamond p))$ but $\mathfrak{M}_0 \notin \mathrm{Mod}(\mathrm{dep}(\Diamond p))$. Thus we conclude that $\mathrm{Mod}(\mathrm{dep}(\Diamond p))$ is not definable in $\mathcal{MDL}$.

References

1. van Benthem, J.: Notes on modal definability. Notre Dame J. Formal Log. **30**(1), 20–35 (1988)
2. Blackburn, P., de Rijke, M., Venema, Y.: Modal Logic. Cambridge University Press, New York (2001)
3. Durand, A., Kontinen, J., Vollmer, H.: Expressivity and Complexity of Dependence Logic. Springer (2016) (In Press)
4. Ebbing, J., Hella, L., Meier, A., Müller, J.-S., Virtema, J., Vollmer, H.: Extended modal dependence logic $\mathcal{EMDL}$. In: Libkin, L., Kohlenbach, U., de Queiroz, R. (eds.) WoLLIC 2013. LNCS, vol. 8071, pp. 126–137. Springer, Heidelberg (2013)
5. Galliani, P.: Inclusion and exclusion dependencies in team semantics - on some logics of imperfect information. Ann. Pure Appl. Log. **163**(1), 68–84 (2012)
6. Gargov, G., Goranko, V.: Modal logic with names. J. Philos. Log. **22**, 607–636 (1993)
7. Goldblatt, R.I., Thomason, S.K.: Axiomatic classes in propositional modal logic. In: Crossley, J.N. (ed.) Algebra and Logic, pp. 163–73. Springer, Heidelberg (1975)

8. Goranko, V., Otto, M.: Model theory of modal logic. In: Blackburn, P., Van Benthem, J., Wolter, F. (eds.) Handbook of Modal Logic. Studies in Logic and Practical Reasoning, vol. 3, pp. 249–329. Elsevier, Amsterdam (2007)
9. Goranko, V., Passy, S.: Using the universal modality: gains and questions. J. Log. Comput. **2**(1), 5–30 (1992)
10. Grädel, E., Väänänen, J.: Dependence and independence. Stud. Logica **101**(2), 399–410 (2013)
11. Hella, L., Luosto, K., Sano, K., Virtema, J.: The expressive power of modal dependence logic. In: AiML 2014 (2014)
12. Hella, L., Stumpf, J.: The expressive power of modal logic with inclusion atoms. In: GandALF 2015 (2015)
13. Hennessy, M., Milner, R.: Algebraic laws for nondeterminism and concurrency. J. ACM **32**(1), 137–161 (1985)
14. Hintikka, J., Sandu, G.: Informational independence as a semantical phenomenon. In: Fenstad, J.E., et al. (eds.) Logic, methodology and philosophy of science, VIII (Moscow, 1987). Studies in Logic and the Foundations of Mathematics, vol. 126, pp. 571–589. North-Holland, Amsterdam (1989)
15. Hodges, W.: Some strange quantifiers. In: Mycielski, J., Rozenberg, G., Salomaa, A. (eds.) Structures in Logic and Computer Science. LNCS, vol. 1261, pp. 51–65. Springer, Heidelberg (1997)
16. Kontinen, J., Müller, J.-S., Schnoor, H., Vollmer, H.: Modal independence logic. In: AiML 2014 (2014)
17. Kontinen, J., Müller, J.-S., Schnoor, H., Vollmer, H.: A van Benthem theorem for modal team semantics. In: 24th EACSL Annual Conference on Computer Science Logic (2015)
18. Sano, K., Virtema, J.: Characterizing frame definability in team semantics via the universal modality. In: de Paiva, V., de Queiroz, R., Moss, L.S., Leivant, D., de Oliveira, A. (eds.) WoLLIC 2015. LNCS, vol. 9160, pp. 140–155. Springer, Heidelberg (2015)
19. Väänänen, J.: Dependence Logic - A New Approach to Independence Friendly Logic. London Mathematical Society student texts, vol. 70. Cambridge University Press, New York (2007)
20. Väänänen, J.: Modal dependence logic. In: Apt, K.R., van Rooij, R. (eds.) New Perspectives on Games and Interaction. Texts in Logic and Games, vol. 4, pp. 237–254. Amsterdam University Press, Amsterdam (2008)

Negation and Partial Axiomatizations of Dependence and Independence Logic Revisited

Fan Yang(✉)

Delft University of Technology, Delft, The Netherlands
fan.yang.c@gmail.com

Abstract. In this paper, we axiomatize the negatable consequences in dependence and independence logic by extending the natural deduction systems of the logics given in [10,20]. We give a characterization for negatable formulas in independence logic and negatable sentences in dependence logic, and identify an interesting class of formulas that are negatable in independence logic. Dependence and independence atoms, first-order formulas belong to this class.

1 Introduction

Negation and partial axiomatizations of dependence and independence logic have been studied in the literature. In this paper, we take a new look at these topics.

Dependence logic was introduced by Väänänen [23] as a development of *Henkin quantifier* [11] and *independence-friendly logic* [12]. Recently, Grädel and Väänänen [9] defined a variant of dependence logic, called *independence logic*. The two logics add to first-order logic new types of atomic formulas $=\!\!(\vec{x}, y)$ and $\vec{x} \perp_{\vec{z}} \vec{y}$, called *dependence atom* and *independence atom*, to explicitly specify the dependence and independence relations between variables. Intuitively, $=\!\!(\vec{x}, y)$ states that "the value of y is completely determined by the values of the variables in the tuple $\vec{x}$", and $\vec{x} \perp_{\vec{z}} \vec{y}$ states that "given the values of the variables $\vec{z}$, the values of $\vec{x}$ and the values of $\vec{y}$ are completely independent of each other". These properties cannot be meaningfully manifested in *single* assignments of the variables. Therefore unlike in the case of the usual Tarskian semantics, formulas of dependence and independence logic are evaluated on *sets* of assignments (called *teams*) instead. This semantics is called *team semantics* and was introduced by Hodges [13,14].

Dependence and independence logic are known to have the same expressive power as existential second-order logic Σ^1_1 (see [5,18]). This fact has two negative consequences: The logics are not closed under classical negation and are not axiomatizable. The aim of this paper is to shed some new light on these problems.

Regarding the first problem, "negation", which is usually a desirable connective for a logic, turns out to be a tricky connective in the context of team semantics. The negation that dependence and independence logic inherit from first-order logic (denoted by $\neg$) is a type of "syntactic negation", in the

J. Väänänen et al. (Eds.): WoLLIC 2016, LNCS 9803, pp. 410–431, 2016.
DOI: 10.1007/978-3-662-52921-8_25

sense that in order to compute the meaning of the formula $\neg\phi$, the negation $\neg$ has to be brought to the very front of atomic formulas by applying De Morgen's laws and the double negation law. It was proved that this negation $\neg$ is actually not a semantic operator [19], meaning that ϕ and ψ are semantically equivalent does not necessarily imply that $\neg\phi$ and $\neg\psi$ are semantically equivalent. The *classical (contradictory) negation* (denoted by $\sim$ in the literature), on the other hand, is a semantic operator. Since the Σ_1^1 fragment of second-order logic is not closed under classical negation, neither dependence nor independence logic is closed under classical negation. Dependence logic extended with the classical negation $\sim$ is called *team logic* in the literature, and it has the same expressive power as full second-order logic (see [17,23]).

Since every formula of dependence and independence logic is satisfied on the empty team, the classical contradictory negation $\sim\phi$ of any formula will not be satisfied on the empty team, implying that $\sim\phi$ cannot possibly be definable in dependence or independence logic for any single formula ϕ. This technical subtlety makes the classical contradictory negation $\sim$ less interesting. In this paper, we will, instead, consider the *weak classical negation*, denoted by $\dot\sim$, which behaves exactly as the classical negation except that on the empty team $\dot\sim\phi$ is always satisfied. We will give a characterization for negatable formulas in independence logic and negatable sentences in dependence logic by generalizing an argument in [23]. We also identify an interesting class of formulas that are negatable in independence logic. First-order formulas, dependence and independence atoms belong to this class. Formulas of this class are closely related to the *dependency notions* considered in [6] and the *generalized dependence atoms* studied in [16,21].

As for the axiomatization problem, since Σ_1^1 is not axiomatizable, dependence and independence logic cannot possibly be axiomatized in full. Nevertheless, [10,20] defined natural deduction systems for the logics such that the equivalence

$$\Gamma \models \phi \iff \Gamma \vdash \phi \tag{1}$$

holds if Γ is a set of sentences of dependence or independence logic and ϕ is a first-order sentence. It was left open whether these partial axiomatizations can be generalized such that the above equivalence (1) holds if Γ is a set of formulas (that possibly contain free variables) and ϕ is a (possibly open) first-order formula. Kontinen [15] gave such a generalization by expanding the signature with an extra relation symbol so as to interpret the teams associated with the free variables. In this paper, we will generalize the partial axiomatization results in [10,20] via a different approach, an approach that makes use of the weak classical negation. We will define extensions of the systems given in [10,20] such that the equivalence (1) holds if Γ is a set of formulas and ϕ is a formula that is negatable in the logics.

2 Preliminaries

Let us start by recalling the syntax and semantics (i.e. team semantics) of dependence and independence logic.

Although team semantics is intended for extensions of first-order logic obtained by adding dependence or independence atoms, for the sake of comparison we will now introduce the team semantics for first-order logic too. First-order atomic formulas α for a given signature $\mathcal{L}$ are defined as usual. Well-formed formulas of first-order logic, also called *first-order formulas*, (in *negation normal form*) are defined by the following grammar:

$$\phi ::= \alpha \mid \neg\alpha \mid \bot \mid \phi \wedge \phi \mid \phi \vee \phi \mid \exists x\phi \mid \forall x\phi$$

Formulas will be evaluated on the usual first-order models over an appropriate signature $\mathcal{L}$. We will use the same notation M for both a model and its domain. Let R be a fresh k-ary relation symbol and R^M a k-ary relation on M. We write $\mathcal{L}(R)$ for the expanded signature and (M, R^M) denotes the $\mathcal{L}(R)$-expansion of M in which the relation symbol R is interpreted as R^M. We write $\phi(R)$ to emphasize that the relation symbol R occurs in the formula ϕ.

Definition 2.1. *Let M be a model and V a set of first-order variables. A* team *X of M over V is a set of assignments of M over V, i.e., a set of functions $s : V \to M$. The set V is called the* domain *of X, denoted by* $\mathrm{dom}(X)$.

There is one and only one assignment of M over the empty domain, namely the empty assignment $\emptyset$. The singleton of the empty assignment $\{\emptyset\}$ is a team of M, and the empty set $\emptyset$ is a team of M over any domain.

Let s be an assignment of M over V and $a \in M$. We write $s(a/x)$ for the assignment of M over $V \cup \{x\}$ defined as $s(a/x)(x) = a$ and $s(a/x)(y) = s(y)$ for all $y \in V \setminus \{x\}$. For any set $N \subseteq M$ and any function $F : X \to \wp(M) \setminus \{\emptyset\}$, define

$$X(N/x) = \{s(a/x) : a \in N,\ s \in X\} \text{ and } X[F/x] = \{s(a/x) : s \in X \text{ and } a \in F(s)\}$$

We write $\vec{x}$ for a sequence $x_1, \dots, x_n$ of variables and the length n will always be clear from the context or does not matter; similarly for a sequence $\vec{F}$ of functions and a sequence $\vec{s}$ of assignments. A team $X(M/x_1)\dots(M/x_n)$ will sometimes be abbreviated as $X(M/\vec{x})$, and $X[F_1/x_1]\dots[F_n/x_n]$ as $X[F_1/x_1, \dots, F_n/x_n]$ or $X[\vec{F}/\vec{x}]$.

We now define the team semantics for first-order formulas. Note that our version of the team semantics for disjunction and existential quantifier is known as the *lax semantics* in the literature.

Definition 2.2. *Define inductively the notion of a first-order formula ϕ being* satisfied *on a model M and a team X, denoted by $M \models_X \phi$, as follows:*

– $M \models_X \alpha$ *with* α *a first-order atomic formula iff for all* $s \in X$, $M \models_s \alpha$ *in the usual sense*
– $M \models_X \neg\alpha$ *with* α *a first-order atomic formula iff for all* $s \in X$, $M \models_s \neg\alpha$ *in the usual sense*
– $M \models_X \bot$ *iff* $X = \emptyset$
– $M \models_X \phi \wedge \psi$ *iff* $M \models_X \phi$ *and* $M \models_X \psi$
– $M \models_X \phi \vee \psi$ *iff there exist* $Y, Z \subseteq X$ *with* $X = Y \cup Z$ *such that* $M \models_Y \phi$ *and* $M \models_Z \psi$
– $M \models_X \exists x\phi$ *iff* $M \models_{X[F/x]} \phi$ *for some function* $F : X \to \wp(M) \setminus \{\emptyset\}$
– $M \models_X \forall x\phi$ *iff* $M \models_{X(M/x)} \phi$

A routine inductive proof shows that first-order formulas have the downward closure property and the union closure property:

(Downward Closure Property). $M \models_X \phi$ and $Y \subseteq X$ imply $M \models_Y \phi$
(Union Closure Property). $M \models_{X_i} \phi$ for all $i \in I$ implies $M \models_{\bigcup_{i\in I} X_i} \phi$

which combined are equivalent to the flatness property:

(Flatness Property). $M \models_X \phi \iff M \models_{\{s\}} \phi$ for all $s \in X$

It follows easily from the flatness property that the team semantics for first-order formulas coincides with the usual single-assignment semantics in the sense that

$$M \models_{\{s\}} \phi \iff M \models_s \phi \tag{2}$$

holds for any model M, any assignment s and any first-order formula ϕ. If ϕ is a first-order formula, then the string $\neg\phi$, called the *syntactic negation* of ϕ, can be viewed as a first-order formula in negation normal form obtained in the usual way (i.e. by applying De Morgan's laws, the double negation law, etc.), and we write $\phi \to \psi$ for the formula $\neg\phi \vee \psi$. Since first-order formulas satisfy the Law of Excluded Middle $\phi \vee \neg\phi$ under the usual single-assignment semantics, Expression (2) implies that $M \models_{\{s\}} \phi \vee \neg\phi$ always holds, which, together with the flatness property, implies that $M \models_X \phi \vee \neg\phi$ holds for all teams X and all models M, namely, the Law of Excluded Middle holds for first-order formulas also in the sense of team semantics.

We now turn to dependence and independence logic. Well-formed formulas of independence logic ($\mathcal{I}$) are defined by the following grammar:

$$\begin{aligned}\phi ::= {} & \alpha \mid \neg\alpha \mid \bot \mid x_1 \ldots x_n \perp_{z_1\ldots z_k} y_1 \ldots y_m \mid {=}(x_1, \ldots, x_n, y) \mid x_1 \ldots x_n \subseteq y_1 \ldots y_n \mid \\ & \phi \wedge \phi \mid \phi \vee \phi \mid \exists x\phi \mid \forall x\phi\end{aligned}$$

where α ranges over first-order atomic formulas. The formulas ${=}(\vec{x}, y)$, $\vec{x} \perp_{\vec{z}} \vec{y}$ and $\vec{x} \subseteq \vec{y}$ are called *dependence atom*, *independence atom* and *inclusion atom*, respectively. We refer to any of these atoms as *atoms of dependence and independence*. For the convenience of our argument in the paper, the independence logic as defined has a richer syntax than the standard one in the literature, which has the same syntax as first-order logic extended with independence atoms only. The other atoms are definable in the standard independence logic; for a proof see e.g., [4]. *Dependence logic* ($\mathcal{D}$), which is a fragment of $\mathcal{I}$, is defined as first-order logic extended with dependence atoms, and first-order logic extended with inclusion atoms is called *inclusion logic*. In this paper we will only concentrate on dependence logic and independence logic.

The set $\text{Fv}(\phi)$ of free variables of a formula ϕ of $\mathcal{I}$ is defined as usual and we also have the new cases for dependence and independence atoms:

- $\text{Fv}(x_1 \dots x_n \perp_{z_1 \dots z_k} y_1 \dots y_m) = \{x_1, \dots, x_n, y_1, \dots, y_m, z_1, \dots, z_k\}$
- $\text{Fv}(=\!\!(x_1, \dots, x_n, y)) = \{x_1, \dots, x_n, y\}$
- $\text{Fv}(x_1, \dots, x_n \subseteq y_1, \dots, y_n) = \{x_1, \dots, x_n, y_1, \dots, y_n\}$

We write $\phi(\vec{x})$ to indicate that the free variables occurring in ϕ are among $\vec{x}$. A formula ϕ is called a *sentence* if it has no free variable.

Definition 2.3. *Define inductively the notion of a formula ϕ of $\mathcal{I}$ being* satisfied *on a model M and a team X, denoted by $M \models_X \phi$. All the cases are identical to those defined in Definition 2.2 and additionally:*

- *$M \models_X \vec{x} \perp_{\vec{z}} \vec{y}$ iff for all $s, s' \in X$, $s(\vec{z}) = s'(\vec{z})$ implies that there exists $s'' \in X$ such that*
$$s''(\vec{z}) = s(\vec{z}) = s'(\vec{z}),\ s''(\vec{x}) = s(\vec{x}) \textit{ and } s''(\vec{y}) = s'(\vec{y}).$$
- *$M \models_X =\!\!(\vec{x}, y)$ iff for all $s, s' \in X$, $s(\vec{x}) = s'(\vec{x})$ implies $s(y) = s'(y)$.*
- *$M \models_X \vec{x} \subseteq \vec{y}$ iff for all $s \in X$, there exists $s' \in X$ such that $s'(\vec{y}) = s(\vec{x})$.*

We write $\vec{x} \perp \vec{y}$ for $\vec{x} \perp_{\langle\rangle} \vec{y}$, and note that the semantic clause for $\vec{x} \perp \vec{y}$ reduces to

- $M \models_X \vec{x} \perp \vec{y}$ iff for all $s, s' \in X$, there exist $s'' \in X$ such that
$$s''(\vec{x}) = s(\vec{x}) \text{ and } s''(\vec{y}) = s'(\vec{y}).$$

A sentence ϕ is said to be *true* in M, written $M \models \phi$, if $M \models_{\{\emptyset\}} \phi$. We write $\Gamma \models \psi$ if for any model M and any team X, $M \models_X \phi$ for all $\phi \in \Gamma$ implies $M \models_X \psi$. We also write $\phi \models \psi$ for $\{\phi\} \models \psi$. If $\phi \models \psi$ and $\psi \models \phi$, then we write $\phi \equiv \psi$.

We leave it for the reader to verify that formulas of dependence logic have the downward closure property and formulas of independence logic have the empty team property and the locality property:

(Empty Team Property). $M \models_\emptyset \phi$
(Locality Property). If $\{s \restriction \text{Fv}(\phi) \mid s \in X\} = \{s \restriction \text{Fv}(\phi) \mid s \in Y\}$[1], then

$$M \models_X \phi \iff M \models_Y \phi.$$

[1] For an assignment $s : V \to M$ and a set $V' \subseteq V$ of variables, we write $s \restriction V'$ for the restriction of s to the domain V'.

Recall that the existential second-order logic (Σ_1^1) consists of those formulas that are equivalent to some formulas of the form $\exists R_1 \ldots \exists R_k \phi$, where ϕ is a first-order formula. An $\mathcal{L}(R)$-sentence $\phi(R)$ of Σ_1^1 is said to be *downward monotone* with respect to R if $(M, Q) \models \phi(R)$ and $Q' \subseteq Q$ imply $(M, Q') \models \phi(R)$. It is known that $\phi(R)$ is downward monotone with respect to R if and only if R occurs in $\phi(R)$ only negatively (see e.g., [18]). A team X of M over $\{x_1, \ldots, x_n\}$ induces an n-ary relation

$$rel(X) := \{(s(x_1), \ldots, s(x_n)) \mid s \in X\}$$

on M; conversely, an n-ary relation R on M induces a team

$$X_R := \{\{(x_1, a_1), \ldots, (x_n, a_n)\} \mid (a_1, \ldots, a_n) \in R\}.$$

Theorem 2.4 (see [5,18,23])

(i) *Every $\mathcal{L}$-sentence ϕ of $\mathcal{D}$ or $\mathcal{I}$ is equivalent to an $\mathcal{L}$-sentence τ_ϕ of Σ_1^1, i.e.,*

$$M \models \phi \iff M \models \tau_\phi$$

holds for any model M; and conversely, every $\mathcal{L}$-sentence of Σ_1^1 is equivalent to an $\mathcal{L}$-sentence $\rho(\psi)$ of $\mathcal{D}$ or $\mathcal{I}$.

(ii) *For every $\mathcal{L}$-formula ϕ of $\mathcal{I}$, there is an $\mathcal{L}(R)$-sentence $\tau_\phi(R)$ of Σ_1^1 such that for all models M and all teams X,*

$$M \models_X \phi \iff (M, rel(X)) \models \tau_\phi(R).$$

If, in particular, ϕ is a formula of $\mathcal{D}$, then the relation symbol R occurs in the sentence $\tau_\phi(R)$ only negatively.

(iii) *For every $\mathcal{L}(R)$-sentence $\psi(R)$ of Σ_1^1 that is downward monotone with respect to R, there is an $\mathcal{L}$-formula $\rho(\psi)$ of $\mathcal{D}$ such that for all models M and all teams X,*

$$M \models_X \rho(\psi) \iff (M, rel(X)) \models \psi(R) \vee \forall \vec{x} \neg R\vec{x}. \tag{3}$$

(iv) *For every $\mathcal{L}(R)$-sentence $\psi(R)$ of Σ_1^1, there is an $\mathcal{L}$-formula $\rho(\psi)$ of $\mathcal{I}$ such that (3) holds for all models M and all teams X.*

In the sequel, we will use the notations τ_ϕ and $\tau_\phi(R)$ to denote the (up to semantic equivalence) unique formulas obtained in the above theorem and refer to them as the Σ_1^1-*translations* of the formulas ϕ of $\mathcal{D}$ or $\mathcal{I}$.

3 First-Order Formulas and Negatable Formulas

Formulas of dependence and independence logic can be translated into Σ_1^1 (Theorem 2.4). Therefore in the environment of team semantics a first-order formula ϕ has two identities: It can be viewed either as a formula of $\mathcal{D}$ or $\mathcal{I}$ that is to be evaluated on teams, or as a usual formula of first-order logic that is to be evaluated on single assignments and is possibly (equivalent to) the Σ_1^1-translation τ_ψ of some formula ψ of $\mathcal{D}$ or $\mathcal{I}$. With the latter reading of a first-order formula ϕ, for all models M and all assignments s, $M \models_s \neg\phi$ iff $M \not\models_s \phi$ holds. In this sense, the formula $\neg\phi$ can be interpreted as the "classical (contradictory) negation" of ϕ. However, on the team semantics side, unless the team X is a singleton, $M \not\models_X \phi$ is in general not equivalent to $M \models_X \neg\phi$. To express the contradictory negation in the team semantics setting, let us define the *classical negation* $\sim$ and the *weak classical negation* $\dot\sim$ as follows:

- $M \models_X \sim \phi$ iff $M \not\models_X \phi$
- $M \models_X \dot\sim \phi$ iff either $M \not\models_X \phi$ or $X = \emptyset$

Since formulas of dependence and independence logic have the empty team property, the classical negation $\sim \phi$ of any formula ϕ is not definable in the logics and we are therefore not interested in the classical negation $\sim$ in this paper. On the other hand, the weak classical negation $\dot\sim \phi$ can be definable in the logics for some formulas ϕ. We say that a formula ϕ is *negatable* in $\mathcal{I}$ (or $\mathcal{D}$) if there is a formula ψ of $\mathcal{I}$ (or $\mathcal{D}$) such that $\dot\sim \phi \equiv \psi$. If a formula ϕ of $\mathcal{I}$ is negatable in $\mathcal{I}$, we also say that ϕ is a negatable formula in $\mathcal{I}$ or the formula ϕ of $\mathcal{I}$ is negatable; similarly for $\mathcal{D}$.

For any first-order sentence ϕ, we have $M \not\models_{\{\emptyset\}} \phi$ iff $M \models_{\{\emptyset\}} \neg\phi$ by the Law of Excluded Middle. Thus $\dot\sim \phi \equiv \neg\phi$, meaning that first-order sentences are negatable both in $\mathcal{D}$ and in $\mathcal{I}$. Next, we prove that negatable formulas in $\mathcal{D}$ are, actually, all flat.

Fact 3.1 *If a formula ϕ of $\mathcal{D}$ is negatable in $\mathcal{D}$, then it is upward closed (i.e. $M \models_X \phi$ and $\emptyset \neq X \subseteq Y$ imply $M \models_Y \phi$), and thus flat.*

Proof. Suppose ϕ is a formula of $\mathcal{D}$ that is not upward closed. Then, there exist a model M and two teams $X \neq \emptyset$ and $Y \supseteq X$ such that $M \models_X \phi$ and $M \not\models_Y \phi$. But this means that $\dot\sim \phi$ is not downward closed and thus not definable in $\mathcal{D}$.

We will see in the sequel that the above fact does not apply to independence logic. Also note that sentences are always upward closed (since to evaluate a sentence it is sufficient to consider the nonempty team $\{\emptyset\}$ only). Thus, the other direction of the above fact, if true, would imply that all sentences of $\mathcal{D}$ are negatable. But this is not the case, as we will see in the following characterization theorem for negatable sentences in $\mathcal{D}$ and negatable formulas in $\mathcal{I}$.

Theorem 3.2

(i) *An $\mathcal{L}$-formula ϕ of $\mathcal{I}$ is negatable in $\mathcal{I}$ if and only if its Σ_1^1-translation $\tau_\phi(R)$ is equivalent to a first-order sentence.*
(ii) *An $\mathcal{L}$-sentence ϕ of $\mathcal{D}$ is negatable in $\mathcal{D}$ if and only if its Σ_1^1-translation τ_ϕ is equivalent to a first-order sentence.*

The above theorem states that negatable formulas in $\mathcal{I}$ are exactly those formulas that have first-order translations, and negatable sentences in $\mathcal{D}$ are exactly those sentences that have first-order translations. Therefore the problem of determining whether a formula of $\mathcal{I}$ or a sentence of $\mathcal{D}$ is negatable reduces to the problem of determining whether a Σ_1^1-sentence (τ_ϕ) is equivalent to a first-order formula, or whether the second-order quantifiers in a Σ_1^1-sentence can be eliminated. This problem is known to be undecidable (this follows from e.g., [3]).

We devote the remainder of this section to the proof of Theorem 3.2. The item (ii) actually follows implicitly from the results in [23], and the item (i) can be proved by essentially the argument of Theorem 6.7 in [23]. To proceed, let us first direct our attention to the Σ_1^1 counterpart of dependence and independence logic and prove a general theorem for Σ_1^1. The proof below is inspired by Theorem 6.7 in [23].

Theorem 3.3

(i) *Let $\phi(R)$ be an $\mathcal{L}(R)$-formula of Σ_1^1 such that $(M,\emptyset) \models \phi(R)$ for any $\mathcal{L}$-model M. The formula $\neg\phi\vee\forall\vec{x}\neg R\vec{x}$ belongs to Σ_1^1 if and only if ϕ is equivalent to a first-order formula.*

(ii) *Let ϕ be an $\mathcal{L}$-formula of Σ_1^1. The $\mathcal{L}$-formula $\neg\phi$ belongs to Σ_1^1 if and only if ϕ is equivalent to a first-order formula.*

Proof. (i) It suffices to prove the direction "$\Longrightarrow$". Suppose both ϕ and $\neg\phi \vee \forall\vec{x}\neg R\vec{x}$ belong to Σ_1^1. We may assume without loss of generality that $\phi \equiv \exists S_1 \dots \exists S_k\psi$ and $(\neg\phi \vee \forall\vec{x}\neg R\vec{x}) \equiv \exists T_1 \dots \exists T_m\chi$ for some first-order formulas ψ and χ, and the relation variables $S_1, \dots, S_k, T_1, \dots, T_m$ are pairwise distinct. Assume also that $\phi(R)$ and $\exists S_1 \dots \exists S_k\psi$ are $\mathcal{L}_1(R)$-formulas, and $\neg\phi(R)\vee\forall\vec{x}\neg R\vec{x}$ and $\exists T_1 \dots \exists T_m\chi$ are $\mathcal{L}_2(R)$-formulas.

Claim 1: $\psi \models \neg\chi \vee \forall\vec{x}\neg R\vec{x}$.

Proof of Claim 1. Put $\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2 \cup \{R, S_1, \dots, S_k, T_1, \dots, T_m\}$. For any $\mathcal{L}$-model M such that $M \models \psi$, we have $M \models \exists S_1 \dots \exists S_k\psi$. If $R^M = \emptyset$, then $M \models \forall\vec{x}\neg R\vec{x}$, thereby $M \models \neg\chi \vee \forall\vec{x}\neg R\vec{x}$. If $R^M \neq \emptyset$, then we have $M \models \neg\forall\vec{x}\neg R\vec{x}$. By the assumption, we also have $M \models \phi$. Thus, we derive

$$\begin{aligned} M \models \neg\neg\phi \wedge \neg\forall\vec{x}\neg R\vec{x} &\Longrightarrow M \models \neg(\neg\phi \vee \forall\vec{x}\neg R\vec{x}) \Longrightarrow M \models \neg\exists T_1 \dots \exists T_m\chi \\ &\Longrightarrow M \models \forall T_1 \dots \forall T_m\neg\chi \Longrightarrow M \models \neg\chi \\ &\Longrightarrow M \models \neg\chi \vee \forall\vec{x}\neg R\vec{x} \end{aligned}$$

as required.

Now, by Craig's Interpolation Theorem of first-order logic, there exists a first-order $\mathcal{L}_1(R) \cap \mathcal{L}_2(R)$-formula θ such that $\psi \models \theta$ and $\theta \models \neg\chi \vee \forall\vec{x}\neg R\vec{x}$.

Claim 2: $\phi \equiv \theta$.

Proof of Claim 2. For any $\mathcal{L}_1(R)$-model M, if $M \models \phi$, then $(M, S_1^M, \dots, S_k^M) \models \psi$ for some relations $S_1^M, \dots, S_k^M$ on M. Hence, $M \models \theta$.

Conversely, for any $\mathcal{L}_1(R)$-model M such that $M \not\models \phi$, we have $R^M \neq \emptyset$ and $M \models \neg\phi \vee \forall\vec{x}\neg R\vec{x}$. The latter implies $(M, T_1^M, \ldots, T_m^M) \models \chi$ for some relations $T_1^M, \ldots, T_m^M$ on M. It then follows that $(M, T_1^M, \ldots, T_m^M) \not\models \neg\chi \vee \forall\vec{x}\neg R\vec{x}$. Hence, $M \not\models \theta$.

(ii) The nontrivial direction "$\Longrightarrow$" follows from a similar and simplified argument. Instead of proving Claim 1 as above, one proves $\psi \models \neg\chi$. $\square$

Now, we are ready to give the proof of Theorem 3.2.

Proof (of Theorem 3.2). (i) Let ϕ be an $\mathcal{L}$-formula of $\mathcal{I}$. By Theorem 2.4(ii) there exists an $\mathcal{L}(R)$-sentence $\tau_\phi(R)$ of Σ_1^1 such that for any model M and any team X,

$$M \models_X \dot{\sim}\phi \iff M \not\models_X \phi \text{ or } X = \emptyset \iff (M, rel(X)) \models \neg\tau_\phi(R) \vee \forall\vec{x}\neg R\vec{x}. \tag{4}$$

Now, to prove the direction "$\Longleftarrow$", assume that $\tau_\phi(R)$ is equivalent to a first-order sentence. Then, the sentence $\neg\tau_\phi(R)$ is also equivalent to a first-order sentence, and thus by Theorem 2.4(iv) there exists a formula $\rho(\neg\tau_\phi)$ of $\mathcal{I}$ such that for all $\mathcal{L}$-models M and all teams X,

$$M \models_X \rho(\neg\tau_\phi) \iff (M, rel(X)) \models \neg\tau_\phi(R) \vee \forall\vec{x}\neg R\vec{x}.$$

It then follows from (4) that $\rho(\neg\tau_\phi) \equiv \dot{\sim}\phi$.

Finally, to prove the direction "$\Longrightarrow$", assume that $\dot{\sim}\phi \equiv \psi$ for some formula ψ of $\mathcal{I}$. By Theorem 2.4(ii) there exists an $\mathcal{L}(R)$-sentence $\tau_\psi(R)$ of Σ_1^1 such that for all models M and all teams X,

$$M \models_X \psi \iff (M, rel(X)) \models \tau_\psi(R).$$

By (4), $\tau_\psi(R) \equiv \neg\tau_\phi(R) \vee \forall\vec{x}\neg R\vec{x}$ and thereby the formula $\neg\tau_\phi(R) \vee \forall\vec{x}\neg R\vec{x}$ belongs to Σ_1^1. For any model M, since $M \models_\emptyset \phi$, we have $(M, \emptyset) \models \tau_\phi(R)$. Then, by Theorem 3.3(i), we conclude that $\tau_\phi(R)$ is equivalent to a first-order formula.

(ii) This item is proved by a similar argument that makes use of Theorem 2.4(i) and Theorem 3.3(ii). $\square$

4 Axiomatizing Negatable Consequences in Dependence and Independence Logic

Dependence and independence logic are not axiomatizable, meaning that the consequence relation $\Gamma \models \phi$ cannot be effectively axiomatized. Nevertheless, if we restrict $\Gamma \cup \{\phi\}$ to a set of sentences and ϕ to a first-order sentence, the consequence relation $\Gamma \models \phi$ is axiomatizable and explicit axiomatizations for $\mathcal{D}$ and $\mathcal{I}$ are given in [10,20]. Throughout this section, let L denote one of the logics of $\mathcal{D}$ and $\mathcal{I}$, and $\vdash_{\mathsf{L}}$ denote the syntactic consequence relation associated with the deduction system of L defined in [20] or in [10].

Theorem 4.1 (see [10,20]). *Let Γ be a set of sentences of* L*, and ϕ a first-order formula. We have $\Gamma \models \phi \iff \Gamma \vdash_{\mathsf{L}} \phi$. In particular, $\Gamma \models \bot \iff \Gamma \vdash_{\mathsf{L}} \bot$.*

Kontinen [15] generalized the above axiomatization result to cover also the case when $\Gamma \cup \{\phi\}$ is a set of formulas (that possibly contain free variables) by adding a new relation symbol to interpret the teams. In this section, we will generalize Theorem 4.1 without expanding the signature to cover the case when $\Gamma \cup \{\phi\}$ is a set of formulas (that possibly contain free variables) and ϕ is negatable.

We first prove that under certain constraint the (possibly open) formula ψ in the entailment $\Delta, \psi \models \theta$ can be turned into a sentence without affecting the entailment relation.

Lemma 4.2. *Let $\Delta \cup \{\chi, \theta\}$ be a set of formulas of* L*. Let* $\mathrm{Fv}(\chi) = \{x_1, \dots, x_n\}$ *and* $\mathrm{Fv}(\Delta) = \bigcup_{\delta \in \Delta} \mathrm{Fv}(\delta)$*. Suppose that* $\mathrm{Fv}(\chi) \cap \mathrm{Fv}(\Delta) = \emptyset$ *and* $\mathrm{Fv}(\chi) \cap \mathrm{Fv}(\theta) = \emptyset$*. We have $\Delta, \chi \models \theta \iff \Delta, \exists x_1 \dots \exists x_n \chi \models \theta$.*

Proof. "$\Longrightarrow$": Suppose $\Delta, \chi \models \theta$. If $M \models_X \delta$ for all $\delta \in \Delta$ and $M \models_X \exists \vec{x} \chi$, then $M \models_{X[\vec{F}/\vec{x}]} \chi$ for some appropriate sequence of functions $\vec{F}$. Since $\mathrm{Fv}(\chi) \cap \mathrm{Fv}(\Delta) = \emptyset$, we derive $M \models_{X[\vec{F}/\vec{x}]} \delta$ for all $\delta \in \Delta$ by the locality property. Thus, by the assumption, we conclude that $M \models_{X[\vec{F}/\vec{x}]} \theta$, which implies $M \models_X \theta$ since $\mathrm{Fv}(\chi) \cap \mathrm{Fv}(\theta) = \emptyset$.

"$\Longleftarrow$": Suppose $\Delta, \exists \bar{x} \chi \models \theta$, and suppose $M \models_X \delta$ for all $\delta \in \Delta$ and $M \models_X \chi$. Then, we have $M \models_X \exists \vec{x} \chi$, which implies $M \models_X \theta$ by the assumption. $\square$

To understand why Theorem 4.1 can be generalized, let us consider a set $\Gamma \cup \{\phi\}$ of formulas of L. Since Σ^1_1 admits the Compactness Theorem, we may assume that Γ is a finite set. Clearly, $\Gamma \models \phi$ is equivalent to $\Gamma, \dot\sim \phi \models \bot$, and further to $\exists \vec{x}(\bigwedge \Gamma \wedge \dot\sim \phi) \models \bot$ by Lemma 4.2, where $\mathrm{Fv}(\bigwedge \Gamma \wedge \dot\sim \phi) = \{x_1, \dots, x_n\}$. Adding appropriate rules to the deduction system to guarantee the equivalence of $\exists \vec{x}(\bigwedge \Gamma \wedge \dot\sim \phi) \vdash \bot$ and $\Gamma \vdash \phi$, the Completeness Theorem can be restated as $\exists \vec{x}(\bigwedge \Gamma \wedge \dot\sim \phi) \nvdash \bot \Longrightarrow \exists \vec{x}(\bigwedge \Gamma \wedge \dot\sim \phi) \not\models \bot$. Now, assuming that $\exists \vec{x}(\bigwedge \Gamma \wedge \dot\sim \phi)$ is deductively consistent, the problem reduces to the problem of constructing a model for the sentence $\exists \vec{x}(\bigwedge \Gamma \wedge \dot\sim \phi)$. If $\dot\sim \phi$ is definable in L, then the problem further reduces to the problem of constructing a model for the Σ^1_1 sentence $\tau_{\exists \vec{x}(\bigwedge \Gamma \wedge \dot\sim \phi)}$, which can in principle be done in first-order logic. This argument shows that via the trick of weak classical negation Theorem 4.1 can, in principle, be generalized. Note that if Γ is a set of sentences and ϕ is a first-order sentence, then $\neg \phi \equiv \dot\sim \phi$ and the foregoing argument reduces to the argument given in [20].

Let us now make this idea precise. Given the Completeness Theorems in [10,20], it suffices to extend the natural deduction systems of [10,20] by adding the two rules below to ensure the equivalence of $\Gamma \vdash \phi$ and $\exists \vec{x}(\bigwedge \Gamma \wedge \dot\sim \phi) \vdash \bot$, where $\dot\sim \phi$ denotes the formula of L that is equivalent to the weak negation of ϕ.

RULES

Weak classical negation transition	Weak classical negation elimination
$\dfrac{\begin{array}{cc} & [\exists\vec{x}(\psi\wedge\dot{\sim}\phi)] \\ D_1 & D_2 \\ \psi & \bot \end{array}}{\phi}\,(*)\quad \dot{\sim}\,\mathsf{Tr}$	$\dfrac{\begin{array}{cc} & [\psi] \\ D_1 & D_2 \\ \exists\vec{x}(\psi\wedge\dot{\sim}\phi) & \phi \end{array}}{\bot}\,(*)\quad \dot{\sim}\,\mathsf{E}$
$(*)$ where the variables $x_1,\dots,x_n$ do not occur freely in any formula in the undischarged assumptions in the derivation D_2	

Let $\vdash^*_{\mathsf{L}}$ denote the syntactic consequence relation associated with the system of L extended with the rules $\dot{\sim}\,\mathsf{Tr}$ and $\dot{\sim}\,\mathsf{E}$. We now prove the Soundness and Completeness Theorem for this extended system.

Theorem 4.3. *Let $\Gamma\cup\{\phi\}$ be a set of formulas of L such that ϕ is negatable in L. We have $\Gamma\models\phi \iff \Gamma\vdash^*_{\mathsf{L}}\phi$.*

Proof. "$\Longleftarrow$": The Soundness of the system follows from Lemma 4.2; see Appendix I for the detailed proof.

"$\Longrightarrow$": Since L is compact, without loss of generality we may assume that Γ is finite. By Lemma 4.2 and the Completeness Theorem of L (Theorem 4.1), we derive

$$\Gamma\models\phi \iff \exists\bar{x}(\bigwedge\Gamma\wedge\dot{\sim}\phi)\models\bot \iff \exists\bar{x}(\bigwedge\Gamma\wedge\dot{\sim}\phi)\vdash_{\mathsf{L}}\bot \iff \Gamma\vdash^*_{\mathsf{L}}\phi$$

by applying the rules $\dot{\sim}\,\mathsf{Tr}$ and $\dot{\sim}\,\mathsf{E}$. □

A key issue in the application of the extended system is the issue of computing the weak negation of formulas in L, or, as the first step, deciding which formulas are negatable in L. As we already remarked, even if we have established in Theorem 3.2 a characterization for negatable formulas, the latter problem is undecidable. Nevertheless, it is possible to identify some interesting classes of negatable formulas. This is what we will pursue in the next section. Let us proceed now to prove that first-order formulas are negatable in $\mathcal{I}$. This will show that for independence logic Theorem 4.3 is indeed a generalization of Theorem 4.1 and also [15].

Given a first-order formula ϕ, consider its syntactic negation $\neg\phi$. By the flatness property and the Law of Excluded Middle, we have

$$M\models_X\neg\phi \iff M\not\models_{\{s\}}\phi \text{ for all } s\in X \tag{5}$$

for all models M and all nonempty teams X. This also shows that $\neg\phi$ is in general not equivalent to $\dot{\sim}\phi$, not even for atomic first-order formulas. Moreover, that the Σ^1_1-translation $\tau_\phi(R)$ of a first-order formula ϕ is equivalent to a first-order sentence is not a trivial consequence of Theorem 2.4 either, because, for instance, the translation of a first-order disjunction $\phi\vee\psi$, as given in [23], is $\tau_{\phi\vee\psi}(R)=\exists S\exists S'(\tau_\phi(S)\wedge\tau_\psi(S')\wedge\forall\vec{x}(R\vec{x}\to(S\vec{x}\vee S'\vec{x})))$.

Proposition 4.4. *If ϕ is a first-order formula, then $\dot\sim\phi(\vec{x}) \equiv \exists\vec{w}(\vec{w} \subseteq \vec{x} \wedge \neg\phi(\vec{w}))$. In particular, first-order formulas are negatable in $\mathcal{I}$.*

Proof. For all models M and all teams X, since ϕ is flat,

$$M \models_X \dot\sim\phi \iff X = \emptyset \text{ or } M \not\models_X \phi \iff X = \emptyset \text{ or } \exists s \in X(M \not\models_{\{s\}} \phi(\vec{x})).$$

By the empty team property of independence logic, it suffices to show that

$$\exists s \in X(M \not\models_{\{s\}} \phi(\vec{x})) \iff M \models_X \exists\vec{w}(\vec{w} \subseteq \vec{x} \wedge \neg\phi(\vec{w})).$$

for all models M and all nonempty teams X.

"$\Longrightarrow$": Assume $M \not\models_{\{s\}} \phi(\vec{x})$ for some $s \in X$ and $\vec{x} = x_1 \dots x_n$. For each $1 \leq i \leq n$, inductively define a constant function F_i as follows:

- $F_1 : X \to \wp(M) \setminus \{\emptyset\}$ is defined as $F_1(t) = \{s(x_1)\}$;
- $F_i : X[F_1/w_1, \dots, F_{i-1}/w_{i-1}] \to \wp(M) \setminus \{\emptyset\}$ is defined as $F_i(t) = \{s(x_i)\}$.

Consider the team $X[\vec{F}/\vec{w}]$ (see Fig. 1 in Appendix II for an example of such a team). Clearly, $M \models_{X[\vec{F}/\vec{w}]} \vec{w} \subseteq \vec{x}$. On the other hand, for any $t \in X[\vec{F}/\vec{w}]$, since $t(\vec{w}) = s(\vec{x})$ and $M \not\models_{\{s\}} \phi(\vec{x})$, we obtain $M \not\models_{\{t\}} \phi(\vec{w})$ by the locality property. Hence, $M \models_{X[\vec{F}/\vec{w}]} \neg\phi(\vec{w})$ by (5).

"$\Longleftarrow$": Conversely, suppose $M \models_X \exists\vec{w}(\vec{w} \subseteq \vec{x} \wedge \neg\phi(\vec{w}))$. Then there are appropriate functions F_i for each $1 \leq i \leq n$ such that $M \models_{X[\vec{F}/\vec{w}]} \vec{w} \subseteq \vec{x}$ and $M \models_{X[\vec{F}/\vec{w}]} \neg\phi(\vec{w})$. By (5), the latter implies that $M \not\models_{\{t\}} \phi(\vec{w})$ for some $t \in X[\vec{F}/\vec{w}]$. By the former, there exists $s' \in X[\vec{F}/\vec{w}]$ such that $s'(\vec{x}) = t(\vec{w})$. This means, by the definition of $X[\vec{F}/\vec{w}]$, that there exists $s \in X$ such that $s(\vec{x}) = s'(\vec{x}) = t(\vec{w})$. Hence, $M \not\models_{\{s\}} \phi(\vec{x})$ by the locality property. $\square$

We remarked that the Σ_1^1-translation of a disjunction $\phi \vee \psi$ of two negatable formulas ϕ and ψ is not itself a first-order formula. In the literature there is another disjunction $\mathbin{\veebar}$, defined as follows, under which the set of negatable formulas is closed:

- $M \models_X \phi \mathbin{\veebar} \psi$ iff $M \models_X \phi$ or $M \models_X \psi$

In the presence of the downward closure property this disjunction is called *intuitionistic disjunction*, and in the environment of $\mathcal{I}$ we shall call it *Boolean disjunction*. The disjunction is uniformly definable in $\mathcal{D}$ or $\mathcal{I}$ since

$$\phi \mathbin{\veebar} \psi \equiv \exists w \exists u(=\!(w) \wedge =\!(u) \wedge ((w = u) \vee \phi) \wedge ((w \neq u) \vee \psi)),$$

and clearly $\dot\sim(\phi \wedge \psi) \equiv \dot\sim\phi \mathbin{\veebar} \dot\sim\psi$ and $\dot\sim(\phi \mathbin{\veebar} \psi) \equiv \dot\sim\phi \wedge \dot\sim\psi$.

Without going into detail we remark that the extended system can be applied to give a new formal proof of Arrow's Impossibility Theorem [2] in social choice theory. In [22] the theorem is formulated as an entailment $\Gamma_{\mathsf{Arrow}} \models \phi_{\mathsf{dictator}}$ in independence logic, where Γ_{Arrow} is a set of formulas expressing the conditions in Arrow's Impossibility Theorem and ϕ_{dictator} is a formula expressing the existence of a dictator. The formula ϕ_{dictator} is of the form $\bigveebar_{i=1}^{n}\phi_i$, where ϕ_i is a first-order formula expressing that voter i is a dictator (among n voters). By what we just obtained, the formula ϕ_{dictator} is negatable in $\mathcal{I}$ and the Completeness Theorem guarantees that $\Gamma_{\mathsf{Arrow}} \vdash_{\mathcal{I}}^{*} \phi_{\mathsf{dictator}}$ is derivable in our extended system.

5 A Hierarchy of Negatable Atoms

In this section, we define an interesting class of formulas that are negatable in $\mathcal{I}$. This class will be presented in the form of an alternating hierarchy of atoms that are definable in $\mathcal{I}$. These atoms are closely related to the *dependency notions* considered in [6], and the *generalized dependence atoms* studied in [16,21]. We will demonstrate that all first-order formulas, dependence atoms, independence atoms and inclusion atoms belong to this class. It then follows from the completeness result we obtained in the previous section that consequences of these types in $\mathcal{I}$ are derivable in the extended system.

Let us start by defining the notion of abstract relation. A k-ary *relation* R is a class of pairs (M, R^M) that is closed under taking isomorphic images, where M ranges over first-order models and $R^M \subseteq M^k$. For instance, the familiar equality $=$ is a binary relation defined by the class

$$\{(M, =^M) \mid M \text{ is a first-order model}\}, \text{ where } =^M := \{(a,a) \mid a \in M\}.$$

Every first-order formula $\phi(x_1, \dots, x_k)$ with k free variables is associated with a k-ary relation

$$\boldsymbol{\phi} := \{(M, \boldsymbol{\phi}^M) \mid M \text{ is a first-order model}\},$$

where $\boldsymbol{\phi}^M := \{(s(x_1), \dots, s(x_k)) \mid M \models_s \phi\}$. A k-ary relation R is said to be *(first-order) definable* if there exists a (first-order) formula $\phi_R(w_1, \dots, w_k)$ such that for all models M and all assignments s,

$$s(\vec{w}) \in R^M \iff M \models_s \phi_R(\vec{w}).$$

Clearly, the first-order formula $w = u$ defines the equality relation, and every first-order formula ϕ defines its associated relation $\boldsymbol{\phi}$.

If R is a k-ary relation, then we write $\overline{R}$ for the complement of R that is defined by letting $\overline{R}^M = M^k \setminus R^M$ for all models M. Clearly, if a first-order formula ϕ defines R, then its negation $\neg\phi$ defines $\overline{R}$.

If $\vec{s} = \langle s_1, \dots, s_k \rangle$, then we write $\vec{s}(\vec{x})$ for $\langle s_1(\vec{x}), \dots, s_k(\vec{x}) \rangle$. For every sequence $\mathsf{k} = \langle k_1, \dots, k_n \rangle$ of natural numbers and every $(k_1 + \cdots + k_n)m$-ary relation R, we introduce two new atomic formulas $\Sigma^R_{n,\mathsf{k}}(x_1, \dots, x_m)$ and $\Pi^R_{n,\mathsf{k}}(x_1, \dots, x_m)$ with the semantics defined as follows:

- $M \models_\emptyset \Sigma^R_{n,\mathsf{k}}(\vec{x})$ and $M \models_\emptyset \Pi^R_{n,\mathsf{k}}(\vec{x})$.
- If n is odd, then define for any model M and any *nonempty* team X
  - $M \models_X \Sigma^R_{n,\mathsf{k}}(\vec{x})$ iff there exist $s_{11}, \dots, s_{1k_1} \in X$ such that for all $s_{21}, \dots, s_{2k_2} \in X$, ... there exist $s_{n1}, \dots, s_{nk_n} \in X$ such that $(\vec{s_1}(\vec{x}), \dots, \vec{s_n}(\vec{x})) \in R^M$;
  - $M \models_X \Pi^R_{n,\mathsf{k}}(\vec{x})$ iff for all $s_{11}, \dots, s_{1k_1} \in X$, there exist $s_{21}, \dots, s_{2k_2} \in X$ such that ... for all $s_{n1}, \dots, s_{nk_n} \in X$, it holds that $(\vec{s_1}(\vec{x}), \dots, \vec{s_n}(\vec{x})) \in R^M$.
- Similarly if n is even.

Fact 5.1. $\dot\sim \Sigma^R_{n,\mathsf{k}}(\vec{x}) \equiv \Pi^{\overline{R}}_{n,\mathsf{k}}(\vec{x})$ *and* $\dot\sim \Pi^R_{n,\mathsf{k}}(\vec{x}) \equiv \Sigma^{\overline{R}}_{n,\mathsf{k}}(\vec{x})$.

Let us now give some examples of the $\Sigma^R_{n,\mathsf{k}}$ and $\Pi^R_{n,\mathsf{k}}$ atoms.

Example 5.2

(a) *The dependence atom* $=\!\!(x_1, \dots, x_k, y)$ *is a* $\Pi^{\mathbf{dep}_k}_{1,\langle 2\rangle}(x_1, \dots, x_k, y)$ *atom, where* $\mathbf{dep}_k$ *is a* $2(k+1)$*-ary relation defined as*

$$(a_1, \dots, a_k, b, a'_1, \dots, a'_k, b') \in (\mathbf{dep}_k)^M \quad \textit{iff} \quad [\,(a_1, \dots, a_k) = (a'_1, \dots, a'_k) \Longrightarrow b = b'\,].$$

The first-order formula $\big((w_1 = w'_1) \wedge \dots \wedge (w_k = w'_k)\big) \to (u = u')$ *defines* $\mathbf{dep}_k$.

(b) *The independence atom* $x_1, \dots, x_k \perp_{z_1,\dots,z_n} y_1, \dots, y_m$ *is a*

$$\Pi^{\mathbf{ind}_{k,m,n}}_{2,\langle 2,1\rangle}(x_1, \dots, x_k, y_1, \dots, y_m, z_1, \dots, z_n)$$

atom, where $\mathbf{ind}_{k,m,n}$ *is a (first-order definable)* $(2{+}1)(k{+}m{+}n)$*-ary relation defined as* $(\vec{a}, \vec{b}, \vec{c}, \vec{a'}, \vec{b'}, \vec{c'}, \vec{a''}, \vec{b''}, \vec{c''}) \in (\mathbf{ind}_{k,m,n})^M$ *iff*

$$\begin{aligned}(c_n, \dots, c_n) =\,& (c'_1, \dots, c'_n) = (c''_1, \dots, c''_n)\\ &\Longrightarrow [\,(a''_1, \dots, a''_k) = (a_1, \dots, a_k) \textit{ and } (b''_1, \dots, b''_m) = (b'_1, \dots, b'_m)\,].\end{aligned}$$

(c) *The inclusion atom* $x_1, \dots, x_k \subseteq y_1, \dots, y_k$ *is a* $\Pi^{\mathbf{inc}_k}_{2,\langle 1,1\rangle}(x_1, \dots, x_k, y_1, \dots, y_k)$ *atom, where* $\mathbf{inc}_k$ *is a (first-order definable)* $(1+1)2k$*-ary relation defined as*

$$(a_1, \dots, a_k, b_1, \dots, b_k, a'_1, \dots, a'_k, b'_1, \dots, b'_k) \in (\mathbf{inc}_k)^M \ \textit{iff}\ (a_1, \dots, a_k) = (b'_1, \dots, b'_k).$$

(d) *Every first-order formula* $\phi(x_1, \dots, x_k)$ *is a* $\Pi^{\boldsymbol{\phi}}_{1,\langle 1\rangle}(x_1, \dots, x_k)$ *atom, where* $\boldsymbol{\phi}$ *is a (first-order definable)* $1 \cdot k$*-ary relation defined as*

$$(a_1, \dots, a_k) \in \boldsymbol{\phi}^M \quad \textit{iff} \quad M \models_{s_{\vec{a}}} \phi \textit{ where } s_{\vec{a}}(x_i) = a_i \textit{ for all } i.$$

In what follows, let $\mathsf{k} = \langle k_1, \dots, k_n\rangle$ be an arbitrary sequence of natural numbers, $\vec{x} = \langle x_1, \dots, x_m\rangle$ an arbitrary sequence of variables, and R an arbitrary $(k_1{+}\dots{+}k_n)m$-ary relation. Suppose R is definable by a formula $\phi_R(\vec{\mathsf{w_1}}, \dots, \vec{\mathsf{w_n}})$, where $\vec{\mathsf{w_i}} = \langle \mathsf{w}_{\mathsf{i},1}, \dots, \mathsf{w}_{\mathsf{i},\mathsf{k_i}}\rangle$ and $\mathsf{w}_{\mathsf{i},\mathsf{j}} = \langle w_{i,j,1}, \dots, w_{i,j,m}\rangle$. The $\Sigma^R_{n,\mathsf{k}}(\vec{x})$ and $\Pi^R_{n,\mathsf{k}}(\vec{x})$ atoms can be translated into second-order logic in the same manner as in Theorem 2.4. For instance, if n is even, let S be a fresh m-ary relation symbol and let $\tau_{\Sigma^R_{n,\mathsf{k}}(\vec{x})}(S) :=$

$$\begin{aligned}&\exists\vec{\mathsf{w_1}}\Big(S(\mathsf{w}_{1,1}) \wedge \dots \wedge S(\mathsf{w}_{1,\mathsf{k_1}}) \wedge \forall\vec{\mathsf{w_2}}\Big(S(\mathsf{w}_{2,1}) \wedge \dots \wedge S(\mathsf{w}_{2,\mathsf{k_2}}) \to \exists\vec{\mathsf{w_3}} \cdots\\ &\qquad\cdots \exists\vec{\mathsf{w_n}}\Big(S(\mathsf{w}_{\mathsf{n},1}) \wedge \dots \wedge S(\mathsf{w}_{\mathsf{n},\mathsf{k_n}}) \wedge \phi_R(\vec{\mathsf{w_1}}, \dots, \vec{\mathsf{w_n}})\underbrace{\Big)\cdots\Big)\Big)}_{n}.\end{aligned}$$

Then, we have $M \models_X \Sigma^R_{n,\mathsf{k}}(\vec{x}) \iff (M, rel(X)) \models \tau_{\Sigma^R_{n,\mathsf{k}}(\vec{x})}(S)$ for any model M and any team X. If $\phi_R(\overrightarrow{\mathsf{w_1}}, \dots, \overrightarrow{\mathsf{w_n}})$ is a first-order formula, i.e., if R is first-order definable, then $\tau_{\Sigma^R_{n,\mathsf{k}}(\vec{x})}(S)$ is a first-order sentence. This shows, by Theorem 3.2(i), that $\Sigma^R_{n,\mathsf{k}}(\vec{x})$ and $\Pi^R_{n,\mathsf{k}}(\vec{x})$ atoms are negatable in $\mathcal{I}$ as long as R is first-order definable.

Yet, in order to apply the rules of the extended deduction system defined in Sect. 4 to derive the $\Sigma^R_{n,\mathsf{k}}(\vec{x})$ and $\Pi^R_{n,\mathsf{k}}(\vec{x})$ consequences in $\mathcal{I}$, one needs to compute the formulas that are equivalent to the weak classical negations of the $\Sigma^R_{n,\mathsf{k}}(\vec{x})$ and $\Pi^R_{n,\mathsf{k}}(\vec{x})$ atoms in the original language of $\mathcal{I}$. This can be done by applying Fact 5.1 and going through the Σ^1_1-translation (i.e. applying Theorem 2.4(ii) and (iv)). However, as the Σ^1_1-translation creates a number of dummy symbols (see [5,23]), such an algorithm is inefficient. In the remainder of this section, we will give a direct definition of the atoms $\Sigma^R_{n,\mathsf{k}}(\vec{x})$ and $\Pi^R_{n,\mathsf{k}}(\vec{x})$ in the original language of $\mathcal{I}$.

For each $1 \leq i \leq n$, define

- $\mathsf{inc}(\mathsf{w_{i,1}}, \dots, \mathsf{w_{i,k_i}}; \vec{x}) := \bigwedge_{j=1}^{k_i} (\mathsf{w_{i,j}} \subseteq \vec{x})$
- $\mathsf{pro}(\overrightarrow{\mathsf{w_1}}, \dots, \overrightarrow{\mathsf{w_{i-1}}}; \vec{x}; \mathsf{w_{i,1}}, \dots, \mathsf{w_{i,k_i}}) :=$

$$\left(\bigwedge_{j=1}^{k_i} (\vec{x} \subseteq \mathsf{w_{i,j}})\right) \wedge \left(\bigwedge_{j=1}^{k_i} (\langle \mathsf{w_{i,j'}} \mid j' \neq j \rangle \perp \mathsf{w_{i,j}})\right) \wedge (\overrightarrow{\mathsf{w_1}} \dots \overrightarrow{\mathsf{w_{i-1}}} \perp \mathsf{w_{i,1}} \dots \mathsf{w_{i,k_i}})$$

and inductively define formulas σ_i and π_i as follows:

- $\sigma_1[\vec{x}; \phi_R(\overrightarrow{\mathsf{w_1}}, \dots, \overrightarrow{\mathsf{w_n}})] := \exists \overrightarrow{\mathsf{w_n}} \Big(\mathsf{inc}(\mathsf{w_{n,1}}, \dots, \mathsf{w_{n,k_n}}; \vec{x}) \wedge \phi_R(\overrightarrow{\mathsf{w_1}}, \dots, \overrightarrow{\mathsf{w_n}})\Big)$
- $\pi_1[\vec{x}; \phi_R(\overrightarrow{\mathsf{w_1}}, \dots, \overrightarrow{\mathsf{w_n}})] := \exists \overrightarrow{\mathsf{w_n}} \Big(\mathsf{pro}(\overrightarrow{\mathsf{w_1}}, \dots, \overrightarrow{\mathsf{w_{n-1}}}; \vec{x}; \mathsf{w_{n,1}}, \dots, \mathsf{w_{n,k_n}})$
  $\wedge\, \phi_R(\overrightarrow{\mathsf{w_1}}, \dots, \overrightarrow{\mathsf{w_n}})\Big)$
- $\sigma_{i+1}[\vec{x}; \phi_R(\overrightarrow{\mathsf{w_1}}, \dots, \overrightarrow{\mathsf{w_n}})] := \exists \overrightarrow{\mathsf{w_{n-i}}} \Big(\mathsf{inc}(\mathsf{w_{n-i,1}}, \dots, \mathsf{w_{n-i,k_{n-i}}}; \vec{x})$
  $\wedge \pi_i[\vec{x}; \phi_R(\overrightarrow{\mathsf{w_1}}, \dots, \overrightarrow{\mathsf{w_n}})]\Big)$
- $\pi_{i+1}[\vec{x}; \phi_R(\overrightarrow{\mathsf{w_1}}, \dots, \overrightarrow{\mathsf{w_n}})] := \exists \overrightarrow{\mathsf{w_{n-i}}} \Big(\mathsf{pro}(\overrightarrow{\mathsf{w_1}}, \dots, \overrightarrow{\mathsf{w_{n-i-1}}}; \vec{x}; \mathsf{w_{n-i,1}}, \dots, \mathsf{w_{n-i,k_{n-i}}})$
  $\wedge \sigma_i[\vec{x}; \phi_R(\overrightarrow{\mathsf{w_1}}, \dots, \overrightarrow{\mathsf{w_n}})]\Big)$[2]

Theorem 5.3. *Let R and ϕ_R be as above. Then*

- $\Sigma^R_{n,\mathsf{k}}(x_1, \dots, x_m) \equiv \sigma_n[\vec{x}; \phi_R(\overrightarrow{\mathsf{w_1}}, \dots, \overrightarrow{\mathsf{w_n}})]$
- $\Pi^R_{n,\mathsf{k}}(x_1, \dots, x_m) \equiv \pi_n[\vec{x}; \phi_R(\overrightarrow{\mathsf{w_1}}, \dots, \overrightarrow{\mathsf{w_n}})]$

[2] If $i + 1 = n$, then $\overrightarrow{\mathsf{w_1}} \dots \overrightarrow{\mathsf{w_{n-i-1}}}$ denotes the empty sequence $\langle\rangle$ and we stipulate $\langle\rangle \perp \vec{y} := \top$.

Proof. We only give the detailed proof for $\Sigma^R_{n,\mathsf{k}}(x_1,\dots,x_m)$ when n is odd. The other case and the other equivalence can be proved analogously.

Our proof makes use of Lemma A in Appendix III. First, note that

$$\sigma_n[\vec{x};\phi_R(\overrightarrow{\mathsf{w_1}},\dots,\overrightarrow{\mathsf{w_n}})] := \exists\overrightarrow{\mathsf{w_1}}\Big(\mathsf{inc}(\mathsf{w}_{1,1},\dots,\mathsf{w}_{1,\mathsf{k}_1};\vec{x})\wedge \exists\overrightarrow{\mathsf{w_2}}\Big(\mathsf{pro}(\overrightarrow{\mathsf{w_1}};\vec{x};\mathsf{w}_{2,1},\dots,\mathsf{w}_{2,\mathsf{k}_2})\wedge\cdots\cdots\wedge \exists\overrightarrow{\mathsf{w_n}}\Big(\mathsf{inc}(\mathsf{w}_{\mathsf{n},1},\dots,\mathsf{w}_{\mathsf{n},\mathsf{k}_\mathsf{n}};\vec{x})\wedge\phi_R(\overrightarrow{\mathsf{w_1}},\dots,\overrightarrow{\mathsf{w_n}})\underbrace{\Big)\cdots\Big)\Big)}_{n}$$

Suppose $M\models_X \Sigma^R_{n,\mathsf{k}}(\vec{x})$ for some model M and some nonempty team X. Then

$$(\exists\overrightarrow{s_1}\in X^{k_1})(\forall\overrightarrow{s_2}\in X^{k_2})\cdots\cdots(\exists\overrightarrow{s_n}\in X^{k_n})(\overrightarrow{s_1}(\vec{x}),\dots,\overrightarrow{s_n}(\vec{x}))\in R^M. \tag{6}$$

Let $\Gamma_1=\langle\gamma_{1,1},\dots,\gamma_{1,k_1}\rangle$ be a sequence of constant choice functions $\gamma_{1,j}:X\to X$ defined as $\gamma_{1,j}(t)=s_{1,j}$. Let $\overrightarrow{F_{1,1}},\dots,\overrightarrow{F_{1,k_1}}$ be the group of simulating functions for $\Gamma_1[X]\upharpoonright\vec{x}$ on $\mathsf{w}_{1,1},\dots,\mathsf{w}_{1,\mathsf{k}_1}$ and Y_1 its associated team defined as in Lemma A(i) in Appendix III. Then, $M\models_{Y_1}\mathsf{inc}(\mathsf{w}_{1,1},\dots,\mathsf{w}_{1,\mathsf{k}_1};\vec{x})$. It then remains to show that $M\models_{Y_1}\pi_{n-1}[\vec{x};\phi_R(\overrightarrow{\mathsf{w_1}},\dots,\overrightarrow{\mathsf{w_n}})]$.

Let $\vec{F}_{2,1},\dots,\vec{F}_{2,k_2}$ be the group of duplicating functions for $Y_1\upharpoonright\vec{x}$ on $\mathsf{w}_{2,1},\dots,\mathsf{w}_{2,\mathsf{k}_2}$, and Y_2 its associated team defined as in Lemma A(ii) in Appendix III. Then, $M\models_{Y_2}\mathsf{pro}(\overrightarrow{\mathsf{w_1}};\vec{x};\mathsf{w}_{2,1},\dots,\mathsf{w}_{2,\mathsf{k}_2})$.

It remains to show that $M\models_{Y_2}\sigma_{n-2}[\vec{x};\phi_R(\overrightarrow{\mathsf{w_1}},\dots,\overrightarrow{\mathsf{w_n}})]$. By Lemma A(ii), for each $t\in Y_2$, there exists $\overrightarrow{s_{t,2}}=(s^t_{2,1},\dots,s^t_{2,k_2})\in X^{k_2}$ satisfying

$$s^t_{2,1}(\vec{x})=t(\mathsf{w}_{2,1}),\dots,s^t_{2,k_2}(\vec{x})=t(\mathsf{w}_{2,\mathsf{k}_2}).$$

Hence, by (6), there exists $\overrightarrow{s_{t,3}}=(s^t_{3,1},\dots,s^t_{3,k_3})\in X^{k_3}$ such that

$$(\forall\overrightarrow{s_4}\in X^{k_4})\cdots\cdots(\exists\overrightarrow{s_n}\in X^{k_n})(\overrightarrow{s_1}(\vec{x}),\overrightarrow{s_{t,2}}(\vec{x}),\overrightarrow{s_{t,3}}(\vec{x}),\overrightarrow{s_4}(\vec{x})\dots,\overrightarrow{s_n}(\vec{x}))\in R^M.$$

Let $\Gamma_3=\langle\gamma_{3,1},\dots,\gamma_{3,k_3}\rangle$ be a sequence of choice functions $\gamma_{3,j}:Y_2\to Y_2$ defined as $\gamma_{3,j}(t)=s^t_{3,j}$. Let $\overrightarrow{F_{3,1}},\dots,\overrightarrow{F_{3,k_3}}$ be the group of simulating functions for $\Gamma_3[Y_2]\upharpoonright\vec{x}$ on $\mathsf{w}_{3,1},\dots,\mathsf{w}_{3,\mathsf{k}_3}$, and Y_3 its associated team defined as in Lemma A(i) in Appendix III. Then, $M\models_{Y_3}\mathsf{inc}(\mathsf{w}_{3,1},\dots,\mathsf{w}_{3,\mathsf{k}_3};\vec{x})$ and it remains to show that $M\models_{Y_3}\pi_{n-3}[\vec{x};\phi_R(\overrightarrow{\mathsf{w_1}},\dots,\overrightarrow{\mathsf{w_n}})]$.

Repeat the argument n times. In the last step we have Y_n and Γ_n defined and $M\models_{Y_n}\mathsf{inc}(\mathsf{w}_{\mathsf{n},1},\dots,\mathsf{w}_{\mathsf{n},\mathsf{k}_\mathsf{n}};\vec{x})$ by Lemma A(i). It then only remains to show that $M\models_{Y_n}\phi_R(\overrightarrow{\mathsf{w_1}},\dots,\overrightarrow{\mathsf{w_n}})$. Since ϕ_R is flat, it suffices to show that $M\models_{\{t\}}\phi_R$ holds for all $t\in Y_n$. By the definition of Y_n and Lemma A(i)(ii), we have

$$(\overrightarrow{s_1}(\vec{x}),\overrightarrow{s_{t,2}}(\vec{x}),\overrightarrow{s_{t,3}}(\vec{x}),\overrightarrow{s_{t,4}}(\vec{x})\dots,\overrightarrow{s_{t,n}}(\vec{x}))\in R^M$$

$$\text{and } t(\overrightarrow{\mathsf{w_1}})=\overrightarrow{s_1}(\vec{x}),\ t(\overrightarrow{\mathsf{w_2}})=\overrightarrow{s_{t,2}}(\vec{x}),\ \dots,\ t(\overrightarrow{\mathsf{w_n}})=\overrightarrow{s_{t,n}}(\vec{x}).$$

Thus, $M\models_{\{t\}}\phi_R(\overrightarrow{\mathsf{w_1}},\dots,\overrightarrow{\mathsf{w_n}})$, as the first-order formula ϕ_R defines R.

Conversely, suppose $M \models_X \sigma_n[\vec{x}; \phi_R(\overrightarrow{\mathsf{w_1}}, \ldots, \overrightarrow{\mathsf{w_n}})]$ for some model M and some nonempty team X. Let Y be a team generated by the formula $\sigma_n[\vec{x}; \phi_R(\overrightarrow{\mathsf{w_1}}, \ldots, \overrightarrow{\mathsf{w_n}})]$ from X such that $M \models_Y \phi_R(\overrightarrow{\mathsf{w_1}}, \ldots, \overrightarrow{\mathsf{w_n}})$.

Pick any $t \in Y$. Since $M \models_Y \mathsf{inc}(\mathsf{w}_{1,1}, \ldots, \mathsf{w}_{1,\mathsf{k}_1}; \vec{x})$, there exist $s_{1,1}, \ldots, s_{1,k_1} \in X$ such that

$$s_{1,1}(\vec{x}) = t(\mathsf{w}_{1,1}), \ldots, s_{1,k_1}(\vec{x}) = t(\mathsf{w}_{1,k_1}).$$

Let $s_{2,1}, \ldots, s_{2,k_2} \in X$ be arbitrary. Since $M \models_Y \mathsf{pro}(\overrightarrow{\mathsf{w_1}}; \vec{x}; \mathsf{w}_{2,1}, \ldots, \mathsf{w}_{2,\mathsf{k}_2})$, it is not hard to see that there exist $t_2 \in Y$ such that

$$t_2(\overrightarrow{\mathsf{w_1}}) = t(\overrightarrow{\mathsf{w_1}}) = \overrightarrow{s_1}(\vec{x}) \text{ and } s_{2,1}(\vec{x}) = t_2(\mathsf{w}_{2,1}), \ldots, s_{2,k_2}(\vec{x}) = t_2(\mathsf{w}_{2,k_2}).$$

Repeat the argument n times to find in the same manner the corresponding assignments $\overrightarrow{s_3} \in X^{k_3}, \overrightarrow{s_5} \in X^{k_5}, \ldots, \overrightarrow{s_n} \in X^{k_n}$ and the corresponding assignments $t_4, t_6, \ldots, t_{n-1} \in Y$ for arbitrary $\overrightarrow{s_4} \in X^{k_4}, \overrightarrow{s_6} \in X^{k_6}, \ldots, \overrightarrow{s_{n-1}} \in X^{k_{n-1}}$. In the last step we have

$$t_{n-1}(\overrightarrow{\mathsf{w_1}}) = \overrightarrow{s_1}(\vec{x}), \ldots, t_{n-1}(\overrightarrow{\mathsf{w}_{n-1}}) = \overrightarrow{s_{n-1}}(\vec{x})$$

and there exist $s_{n,1}, \ldots, s_{n,k_n} \in X$ such that

$$s_{n,1}(\vec{x}) = t_{n-1}(\mathsf{w}_{n,1}), \ldots, s_{n,k_n}(\vec{x}) = t_{n-1}(\mathsf{w}_{n,k_n}).$$

Since $M \models_Y \phi_R(\overrightarrow{\mathsf{w_1}}, \ldots, \overrightarrow{\mathsf{w_n}})$, we have $M \models_{\{t_{n-1}\}} \phi_R(\overrightarrow{\mathsf{w_1}}, \ldots, \overrightarrow{\mathsf{w_n}})$ by the downward closure property. Since the first-order formula ϕ_R defines R, we conclude

$$(t_{n-1}(\overrightarrow{\mathsf{w_1}}), \ldots, t_{n-1}(\overrightarrow{\mathsf{w}_n})) \in R^M \text{ yielding } (\overrightarrow{s_1}(\vec{x}), \ldots, \overrightarrow{s_n}(\vec{x})) \in R^M.$$

□

6 Concluding Remarks

In this paper, we have extended the natural deduction systems of dependence and independence logic defined in [10,20] and obtained complete axiomatizations of the negatable consequences in these logics. We also gave a characterization of negatable formulas in $\mathcal{I}$ and negatable sentences in $\mathcal{D}$. Determining whether a formula of $\mathcal{I}$ or $\mathcal{D}$ is negatable is an undecidable problem. Nevertheless, we identified an interesting class of negatable formulas. Formulas in this class are presented as $\Sigma^R_{n,\mathsf{k}}$ and $\Pi^R_{n,\mathsf{k}}$ atoms. First-order formulas, dependence and independence atoms belong to this class. Since the set of negatable formulas is closed under the Boolean connectives $\wedge$ and $\mathbin{\mathchoice{\vee}{\vee}{\vee}{\vee}\!\!\vee}$, Boolean combinations of $\Sigma^R_{n,\mathsf{k}}$ and $\Pi^R_{n,\mathsf{k}}$ atoms are also negatable.

An interesting corollary of the paper is that Armstrong's Axioms [1] that characterize dependence atoms and the Geiger-Paz-Pearl axioms [8] that characterize independence atoms can be derived in our extended system of $\mathcal{I}$. We leave the derivations of these axioms for future work.

The results of this paper can be generalized in two directions. The first direction is to identify other negatable formulas than those in the set of the Boolean combinations of atoms from our hierarchy. The other direction is to analyze the $\Sigma^R_{n,\mathsf{k}}$ and $\Pi^R_{n,\mathsf{k}}$ atoms in more detail. As we saw in Example 5.2, first-order formulas and the atoms of dependence and independence situate only on the Π_1 or Π_2 level. Identifying interesting properties that situate on higher levels of the hierarchy and studying the logics that the higher level atoms induce would be an interesting topic for future research. For example, it is easy to verify that $\Pi^R_{1,\mathsf{k}}$ atoms (including first-order formulas and dependence atoms) are closed downward, and $\Sigma^R_{1,\mathsf{k}}$ atoms are closed upward. First-order logic extended with upward closed atoms is shown in [7] to be equivalent to first-order logic. Adding other such atoms to first-order logic results in many new logics that are expressively less than Σ^1_1 or independence logic and possibly stronger than first-order logic. These logics are potentially interesting, because, for instance, by the argument of this paper, the negatable consequences in these logics can in principle be axiomatized.

Acknowledgements. The author is in debt to Eric Pacuit for the discussions on formalizing Arrow's Theorem in independence logic, which in the end led to the results of this paper unexpectedly. The author would also like to thank Juha Kontinen and Jouko Väänänen for stimulating discussions concerning the technical details of the paper.

Appendix I

Proof (of the direction "$\Longleftarrow$" of Theorem 4.3*).* It suffices to show by induction that $\Gamma \models \phi$ holds for each derivation D in the extended system with the conclusion ϕ and the hypotheses in Γ. We only give the proof for the induction step when the rule $\dot{\sim}\,\mathsf{Tr}$ is applied. The case when the rule $\dot{\sim}\,\mathsf{E}$ is applied can be proved similarly, and all the other cases follow from the arguments in [20] and in [10].

Assume that D_2 is a derivation for $\Delta, \exists\vec{x}(\psi \wedge \dot{\sim}\,\phi) \vdash^*_{\mathsf{L}} \bot$ and D_1 is a derivation for $\Pi \vdash^*_{\mathsf{L}} \psi$, where $\mathrm{Fv}(\Delta) \cap \{x_1, \dots, x_n\} = \emptyset$. We show that $\Delta, \Pi \models \phi$. By the induction hypothesis, we have $\Delta, \exists\vec{x}(\psi \wedge \dot{\sim}\,\phi) \models \bot$ and $\Pi \models \psi$. From the former and Lemma 4.2 we obtain $\Delta, \psi \wedge \dot{\sim}\,\phi \models \bot$, which is equivalent to $\Delta, \psi \models \phi$. Since $\Pi \models \psi$, we conclude $\Delta, \Pi \models \phi$, as desired. $\square$

Appendix II

	$x_1 \ldots x_n$
	$a_1 \ldots a_n$
s	$b_1 \ldots b_n$
	$c_1 \ldots c_n$

	$x_1 \ldots x_n$	$w_1 \ldots w_n$
	$a_1 \ldots a_n$	$b_1 \ldots b_n$
s'	$b_1 \ldots b_n$	$b_1 \ldots b_n$
	$c_1 \ldots c_n$	$b_1 \ldots b_n$

Fig. 1. (a) A team X. (b) A team $X[F/\vec{w}]$

(a)

	$\vec{x}$	$\vec{y}$
s_1	$\vec{a}$	$\vec{a}$
s_2	$\vec{a}$	$\vec{b}$
s_3	$\vec{a}$	$\vec{c}$
s_4	$\vec{b}$	$\vec{a}$
s_5	$\vec{b}$	$\vec{b}$
s_6	$\vec{b}$	$\vec{c}$

(b)

	$\vec{x}$	$\vec{y}$	$\overrightarrow{w_1}$
s'_1	$\vec{a}$	$\vec{a}$	$\vec{b}$
s'_2	$\vec{a}$	$\vec{b}$	$\vec{b}$
s'_3	$\vec{a}$	$\vec{c}$	$\vec{a}$
s'_4	$\vec{b}$	$\vec{a}$	$\vec{b}$
s'_5	$\vec{b}$	$\vec{b}$	$\vec{b}$
s'_6	$\vec{b}$	$\vec{c}$	$\vec{a}$

(c)

	$\vec{x}$	$\vec{y}$	$\overrightarrow{w_1}$	$\overrightarrow{w_2}$
s''_1	$\vec{a}$	$\vec{a}$	$\vec{b}$	$\vec{a}$
s''_2	$\vec{a}$	$\vec{b}$	$\vec{b}$	$\vec{b}$
s''_3	$\vec{a}$	$\vec{c}$	$\vec{a}$	$\vec{b}$
s''_4	$\vec{b}$	$\vec{a}$	$\vec{b}$	$\vec{a}$
s''_5	$\vec{b}$	$\vec{b}$	$\vec{b}$	$\vec{b}$
s''_6	$\vec{b}$	$\vec{c}$	$\vec{a}$	$\vec{b}$

Fig. 2. (a) A team X (b) A team $X[\vec{F_1}/\overrightarrow{w_1}]$ (c) A team $X[\vec{F_1}/\overrightarrow{w_1}, \vec{F_2}/\overrightarrow{w_2}]$

(a)

	$\vec{x}$	$\vec{y}$
s_1	$\vec{a}$	$\vec{b}$
s_2	$\vec{b}$	$\vec{a}$

(b)

	$\vec{x}$	$\vec{y}$	$\overrightarrow{w_1}$
s_{11}	$\vec{a}$	$\vec{b}$	$\vec{a}$
s_{12}	$\vec{a}$	$\vec{b}$	$\vec{b}$
s_{21}	$\vec{b}$	$\vec{a}$	$\vec{a}$
s_{22}	$\vec{b}$	$\vec{a}$	$\vec{b}$

(c)

	$\vec{x}$	$\vec{y}$	$\overrightarrow{w_1}$	$\overrightarrow{w_2}$
s_{111}	$\vec{a}$	$\vec{b}$	$\vec{a}$	$\vec{a}$
s_{112}	$\vec{a}$	$\vec{b}$	$\vec{a}$	$\vec{b}$
s_{121}	$\vec{a}$	$\vec{b}$	$\vec{b}$	$\vec{a}$
s_{122}	$\vec{a}$	$\vec{b}$	$\vec{b}$	$\vec{b}$
s_{211}	$\vec{b}$	$\vec{a}$	$\vec{a}$	$\vec{a}$
s_{212}	$\vec{b}$	$\vec{a}$	$\vec{a}$	$\vec{b}$
s_{221}	$\vec{b}$	$\vec{a}$	$\vec{b}$	$\vec{a}$
s_{222}	$\vec{b}$	$\vec{a}$	$\vec{b}$	$\vec{b}$

Fig. 3. (a) A team X (b) A team $X[\vec{F_1}/\overrightarrow{w_1}]$ (c) A team $X[\vec{F_1}/\overrightarrow{w_1}, \vec{F_2}/\overrightarrow{w_2}]$

Appendix III

Lemma A. *Let X be a nonempty team of a model M with $x_1, \dots, x_m \in \mathrm{dom}(X)$.*

(i) *Let $\gamma : X \to X$ be a choice function. Define inductively functions $F_1, \dots, F_m$ to simulate assignments in $\gamma[X]$ restricted to $\vec{x}$ on a sequence $\vec{w} = \langle w_1, \dots, w_m \rangle$ of new variables as follows:*
 – *Define the function $F_1 : X \to \wp(M) \setminus \{\emptyset\}$ as $F_1(t) = \{\gamma(t)(x_1)\}$.*
 – *For each $2 \le i \le m$, define the function $F_i : X[F_1/w_1, \dots, F_{i-1}/w_{i-1}] \to \wp(M) \setminus \{\emptyset\}$ as $F_i(t) = \{\gamma(t)(x_i)\}$.*
 We call $\vec{F} = \langle F_1, \dots, F_m \rangle$ the sequence of simulating functions for $\gamma[X] \upharpoonright \vec{x}$ *on $\vec{w}$. Let $Y = X[\vec{F}/\vec{w}]$ (see Fig. 1 in Appendix II for an example of such a team with a constant choice function $\gamma(t) = s$ for all $t \in X$, or Fig. 2(b) for another example with an obvious choice function). Then, $t(\vec{w}) = \gamma(t)(\vec{x})$ for all $t \in Y$ and $M \models_Y \mathsf{inc}(\vec{w}; \vec{x})$.*
 For a sequence $\Gamma = \langle \gamma_1, \dots, \gamma_k \rangle$ of choice functions $\gamma_i : X \to X$,
 – *let $\vec{F_1}$ be the sequence of simulating functions for $\gamma_1[X] \upharpoonright \vec{x}$ on $\overrightarrow{w_1}$,*
 – *and for each $2 \le i \le k$, let $\vec{F_i}$ be the sequence of simulating functions for $\gamma_i[X[\overrightarrow{F_1}/\overrightarrow{w_1}, \dots, \overrightarrow{F_{i-1}}/\overrightarrow{w_{i-1}}]] \upharpoonright \vec{x}$ on $\vec{w_i}$.*
 We call $\vec{F_1}, \dots, \vec{F_k}$ the group of simulating functions for $\Gamma[X] \upharpoonright \vec{x}$ *on $\overrightarrow{w_1}, \dots, \overrightarrow{w_k}$, and the team $Y = X[\vec{F_1}/\overrightarrow{w_1}, \dots, \vec{F_k}/\overrightarrow{w_k}]$ its* associated team *(see Fig. 2 in Appendix II for examples of such teams). Then, $M \models_Y \mathsf{inc}(\overrightarrow{w_1}, \dots, \overrightarrow{w_k}; \vec{x})$.*

(ii) *Define inductively functions $F_1, \dots, F_m$ to duplicate assignments in X restricted to $\vec{x}$ on a sequence $\vec{w} = \langle w_1, \dots, w_m \rangle$ of new variables as follows:*
 – *Define the function $F_1 : X \to \wp(M) \setminus \{\emptyset\}$ as $F_1(t) = \{s(x_1) \mid s \in X\}$.*
 – *For each $2 \le i \le m$, define the function $F_i : X[F_1/w_1, \dots, F_{i-1}/w_{i-1}] \to \wp(M) \setminus \{\emptyset\}$ as*

$$F_i(t) = \{s(x_i) \mid s \in X \text{ and } s \upharpoonright \{x_1, \dots, x_{i-1}\} = t \upharpoonright \{w_1, \dots, w_{i-1}\}\}.$$

 We call $\vec{F} = \langle F_1, \dots, F_m \rangle$ the sequence of duplicating functions for $X \upharpoonright \vec{x}$ *on $\vec{w}$. (see Fig. 3(b) in Appendix II for an example of a team $X[\vec{F}/\vec{w}]$).*
 For a team X,
 – *let $\vec{F_1}$ be the sequence of duplicating functions for $X \upharpoonright \vec{x}$ on $\overrightarrow{w_1}$,*
 – *and for each $i = 2, \dots, k$, let $\vec{F_i}$ be the sequence of duplicating functions for $X[\overrightarrow{F_1}/\overrightarrow{w_1}, \dots, \overrightarrow{F_{i-1}}/\overrightarrow{w_{i-1}}] \upharpoonright \vec{x}$ on $\vec{w_i}$.*
 We call $\vec{F_1}, \dots, \vec{F_k}$ the group of duplicating functions for $X \upharpoonright \vec{x}$ *on $\overrightarrow{w_1}, \dots, \overrightarrow{w_k}$. and the team $Y = X[\vec{F_1}/\overrightarrow{w_1}, \dots, \vec{F_k}/\overrightarrow{w_k}]$ its* associated team *(see Fig. 3 in Appendix II for examples of such teams). Then, $M \models_Y \mathsf{pro}(\vec{y}; \vec{x}; \overrightarrow{w_1}, \dots, \overrightarrow{w_k})$ for any sequence $\vec{y}$ of variables in $\mathrm{dom}(X)$ that has no variable in common with $\vec{x}$ and $\overrightarrow{w_1}, \dots, \overrightarrow{w_k}$, and for any $t \in Y$, there exist $s_1, \dots, s_k \in X$ such that $s_1(\vec{x}) = t(\overrightarrow{w_1}), \dots, s_k(\vec{x}) = t(\overrightarrow{w_k})$.*

Proof. We only give the detailed proof for $M \models_Y \mathsf{pro}(\vec{y};\vec{x};\overrightarrow{w_1},\ldots,\overrightarrow{w_k})$ in the item (ii), i.e.,

$$M \models_Y \bigwedge_{i=1}^{k}(\vec{x} \subseteq \overrightarrow{w_i}) \wedge \left(\bigwedge_{i=1}^{k}(\langle\overrightarrow{w_j} \mid j \neq i\rangle \perp \overrightarrow{w_i})\right) \wedge (\vec{y} \perp \overrightarrow{w_1}\ldots\overrightarrow{w_k}) \tag{7}$$

To show that Y satisfies the first conjunct of the formula in (7), it suffices to show that $M \models_{Y_i} \vec{x} \subseteq \overrightarrow{w_i}$ for each $1 \leq i \leq k$ and $Y_i = X[\vec{F_1}/\overrightarrow{w_1},\ldots,\vec{F_i}/\overrightarrow{w_i}]$.

For any $t \in Y_i$, by the definition of $Y_i = Y_{i-1}[\vec{F_i}/\overrightarrow{w_i}]$, there exists $s \in X$ such that $s(\vec{x}) = t(\vec{x})$, and

$$t' = s \cup \{(w_{i,1}, s(x_1)),\ldots,(w_{i,m}, s(x_m))\} \in Y_{i-1}[F_{i,1}/w_{i,1},\ldots,F_{i,m}/w_{i,m}]$$

Thus, $t'(\overrightarrow{w_i}) = s(\vec{x}) = t(\vec{x})$, as required.

To prove that Y satisfies the second and the third conjuncts of the formula in (7), we prove a more general property that $M \models_Y \overrightarrow{w_{i_1}}\ldots\overrightarrow{w_{i_a}} \perp \overrightarrow{w_{j_1}}\ldots\overrightarrow{w_{j_b}}v_1\ldots v_c$ holds for any disjoint subsequences $\overrightarrow{w_{i_1}}\ldots\overrightarrow{w_{i_a}}$ and $\overrightarrow{w_{j_1}}\ldots\overrightarrow{w_{j_b}}$ of $\overrightarrow{w_1}\ldots\overrightarrow{w_k}$ and any variables $v_1\ldots v_c \in \mathrm{dom}(X)$. Assume that $\{\overrightarrow{w_{i_1}}\ldots\overrightarrow{w_{i_a}},\overrightarrow{w_{j_1}}\ldots\overrightarrow{w_{j_b}}\} = \{\overrightarrow{w_{l_1}}\ldots\overrightarrow{w_{l_d}}\}$ with $l_1 < \cdots < l_d$.

Let $s, s' \in Y$ be arbitrary. We need to find an $s'' \in Y$ such that $s''(\overrightarrow{w_{i_1}}\ldots\overrightarrow{w_{i_a}}) = s(\overrightarrow{w_{i_1}}\ldots\overrightarrow{w_{i_a}})$ and $s''(\overrightarrow{w_{j_1}}\ldots\overrightarrow{w_{j_b}}v_1\ldots v_c) = s'(\overrightarrow{w_{j_1}}\ldots\overrightarrow{w_{j_b}}v_1\ldots v_c)$. Let f be a function satisfying

$$f(\overrightarrow{w_{l_\xi}}) = \begin{cases} s(\overrightarrow{w_{l_\xi}}) & \text{if } l_\xi \in \{i_1,\ldots,i_a\} \\ s'(\overrightarrow{w_{l_\xi}}) & \text{if } l_\xi \in \{j_1,\ldots,j_b\} \end{cases}$$

There exists $s_1 \in X$ such that $s_1(\vec{x}) = f(\overrightarrow{w_{l_1}})$. Put $Y_{l_1-1} = X[\vec{F_1}/\overrightarrow{w_1},\ldots,\overrightarrow{F_{l_1-1}}/\overrightarrow{w_{l_1-1}}]$ and $t = s' \upharpoonright \mathrm{dom}(Y_{l_1-1})$. By the construction,

$$t_{l_1} = t \cup \{(w_{l_1,1}, s_1(x_1)),\ldots,(w_{l_1,m}, s_1(x_m))\} \in Y_{l_1-1}[\overrightarrow{F_{l_1}}/\overrightarrow{w_{l_1}}] = Y_{l_1}.$$

Thus

$$t_{l_1}(\overrightarrow{w_{l_1}}) = s_1(\vec{x}) = f(\overrightarrow{w_{l_1}}) \text{ and } t_{l_1}(\vec{v}) = t(\vec{v}) = s'(\vec{v}).$$

Repeat the same argument for $f(\overrightarrow{w_{l_2}}),\ldots,f(\overrightarrow{w_{l_d}})$, we can find $t_{l_d} \in Y_{l_d}$ such that

$$t_{l_d}(\overrightarrow{w_{i_1}}\ldots\overrightarrow{w_{i_a}}) = s(\overrightarrow{w_{i_1}}\ldots\overrightarrow{w_{i_a}}) \text{ and } t_{l_d}(\overrightarrow{w_{j_1}}\ldots\overrightarrow{w_{j_b}}v_1\ldots v_c) = s'(\overrightarrow{w_{j_1}}\ldots\overrightarrow{w_{j_b}}v_1\ldots v_c).$$

Finally, by the construction of Y, there exists $s'' \in Y$ such that $s'' \upharpoonright \mathrm{dom}(Y_{l_d}) = t_{l_d}$. Hence, s'' is the desired assignment. □

References

1. Armstrong, W.W.: Dependency structures of data base relationships. In: IFIP Congress, pp. 580–583 (1974)
2. Arrow, K.J.: Social Choice and Individual Values. Yale University Press, New Haven (1951)

3. Chagrova, L.A.: An undecidable problem in correspondence theory. J. Symbol. Logic **56**, 1261–1272 (1991)
4. Galliani, P.: The dynamics of imperfect information. Ph.D. thesis, University of Amsterdam (2012)
5. Galliani, P.: Inclusion and exclusion in team semantics: on some logics of imperfect information. Ann. Pure Appl. Logic **163**(1), 68–84 (2012)
6. Galliani, P.: On strongly first-order dependencies (2014). CoRR abs/1403.3698
7. Galliani, P.: Upwards closed dependencies in team semantics. J. Inf. Comput. **24**(C), 124–135 (2015)
8. Geiger, D., Paz, A., Pearl, J.: Axioms and algorithms for inferences involving probabilistic independence. Inf. Comput. **91**(1), 128–141 (1991)
9. Grädel, E., Väänänen, J.: Dependence and independence. Stud. Logica. **101**(2), 399–410 (2013)
10. Hannula, M.: Axiomatizing first-order consequences in independence logic. Ann. Pure Appl. Logic **166**(1), 61–91 (2015)
11. Henkin, L.: Some remarks on infinitely long formulas. In: Infinitistic Methods, Proceedings Symposium Foundations of Mathematics, Warsaw, pp. 167–183. Pergamon (1961)
12. Hintikka, J.: The Principles of Mathematics Revisited. Cambridge University Press, Cambridge (1998)
13. Hodges, W.: Compositional semantics for a language of imperfect information. Logic J. IGPL **5**, 539–563 (1997)
14. Hodges, W.: Some strange quantifiers. In: Mycielski, J., Rozenberg, G., Salomaa, A. (eds.) Structures in Logic and Computer Science. LNCS, vol. 1261, pp. 51–65. Springer, Heidelberg (1997)
15. Kontinen, J.: On natural deduction in dependence logic. In: Villaveces, A., Roman Kossak, J.K., Hirvonen, Å. (eds.) Logic Without Borders,: Essays on Set Theory, Model Theory, Philosophical Logic and Philosophy of Mathematics, pp. 297–304. De Gruyter (2015)
16. Kontinen, J., Müller, J., Schnoor, H., Vollmer, H.: Modal independence logic. In: Advances in Modal Logic, vol. 10, pp. 353–372. College Publications (2014)
17. Kontinen, J., Nurmi, V.: Team logic and second-order logic. Fundamenta Informaticae **106**, 259–272 (2011)
18. Kontinen, J., Väänänen, J.: On definability in dependence logic. J. Logic Lang. Inf. **18**(3), 317–332 (2009). (Erratum: The Same Journal **20**(1), 133–134 (2011))
19. Kontinen, J., Väänänen, J.: A remark on negation in dependence logic. Notre Dame J. Formal Logic **52**(1), 55–65 (2011)
20. Kontinen, J., Väänänen, J.: Axiomatizing first-order consequences in dependence logic. Ann. Pure Appl. Logic **164**, 11 (2013)
21. Kuusisto, A.: A double team semantics for generalized quantifiers. J. Logic Lang. Inform. **24**(2), 149–191 (2015)
22. Pacuit, E., Yang, F.: Dependence and independence in social choice: arrow's theorem. In: Abramsky, H.V.S., Kontinen, J., Väänänen, J. (eds.) Dependence Logic: Theory and Application. Progress in Computer Science and Applied Logic, pp. 227–251. Birkhauser (2016)
23. Väänänen, J.: Dependence Logic: A New Approach to Independence Friendly Logic. Cambridge University Press, Cambridge (2007)

Anaphors and Quantifiers

R. Zuber(✉)

CNRS, Laboratoire de Linguistique Formelle, Paris, France
Richard.Zuber@linguist.univ-paris-diderot.fr

Abstract. Various properties of functions denoted by anaphors and anaphoric determiners are studied in this paper. These properties indicate similarities (conservativity, intersectivity) and differences (predicate invariance, anaphoric conservativity, anaphoric intersectivity) between anaphoric functions and quantifiers and show that anaphors essentially extend the expressive power of natural languages.

1 Introduction

Anaphors belong to expressions which can play the role of verbal arguments like (ordinary) noun phrases (NPs) and whose referential meaning depends on the meaning of other expressions called *antecedents of anaphors*. The class of anaphors we will consider is represented by reflexives and reciprocals. They cannot occur in the subject position of sentences. The case we will consider is when they occur on the direct object position and the subject NPs of sentences in which they occur are their antecedents.

A typical example of a reflexive is the (anaphoric) pronoun like *himself* and a typical example of a reciprocal is the reciprocal pronoun *each other*. Other examples of reflexives and reciprocals are given by complex expressions containing *himself* or *each other*. Such complex examples can in particular be Boolean compounds of anaphoric pronouns with anaphoric or non-anaphoric noun phrases. For instance *himself but not most students* is such a reflexive and *each other and ten philosophers* is such a reciprocal.

Finally, there are also complex reflexives and reciprocals which are not Boolean compounds. Recall that a large class NPs is obtained by the application of a determiner to a common noun (CN). Thus from determiners like *all...but ten, most but not all, at least two* and the common noun *logician(s)* one can obtain the complex NPs like *all logicians but ten, most, but not all logicians, at least two logicians*. Interestingly, there is also a class of complex reflexives and reciprocals which have a similar structure. Thus we have reflexive (anaphoric) determiners (RefDets) like for instance *every... except herself* and *most,..., including Socrates and himself* and reciprocal determiners (RecDets) like *no... except each other, most..., including each other* which can apply to a CN and give complex reflexives and reciprocals like *no teacher, except herself* or *every logician except each other* (as in *Dan and Leo admire every logician excpt each other*).

J. Väänänen et al. (Eds.): WoLLIC 2016, LNCS 9803, pp. 432–445, 2016.
DOI: 10.1007/978-3-662-52921-8_26

We will assume that reflexives and reciprocals belong to the class of anaphoric NPs (ANPs). RefDets and RecDets, which can form ANPs when applied to CNs, belong to the class of anaphoric determiners (ADets). ANPs themselves belong to the class of generalised NPs (GNPs) that is expressions which can play the role of nominal arguments of transitive verbs.

A general form of sentences in which ANPs, RefDets and RecDets occur is given in (1), where TVP is a transitive verb phrase which denotes a binary relation and ANP can be of the form RefDet(CN) or RecDet(CN):

(1) $NP\ TVP\ ANP$

I will be interested in logical properties of functions denoted reflexive and reciprocal ANPs, by RefDets and by RecDets. These functions, will be called *anaphoric functions*. The properties we will discuss indicate formal similarities and differences between quantifiers denotes by NPs and anaphoric functions denoted by ANPs. In the same way we will indicate formal differences between quantifiers denoted by "ordinary" determiners (those forming "ordinary" NPs with a CN) and anaphoric functions denoted by ADets

To see informally one logical difference between an ANP and an ordinary NP consider the following examples:

(2) a. Leo and Lea hug each other.
 b. Bill and Sue hug each other.

(3) Leo, Lea, Bill and Sue hug each other.

Clearly (2a) in conjunction with (2b) does not entail (3). However, if we replace *each other* by an ordinary NP, the entailment holds. Some differences of logical nature, at informal level, between reflexives and ordinary NPs on object position are given in Keenan 2007 and Zuber 2010a.

Anaphors are often discussed in a more general setting: anaphoric interpretation is associated with an "ordinary" pronoun when its antecedent occurs in another sentence or another clause than the one in which the pronoun occurs. In particular medieval scholars discussed anaphoric use of pronouns in the so called *donkey sentences* as recalled and introduced into modern discussion by Geach (1968, p. 117). I will not discuss here this possibility of the anaphoric use of pronouns (for historical remarks concerning this problem see Egli 2000).

Geach (1968, p, 132 f) also indicates various logical peculiarities of the pronoun *himself* and Keenan (2007) makes some of them precise, taking into account the fact that *himself* can be a syntactic part of more complex reflexives. Logical properties of *each other* have been often discussed (Dalrymple *et al.* 1998, Peters and Westerståhl 2006, Sabato and Winter 2012, a.o.) but only in its syntactically simple form. As far as I know, ADets have not been subject to logical analysis and RecDets probably have not been even distinguished.

In the next section we recall some basic notions from the generalised quantifier theory and, more importantly, we show how they can be extended so that they apply to anaphoric functions. In Sect. 3, after providing some simple examples of anaphoric functions, we indicate a series of properties specific to anaphoric functions. Section 4 deals with a specific class of type $\langle 2\rangle$ quantifiers (that is sets of binary relations) and indicates some conditions for their representability with the help of anaphoric functions.

Since, technically, results presented in this paper are simple, proofs are omitted in most cases.

2 Formal Preliminaries

We will consider binary relations and functions over a universe E, assumed to be finite throughout this paper. $D(R)$ denotes the domain of R. The relation I is the identity relation: $I = \{\langle x, y\rangle : x = y\}$. If R is a binary relation and X a set then $R/X = R \cap (X \times X)$. The binary relation R^S is the greatest symmetric relation included in R, that is $R^S = R \cap R^{-1}$ and $R^{S-} = R^S \cap I'$. If R is an irreflexive symmetric relation (i.e. $R \cap R^{-1} \cap I = \emptyset$) then $\Pi(R)$ is the least fine partition of R such that every of its blocks is of the form $(A \times A) \cap I'$. A partition is 1. *trivial* iff it contains only one block. Observe that if R is an irreflexive symmetric relation and $\Pi(R)$ is not trivial than every block of $\Pi(R)$ contains at least two elements.

If a function takes only a binary relation as argument, its type is noted $\langle 2 : \tau\rangle$, where τ is the type of the output; if a function takes a set and a binary relation as arguments, its type is noted $\langle 1, 2 : \tau\rangle$. If $\tau = 1$ then the output of the function is a set of individuals and thus its type is $\langle 2 : 1\rangle$ or $\langle 1, 2 : 1\rangle$. The function $SELF$, denoted by the reflexive *himself* and defined as $SELF(R) = \{x : \langle x, x\rangle \in R\}$, is of type $\langle 2 : 1\rangle$ and the function denoted by the anaphoric determiner *every...but himself* is of type $\langle 1, 2 : 1\rangle$. We will consider here also the case when τ corresponds to a set of type $\langle 1\rangle$ quantifiers and thus τ equals, in Montagovian notation, $\langle\langle\langle e, t\rangle t\rangle t\rangle$. The type of such functions will be noted either $\langle 2 : \langle 1\rangle\rangle$ - functions from binary relations to sets of type $\langle 1\rangle$ quantifiers or $\langle 1, 2 : \langle 1\rangle\rangle$ - functions from sets and binary relations to sets of type $\langle 1\rangle$ quantifiers.

Basic type $\langle 1\rangle$ quantifiers are functions from sets to truth-values. In this case they are denotations of subject NPs. However, NPs can also occur in the direct object position and in this case their denotations do not take sets (denotations of VPs) as arguments but denotations of TVPs (relations) as arguments. To account for this eventuality the domain of application of basic type $\langle 1\rangle$ quantifiers is extended in the way that it contains in addition the set of binary relations. When a quantifier Q acts as a "direct object" we get its *accusative case extension* Q_{acc} (Keenan and Westerstahl 1997):

Definition 1. *For each type $\langle 1\rangle$ quantifier Q, $Q_{acc}R = \{a : Q(aR) = 1\}$, where $aR = \{y : \langle a, y\rangle \in R\}$.*

A type $\langle 1\rangle$ quantifier Q is *positive*, $Q \in POS$, iff $\emptyset \notin Q$; Q is *natural* iff either $Q \in POS$ and $E \in Q$ or $Q \notin POS$ and $E \notin Q$; Q is plural, $Q \in PL$, iff if $X \in Q$ then $|X| \geq 2$. Q_A is the atomic quantifier true of just A.

A special class of type $\langle 1\rangle$ quantifiers is formed by *individuals*: I_a is an individual (generated by $a \in E$) iff $I_a = \{X : a \in X\}$. They are denotations of proper names. More generally, $Ft(A)$, the *(principal) filter generated by the set.* A, is defined as $Ft(A) = \{X : X \subseteq E \wedge A \subseteq X\}$. NPs of the form *Every CN* denote principal filters generated by the denotation of *CN*. Meets of two principal filters are principal filters: $Ft(A) \cap Ft(B) = Ft(A \cup B)$.

We will use also the property of *living on* (cf. Barwise and Cooper 1981). The basic type $\langle 1\rangle$ quantifier lives on a set A (where $A \subseteq E$) iff for all $X \subseteq E$, $Q(X) = Q(X \cap A)$. Q *weakly* lives on A iff if $X \in Q$ then $X \cap A \in Q$. We extend the notion of living on to the type $\langle 2:1\rangle$ functions. Thus a type $\langle 2:1\rangle$ function F lives on the relation S iff $F(R) = F(R \cap S)$ for any binary relation R. It is easy to see that Q lives on A iff Q_{acc} lives on $E \times A$.

If E is finite then there is always a smallest set on which a quantifier Q lives. If A is a set on which Q lives we will write $Li(Q, A)$ and the smallest set on which Q lives will be noted $SLi(Q)$. A related notion is the notion of a witness set of the quantifier Q, relative to the set A on which Q lives:

Definition 2. $W \in Wt_Q(A)$ *iff* $W \in Q \wedge W \subseteq A \wedge Li(Q, A)$.

Observe that any principal filter lives on the set by which it is generated, and, moreover, this set is its witness set. Atomic quantifiers live on the universe E only and weakly live on their unique elements.

"Ordinary" determiners denote functions from sets to type $\langle 1\rangle$ quantifiers. They are thus type $\langle 1,1\rangle$ quantifiers.

Accusative extensions of type $\langle 1\rangle$ quantifiers are specific type $\langle 2:1\rangle$ functions. They satisfy the invariance condition called *accusative extension condition* **EC** (Keenan and Westerstahl 1997):

Definition 3. *A type $\langle 2:1\rangle$ function F satisfies **EC** iff for R and S binary relations, and $a, b \in E$, if $aR = bS$ then $a \in F(R)$ iff $b \in F(S)$.*

Observe that if F satisfies **EC** then for all $X \subseteq E$ either $F(E \times X) = \emptyset$ or $F(E \times X) = E$. Given that $SELF(E \times A) = A$ the function $SELF$ does not satisfy **EC**. The function $SELF$ satisfies the following weaker predicate invariance condition **PI** (Keenan 2007):

Definition 4. *A type $\langle 2:1\rangle$ function F is predicate invariant (**PI**) iff for R and S binary relations, and $a \in E$, if $aR = aS$ then $a \in F(R)$ iff $a \in F(S)$.*

This condition is also satisfied for instance by the function $ONLY\text{-}SELF$ defined as follows: $ONLY\text{-}SELF(R) = \{x : xR = \{x\}\}$. Given that $ONLY\text{-}SELF(E \times \{a\}) = \{a\}$, the function $ONLY\text{-}SELF$ does not satisfy **EC**.

The following proposition indicates another way to define **PI**:

Proposition 1. *A type $\langle 2:1\rangle$ function F is predicate invariant iff for any $x \in E$ and any binary relation R, $x \in F(R)$ iff $x \in F(\{x\} \times xR)$.*

We will use the following property of **PI** functions:

Proposition 2. *Functions satisfying **PI** form an atomic Boolean algebra. Its atoms are functions $h_{a,A}$, (for $a \in E$ and $A \subseteq E$),defined as follows: $h_{a,A}(R) = \{a\}$ if $aR = A$ and $= \emptyset$ otherwise.*

The conditions **EC** and **PI** concern type $\langle 2:1 \rangle$ functions, considered here as being denoted by full ANPs. We also need a similar definition for type $\langle 1,2:1 \rangle$ functions denoted by RefDets. Thus:

Definition 5. *A type $\langle 1,2:1 \rangle$ function F satisfies **D1EC** iff for R and S binary relations, $X \subseteq E$ and $a,b \in E$, if $aR \cap X = bS \cap X$ then $a \in F(X,R)$ iff $b \in F(X,S)$.*

Observe that if $F(X,R)$ satisfies **D1EC** then for all $X, A \subseteq E$ either $F(X, E \times A) = \emptyset$ or $F(X, E \times A) = E$. Denotations of ordinary determiners occurring in NPs which take direct object position satisfy **D1EC**. More precisely, if D is a type $\langle 1,1 \rangle$ conservative quantifier, then the function $F(X,R) = D(X)_{acc}(R)$ satisfies **D1EC**: in this case $F(X,R) = \{y : D(X)(yR \cap X) = 1\}$ and $F(X,S) = \{y : D(X)(yS \cap X) = 1\}$. So if $aR \cap X = bS \cap X$ then $a \in F(X,R)$ iff $b \in F(X,S)$.

Functions denoted by properly anaphoric determiners (ones which form ANPs denoting functions satisfying **PI** but failing **EC**) do not satisfy **D1EC**. For instance the function $F(X,R) = \{y : X \cap yR = \{y\}\}$ denoted by the anaphoric determiner *no... except himself/herself* does not satisfy **D1EC**. To see this observe that for $A = \{a\}$ and X such that $a \in X$ one has $F(X, E \times A) = \{a\}$ and thus $F(X, E \times X) \neq \emptyset$ and $F(X, E \times X) \neq E$.

Type $\langle 1,2:1 \rangle$ functions denoted by anaphoric determiners do not satisfy **D1EC**. They satisfy the following weaker condition (Zuber 2010b):

Definition 6. *A type $\langle 1,2:1 \rangle$ function F satisfies **D1PI** (*predicate invariance for unary determiners*) iff for R and S binary relations $X \subseteq E$, and $x \in E$, if $xR \cap X = xS \cap X$ then $x \in F(X,R)$ iff $x \in F(X,S)$.*

The following proposition indicates an equivalent way to define **D1PI** :

Proposition 3. *A type $\langle 1,2:1 \rangle$ function F satisfies **D1PI** iff for any $x \in E$, $X \subseteq E$, any binary relation R one has $x \in F(X,R)$ iff $x \in F(X, (\{x\} \times X) \cap R)$*

The above invariance principles concern type $\langle 2:1 \rangle$ and type $\langle 1,2:1 \rangle$ functions. We need to present similar "higher order" invariance principles for type $\langle 2:\langle 1 \rangle \rangle$ and type $\langle 1,2:\langle 1 \rangle \rangle$ functions that is functions having as output a set of type $\langle 1 \rangle$ quantifiers. This is necessary because, as we will see, some type $\langle 1,2:\langle 1 \rangle \rangle$ functions are denotations of RecDets.

One can distinguish various kinds of type $\langle 2:\langle 1 \rangle \rangle$ and type $\langle 1,2:\langle 1 \rangle \rangle$ functions. Observe first that any type $\langle 2:1 \rangle$ function whose output is denoted by a VP can be lifted to a type $\langle 2:\langle 1 \rangle \rangle$ function. The accusative extension of a type $\langle 1 \rangle$ quantifier Q can be lifted to type $\langle 2:\langle 1 \rangle \rangle$ function in the way indicated in (4). Such functions will be called *accusative lifts*. More generally, if F is a type $\langle 2:1 \rangle$ function, its lift F^L, a type $\langle 2:\langle 1 \rangle \rangle$ function, is defined in (5):

(4) $Q^L_{acc}(R) = \{Z : Z(Q_{acc}(R)) = 1\}$.
(5) $F^L(R) = \{Z : Z(F(R)) = 1\}$.

The variable Z above runs over the set of type $\langle 1 \rangle$ quantifiers.

For type $\langle 2 : \langle 1 \rangle\rangle$ functions which are lifts of type $\langle 2 : 1\rangle$ functions we have:

Proposition 4. *If a type $\langle 2 : \langle 1 \rangle\rangle$ function F is a lift of a type $\langle 2 : 1\rangle$ function then for any type $\langle 1 \rangle$ quantifiers Q_1 and Q_2 and any binary relation R, if $Q_1 \in F(R)$ and $Q_2 \in F(R)$ then $(Q_1 \wedge Q_2) \in F(R)$*

For type $\langle 2 : \langle 1 \rangle\rangle$ functions which are accusative lifts we have:

Proposition 5. *Let F be a type $\langle 2 : \langle 1 \rangle\rangle$ function which is an accusative lift. Then for any $A, B \subseteq E$, any binary relation R, $Ft(A) \in F(R)$ and $Ft(B) \in F(R)$ iff $Ft(A \cup B) \in F(R)$.*

Accusative lifts satisfy the following higher order extension condition **HEC** (Zuber 2014):

Definition 7. *A type $\langle 2 : \langle 1 \rangle\rangle$ function F satisfies **HEC** (higher order extension condition) iff for any natural type $\langle 1 \rangle$ quantifiers Q_1 and Q_2 with the same polarity, any $A, B \subseteq E$, any binary relations R, S, if $Li(Q_1, A)$, $Li(Q_2, B)$ and $\forall_{a \in A} \forall_{b \in B}(aR = bS)$ then $Q_1 \in F(R)$ iff $Q_2 \in F(S)$.*

Functions satisfying **HEC** have the following property::

Proposition 6. *Let F satisfies **HEC** and let $R = E \times C$, for $C \subseteq E$ arbitrary. Then for any $X \subseteq E$ either $Ft(X) \in F(R)$ or for any X, $Ft(X) \notin F(R)$*

Thus a function satisfying **HEC** condition and whose argument is the cross-product relation of the form $E \times A$, has in its output either all principal filters or no principal filter. We will see that the function denoted by the ANP *each other* does not satisfy **HEC**.

It follows from Proposition 6 that lifts of genuine predicate invariant functions do not satisfy **HEC**. They satisfy the following weaker condition (Zuber 2014):

Definition 8. *A type $\langle 2 : \langle 1 \rangle\rangle$ function F satisfies **HPI** (higher order predicate invariance) iff for type $\langle 1 \rangle$ quantifier Q, any $A \subseteq E$, any binary relations R, S, if $Li(Q, A)$ and $\forall_{a \in A}(aR = aS)$ then $Q \in F(R)$ iff $Q \in F(S)$.*

An equivalent way to define **HPI** is given in Proposition 7:

Proposition 7. *Function F satisfies **HPI** iff if $Li(Q, A)$ then $Q \in F(R)$ iff $Q \in F((A \times E) \cap R)$*

The above definitions of **HEC** and of **HPI** easily extend to type $\langle 1, 2 : \langle 1 \rangle\rangle$ functions, which are, as we will see, denotations of RecDets:

Definition 9. *A type $\langle 1,2 : \langle 1\rangle\rangle$ function F satisfies **D1HEC** (higher order extension condition for unary dets) iff for any natural type $\langle 1\rangle$ quantifiers Q_1 and Q_2 with the same polarity, any $A, B \subseteq E$, any binary relations R, S, if $Li(Q_1, A)$, $Li(Q_2, B)$ and $\forall_{a\in A}\forall_{b\in B}(aR \cap X = bS \cap X)$ then $Q_1 \in F(X, R)$ iff $Q_2 \in F(X, S)$.*

Definition 10. *A type $\langle 1,2 : \langle 1\rangle\rangle$ function F satisfies **D1HPI** (higher order predicate invariance for unary dets) iff for any type $\langle 1\rangle$ quantifier Q, any $A \subseteq E$, any binary relations R, S, if $Li(Q, A)$ and $\forall_{a\in A}(aR \cap X = aS \cap X)$ then $Q \in F(X, R)$ iff $Q \in F(X, S)$.*

The condition **D1HPI** can also be characterised as in:

Proposition 8. *$F(X, R)$ satisfies **D1HPI** iff if Q lives on A then $Q \in F(X, R)$ iff $Q \in F(X, (A \times X) \cap R)$*

The second series of properties of functions we will discuss concerns conservativity. Recall first the constraint of conservativity for type $\langle 1, 1\rangle$ quantifiers:

Definition 11. *$F \in CONS$ iff $F(X, Y) = F(X, X \cap Y)$ for any $X, Y \subseteq E$*

Conservative quantifiers have two important sub-classes: intersective and co-intersective quantifiers (Keenan 1993): a type $\langle 1, 1\rangle$ quantifier F is intersective (resp. co-intersective) iff $F(X_1, Y_1) = F(X_2, Y_2)$ whenever $X_1 \cap Y_1 = X_2 \cap Y_2$ (resp. $X_1 \cap Y_1' = X_2 \cap Y_2'$).

All the above properties of quantifiers can be generalised so that they apply to type $\langle 1,2 : 1\rangle$ and type $\langle 1,2 : \langle 1\rangle\rangle$ functions (Zuber 2010a):

Definition 12. *A function F of type $\langle 1,2 : \tau\rangle$ is conservative iff $F(X, R) = F(X, (E \times X) \cap R)$*

For instance the function $F(X, R) = MOST(X)_{acc}(R)$ is conservative (where $MOST(X)(Y) = 1$ iff $|X \cap Y| > |X \cap Y'|$). In fact the type $\langle 1,2 : 1\rangle$ function $F(X, R) = D(X)_{acc}(R)$ and the type $\langle 1,2 : \langle 1\rangle\rangle$ function $F(X, R) = D(X)^L_{acc}(R)$ are conservative iff D is a conservative type $\langle 1, 1\rangle$ quantifier.

Definition 13. *A type $\langle 1,2 : \tau\rangle$ function is intersective iff $F(X_1, R_1) = F(X_2, R_2)$ whenever $(E \times X_1) \cap R_1 = (E \times X_2) \cap R_2$.*

Definition 14. *A type $\langle 1,2 : \tau\rangle$ function is co-intersective iff $F(X_1, R_1) = F(X_2, R_2)$ whenever $(E \times X_1) \cap R_1' = (E \times X_2) \cap R_2'$.*

As in the case of type $\langle 1, 1\rangle$ quantifiers it is possible to give other, equivalent, definitions of intersectivity and co-intersectivity for type $\langle 1,2 : \tau\rangle$ functions:

Proposition 9. *F is intersective iff $F(X, R) = F(E, (E \times X) \cap R)$.*

Proposition 10. *F is co-intersective iff $F(X, R) = F(E, (E \times X') \cup R)$.*

One can show that the type $\langle 1,2:1\rangle$ function $F(X,R)=D(X)_{acc}(R)$ and the type $\langle 1,2:\langle 1\rangle\rangle$ function $F(X,R)=D(X)^{L}_{acc}(R)$ are intersective (resp. co-intersective) if D is an intersective (resp. co-intersective) type $\langle 1,1\rangle$ quantifier.

One can notice that intersective and co-intersective functions are conservative. Interestingly for functions satisfying **D1PI** or **D1HPI** we have:

Proposition 11. *Any function satisfying* ***D1PI*** *or* ***D1HPI*** *is conservative*

Since the above definitions do not depend on the type τ, they apply to type $\langle 1,2:1\rangle$ and type $\langle 1.2:\langle 1\rangle\rangle$ functions.

3 Properties of Anaphoric Functions

The properties discussed in the preceding section characterise anaphoric functions "negatively". Since any function satisfying, say, **AC** also satisfies **PI** we can distinguish a class of "genuine" functions satisfying **PI** as those which satisfy **PI** but do not satisfy **AC**. We have seen that this is the case for $SELF$ and $ONLY$-$SELF$. To see that this is also the case with anaphoric functions of other types we need some examples of such functions.

For simplicity we will consider that reciprocals give rise only to full (logical) reciprocity. Thus we exclude the readings of *each other* as found for instance in *follow each other*. (cf. Dalrymple *et al.* 1998).

Let us recall briefly types of functions we are interested in. Reflexives denotes functions of type $\langle 2:1\rangle$ (like $SELF$ or $ONLY$-$SELF$) because they form VPs when applied to a TVP. RefDets denote functions of type $\langle 1,2:1\rangle$ because they form reflexives by applying to CNs. Reciprocals and RecDets differ in many respects from reflexives and RefDets respectively. Both these classes also differ from ordinary determiners and ordinary NPs. Thus, given Proposition 4 and examples in (2) and (3), functions denoted by reciprocals are not lifts of type $\langle 2:1\rangle$ functions and the conjunction *and* is not understood pointwise. Hence, to avoid the type mismatch and get the right interpretations we will consider that reciprocals *each other* denotes a type $\langle 2:\langle 1\rangle\rangle$ function (because they apply to TVPs and give a lifted VP) and RecDets denote type $\langle 1,2:\langle 1\rangle\rangle$ functions.

A class of anaphoric type $\langle 1,2:1\rangle$ functions is given by the schema in (6):

(6) $F(X,R)=D(X)_{INCL-SELF}(R)=\{y: y\in X\wedge\langle y,y\rangle\in R\wedge y\in D(X)_{acc}(R)\}$, where D is a monotone on the second argument type $\langle 1,1\rangle$ quantifier.

Replacing in (6) D by $MOST$ or TEN we obtain the anaphoric function denoted by the RefDets like *most,..., including himself* or *ten,..., including himself.*

Two other anaphoric type $\langle 1,2:1\rangle$ functions are given in (7) and (8):

(7) $F_{NO}(X,R)=\{x: X\cap xR=\{x\}\}$

(8) $F_{EVERY-BUT-\{L\}}(X,R)=\{x: X\cap xR'=\{x,L\}\}$

The RefDet *no...except himself* denotes the function in (7) and the RefDet *every... except Leo and himself*, denotes the function in (8).

To define the type $\langle 2 : \langle 1 \rangle \rangle$ function EA denoted by the reciprocal *each other* we use the partition $\Pi(R^{S-})$. Our definition is the definition "be cases" which depend on whether the partition $\Pi(R^{S-})$ is trivial or non-trivial. Thus

(9) (i) $EA(R) = \{Q : Q \in PL \wedge \neg 2(E) \subseteq Q\}$ if $R^{S-} = \emptyset$

(ii) $EA(R) = \{Q : Q \in PL \wedge Q_{D(B)} \subseteq Q\}$, if $\Pi(R^{S-})$ is trivial with B as its only block

(iii) $EA(R) = \{Q : Q \in PL \wedge \exists_B(B \in \Pi(R^{S-}) \wedge Q(D(B) = 1\} \cup \{Q : Q \in PL \wedge \exists_B(B \in \Pi(R^{S-}) \wedge Q = \neg Q_{D(B)}\}$ if $\Pi(R^{S-})$ is non-trivial.

In a similar way we can obtain type $\langle 1, 2 : \langle 1 \rangle \rangle$ anaphoric functions. In (10) we have the function denoted by the reciprocal determiner *no...except each other*:

(10) (i) $NO_{BUT-EA}(X, R) = \{Q : Q \in PL \wedge \neg TWO(E) \subseteq Q\}$ if $R^{S-} = \emptyset$

(ii) $NO_{BUT-EA}(X, R) = \{Q : Q \in PL \wedge D(B) \times D'(B) \cap R = \emptyset \wedge Q_{D(B} \subseteq Q\}$ if $\Pi(R^{S-}/X)$ is trivial with B as its only block.

(iii) $F(X, R) = \{Q : Q \in PL \wedge \exists_B(B \in \Pi(R^S/X))\exists_W(W \in Wt_Q(SLi(Q) \wedge (W \times W) \cap I') = B \wedge D(B) \times D'(B) \cap R = \emptyset\}$ if $\Pi(R^{S-}/X)$ is non-trivial.

To obtain the function $EVERY_{BUT-EA}$ denoted by *every... except each other* one can use the fact that, roughly, $EVERY$ is related to NO by the negation of the second argument: $EVERY(X, Y) = NO(X, Y')$. Thus we have:

(11) $EVERY_{BUT-EA}(X, R) = NO_{BUT-EA}(X, R')$

The functions described above are anaphoric in the sense that they satisfy predicate invariance conditions **PI**, **HPI**, **D1PI** or **D1HPI** and do not satisfy stronger conditions **EC**, **HEC**, **D1EC** or **D1HEC**. we have already seen this for $SELF$ and $ONLY$-$SELF$. This is easy to see for functions in (6), (7) and (8) because their values for $R = E \times A$ may differ from E and $\emptyset$. Similarly, using Proposition 6 and Definition 8 we show that the function EA in (9) is anaphoric (because for $R = E \times A$ the partition $\Pi(R^{S-})$ is trivial).

To show that functions denoted by RecDets do not satisfy **D1HEC** we can use Proposition 12, analogous to Proposition 6:

Proposition 12. *Let F satisfies **D1HEC** and let $R = E \times C$, for $C \subseteq E$. Then for any $A \subseteq E$ either $Ft(A) \in F(X, R)$ or for any X, $Ft(A) \notin F(X, R)$*

Using Propositions 12 and 8 one can show that functions in (10) and (11) are anaphoric.

Examples of RefDets discussed above suggest that functions they denote satisfy a constraint stronger than conservativity. Observe that anaphoric functions given in (6), (7) and (8) all have the property given in (12):

(12) $F(X, R) \subseteq X$.

Interestingly, the anaphoric condition **D1PI** and the condition given in (12) entail a specific version of conservativity, *anaphoric conservativity* (a-conservativity), proper to anaphoric determiners. It is defined as follows:

Definition 15. *A type $\langle 1, 2 : \tau\rangle$ function F is* a-conservative *iff* $F(X, R) = F(X, (X \times X) \cap R)$.

The following proposition makes clearer what a-conservativity is:

Proposition 13. *A type $\langle 1, 2 : \tau\rangle$ function F is* a-conservative *iff for any $X \subseteq E$ and any binary relations R_1 and R_2 if $(X \times X) \cap R_1 = (X \times X) \cap R_2$ then $F(X, R_1) = F(X, R_2)$.*

Any a-conservative function is conservative. Ordinary determiners in the object position in general do not denote a-conservativce functions: if D is a (conservative) type $\langle 1, 1\rangle$ quantifier, then the type $\langle 1, 2 : 1\rangle$ function $F(R, X) = D(X)_{acc}(R)$ is not a-conservative. For instance if $D = ALL$ and $R = E \times A$ then $F(X, R) = ALL(X)_{acc}(E \times A) = E$ if $X \subseteq A$ but in this case $F(X, (X \times X) \cap R) = ALL(X)_{acc}((X \times X) \cap (E \times A) = X$. Thus $F(X, R) \neq F(X, (X \times X) \cap R)$ which means that $F(X, R) = ALL(X)_{acc}(R)$ is not a-conservative (though it is conservative).

Concerning RefDets and a-conservativity we have:

Proposition 14. *A type $\langle 1, 2 : 1\rangle$ function F satisfying **D1PI** such that $F(X, R) \subseteq X$ is a-conservative.*

Thus the functions denoted by RefDets are a-conservative.

When one looks at type $\langle 1, 2 : \langle 1\rangle\rangle$ functions $F(X, R)$, denotations of non-possessive RecDets, one observes that they have the property given in (13):

(13) If $Q \in F(X, R)$, then Q weakly lives on X.

For functions denoted by RecDets satisfying the condition in (13) we have:

Proposition 15. *Any type $\langle 1, 2 : \langle 1\rangle\rangle$ conservative functions satisfying **D1HPI** and the condition in (13) is a-conservative.*

Thus functions denoted by ADets are a-conservative..

More can be said with respect to the class of functions denoted by anaphoric determiners formed from *no* or *every*. Since they are related either to "ordinary" intersective determiners (like *no... except Leo*) or to "ordinary" co-intersective determiners (like *every... except Lea*) they are provably either intersective or co-intersective (in the sense of definitions D13 and D14 respectively). The function in (10) is intersective and the function in (11) is co-intersective.

In addition, given that the functions we consider satisfy predicate invariance and condition like (12) or (13), they have a stronger property than just intersectivity or co-intersectivity: they are a-intersective or a-co-intersective in the following sense:

Definition 16. *A type* $\langle 1, 2 : \tau \rangle$ *function* F *is a-intersective iff* $F(X_1, R_1) = F(X_2, R_2)$ *whenever* $(X_1 \times X_1) \cap R_1 = (X_2 \times X_2) \cap R_2$

Definition 17. *A type* $\langle 1, 2 : \tau \rangle$ *function* F *is a-co-intersective iff* $F(X_1, R_1) = F(X_2, R_2)$ *whenever* $(X_1 \times X_1) \cap R_1' = (X_2 \times X_2) \cap R_2'$

The following proposition gives another characterisation of the a-intersectivity and a-co-intersectivity:

Proposition 16. *A type* $\langle 1, 2 : \tau \rangle$ *function* F *is a-intersective iff* $F(X, R) = F(E, (X \times X) \cap R)$

Proposition 17. *A type* $\langle 1, 2 : \tau \rangle$ *function* F *is a-co-intersective iff* $F(X, R) = F(E, ((X \times X)') \cup R)$

Functions which are a-intersective or a-co-intersective are a-conservative. The function in (7) and in (10) is a-intersective and the function in (8) and in (11) is a-cointersective.

The property of conservativity is related to the property of living on. Let F be a type $\langle 1, 2 : \tau \rangle$ function and let F_A be a type $\langle 2 : \tau \rangle$ function defined as $F_A(R) = F(A, R)$. Then, clearly, if F is a-conservative, F_A lives on $S = A \times A$. The question one can ask now is whether all anaphoric type $\langle 2 : \tau \rangle$ functions live on a (non-trivial) relation. For instance the function $SELF$ lives on the relation I and the function EA lives on the relation I'. One can observe, however, that the anaphoric type $\langle 2 : 1 \rangle$ function $F(R) = D(R)$ does not live on any non-trivial relation. This observation indicates that "linguistically natural" anaphoric type $\langle 2 : \tau \rangle$ functions (the function $F(R) = D(R)$ is not "linguistically natural") should be additionally characterised by the property of living on (a relation).

4 Predicate Invariant Reducibility

A set of binary relations is a type $\langle 2 \rangle$ quantifier. Among them one can distinguish the following sub-class (cf. Keenan 1992):

Definition 18. *A type* $\langle 2 \rangle$ *quantifier* F *is Fregean, or Frege reducible, iff there exist two type* $\langle 1 \rangle$ *quantifiers* Q *and* Q_1 *such that* $F(R) = Q_1(Q_{acc}(R))$.

A type $\langle 2 \rangle$ quantifier is non-Fregean iff it is not Frege reducible.

Various tests showing that a type $\langle 2 \rangle$ quantifier is Fregean have been established and various type $\langle 2 \rangle$ quantifiers have been shown to be non-Fregean (Keenan 1992, van Eijck 2005) with their help. In these tests essential role play cross-product binary relations, that is binary relations of the form $A \times B$. Thus Keenan 1992 proved the following theorem which can be used to show that some functions are not Fregean (see also van Eijck 2005):

Proposition 18. *(Keenan) If* F_1 *and* F_2 *are Fregean (type* $\langle 2 \rangle$*) quantifiers then* $F_1 = F_2$ *iff for all* $A, B \subseteq E$ *it holds that* $F_1(A \times B) = F_2(A \times B)$

To illustrate Proposition 18 we indicate a class of examples of non-Fregean quantifiers corresponding to the functions of the form $Q(h)$ for $Q = Ft(C)$ and $h = SELF$. First, we have (Zuber 2012):

Proposition 19. *Let $Q = Ft(C)$ for some $C \subseteq E$, $|C| \geq 2$. Then:*
(i) $Q(SELF(X \times Y)) = Q(Q_{acc}(X \times Y)$, for any $X, Y \subseteq E$
(ii) $Q(SELF((C \times C) \cap I)) \neq Q(Q_{acc}((C \times C) \cap I))$, where $I = \{\langle x, x\rangle : x \in E\}$

Given that $(C \times C) \cap I$ is not a cross-product relation, it follows from Proposition 19 that the quantifier $Ft(C)(SELF)$, for $|C| \geq 2$ is not Freagean.

We will now generalise the notion of Frege irreducible quantifiers by considering the possibility of representing a type $\langle 2\rangle$ quantifier by the composition of a type $\langle 1\rangle$ quantifier with a specific predicate invariant function, by analogy with Frege reducible quantifiers which are compositions of a type $\langle 1\rangle$ quantifier with the accusative extension of a type $\langle 1\rangle$ quantifier.

When one looks at the predicate invariant functions discussed above, one observes that they are usually positive or negative that is their value on the empty relation is the empty set. or the whole universe E. For that reason we will consider in what follows only *natural* predicate invariant functions, that is predicate invariant functions h such that $h(\emptyset \times \emptyset) = \emptyset$ or $h(\emptyset \times \emptyset) = E$. Consequently we have the following definition:

Definition 19. *A type $\langle 2\rangle$ quantifier F is **PI**-reducible (*predicate invariant reducible*) iff there exists a type $\langle 1\rangle$ quantifier Q and a natural predicate invariant function h such that $F(R) = Q(h(R))$.*

We will say that the function F is induced by Q and h specifying the Q is the first inducer of F and h its second inducer and write $F = Q(h)$.

Observe that any **PI**-reducible function F can be represented in a standard way as $Q(h)$, where h is positive. The reason is that $Q(h) = Q\neg(h')$ and either h or h' is positive. Since any accusative extension of a type $\langle 1\rangle$ positive quantifier is a positive invariant function, any Frege reducible quantifier is **PI**-reducible. Thus **PI**-reducibility is a generalisation of Frege reducibility.

In order to give a sufficient and necessary condition for a type $\langle 2\rangle$ quantifier to be **PI**-reducible we need the following definition:

Definition 20. *A positive predicate invariant function refines a type $\langle 2\rangle$ quantifier F iff for any binary relations R and S if $h(R) = h(S)$ then $F(R) = F(S)$.*

We can prove now the following sufficient and necessary condition:

Proposition 20. *A type $\langle 2\rangle$ quantifier F is **PI**-reducible iff there exists a positive predicate invariant function h which refines F.*

Proof. (i) If F is **PI**-reducible then $F = Q(h)$ for some type $\langle 1\rangle$ quantifier Q and a positive predicate invariant function h. It is easy to see that h refines F. (ii) Suppose now that there is a positive predicate invariant function h which refines F. Define a type $\langle 1\rangle$ quantifier Q_h by $Q_h(P) = 1$ iff $\exists_S F(S) = 1 \wedge h(S) = P$. Then $Q_h h(R) = 1$ iff $\exists_S F(S) = 1 \wedge h(S) = h(R)$ iff $F(R) = 1$. Thus $F = Q_h(h)$ which means that F is **PI**-reducible. □

To illustrate Proposition 20 consider the type $\langle 2\rangle$ quantifier F_a, $a \in E$, defined as $F_a(R) = 1$ iff $aR = \{a\}$. One can see that the function $h(R) = \{x : xR = \{x\}$ refines F_a. Thus F_a is **PI**-reducible. Moreover, we have $F_a = I_a(h)$.

Proposition 20 allows us also to prove:

Proposition 21. *Let F_R be an atomic type $\langle 2\rangle$ quantifier (that is F_R is true of just R). Then F_R is **PI**-reducible.*

Proof. We have to show that there exists a predicate invariant function h_R which refines F_R. We associate with the relation R two positive **PI** functions h_R^+ and h_R^- defined by unions of their atoms (cf. Proposition 2). Thus $h_R^+ = \bigcup h_{x,xR}$ for all $x \in D(R)$ and $h_R^- = \bigcup h_{x,Y}$ for all $x \notin D(R)$ and all $Y \subseteq E$, $Y \neq \emptyset$. Let $h_R = h_R^+ \cup h_R^-$. One can check that $h_R(R) = D(R)$ and for S a binary relation if $h_R(S) = D(R)$ then $S = R$. This entails that h_R refines F_R and thus F_R is **PI**-reducible. □

In fact the above proof shows how to construct the sequence $Q(h)$ equivalent to F_R: given that $h_R(R) = D(R)$ we have $F_R = Q_{D(R)}(h_R)$, for any atomic type $\langle 2\rangle$ quantifier F_R, where $Q_{D(R)}$ is the atomic type $\langle 1\rangle$ quantifier whose the only member is the set $D(R)$.

Recall (Keenan 1992) that an atomic type $\langle 2\rangle$ quantifier F_R is Frege reducible iff $R = E \times A$ for some $A \subseteq E$. Thus, according to Proposition 21 the atomic quantifier $F_{A\times B}$, for $A \neq E$, is **PI**-reducible though it is not Frege reducible.

As an illustration of Proposition 21 consider (14):

(14) Leo admires only himself and nobody else admires anybody else.

The first conjunct of (14) says that a certain object l is in a certain relation with itself only and the second conjunct says that no other object is in this relation. The result is that the type $\langle 2\rangle$ quantifier F_R expressed by (14) is atomic true just of the relation $R = \{\langle l, l\rangle\}$. So given Proposition 21 the quantifier F_R induced in (14), which is not Frege reducible, is **PI**-reducible (for$|E| \geq 2$).

5 Conclusive Remarks

Functions denoted by anaphors, more specifically by reflexives, reciprocals, and determiners forming them, have been discussed and compared with quantifiers. All these functions necessarily take a binary relation as an argument since, informally, anaphoric relations relate direct objects of transitive sentences to their subjects. Formally, such functions resemble quantifiers because they are, like quantifiers, conservative or have the property of living on. However, they are different from quantifiers because they do not satisfy the extension condition satisfied by quantifiers but only a weaker property of predicate invariance. Moreover, they display properties specific to anaphoric functions like a-conservativity (or even stronger properties of a-intersectivity and a-co-intersectivity) or the property of living on a relation which can also be related to their anaphoricity.

Informally, these properties can be considered as being inherited from the properties of their parts because, for instance, anaphoric determiners are composed of, on the one hand, quantifiers and of "simple" anaphors, on the other hand.

The results presented in this paper show that though the existence of anaphors and anaphoric determiners extends the expressive power of NLs because the anaphoric functions they denote lie outside the class of classically defined generalised quantifiers, these functions resemble quantifiers in certain important ways.

References

Barwise, J., Cooper, R.: Generalised quantifiers and natural language. Linguist. Philos. **4**, 159–219 (1981)

Dalrymple, M., et al.: Reciprocal expressions and the concept of reciprocity. Linguist. Philos. **21**, 151–210 (1998)

Egli, U.: Anaphora from Athens to Amsterdam. In: von Heusinger, K., Egli, U. (eds.) Reference and Anaphoric Relations, pp. 17–29. Kluwer, Dordrecht (2000)

van Eijck, J.: Normal forms for characteristic functions. J. Logic Comput. **15**(2), 85–98 (2005)

Geach, P.T.: Reference and Generality. Cornell University Press, Ithaca (1968)

Keenan, E.L.: Beyond the Frege boundary. Linguist. Philos. **15**, 199–221 (1992)

Keenan, E.L.: On the denotations of anaphors. Res. Lang. Comput. **5–1**, 5–17 (2007)

Keenan, E.L., Westerståhl, D.: Generalized quantifiers in linguistics and logic. In: van Benthem, J., ter Meulen, A. (eds.) Handbook of Logic and Language, pp. 837–893. Elsevier, Amsterdam (1997)

Peters, S., Westerståhl, D.: Quantifiers in Language and Logic. Oxford University Press, Oxford (2006)

Sabato, S., Winter, Y.: Relational domains and the interpretation of reciprocals. Linguist. Philos. **35**, 191–241 (2012)

Zuber, R.: Generalising conservativity. In: Dawar, A., de Queiroz, R. (eds.) WoLLIC 2010. LNCS(LNAI), vol. 6188, pp. 247–258. Springer, Heidelberg (2010a)

Zuber, R.: Semantic constraints on anaphoric determiners. Res. Lang. Comput. **8**, 255–271 (2010b)

Zuber, R.: Reflexives and non-Fregean quantifiers. In: Graf, T., et al. (eds.) Theories of Everything: in Honor of Ed Keenan, UCLA Working Papers in Linguistics 17, pp. 439–445 (2012)

Zuber, R.: Generalising predicate and argument invariance. In: Asher, N., Soloviev, S. (eds.) LACL 2014. LNCS, vol. 8535, pp. 163–176. Springer, Heidelberg (2014)

Author Index