LNCS 13923

Helle Hvid Hansen
Andre Scedrov
Ruy J.G.B. de Queiroz (Eds.)

Logic, Language, Information, and Computation

29th International Workshop, WoLLIC 2023
Halifax, NS, Canada, July 11–14, 2023
Proceedings

Springer

More information about this series at https://link.springer.com/bookseries/558

Helle Hvid Hansen · Andre Scedrov ·
Ruy J.G.B. de Queiroz
Editors

Logic, Language, Information, and Computation

29th International Workshop, WoLLIC 2023
Halifax, NS, Canada, July 11–14, 2023
Proceedings

Springer

Editors
Helle Hvid Hansen
University of Groningen
Groningen, The Netherlands

Andre Scedrov
University of Pennsylvania
Philadelphia, PA, USA

Ruy J.G.B. de Queiroz
Universidade Federal de Pernambuco
Recife, Pernambuco, Brazil

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-031-39783-7 ISBN 978-3-031-39784-4 (eBook)
https://doi.org/10.1007/978-3-031-39784-4

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

Preface

This volume contains the papers presented at the 29th Workshop on Logic, Language, Information and Computation (WoLLIC 2023), held on July 11−14, 2023 at the Department of Mathematics and Statistics of Dalhousie University, Halifax, Nova Scotia, Canada. 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 which 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.

For WoLLIC 2023 there were 43 submissions. The committee decided to accept 21 papers. Submissions and the reviewing process were handled using Easychair. Each submission received at least 3 extensive reviews. PC members had 7 weeks for the reviewing and discussion. We intentionally chose a large PC to keep the review load light, in order to keep up the high quality of reviewing that has been the standard at WoLLIC. PC members were allowed to submit. PC co-chairs did not submit. In order to safeguard the integrity of the reviewing process, PC members, including the co-chairs, had to declare conflict with a submission if they were a co-author, shared an affiliation or recent or upcoming collaboration with one of the co-authors, or if it could be otherwise perceived that the PC member would have a bias towards the decision on the submission. Using an Easychair functionality, PC members were blocked from seeing the reviews and the discussion on submissions for which they had declared a conflict.

This volume includes all the accepted papers, together with the abstracts of the invited lectures at WoLLIC 2023:

- Thomas Bolander (Technical University of Denmark, Denmark)
- Makoto Kanazawa (Hosei University, Japan)
- Michael Moortgat (Utrecht University, The Netherlands)
- Magdalena Ortiz (University of Umeå, Sweden)
- Aybüke Özgün (University of Amsterdam, The Netherlands)
- Dusko Pavlovic (University of Hawaii, USA)
- Richard Zach (University of Calgary, Canada)

Four of the invited lecturers gave a tutorial to prepare the audience for their main talk. The volume also includes the abstracts for these tutorials, which were given by:

- Michael Moortgat (Utrecht University, The Netherlands)
- Magdalena Ortiz (University of Umeå, Sweden)
- Aybüke Özgün (University of Amsterdam, The Netherlands)
- Dusko Pavlovic (University of Hawaii, USA)

The invited speakers were also given the option to submit papers related to their invited talks. This volume includes 3 invited submissions from the invited speakers who chose to submit their papers (Kanazawa, Ortiz, Pavlovic). Their submissions were reviewed by a separate reviewing process, each by 2 different reviewers from the PC.

We would like to thank all the people who contributed to making WoLLIC 2023 a success. We thank all the invited speakers and the authors for their excellent contributions. We thank the Program Committee and all additional reviewers for the work they put into reviewing the submissions. One of the additional reviewers was Line van den Berg, who tragically passed away in May 2023. We are grateful for having received one of her last services to the logic community.

We thank the Steering Committee and the Advisory Committee for their advice, and the Local Organizing Committee (especially Peter Selinger and Julien Ross) for their great support. The submission and reviewing process was faciliated by the Easychair system by Andrei Voronkov under a professional license.

We gratefully acknowledge financial support for WoLLIC 2023 from the Atlantic Association for Research in the Mathematical Sciences (AARMS) and Dalhousie University. We also would like to acknowledge the scientific sponsorship of the following organizations: Interest Group in Pure and Applied Logics (IGPL), 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) and the Brazilian Logic Society (SBL).

June 2023

Helle Hvid Hansen
Andre Scedrov
Ruy J.G.B. de Queiroz

Organization

Program Committee Chairs

Helle Hvid Hansen	University of Groningen, The Netherlands
Andre Scedrov	University of Pennsylvania, USA

General Chair

Ruy de Queiroz	Universidade Federal de Pernambuco, Brazil

Steering Committee

Samson Abramsky	University College London, UK
Agata Ciabattoni	TU Wien, Austria
Anuj Dawar	University of Cambridge, UK
Juliette Kennedy	University of Helsinki, Finland
Ulrich Kohlenbach	Technische Universität Darmstadt, Germany
Daniel Leivant	Indiana University Bloomington, USA
Leonid Libkin	University of Edinburgh, UK
Lawrence Moss	Indiana University Bloomington, USA
Luke Ong	Nanyang Technological University, Singapore
Valeria de Paiva	Topos Institute, USA
Elaine Pimentel	University College London, UK
Ruy de Queiroz	Universidade Federal de Pernambuco, Brazil
Alexandra Silva	Cornell University, USA
Renata Wassermann	Universidade de São Paulo, Brazil

Advisory Committee

Johan van Benthem	University of Amsterdam, The Netherlands, and Stanford University, USA
Joe Halpern	Cornell University, USA
Wilfrid Hodges	Queen Mary, University of London, UK
Angus Macintyre	Queen Mary, University of London, UK
Hiroakira Ono	Japan Advanced Institute of Science and Technology, Japan
Jouko Väänänen	University of Helsinki, Finland

Program Committee

Bahareh Afshari	University of Gothenburg, Sweden, and University of Amsterdam, The Netherlands
Zena Ariola	University of Oregon, USA
Adriana Balan	University Politehnica of Bucharest, Romania
Marta Bílková	Czech Academy of Sciences, Czech Republic
Ranald Clouston	Australian National University, Australia
Willem Conradie	University of the Witwatersrand, South Africa
Josée Desharnais	Laval University, Canada
David Fernández-Duque	Universitat de Barcelona, Spain
Santiago Figueira	Universidad de Buenos Aires, Argentina
Silvia Ghilezan	University of Novi Sad & Mathematical Institute SASA, Serbia
Sujata Ghosh	Indian Statistical Institute, India
Nina Gierasimczuk	Technical University of Denmark, Denmark
Helle Hvid Hansen (Co-chair)	University of Groningen, The Netherlands
Andreas Herzig	CNRS, University of Toulouse, France
Juha Kontinen	University of Helsinki, Finland
Roman Kuznets	TU Wien, Austria
Martha Lewis	University of Bristol, UK
Johannes Marti	University of Oxford, UK
George Metcalfe	University of Bern, Switzerland
Cláudia Nalon	University of Brasilia, Brazil
Carlos Olarte	Université Sorbonne Paris Nord, France
Sophie Pinchinat	University of Rennes, France
Francesca Poggiolesi	CNRS, University Paris 1 Panthéon-Sorbonne, France
Sylvain Pogodalla	Inria Nancy, France
Revantha Ramanayake	University of Groningen, The Netherlands
Mehrnoosh Sadrzadeh	University College London, UK
Andre Scedrov (Co-chair)	University of Pennsylvania, USA
Philip Scott	University of Ottawa, Canada
Viorica Sofronie-Stokkermans	University of Koblenz-Landau, Germany
Kazushige Terui	Kyoto University, Japan
Mladen Vuković	University of Zagreb, Croatia

Additional Reviewers

Sergio Abriola	Universidad de Buenos Aires, Argentina
A. Baskar	Birla Institute of Technology and Science, India
Dylan Bellier	École normale supérieure de Rennes, France
Jeremias Berg	University of Helsinki, Finland
Line van den Berg	University of Bern, Switzerland
Vedran Čačić	University of Zagreb, Croatia
David Cerna	Czech Academy of Sciences, Czech Republic

Kaustuv Chaudhuri	Inria Saclay, France
Vikraman Choudhury	University of Glasgow, UK
Diana Costa	Universidade de Lisboa, Portugal
Edwin Hamel-de le Court	Université Libre de Bruxelles, Belgium
Ugo Dal Lago	University of Bologna, Italy
Farzaneh Derakhshan	Carnegie Mellon University, USA
Huimin Dong	Sun Yat-sen University, China
Fredrik Engström	University of Gothenburg, Sweden
Pietro Galliani	Free University of Bozen-Bolzano, Italy
Ronald de Haan	University of Amsterdam, The Netherlands
Willem Heijltjes	University of Bath, UK
Kengo Hirata	Kyoto University, Japan
Marko Horvat	University of Zagreb, Croatia
Purbita Jana	Institute of Mathematical Sciences, India
Mark Jago	Nottingham University, UK
Daniil Kozhemiachenko	Institut National des Sciences Appliquées, France
Xu Li	University of Luxembourg, Luxembourg
Yao Li	Portland State University, USA
Chun-Yu Lin	Czech Academy of Sciences, Czech Republic
John Longley	University of Edinburgh, UK
Meena Mahajan	Institute of Mathematical Sciences, India
Alessio Mansutti	IMDEA Software Institute, Spain
Maria Vanina Martinez	Universidad de Buenos Aires, Argentina
Luka Mikec	University of Zagreb, Croatia
Richard Moot	CNRS, University of Montpellier, France
Andreas Niskanen	University of Helsinki, Finland
Jovana Obradović	Serbian Academy of Sciences and Arts, Serbia
Hitoshi Omori	Ruhr-Universität Bochum, Germany
Victor Pambuccian	Arizona State University, USA
Mattia Panettiere	Vrije Universiteit Amsterdam, The Netherlands
Xavier Parent	TU Wien, Austria
Romain Péchoux	Inria Loria, France
Samuel Pollard	Sandia National Laboratories, USA
Davide Emilio Quadrellaro	University of Helsinki, Finland
Wilmer Ricciotti	University of Edinburgh, UK
S. P. Suresh	Chennai Mathematical Institute, India
Yì Nicholas Wáng	Sun Yat-sen University, China
Dominik Wehr	University of Gothenburg, Sweden

Sponsoring Institutions

Scientific Sponsorship

Interest Group in Pure and Applied Logics (IGPL)
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)
Brazilian Logic Society (SBL)

Financial Support

Atlantic Association for Research in the Mathematical Sciences (AARMS)
Dalhousie University

Abstracts of Invited Talks

From Dynamic Epistemic Logic to Socially Intelligent Robots

Thomas Bolander

Technical University of Denmark, Denmark
tobo@dtu.dk

Dynamic Epistemic Logic (DEL) can be used as a formalism for agents to represent the mental states of other agents: their beliefs and knowledge, and potentially even their plans and goals. Hence, the logic can be used as a formalism to give agents a Theory of Mind allowing them to take the perspective of other agents. In my research, I have combined DEL with techniques from automated planning in order to describe a theory of what I call Epistemic Planning: planning where agents explicitly reason about the mental states of others. One of the recurring themes is implicit coordination: how to successfully achieve joint goals in decentralised multi-agent systems without prior negotiation or coordination. The talk will first motivate the importance of Theory of Mind reasoning to achieve efficient agent interaction and coordination, will then give a brief introduction to epistemic planning based on DEL, address its (computational) complexity, address issues of implicit coordination and, finally, demonstrate applications of epistemic planning in human-robot collaboration.

Learning Context-Free Grammars from Positive Data and Membership Queries

Makoto Kanazawa

Hosei University, Japan
kanazawa@hosei.ac.jp

A key difficulty in learning context-free, as opposed to regular, languages from positive data and membership queries lies in the relationship between the string sets corresponding to the nonterminals of a context-free grammar and the language generated by the grammar. In the case of a regular language, the states of a minimal DFA for the language correspond to the nonempty left quotients of the language. A left quotient of L is a language of the form$u\backslash L = \{x|ux \in L\}$. Whether a string x belongs to $u\backslash L$ can be determined by the membership query "$ux \in L$?". In the case of a context-free language L generated by a context-free grammar G, there seems to be no general recipe for deciding membership in the string set associated with a nonterminal of G using the membership oracle for L.

In this talk, I present some results of recent work (with Ryo Yoshinaka) about learning a special class of context-free grammars whose nonterminals correspond to "relativized extended regular expressions". These expressions translate into polynomial-time reductions of the membership problem for nonterminals to the membership problem for the language generated by the grammar. There is a successful learner for this class that uses these reductions to test postulated productions for adequacy.

It is an interesting problem to determine the scope of this class of context-free grammars. We have not yet found a context-free language that is not inherently ambiguous that has no grammar in this class. Another intriguing open question is whether extended regular expressions can be restricted to star-free expressions without altering the class of context-free languages that are covered.

Lambek Calculus and its Modal Extensions

Michael Moortgat

Utrecht University, The Netherlands
M.J.Moortgat@uu.nl

In this talk, I review the different uses of modalities in extensions of the Lambek calculus and the ensuing challenges for efficient Natural Language Processing.

The Syntactic Calculus, seen as a non-commutative [4] (or also non-associative [5]) precursor of Intuitionistic Linear Logic, is an early representative of substructural logic. With the revival of interest in the Syntactic Calculus came the realization that the original formulation lacked the expressivity required for realistic grammar development. The extended Lambek calculi introduced in the 1990s enrich the type language with modalities for structural control. These categorial modalities have found two distinct uses. On the one hand, they can act as *licenses* granting modally marked formulas access to structural operations that by default would not be permitted. On the other hand, modalities can be used to *block* structural rules that otherwise would be available.

Examples of modalities as licensors relate to various aspects of grammatical resource management: multiplicity, order and structure. As for multiplicity, under the control of modalities limited forms of copying can be introduced in grammar logics that overall are resource-sensitive systems. As for order and structure, modalities may be used to license changes of word order and/or constituent structure that leave the form-meaning correspondence intact. The complementary use of modalities as blocking devices provides the means to seal off phrases as impenetrable locality domains.

Reviewing early results and current work on extensions of the Lambek calculus, one finds two contrasting views on the nature of modalities. One strand of research addresses the licensing type of control taking its inspiration from the '!' exponential of Linear Logic, but introduces sub-exponential refinements providing access to packages of structural rules, see [1] for a recent representative of this approach. Under the alternative view, advocated in [6] and subsequent work, modalities come in residuated pairs (adjoints), unary variants of the residuated triples (product, left and right implications) of the core Syntactic Calculus; the blocking and licensing type of control here share the same logical rules. A reconciliation of these views is suggested by the multi-type approach of [2] who argue on semantic and proof-theoretic grounds that the linear exponential, rather than being treated as a primitive connective, has to be decomposed into a composition of adjoint operations.

The fine-grained type theory of modally enhanced Lambek calculus increases the complexity of Natural Language Processing when it comes to supertagging (assigning words the contextually appropriate type in the light of high lexical type ambiguity) and parsing (associating a string of words with a structural representation that can serve as the scaffolding for semantic interpretation). In the final part of the talk I discuss proposals of [3] that aim to tackle these problems with an integrated neurosymbolic approach.

References

1. Blaisdell, E., Kanovich, M.I., Kuznetsov, S.L., Pimentel, E., Scedrov, A.: Non-associative, non-commutative multi-modal linear logic. In: Blanchette, J., Kovács, L., Pattinson, D. (eds.) Proceedings IJCAR 2022. Lecture Notes in Computer Science, vol. 13385, pp. 449−467. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-10769-6_27
2. Greco, G., Palmigiano, A.: Linear logic properly displayed. ACM Trans. Comput. Log. **24**(2), 13:1−13:56 (2023). https://doi.org/10.1145/3570919
3. Kogkalidis, K.: Dependency as Modality, Parsing as Permutation: A Neurosymbolic Perspective on Categorial Grammars. Ph.D. thesis, Utrecht University (2023). https://doi.org/10.33540/1721
4. Lambek, J.: The mathematics of sentence structure. Am. Math. Mon. **65**, 154–170 (1958)
5. Lambek, J.: On the calculus of syntactic types. In: Jakobson, R. (ed.) Structure of Language and its Mathematical Aspects, Proceedings of Symposia in Applied Mathematics. vol. XII, pp. 166−178. American Mathematical Society (1961)
6. Moortgat, M.: Multimodal linguistic inference. J. Log. Lang. Inf. **5**(3/4), 349–385 (1996). https://doi.org/10.1007/BF00159344

A Short Introduction to SHACL for Logicians

Magdalena Ortiz

University of Umeå, Sweden
magdalena.ortiz@umu.se

The SHACL Shapes Constraint Language was recommended in 2017 by the W3C for describing constraints on web data (specifically, on RDF graphs) and validating them. At first glance, it may not seem to be a topic for logicians, but as it turns out, SHACL can be approached as a formal logic, and actually quite an interesting one. In this talk, we give a brief introduction to SHACL tailored towards logicians. We discuss how SHACL relates to some well-known modal and description logics, and frame the common uses of SHACL as familiar logic reasoning tasks. This connection allows us to infer some interesting results about SHACL. Finally, we summarise some of our recent work in the SHACL world, aiming to shed light on how ideas, results, and techniques from well-established areas of logic can advance the state of the art in this emerging field.

Beliefs Based on Conflicting and Uncertain Evidence: Connecting Dempster-Shafer Theory and the Topology of Evidence

Aybüke Özgün

University of Amsterdam, The Netherlands
A.Ozgun@uva.nl

One problem to solve in the context of information fusion, decision-making, and other artificial intelligence challenges is to compute justified beliefs based on evidence. In real-life examples, this evidence may be inconsistent, incomplete, or uncertain, making the problem of evidence fusion highly non-trivial. In this talk, I will present a new model for measuring degrees of beliefs based on possibly inconsistent, incomplete, and uncertain evidence, by combining tools from Dempster-Shafer Theory and Topological Models of Evidence. Our belief model is more general than the aforementioned approaches in two important ways: (1) it can reproduce them when appropriate constraints are imposed, and, more notably, (2) it is flexible enough to compute beliefs according to various standards that represent agents' evidential demands. The latter novelty allows us to compute an agent's (possibly) distinct degrees of belief, based on the same evidence, in situations when, e.g, the agent prioritizes avoiding false negatives and when it prioritizes avoiding false positives. Finally, I will discuss further research directions and, time permitting, report on the computational complexity of computing degrees of belief using the proposed belief model.

The main part of the talk is based on joint work with Daira Pinto Prieto and Ronald de Haan. The underlying topological formalism for evidence and belief has been developed in collaboration with Alexandru Baltag, Nick Bezhanishvili, and Sonja Smets.

From Incompleteness of Static Theories to Completeness of Dynamic Beliefs, in People and in Bots

Dusko Pavlovic

University of Hawaii, USA
dusko@hawaii.edu

Self-referential statements, referring to their own truth values, have been studied in logic ever since Epimenides. Self-fulfilling prophecies and self-defeating claims, modifying their truth values as they go, have been studied in tragedies and comedies since Sophocles and Aristophanes. In modern times, the methods for steering truth values in marketing and political campaigns have evolved so rapidly that both the logical and the dramatic traditions have been left behind in the dust. In this talk, I will try to provide a logical reconstruction of some of the methods for constructing self-confirming and self-modifying statements.

The reconstruction requires broadening the logical perspective from static deductive theories to dynamic and inductive. While the main ideas are familiar from the theory of computation, the technical prerequisites will also be discussed in the introductory tutorial.

The Epsilon Calculus in Non-classical Logics: Recent Results and Open Questions

Richard Zach

University of Calgary, Canada
rzach@ucalgary.ca

The epsilon operator [1, 3] is mainly studied in the context of classical logic. It is a term forming operator: if $A(x)$ is a formula, then $\varepsilon x\, A(x)$ is a term—intuitively, a witness for $A(x)$ if one exists, but arbitrary otherwise. Its dual $\tau x\, A(x)$ is a counterexample to $A(x)$ if one exists. Classically, it can be defined as $\varepsilon x\, \neg A(x)$. Epsilon and tau terms allow the classical quantifiers to be defined: $\exists x\, A(x)$ as $A(\varepsilon x\, A(x))$ and $\forall x\, A(x)$ as $A(\tau x\, A(x))$.

Epsilon operators are closely related to Skolem functions, and the fundamental so-called epsilon theorems to Herbrand's theorem. Recent work with Matthias Baaz [2] investigates the proof theory of $\varepsilon\tau$-calculi in superintuitionistic logics. In contrast to the classical ε-calculus, the addition of ε- and τ-operators to intuitionistic and intermediate logics is not conservative, and the epsilon theorems hold only in special cases. However, it is conservative as far as the propositional fragment is concerned.

Despite these results, the proof theory and semantics of $\varepsilon\tau$-systems on the basis of non-classical logics remains underexplored.

References

1. Avigad, J., Zach, R.: The epsilon calculus. In: Zalta, E.N. (ed.) Stanford Encyclopedia of Philosophy. Fall 2020 edn (2020). https://plato.stanford.edu/archives/fall2020/entries/epsilon-calculus/
2. Baaz, M., Zach, R.: Epsilon theorems in intermediate logics. J. Symbolic Logic **87** (2), 682–720 (2022). https://doi.org/10.1017/jsl.2021.103
3. Zach, R.: Semantics and proof theory of the epsilon calculus. In: Ghosh, S., Prasad, S. (eds.) Logic and Its Applications. ICLA 2017, pp. 27–47. No. 10119 in LNCS, Springer, Berlin (2017). https://doi.org/10.1007/978-3-662-54069-5_4

Supported by the Natural Sciences and Engineering Council of Canada.

Abstracts of Tutorials

Compositionality: Categorial Variations on a Theme

Michael Moortgat

Utrecht University, The Netherlands
M.J.Moortgat@uu.nl

In the line of work initiated by Richard Montague, natural language syntax and semantics are related by a homomorphism, a structure-preserving map that sends the sorts and operations of a syntactic source algebra to their counterparts in an algebra for composing meanings. In categorial grammar, source and target take the form of deductive systems, logics of syntactic and semantic types respectively. Natural language syntax and semantics often pose conflicting demands on compositional interpretation and different strategies for resolving these conflicts have shaped the development of the field. The tutorial, aimed at researchers with a logic/computer science background, illustrates some of the main design choices: what is the nature of the syntactic calculus - modelling surface form (Lambek) or abstract syntax (Abstract Categorial Grammar); what is the target interpretation - truth-conditional/model-theoretic (formal semantics), or vector spaces/linear maps (distributional semantics); what is the division of labour between lexical and derivational semantics?

Description Logics and Other Decidable Logics for Graph-structured Data

Magdalena Ortiz

University of Umeå, Sweden
magdalena.ortiz@umu.se

In this tutorial, we will introduce a few expressive description logics that can be used to describe graph-shaped structures, and see how these logics relate to well-established modal logics such as graded and hybrid modal logics, and variants of propositional dynamic logic (PDL). We will also summarise some computational properties of these logics, particularly the boundaries of decidability and the complexity of basic reasoning services.

Dempster-Shafer Theory and Topological Models for Evidence

Aybüke Özgün

University of Amsterdam, The Netherlands
A.Ozgun@uva.nl

In the short tutorial preceding the invited talk, I will provide a brief introduction to Dempster-Shafer theory of belief functions and topological models for evidence, and motivate the proposed framework combining the two approaches.

Prerequisites for the Talk on Incompleteness of Static Theories and Completeness of Dynamic Beliefs, in People and in Bots

Dusko Pavlovic

University of Hawaii, USA
dusko@hawaii.edu

The claim of the main talk is that combining the encodings and self-reference leading to the incompleteness results in static logics and the belief updates in dynamic logics leads to suitable completeness results. But the combined formalism needed to prove this claim may seem unfamiliar. I will use this tutorial to explain how this unfamiliar framework arises from familiar formalisms. (I will also do my best to make it possible to follow both the main talk and the tutorial independently, but the presented research is concerned with self-fulfilling and self-deceiving claims, so it is applicable to itself.)

Contents

Invited Papers

Learning Context-Free Grammars from Positive Data and Membership Queries

Makoto Kanazawa(✉)

Faculty of Science and Engineering, Hosei University, Koganei, Japan
kanazawa@hosei.ac.jp
https://makotokanazawa.ws.hosei.ac.jp

Abstract. I review some recent results on learning subclasses of the context-free grammars from positive data and membership queries. I motivate the relevant learning algorithms through comparison with a similar and arguably reasonable learning algorithm for regular languages.

Keywords: Grammatical inference · Membership queries · Context-free grammars · Extended regular closure

1 Introduction

This paper concerns algorithmic learning of context-free languages. The "learning paradigm" we use is *polynomial-time identification in the limit from positive data and membership queries*. The learner receives an infinite stream of positive examples enumerating the target language, and each time it receives a new positive example, it makes polynomially many queries to the oracle for the target language before outputting a hypothesis. The goal is to converge to a correct grammar for the target language. The availability of the membership oracle makes it possible to learn some interesting subclasses of the context-free languages that properly include the regular languages. This paper presents a variant of previously proposed learning algorithms for three such classes [5–7,9].

A key difficulty in learning context-free, as opposed to regular, languages lies in the relationship between the string sets corresponding to the nonterminals of a context-free grammar and the generated language. In the case of a regular language, states of a minimal DFA for the language correspond to its *left quotients*.[1] A left quotient of a language L is a language of the form $u\backslash L = \{x \mid ux \in L\}$, where u is some string. In order to determine whether a string x belongs to $u\backslash L$, the learner can just ask the membership oracle whether ux belongs to L. Furthermore, when L is regular, there are only finitely many left quotients of L, and this makes it possible to identify the set of left quotients of L in the limit.

In the case of a context-free grammar G, the relationship between the set of strings derived from a nonterminal A of G and the language $L = L(G)$ of G is

[1] In this paper, by a "minimal DFA" for a regular language, we mean one with no dead state.

H. H. Hansen et al. (Eds.): WoLLIC 2023, LNCS 13923, pp. 3–18, 2023.
https://doi.org/10.1007/978-3-031-39784-4_1

much less straightforward. Unless A is useless, there is a pair of terminal strings (u, v) such that $S \Rightarrow_G^* u\,A\,v$, so the set $L_G(A) = \{x \mid A \Rightarrow_G^* x\}$ must be a subset of $u\backslash L/v = \{x \mid uxv \in L\}$. (A set of this latter form is called a *quotient* of L.) In general, $L_G(A)$ may be a proper subset of $\bigcap\{u\backslash L/v \mid S \Rightarrow_G^* uAv\}$, and it is not clear whether there is anything further that can be said in general about the relationship between $L_G(A)$ and the quotients of L.

The kind of learning algorithm we look at in this paper simply assumes that the string set associated with each nonterminal of the target grammar G_* can be expressed as the result of applying certain operations to quotients of $L_* = L(G_*)$. We consider three successively larger sets of operations that may be used in these expressions: (i) the set consisting of intersection only, (ii) the set of Boolean operations, and (iii) the set consisting of Boolean and regular operations. With the choice (i), the string sets associated with the nonterminals are in the *intersection closure* of $\mathrm{Quot}(L_*) = \{u\backslash L_*/v \mid (u, v) \in \Sigma^* \times \Sigma^*\}$, the set of quotients of L_*. With (ii), they are in the *Boolean closure* of $\mathrm{Quot}(L_*)$. With (iii), they are in what we call the *extended regular closure* of $\mathrm{Quot}(L_*)$. When K is a language in one of these classes, the membership of a string x in K can be determined by making a finite number of queries of the form "$uyv \in L_*$?", where (u, v) is an element of some fixed set (depending on K) and y is a substring of x. As we will see, the fact that the membership problem for the set associated with a nonterminal reduces to the membership problem for the target language means that the "validity" of a production can be decided in the limit with the help of the oracle for the target language.

Before describing our learning algorithms for the three subclasses of the context-free languages (Sect. 3), it is perhaps instructive to look at how the class of regular languages can be learned within the same paradigm, so we start with the latter.

2 Regular Languages

Let us describe a learning algorithm that identifies an unknown regular language L_* from positive data and membership queries. So as to facilitate comparison with the case of context-free languages, we assume that the learner outputs right-linear context-free grammars. A context-free grammar $G = (N, \Sigma, P, S)$, where N is the set of nonterminals, Σ is the terminal alphabet, P is the set of productions, and S is the start symbol, is said to be *right-linear* if each production in P is either of the form $A \to a\,B$ or of the form $A \to \varepsilon$, where $A, B \in N$ and $a \in \Sigma$. Suppose that $G_* = (N_*, \Sigma, P_*, S)$ is the right-linear context-free grammar corresponding to the minimal DFA for L_*. (This means that G_* has no useless nonterminal.) The sets of terminal strings derived from the nonterminals of G_* are exactly the nonempty left quotients of L_*. For each $B \in N_*$, let u_B be the length-lexicographically first string such that $u_B\backslash L_* = \{x \in \Sigma^* \mid B \Rightarrow_{G_*}^* x\}$.[2] We have $u_S = \varepsilon$, corresponding to $\varepsilon\backslash L_* = L_*$. Productions in P_* are of one of two forms:

[2] The choice of the length-lexicographic order is not essential. Other strict total orders on Σ^* may be used instead, provided that the empty string comes first.

$$A \to a\,B, \qquad \text{where } a \in \Sigma \text{ and } u_A a\backslash L_* = u_B\backslash L_*,$$
$$A \to \varepsilon, \qquad \text{where } u_A \in L_*.$$

(The former type of production means that there is a transition labeled a from the state corresponding to A to the state corresponding to B in the minimal DFA, and the latter type of production means that A corresponds to a final state.) The learner's task is (i) to identify the set $Q_* = \{u_B \mid B \in N_*\}$, and (ii) to determine, for each $u, v \in Q_*$ and $a \in \Sigma$, whether $ua\backslash L_* = v\backslash L_*$.

For $K \subseteq \Sigma^*$, let

$$\mathrm{Pref}(K) = \{u \in \Sigma^* \mid uv \in K \text{ for some } v \in \Sigma^*\},$$
$$\mathrm{Suff}(K) = \{v \in \Sigma^* \mid uv \in K \text{ for some } u \in \Sigma^*\}.$$

One reasonable strategy for the learner is to work under the assumption that the available positive data T is large enough that $Q_* \subseteq \mathrm{Pref}(T)$ and for each pair of distinct nonempty left quotients of L_*, a string in their symmetric difference occurs in $(\{\varepsilon\} \cup \Sigma)\mathrm{Suff}(T)$. (The assumption will eventually be true.) Let $\prec$ be the length-lexicographic strict total order on Σ^*. For $J, E \subseteq \Sigma^*$, define

$$\begin{aligned} Q(J,E) = \{u \mid\ & u \in J \text{ and for every } v \in J,\\ & \text{if } v \prec u, \text{then} (\{\varepsilon\} \cup \Sigma)E \cap (v\backslash L_*) \neq (\{\varepsilon\} \cup \Sigma)E \cap (u\backslash L_*)\}. \end{aligned}$$

Then $Q(\mathrm{Pref}(T), \mathrm{Suff}(T))$ is the set of nonterminals of the grammar the learner hypothesizes. When we use a string u as a nonterminal in a grammar, we write $\langle\!\langle u \rangle\!\rangle$ instead of just u to avoid confusion. A production $\langle\!\langle u \rangle\!\rangle \to a\,\langle\!\langle v \rangle\!\rangle$ should be included in the grammar if and only if $ua\backslash L_* = v\backslash L_*$, but this cannot be decided without knowledge of the identity of L_*, even with the help of the oracle for L_*. It is again reasonable for the learner to assume that the available positive data is large enough to provide any witness to the falsity of this identity. Let

$$\begin{aligned} P(J,E) =\ & \{\langle\!\langle u \rangle\!\rangle \to a\,\langle\!\langle v \rangle\!\rangle \mid u, v \in J, a \in \Sigma, E \cap (ua\backslash L_*) = E \cap (v\backslash L_*)\} \cup \\ & \{\langle\!\langle u \rangle\!\rangle \to \varepsilon \mid u \in J, u \in L_*\}. \end{aligned} \tag{1}$$

Then $P(Q(\mathrm{Pref}(T), \mathrm{Suff}(T)), \mathrm{Suff}(T))$ is the set of productions of the hypothesized grammar.

The learning algorithm in its entirety is listed in Algorithm 1.[3]

It is not difficult to see that the output G_i of Algorithm 1 is isomorphic to G_* whenever the following conditions hold:[4]

(i) $Q_* \subseteq \mathrm{Pref}(T_i)$,
(ii) for each $u, v \in Q_*$,

$$u\backslash L_* \neq v\backslash L_* \Longrightarrow (\{\varepsilon\} \cup \Sigma)\mathrm{Suff}(T_i) \cap ((u\backslash L_*) \triangle (v\backslash L_*)) \neq \varnothing,$$

[3] There is an obvious connection with the work of Angluin [1,2] and many others which I will not discuss here since this algorithm is not the basis of our generalization to context-free languages.

[4] I write $X \triangle Y$ for the symmetric difference of X and Y.

Algorithm 1. Learner for the regular languages.

Data: A positive presentation $t_1, t_2, \dots$ of $L_* \subseteq \Sigma^*$; membership oracle for L_*;
Result: A sequence of grammars $G_1, G_2, \dots$;

$T_0 := \varnothing$;
for $i = 1, 2, \dots$ **do**
  $T_i := T_{i-1} \cup \{t_i\}$; output $G_i := (N_i, \Sigma, P_i, \langle\!\langle \varepsilon \rangle\!\rangle)$ where
  $N_i := Q(\mathrm{Pref}(T_i), \mathrm{Suff}(T_i))$;
  $P_i := P(N_i, \mathrm{Suff}(T_i))$;
end

(iii) for each $u, v \in Q_*$ and $a \in \Sigma$,

$$ua\backslash L_* \neq v\backslash L_* \Longrightarrow \mathrm{Suff}(T_i) \cap ((ua\backslash L_*) \bigtriangleup (v\backslash L_*)) \neq \varnothing.$$

Computing $N(T_i)$ requires membership queries for all elements of $\mathrm{Pref}(T_i)\,(\{\varepsilon\} \cup \Sigma)\,\mathrm{Suff}(T_i)$, while computing $P(T_i)$ requires membership queries for some subset of $\mathrm{Pref}(T_i)\,(\{\varepsilon\} \cup \Sigma)\,\mathrm{Suff}(T_i)$. The number of queries needed to compute G_i is polynomial in the total lengths of the strings in T_i.

Algorithm 1 satisfies the following properties:

(a) It is set-driven: G_i is determined uniquely by $\{t_1, \dots, t_i\}$ (for a fixed L_* but across different positive presentations $t_1, t_2, \dots$ of L_*).
(b) Its conjecture is consistent with the positive data: $\{t_1, \dots, t_i\} \subseteq L(G_i)$.[5]
(c) It updates its conjecture in polynomial time (in the total lengths of the strings in $\{t_1, \dots, t_i\}$).
(d) There is a "characteristic sample" $D \subseteq L_*$ whose total size is polynomial in the representation size of G_* such that G_i is isomorphic to G_* whenever $D \subseteq \{t_1, \dots, t_i\}$.

These characteristics of Algorithm 1 obviously rest on special properties of the regular languages, and not all of them can be maintained as we move to learning algorithms for context-free languages. We keep (c), but abandon (a), (b), and (d) in favor of weaker conditions. Since a context-free language has no canonical grammar and there is no polynomial bound on the length of the shortest string generated by a context-free grammar, we cannot hope to maintain (d), but even the following weakening will not hold of our algorithms:

(d†) There is a "characteristic sample" $D \subseteq L_*$ whose *cardinality* is polynomial in the representation size of G_* such that $L(G_i) = L_*$ whenever $D \subseteq T_i$.

Let us illustrate the kind of change we must make with another learning algorithm for the regular languages. The algorithm will no longer be set-driven. For $J, E \subseteq \Sigma^*$, define

$$\begin{aligned} P'(J,E) = \ &\{\langle\!\langle u \rangle\!\rangle \to a \,\langle\!\langle v \rangle\!\rangle \mid u, v \in J, a \in \Sigma, u\backslash L_* \supseteq a\,(E \cap (v\backslash L_*))\} \cup \\ &\{\langle\!\langle u \rangle\!\rangle \to \varepsilon \mid u \in J, u \in L_*\}. \end{aligned} \tag{2}$$

[5] This requires a proof. Here it is crucial that we had $(\{\varepsilon\} \cup \Sigma)E$ rather than E in the definition of $Q(J, E)$.

Since $u\backslash L_* \supseteq a\,(E \cap (v\backslash L_*))$ is equivalent to $E \cap (ua\backslash L_*) \supseteq E \cap (v\backslash L_*)$, the difference between $P(J,E)$ and $P'(J,E)$ just consists in replacing equality with inclusion. The algorithm updates the set J_i of prefixes of positive examples only when the positive examples received so far are incompatible with the previous conjecture. It is listed in Algorithm 2.

Algorithm 2. Learner for the regular languages, nondeterministic version.

Data: A positive presentation $t_1, t_2, \ldots$ of $L_* \subseteq \Sigma^*$; membership oracle for L_*;
Result: A sequence of grammars $G_1, G_2, \ldots$;

$T_0 := \varnothing$; $E_0 := \varnothing$; $J_0 := \varnothing$; $G_0 := (\{\langle\!\langle \varepsilon \rangle\!\rangle\}, \Sigma, \varnothing, \langle\!\langle \varepsilon \rangle\!\rangle)$;
for $i = 1, 2, \ldots$ **do**
  $T_i := T_{i-1} \cup \{t_i\}$; $E_i := E_{i-1} \cup \mathrm{Suff}(\{t_i\})$;
  if $T_i \subseteq L(G_{i-1})$ **then**
    $J_i := J_{i-1}$;
  else
    $J_i := \mathrm{Pref}(T_i)$;
  end
  $N_i := Q(J_i, E_i)$;
  $P_i := P'(N_i, E_i)$;
  output $G_i := (N_i, \Sigma, P_i, \langle\!\langle \varepsilon \rangle\!\rangle)$;
end

Suppose that the conditions (i), (ii) above and the following condition (iii′) hold of T_i:

(iii′) for each $u, v \in Q_*$ and $a \in \Sigma$,

$$u\backslash L_* \not\supseteq a(v\backslash L_*) \Longrightarrow a\,(\mathrm{Suff}(T_i) \cap (v\backslash L_*)) - (u\backslash L_*) \neq \varnothing.$$

The conditions (i) and (ii) mean that $Q_* = Q\,(\mathrm{Pref}(T_i), E_i)$. There are two cases to consider.

Case 1. $T_i \not\subseteq L(G_{i-1})$. Then $J_i = \mathrm{Pref}(T_i)$ and $N_i = Q_*$. The condition (iii′) then means that

$$\begin{aligned} P_i &= P'(N_i, E_i) \\ &= \{\langle\!\langle u \rangle\!\rangle \to a\,\langle\!\langle v \rangle\!\rangle \mid u, v \in Q_*, a \in \Sigma, u\backslash L_* \supseteq a\,(v\backslash L_*)\} \cup \\ &\quad\ \{\langle\!\langle u \rangle\!\rangle \to \varepsilon \mid u \in Q_*, u \in L_*\}. \end{aligned}$$

The learner's hypothesis G_i is just like G_* (the right-linear grammar corresponding to the minimal DFA for L_*) except that it may have additional productions. In general, the finite automaton corresponding to G_i is nondeterministic, but it accepts exactly the same strings as the minimal DFA. It is in fact the result of adding to the minimal DFA as many transitions as possible without changing the accepted language. (Let us call this NFA the *fattening* of the minimal DFA.) We have $L(G_i) = L_*$, and at all stages $l \geq i$, the sets N_l and P_l, as well as the output grammar G_l, will stay constant.

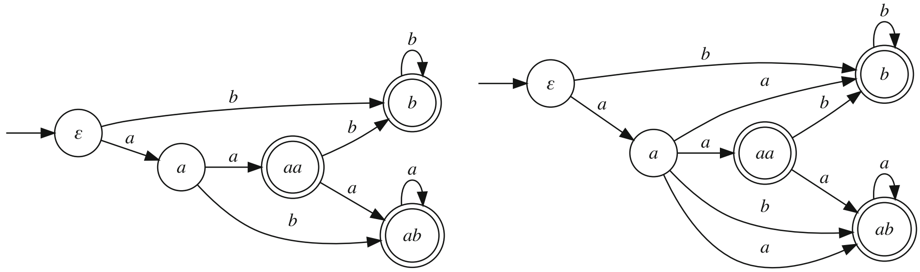


Fig. 1. The minimal DFA for $aba^* \cup bb^* \cup aa(a^* \cup b^*)$ (left) and its fattening (right).

Case 2. $T_i \subseteq L(G_{i-1})$. In this case, J_i may be a proper subset of $\mathrm{Pref}(T_i)$. The condition (ii) implies that N_i is in one-to-one correspondence with some subset of Q_*. That is to say, for each $u' \in N_i$, there is a $u \in Q_*$ such that $u' \backslash L_* = u \backslash L_*$, and if $u', v' \in N_i$ and $u' \backslash L_* = v' \backslash L_*$, then $u' = v'$. The condition (iii′) implies

$$P'(N_i, E_i) = \{\langle\!\langle u \rangle\!\rangle \to a \, \langle\!\langle v \rangle\!\rangle \mid u, v \in N_i, a \in \Sigma, u \backslash L_* \supseteq a \, (v \backslash L_*)\} \cup \\ \{\langle\!\langle u \rangle\!\rangle \to \varepsilon \mid u \in N_i, u \in L_*\}.$$

This means that the NFA corresponding to this right-linear grammar is isomorphic to a subautomaton of the fattening of the minimal DFA, so we must have $L(G_i) \subseteq L_*$. If $L(G_i) = L_*$, then the learner's hypothesis will remain the same at all later stages. If $L_* - L(G_i) \neq \varnothing$, then Case 1 applies at the earliest stage $l \geq i$ such that $t_l \notin L(G_i)$.

Example 1. Suppose that the target language is $L_* = aba^* \cup bb^* \cup aa(a^* \cup b^*)$. The minimal DFA for L_* and its fattening are shown in Fig. 1. On receiving $\{b, aa, ab\}$ (presented in this order), Algorithm 2 outputs the right-linear grammar corresponding to the NFA on the right. On receiving $\{b, aba\}$, it outputs the right-linear grammar corresponding to the NFA in Fig. 2. In both cases, the learner's hypothesis stays constant at all later stages.

Although Algorithm 2 is not set-driven and the grammar it stabilizes on depends on the order of the positive presentation, its behavior is not unreasonable. In a way, it tries to postulate as few nonterminals (states) as possible. It processes all positive examples immediately and does not engage in any "delaying trick" [3,4] just to achieve polynomial update time.

Instead of (a) and (b), Algorithm 2 satisfies the following weaker conditions:

(a′) If $\{t_1, \dots, t_i\} \subseteq L(G_i)$ and $t_{i+1} \in \{t_1, \dots, t_i\}$, then $G_{i+1} = G_i$.
(b′) If $\{t_1, \dots, t_i\} \not\subseteq L(G_{i-1})$, then $\{t_1, \dots, t_i\} \subseteq L(G_i)$.

It is easy to verify (a′). For (b′), if $t_k = a_1 \dots a_n$ with $a_j \in \Sigma$ $(1 \leq j \leq n)$, then $\mathrm{Pref}(\{t_k\}) \subseteq J_i$ means that for each $j = 0, \dots, n$, there is a $u_j \in N_i$ such that

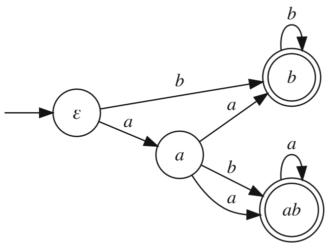


Fig. 2. An NFA for $aba^* \cup bb^* \cup aa(a^* \cup b^*)$.

$(\{\varepsilon\} \cup \Sigma)E_i \cap (u_j \backslash L_*) = (\{\varepsilon\} \cup \Sigma)E_i \cap (a_1 \dots a_j \backslash L_*)$. We have

$$\begin{aligned} a_{j+1}(E_i \cap (v_{j+1} \backslash L_*)) &= a_{j+1}(E_i \cap (a_1 \dots a_{j+1} \backslash L_*)) \\ &= a_{j+1}E_i \cap (a_1 \dots a_j \backslash L_*) \\ &= a_{j+1}E_i \cap (v_j \backslash L_*) \\ &\subseteq v_j \backslash L_*, \end{aligned}$$

so P_i contains the production $\langle\!\langle v_j \rangle\!\rangle \to a_{j+1} \langle\!\langle v_{j+1} \rangle\!\rangle$. Since

$$\varepsilon \in \{\varepsilon\}E_i \cap (a_1 \dots a_n \backslash L_*) = \{\varepsilon\}E_i \cap (v_n \backslash L_*),$$

we have $v_n \in L_*$, which implies that $\langle\!\langle v_n \rangle\!\rangle \to \varepsilon$ is in P_i as well. So we have a derivation $\langle\!\langle \varepsilon \rangle\!\rangle = \langle\!\langle v_0 \rangle\!\rangle \Rightarrow^*_{G_i} a_1 \dots a_n = t_k$.

We cannot prove that Algorithm 2 satisfies (d) (or (d†), for that matter) in the same way we proved (d) for Algorithm 1. This is because when $D \subseteq T_i$, where D is a polynomial-sized set satisfying (i), (ii), and (iii′), we may have $T_i \subseteq L(G_{i-1})$, in which case the algorithm may need an additional string from $L_* - L(G_i)$ in order to reach a correct grammar. This additional string depends on the set $N_i \subset Q_*$, of which there are exponentially many possibilities.[6] We can summarize this behavior of Algorithm 2 as follows:

(d′) There is a finite set $D \subseteq L_*$ whose total size is bounded by a polynomial in the representation size of G_* such that whenever $D \subseteq \{t_1, \dots, t_i\}$, there is a string t, depending on $(t_1, \dots, t_i)$, of length less than $|Q_*|$ such that whenever $t \in \{t_{i+1}, \dots, t_l\}$, G_l is constant and isomorphic to the right-linear grammar corresponding to a subautomaton of the fattening of the minimal DFA corresponding to G_*.

Our learning algorithms for context-free languages resemble Algorithm 2 in many ways, but also differ in some important respects.

6 We only need to worry about *maximal* subsets of Q_* such that the corresponding subautomaton of the fattening of the minimal DFA for L_* fails to accept all strings in L_*. Still, there may be exponentially many such maximal sets.

3 Context-Free Languages

Like Algorithms 1 and 2, our algorithms for learning context-free languages use membership queries to test whether a given string x belongs to the string set associated with a postulated nonterminal. For this to be possible, we must assume that the set *reduces* in polynomial time to the target language L_*. The reduction must be uniform across different target languages—the learner must have a representation of a nonterminal without full knowledge of the target language, and this representation must determine the reduction by which the string set of the nonterminal reduces to the target language.

Let us formally define the three subclasses of context-free grammars we are interested in. If $G = (N, \Sigma, P, S)$ is a context-free grammar, a tuple $(X_B)_{B \in N}$ of sets in $\mathscr{P}(\Sigma^*)$ is a *pre-fixed point* of G if for each production $A \to w_0\, B_1\, w_1\, \ldots\, B_n\, w_n$ in P, we have

$$X_A \supseteq w_0\, X_{B_1}\, w_1\, \ldots\, X_{B_n}\, w_n.$$

The tuple $(X_B)_{B \in N}$ with $X_B = \Sigma^*$ for all $B \in N$ is the greatest pre-fixed point of G, and the tuple $(L_G(B))_{B \in N}$ is the least pre-fixed point (under the partial order of componentwise inclusion). A pre-fixed point $(X_B)_{B \in N}$ is *sound* if $X_S \subseteq L(G)$ (or, equivalently, if $X_S = L(G)$). Let Γ be a set of operations on $\mathscr{P}(\Sigma^*)$ (of varying arity). Then G has the *Γ-closure property* if G has a sound pre-fixed point (SPP) each of whose components belongs to the Γ-closure of $\mathrm{Quot}(L(G))$. Setting Γ to $\{\cap\}$, we get the class of context-free grammars with the *intersection closure property*. With $\Gamma = \{\cap, \bar{\cdot}, \cup\}$ (intersection, complement, and union), we get the context-free grammars with the *Boolean closure property*. If we add to this set $\varnothing$, ε, and a ($a \in \Sigma$) (considered the zero-ary operations producing $\varnothing$, $\{\varepsilon\}$, and $\{a\}$, respectively) and the concatenation and Kleene star operations, we get the context-free grammars with the *extended regular closure property*.

Our learning algorithms targeting context-free grammars with the Γ-closure property use expressions built from *query atoms* $(u, v)^{\triangleleft}$, where $u, v \in \Sigma^*$, and symbols for operations in Γ. These expressions denote subsets of Σ^* relative to L_* in the obvious way:

$$\begin{aligned} [\![(u,v)^{\triangleleft}]\!]^{L_*} &= u \backslash L_* / v, \\ [\![e_1 \cap e_2]\!]^{L_*} &= [\![e_1]\!]^{L_*} \cap [\![e_2]\!]^{L_*}, \\ [\![\overline{e_1}]\!]^{L_*} &= \Sigma^* - [\![e_1]\!]^{L_*}, \\ [\![e_1 \cup e_2]\!]^{L_*} &= [\![e_1]\!]^{L_*} \cup [\![e_2]\!]^{L_*}, \end{aligned} \qquad \begin{aligned} [\![\varnothing]\!]^{L_*} &= \varnothing, \\ [\![\varepsilon]\!]^{L_*} &= \{\varepsilon\}, \\ [\![a]\!]^{L_*} &= \{a\}, \\ [\![e_1 e_2]\!]^{L_*} &= [\![e_1]\!]^{L_*}\, [\![e_2]\!]^{L_*}, \\ [\![e_1^*]\!]^{L_*} &= ([\![e_1]\!]^{L_*})^*. \end{aligned}$$

An important property of these expressions is the following. If e is an expression, let $C_e = \{(u, v) \mid (u, v)^{\triangleleft} \text{occurs in } e\}$.

(*) The truth value of $x \in [\![e]\!]^{L_*}$ only depends on the truth values of $y \in u\backslash L_*/v$ for substrings y of x and $(u, v) \in C_e$.

3.1 Examples

Let us look at some examples.[7] If A is a nonterminal of a grammar G, we often abuse the notation and write just A for the set $L_G(A)$.

Example 2. Consider

$$L_1 = \{a^m b^n \mid m \text{ is even and } m = n\} \cup \{a^m b^n \mid m \text{ is odd and } 2m = n\}.$$

This language is generated by the following grammar:

$$S \to T \mid U, \quad T \to \varepsilon \mid aaTbb, \quad U \to abb \mid aaUbbbb.$$

We have

$$\begin{aligned} S &= L_1 = [\![(\varepsilon, \varepsilon)^{\lhd}]\!]^{L_1}, \\ T &= \{a^m b^n \mid m \text{ is even and } m = n\} = [\![(\varepsilon, \varepsilon)^{\lhd} \cap (aa, bb)^{\lhd}]\!]^{L_1}, \\ U &= \{a^m b^n \mid m \text{ is odd and } 2m = n\} = [\![(\varepsilon, \varepsilon)^{\lhd} \cap (aa, bbbb)^{\lhd}]\!]^{L_1}. \end{aligned}$$

So this grammar has the intersection closure property.

Example 3. Consider

$$L_2 = \{a^m b^n \mid m \text{ is even and } m = n\} \cup \{a^m b^n \mid m \text{ is odd and } 2m \leq n\}.$$

This language is generated by the following grammar:

$$S \to T \mid UB, \quad T \to \varepsilon \mid aaTbb, \quad U \to abb \mid aaUbbbb, \quad B \to \varepsilon \mid Bb.$$

We have

$$\begin{aligned} S &= L_2 = [\![(\varepsilon, \varepsilon)^{\lhd}]\!]^{L_2}, \\ T &= \{a^m b^n \mid m \text{ is even and } m = n\} = \left[\!\!\left[(\varepsilon, \varepsilon)^{\lhd} \cap \overline{(aa, bbbb)^{\lhd}}\right]\!\!\right]^{L_2}, \\ U &= \{a^m b^n \mid m \text{ is odd and } 2m = n\} = \left[\!\!\left[(\varepsilon, \varepsilon)^{\lhd} \cap (aa, bbbb)^{\lhd} \cap \overline{(aa, bbb)^{\lhd}}\right]\!\!\right]^{L_2}, \\ B &= b^* = [\![(abb, \varepsilon)]\!]^{L_2}. \end{aligned}$$

So this grammar has the Boolean closure property. We can show that the set T is not in the intersection closure of $\mathrm{Quot}(L_2)$, and indeed, L_2 has no grammar with the intersection closure property.

[7] In all of these examples, the SPP witnessing the relevant closure property is the least SPP, but there are cases where the witnessing SPP cannot be the least one [5].

Example 4. Consider

$$L_3 = \{a^m b^n \mid m \text{ is even and } m \geq n\} \cup \{a^m b^n \mid m \text{ is odd and } 2m \leq n \leq 3m\}.$$

This language is generated by the following grammar:

$$S \to AT \mid U, \quad A \to \varepsilon \mid Aaa, \quad T \to \varepsilon \mid aaTbb,$$
$$U \to abb \mid V \mid aaUbbbb, \quad V \to abbb \mid aaVbbbbbb.$$

We have

$$S = L_3 = [\![(\varepsilon,\varepsilon)^{\triangleleft}]\!]^{L_3},$$
$$A = (aa)^* = [\![(\varepsilon, aabb)^{\triangleleft}]\!]^{L_3}$$
$$T = \{a^m b^n \mid m \text{ is even and } m = n\} = \left[\!\!\left[(\varepsilon,\varepsilon)^{\triangleleft} \cap (aa)^* b^* \cap \overline{(\varepsilon,b)^{\triangleleft}}\right]\!\!\right]^{L_3},$$
$$U = \{a^m b^n \mid m \text{ is odd and } 2m \leq n \leq 3m\} = [\![(\varepsilon,\varepsilon)^{\triangleleft} \cap (aa)^* ab^*]\!]^{L_3},$$
$$V = \{a^m b^n \mid m \text{ is odd and } n = 3m\} = \left[\!\!\left[(\varepsilon,\varepsilon)^{\triangleleft} \cap (aa)^* ab^* \cap \overline{(\varepsilon,b)^{\triangleleft}}\right]\!\!\right]^{L_3}$$

So this grammar has the extended regular closure property. We can show that L_3 has no grammar with the Boolean closure property.

Example 5. Consider

$$L_4 = \{a^m b^n \mid m \geq n\} \cup \{a^m b^n \mid m \geq 1 \text{ and } 2m \leq n \leq 3m\}.$$

Unlike the previous three examples, this is not a deterministic context-free language. It is generated by the following unambiguous grammar:

$$S \to AT \mid U, \quad A \to \varepsilon \mid Aa, \quad T \to \varepsilon \mid aTb,$$
$$U \to abb \mid V \mid aUbb, \quad V \to abbb \mid aVbbb.$$

We have

$$S = L_4 = [\![(\varepsilon,\varepsilon)^{\triangleleft}]\!]^{L_4},$$
$$A = a^* = [\![(\varepsilon,ab)^{\triangleleft}]\!]^{L_4},$$
$$T = \{a^m b^n \mid m = n\} = \left[\!\!\left[(\varepsilon,\varepsilon)^{\triangleleft} \cap \left(\varepsilon \cup \overline{(a,bb)^{\triangleleft}}\right)\right]\!\!\right]^{L_4},$$
$$U = \{a^m b^n \mid \text{ m } \geq 1 \text{ and } 2m \leq n \leq 3m\} = \left[\!\!\left[(\varepsilon,\varepsilon)^{\triangleleft} \cap \left(ab \cup \overline{(\varepsilon,\varepsilon)^{\triangleleft}}\right) bb^*\right]\!\!\right]^{L_4},$$
$$V = \{a^m b^n \mid \text{ n = 3m}\} = \left[\!\!\left[(\varepsilon,\varepsilon)^{\triangleleft} \cap \left(ab \cup \overline{(\varepsilon,\varepsilon)^{\triangleleft}}\right) bb^* \cap \overline{(\varepsilon,b)^{\triangleleft}}\right]\!\!\right]^{L_4}.$$

So this grammar has the extended regular closure property. Again, we can show that L_4 has no grammar with the Boolean closure property.

Example 6. The inherently ambiguous language

$$\{a^l b^m c^n d^q \mid \ l,m,n,q \geq 1 \text{ and } l = n \vee m > q\}$$

does not have a grammar with the extended regular closure property [7].

3.2 Algorithm

Let us describe our algorithm for learning context-free languages, leaving the set Γ of available operations as an unspecified parameter. The set Γ may be any subset of the Boolean and regular operations, as long as it contains intersection. In this algorithm, quotients of L_* will play a role similar to the role left quotients played in Algorithm 2. An important difference is that since a context-free language has infinitely many quotients (unless it is regular), the learner cannot identify the set of all quotients, even in the limit. It tries to identify a superset of the set of quotients the target grammar "uses", so to speak, and this is done by a strategy similar to Algorithm 2.

Since the learner can only postulate a polynomial number of nonterminals, we place an arbitrary finite bound k on the number of occurrences of symbols for operations in the representation of a nonterminal. Since polynomially many string pairs (u, v) are available as building blocks of nonterminals, this is a necessary restriction.

We also have to place a suitable syntactic restriction on nonterminals that ensures that their denotations relative to L_* are included in $\text{Sub}(L_*) = \{x \mid \text{for some } (u, v), uxv \in L_*\}$. We call a nonterminal obeying this restriction *guarded*:

- $(u, v)^{\triangleleft}$ is guarded.
- If e_1 is guarded, so is $e_1 \cap e_2$.
- If e_1 and e_2 are guarded, so is $e_1 \cup e_2$.

If C is a finite subset of $\Sigma^* \times \Sigma^*$, we write $\mathcal{E}_k(C, \Gamma)$ for the set of all guarded expressions built up from query atoms in $\{(u, v)^{\triangleleft} \mid (u, v) \in C\}$ using operations in Γ up to k times.[8]

As for productions, we place a bound r on the number of occurrences of nonterminals and a bound s on the length of contiguous terminal strings on the right-hand side of a production. So a production postulated by the learner is of the form

$$A \to w_0\, B_1\, w_1\, \ldots\, B_n\, w_n,$$

where $n \leq r$ and $|w_i| \leq s$ $(0 \leq i \leq n)$.[9] The notation $\langle \Sigma^{\leq s} \rangle^{\leq r+1}$ is used to denote the set of possible choices of $(w_0, w_1, \ldots, w_n)$. Note that if $E \subseteq \Sigma^*$ is closed under substring, then the set

$$\Sigma^{\leq s} (E\, \Sigma^{\leq s})^{\leq r} = \bigcup \{ w_0\, E\, w_1\, \ldots\, E\, w_n \mid (w_0, w_1, \ldots, w_n) \in \langle \Sigma^{\leq s} \rangle^{\leq r+1} \}$$

is also closed under substring, and is a superset of E.

8 A reasonable optimization is to put expressions in some suitable "normal form", to avoid including a large number of equivalent expressions in $\mathcal{E}_k(C, \Gamma)$.

9 An alternative is to allow arbitrary terminal strings to surround nonterminals on the right-hand side of productions, as long as they are "observed" in the positive data [5–7].

Define a strict total order $\prec_2$ on $\Sigma^* \times \Sigma^*$ by

$$(u_1, u_2) \prec_2 (v_1, v_2) \iff u_1 \prec v_1 \vee (u_1 = v_1 \wedge u_2 \prec v_2).$$

We write $\preceq_2$ for the reflexive counterpart of $\prec_2$. For $J \subseteq \Sigma^* \times \Sigma^*$ and $E \subseteq \Sigma^*$, let

$$\begin{aligned} Q^{r,s}(J, E) = \\ \{(u_1, u_2) \mid\ & (u_1, u_2) \in J \text{ and for each } (v_1, v_2) \in J, \\ & \text{if } (v_1, v_2) \prec_2 (u_1, u_2), \text{then} \\ & (\Sigma^{\leq s}(E\,\Sigma^{\leq s})^{\leq r}) \cap (v_1 \backslash L_* / v_2) \neq (\Sigma^{\leq s}(E\,\Sigma^{\leq s})^{\leq r}) \cap (u_1 \backslash L_* / u_2)\}. \end{aligned}$$

For $K, E \subseteq \Sigma^*$ and a set N of expressions, define

$$\begin{aligned} \mathrm{Sub}(K) &= \{y \in \Sigma^* \mid uyv \in K \text{ for some } (u, v) \in \Sigma^* \times \Sigma^*\}, \\ \mathrm{Con}(K) &= \{(u, v) \in \Sigma^* \times \Sigma^* \mid uyv \in K \text{ for some } y \in \Sigma^*\}, \\ P^{r,s}(N, E) &= \{\, A \to w_0\, B_1\, w_1\, \dots\, B_n\, w_n \mid \\ &\qquad 0 \leq n \leq r, A, B_1, \dots, B_n \in N, \\ &\qquad (w_0, w_1, \dots, w_n) \in \langle \Sigma^{\leq s} \rangle^{\leq r+1}, \\ &\qquad [\![A]\!]^{L_*} \supseteq w_0\, (E \cap [\![B_1]\!]^{L_*})\, w_1\, \dots\, (E \cap [\![B_n]\!]^{L_*})\, w_n\}. \end{aligned}$$

The inclusion $[\![A]\!]^{L_*} \supseteq w_0\, (E \cap [\![B_1]\!]^{L_*})\, w_1\, \dots\, (E \cap [\![B_n]\!]^{L_*})\, w_n$ in the definition of $P^{r,s}(N, E)$ is analogous to the inclusion $u \backslash L_* \supseteq a\, (E \cap (v \backslash L_*))$ in the definition of $P'(J, E)$ in (2).

With the necessary definitions in place, we can list the learning algorithm in Algorithm 3.

Algorithm 3. Learner for a subclass of the context-free languages.

Parameters: Positive integers r, s, k
Data: A positive presentation $t_1, t_2, \dots$ of $L_* \subseteq \Sigma^*$; membership oracle for L_*;
Result: A sequence of grammars $G_1, G_2, \dots$;

```
T_0 := ∅; E_0 := ∅; J_0 := ∅; G_0 := ({(ε, ε)^◁}, Σ, ∅, (ε, ε)^◁);
for i = 1, 2, ... do
    T_i := T_{i−1} ∪ {t_i}; E_i := E_{i−1} ∪ Sub({t_i});
    if T_i ⊆ L(G_{i−1}) then
        J_i := J_{i−1};
    else
        J_i := Con(T_i);
    end
    N_i := 𝓔_k(Q^{r,s}(J_i, E_i), Γ);
    P_i := P^{r,s}(N_i, E_i);
    output G_i := (N_i, Σ, P_i, (ε, ε)^◁);
end
```

By the property (∗) of expressions used by the learner, membership queries that are needed to compute N_i and P_i are all of the form "$uyv \in L_*$?", where $(u, v) \in \mathrm{Con}(T_i)$ and $y \in \Sigma^{\leq s}(\mathrm{Sub}(T_i)\Sigma^{\leq s})^{\leq r}$. There are only polynomially many of them (in the total lengths of the strings in T_i).

Algorithm 3 is by no means capable of learning all context-free languages. What is the class of context-free languages that the algorithm can learn? Recall that the denotation of each nonterminal B (relative to L_*) is a subset of $\mathrm{Sub}(L_*)$. Because of how $P^{r,s}(N, E)$ is defined, if Algorithm 3 stabilizes on a grammar G_i, then all its productions $A \to w_0\, B_1\, w_1\, \ldots\, B_n\, w_n$ are *valid* in the sense that

$$[\![A]\!]^{L_*} \supseteq w_0\ [\![B_1]\!]^{L_*}\ w_1\ \ldots\ [\![B_n]\!]^{L_*}\ w_n, \tag{3}$$

since $\bigcup_i E_i = \mathrm{Sub}(L_*)$. This means that the tuple of sets $([\![B]\!]^{L_*})_{B \in N_i}$ is a pre-fixed point of G_i. So $L_{G_i}(B) \subseteq [\![B]\!]^{L_*}$. In particular, $L(G_i) = L_{G_i}((\varepsilon, \varepsilon)^{\triangleleft}) \subseteq [\![(\varepsilon, \varepsilon)^{\triangleleft}]\!]^{L_*} = L_*$. The converse inclusion can be shown by appealing to the fact that Algorithm 3 satisfies the property (b′), which can be proved in the same way as with Algorithm 2. If $L_* \not\subseteq L(G_i)$, then at the earliest stage $l > i$ such that $t_l \notin L(G_i)$, we must have $G_l \neq G_i$, since $t_l \in L(G_l)$ by (b′). This contradicts the assumption that Algorithm 3 stabilizes on G_i. So we have $L(G_i) = L_*$ and $([\![B]\!]^{L_*})_{B \in N_i}$ is an SPP of G_i. This shows that G_i has the Γ-closure property.

Theorem 7. *If, given a positive presentation of L_* and the membership oracle for L_*, Algorithm 3 stabilizes on a grammar G, then $L(G) = L_*$ and G has the Γ-closure property.*

We have seen that Algorithm 3 can only learn a context-free language that has a grammar with the Γ-closure property. Conversely, if L_* has a grammar G_* with the Γ-closure property, let r and s be the maximal number of nonterminals and the maximal length of contiguous terminal strings, respectively, on the right-hand side of productions of G_*. Let k be the least number such that G_* has an SPP each of whose components is denoted by a guarded expression containing at most k operations in Γ. Then with this choice of r, s, k, one can show that Algorithm 3 learns L_*. The proof is similar to the proof for Algorithm 2, with some additional complications.[10]

Let Q_* be the set of pairs (u, v) of strings such that the query atom $(u, v)^{\triangleleft}$ occurs in expressions for components of the SPP for G_*. So if N_* is the set of nonterminals of G_*, there is an expression $e_B \in \mathcal{E}_k(Q_*, \Gamma)$ for each $B \in N_*$ such that $([\![e_B]\!]^{L_*})_{B \in N_*}$ is an SPP of G_*. We can safely assume that if $(u, v) \in Q_*$ and $(u', v') \prec_2 (u, v)$, then $u' \backslash L_* / v' \neq u \backslash L_* / v$. In particular, if (u_1, v_1) and (u_2, v_2) are distinct elements of Q_*, we have $u_1 \backslash L_* / v_1 \neq u_2 \backslash L_* / v_2$.

Now assume that $t_1, t_2, \ldots$ is a positive presentation of L_*, and suppose $T_i = \{t_1, \ldots, t_i\}$ is such that $Q_* \subseteq \mathrm{Con}(T_i)$.

10 Unlike the algorithms in [5–7,9], Algorithm 3 tries to avoid using different query atoms that denote the same quotient of L_*. This should be compared with Leiß's [8] algorithm for the case $\Gamma = \{\cap\}$, which tries to minimize the number of nonterminals used.

Case 1. $T_i \not\subseteq L(G_{i-1})$. Then $Q_* \subseteq J_i = \mathrm{Con}(T_i)$. At each stage $l \geq i$, we have $Q_* \subseteq J_l$, and for each $(u, v) \in Q_*$, there is a $(u', v') \in Q^{r,s}(J_l, E_l)$ such that $(u', v') \preceq_2 (u, v)$ and

$$(\Sigma^{\leq s}(E_l \Sigma^{\leq s})^{\leq r}) \cap (u' \backslash L_* / v') = (\Sigma^{\leq s}(E_l \Sigma^{\leq s})^{\leq r}) \cap (u \backslash L_* / v). \tag{4}$$

For each $B \in N_*$, let e'_B be the result of replacing each query atom $(u, v)^{\triangleleft}$ in e_B by $(u', v')^{\triangleleft}$. Then $e'_B \in N_l$ for each $B \in N_*$, and (4) implies

$$(\Sigma^{\leq s}(E_l \Sigma^{\leq s})^{\leq r}) \cap [\![e_{B'}]\!]^{L_*} = (\Sigma^{\leq s}(E_l \Sigma^{\leq s})^{\leq r}) \cap [\![e_B]\!]^{L_*}. \tag{5}$$

For each production

$$A \to w_0\, B_1\, w_1\, \ldots\, B_n\, w_n$$

of G_*, we have

$$[\![e_A]\!]^{L_*} \supseteq w_0\, [\![e_{B_1}]\!]^{L_*}\, w_1\, \ldots\, [\![e_{B_n}]\!]^{L_*}\, w_n,$$

since $([\![e_B]\!]^{L_*})_{B \in N_*}$ is an SPP of G_*. Since $(w_0, w_1, \ldots, w_n) \in \langle \Sigma^{\leq s} \rangle^{\leq r+1}$, we have

$$(\Sigma^{\leq s})(E_l \Sigma^{\leq s})^{\leq r} \cap [\![e_A]\!]^{L_*} \supseteq w_0\, (E_l \cap [\![e_{B_1}]\!]^{L_*})\, w_1\, \ldots\, (E_l \cap [\![e_{B_n}]\!]^{L_*})\, w_n.$$

But since $E_l \subseteq \Sigma^{\leq s}(E_l \Sigma^{\leq s})^{\leq r}$, this together with (5) implies

$$(\Sigma^{\leq s})(E_l \Sigma^{\leq s})^{\leq r} \cap [\![e_{A'}]\!]^{L_*} \supseteq w_0\, (E_l \cap [\![e_{B'_1}]\!]^{L_*})\, w_1\, \ldots\, (E_l \cap [\![e_{B'_n}]\!]^{L_*})\, w_n,$$

so

$$e_{A'} \to w_0\, e_{B'_1}\, w_1\, \ldots\, e_{B'_n}\, w_n$$

is a production in P_l. Since $(\varepsilon, \varepsilon)' = (\varepsilon, \varepsilon)$ and $((\varepsilon, \varepsilon)^{\triangleleft})' = (\varepsilon, \varepsilon)^{\triangleleft}$, this means that G_l contains a homomorphic image of G_*, and so $L_* \subseteq L(G_l)$. It follows that for all $l \geq i$, we have $J_l = J_i$ and $Q^{r,s}(J_l, E_l) \subseteq J_i$. Since $Q^{r,s}(J, E)$ is monotone in E, we see that $Q^{r,s}(J_l, E_l)$ eventually stabilizes, i.e., there is an $m \geq i$ such that for all $l \geq m$, $Q^{r,s}(J_l, E_l) = Q^{r,s}(J_m, E_m)$. Then we have $N_l = N_m$ and $P_l = P^{r,s}(N_m, E_l)$ for all $l \geq m$. Since $P^{r,s}(N, E)$ is antitone in E, it follows that P_l, and hence G_l, eventually stabilize. By Theorem 7, the grammar that the algorithm stabilizes on is a grammar for L_*.

Case 2. $T_i \subseteq L(G_{i-1})$. We distinguish two cases.

Case 2.1. $T_l \subseteq L(G_{l-1})$ for all $l \geq i$. Then $J_l = J_i = J_{i-1}$ for all $l \geq i$, and by a reasoning similar to Case 1, G_l will eventually stabilize to a correct grammar for L_*.

Case 2.2. $T_l \not\subseteq L(G_{l-1})$ for some $l \geq i$. Then we are in Case 1 at the earliest stage $l \geq i$ when this happens.

Theorem 8. *Suppose that $G_* = (N_*, \Sigma, P_*, S)$ is a context-free grammar with the Γ-closure property such that if $B \to w_0\, B_1\, w_1\, \ldots,\, B_n\, w_n$ is a production in P_*, then $n \leq r$ and $|w_j| \leq s$ $(0 \leq j \leq n)$. Let $L_* = L(G_*)$. Given a positive presentation of L_* and the membership oracle for L_*, Algorithm 3 converges to a grammar for L_* with the Γ-closure property.*

Let us close by stating a data efficiency property of Algorithm 3 in the style of (d′). Let D be the subset of L_* with the least cardinality such that $Q_* \subseteq \mathrm{Con}(D)$. Then $|D| \leq |Q_*| \leq k|N_*|$, but there may be strings in D whose length is exponential in the representation size of G_*. Suppose $D \subseteq T_i = \{t_1, \dots, t_i\}$.

Case 1. If $T_i \not\subseteq L(G_{i-1})$, then the size of $J_i = \mathrm{Con}(T_i)$ is polynomial in the total lengths of the strings in T_i. We need additional positive examples to make the set of nonterminals postulated by the learner equal $\mathcal{E}_k(Q^{r,s}(J_i, \mathrm{Sub}(L_*)), \Gamma)$. We need at most $\binom{|J_i|}{2}$ such positive examples. We also need positive examples to eliminate any productions that are not valid in the sense of (3). The number of such productions is at most polynomial in $|J_i|$. We can combine these two types of positive examples and present them in any order, interspersed with other positive examples, to force the learner to stabilize on a correct grammar.

Case 2. $T_i \subseteq L(G_{i-1})$. Then J_i may be a proper subset of $\mathrm{Con}(T_i)$. As soon as we have $T_l \not\subseteq L(G_{l-1})$ $(i \leq l)$, we will be in Case 1. Until this happens, polynomially many positive examples (in the size of J_i) suffice to make the set of nonterminals postulated by the learner equal $\mathcal{E}_k(Q^{r,s}(J_i, \mathrm{Sub}(L_*)), \Gamma)$ and to eliminate all invalid productions formed with these nonterminals. When T_l $(l \geq i)$ contains these positive examples and we are still in Case 2, we have two possibilities.

Case 2.1. $L_* \subseteq L(G_l)$. In this case, G_l has already stabilized.

Case 2.2. $L_* \not\subseteq L(G_l)$. Then any positive example not in $L(G_l)$ puts us in Case 1.

Summarizing, all we seem to be able to say about Algorithm 3 is that it satisfies the following complex property:

(d†′) There exists a finite subset D of L_* whose cardinality is bounded by a polynomial in the representation size of G_* such that

- whenever $D \subseteq \{t_1, \dots, t_i\} \subseteq L_*$, one can find a finite subset D' of L_*, depending on $(t_1, \dots, t_i)$, whose cardinality is polynomial in the total lengths of the strings in $\{t_1, \dots, t_i\}$ such that
  - whenever $D' \subseteq \{t_{i+1}, \dots, t_k\} \subseteq L_*$, one can find a string $t \in L_*$, depending on $(t_1, \dots, t_k)$, such that
    - whenever $t \in \{t_{k+1}, \dots, t_l\} \subseteq L_*$, one can find a finite subset D'' of L_*, depending on $(t_1, \dots, t_l)$, whose cardinality is polynomial in the total lengths of the strings in $\{t_1, \dots, t_l\}$ such that
      - whenever $D'' \subseteq \{t_{l+1}, \dots, t_m\} \subseteq L_*$, the output G_m of Algorithm 3 is constant and is a correct grammar for L_*.

Acknowledgements. I am grateful to Ryo Yoshinaka for the joint work this paper is an outgrowth of. This work was supported by JSPS KAKENHI Grant Number 17K00026.

References

1. Angluin, D.: A note on the number of queries needed to identify regular languages. Inf. Control **51**(1), 76–87 (1981). https://doi.org/10.1016/S0019-9958(81)90090-5
2. Angluin, D.: Learning regular sets from queries and counterexamples. Inf. Comput. **75**(2), 87–106 (1987). https://doi.org/10.1016/0890-5401(87)90052-6
3. Case, J.: Gold-style learning theory. In: Heinz, J., Sempere, J.M. (eds.) Topics in Grammatical Inference, pp. 1–23. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-48395-4_1
4. Eyraud, R., Heinz, J., Yoshinaka, R.: Efficiency in the identification in the limit learning paradigm. In: Heinz, J., Sempere, J.M. (eds.) Topics in Grammatical Inference, pp. 25–46. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-48395-4_2
5. Kanazawa, M., Yoshinaka, R.: The strong, weak, and very weak finite context and kernel properties. In: Drewes, F., Martín-Vide, C., Truthe, B. (eds.) LATA 2017. LNCS, vol. 10168, pp. 77–88. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-53733-7_5
6. Kanazawa, M., Yoshinaka, R.: A hierarchy of context-free languages learnable from positive data and membership queries. In: Chandlee, J., Eyraud, R., Heinz, J., Jardine, A., van Zaanen, M. (eds.) Proceedings of the Fifteenth International Conference on Grammatical Inference. Proceedings of Machine Learning Research, vol. 153, pp. 18–31 (2021). https://www.proceedings.mlr.press/v153/kanazawa21a.html
7. Kanazawa, M., Yoshinaka, R.: Extending distributional learning from positive data and membership queries. In: Proceedings of 16th edition of the International Conference on Grammatical Inference. Proceedings of Machine Learning Research, vol. 217, pp. 8–22 (2023). https://proceedings.mlr.press/v217/kanazawa23a.html
8. Leiß, H.: Learning context free grammars with the finite context property: a correction of A. Clark's algorithm. In: Morrill, G., Muskens, R., Osswald, R., Richter, F. (eds.) Formal Grammar 2014. LNCS, vol. 8612, pp. 121–137. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44121-3_8
9. Yoshinaka, R.: Towards dual approaches for learning context-free grammars based on syntactic concept lattices. In: Mauri, G., Leporati, A. (eds.) DLT 2011. LNCS, vol. 6795, pp. 429–440. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22321-1_37

A Short Introduction to SHACL for Logicians

Magdalena Ortiz(✉)

Umeå University,Umeå, Sweden
magdalena.ortiz@cs.umu.se

Abstract. The Shapes Constraint Language (SHACL) was recommended by the W3C in 2017 for describing constraints on web data (specifically, on the so-called RDF graphs) and validating them. At first glance, it may not seem to be a topic for logicians, but as it turns out, SHACL can be approached as a formal logic, and actually quite an interesting one. In this paper, we give a brief introduction to SHACL tailored towards logicians and frame key uses of SHACL as familiar logic reasoning tasks. We discuss how SHACL relates to description logics, which are the basis of the OWL Web Ontology Languages, a related yet orthogonal standard for web data. Finally, we summarize some of our recent work in the SHACL world, hoping that this may shed light on how ideas, results, and techniques from well-established areas of logic can advance the state of the art in this emerging field.

Keywords: SHACL · semantic web · description logics

1 What is SHACL and Why Do We Need It?

One of the most fundamental changes the world has seen in the last decades is the emergence and ultrafast growth of the Web, where almost inconceivable amounts of data of all shapes and forms is shared and interconnected. The World Wide Web Consortium (W3C) has been a key player in this growth: an international community that develops open standards that are used for building the web from documents and data. A particularly influential set of standards are those developed within the *semantic web* initiative, which aims to build a useful *web of data* that is interoperable and understandable to both humans and machines.

The big bulk of shared data on the web uses the RDF standard [28]. In a nutshell, labelled graphs where nodes represent web resources and data values, connected by arrows labelled with different kinds of properties with a standardized meaning. RDF datasets are often described as sets of triples (s, p, o) where the *subject* s and the object o are data items of resources with a unique identifier, and they are connected by property p. For our purposes, it is enough to think of graphs whose edges are labeled with *properties* from a dedicated alphabet.[1]

[1] In RDF, properties are not necessarily disjoint from the nodes, which can be either web identifiers called IRIs, data values given as *literals*, and the so-called *blank nodes*; we omit RDF details from here and refer to [28].

H. H. Hansen et al. (Eds.): WoLLIC 2023, LNCS 13923, pp. 19–32, 2023.
https://doi.org/10.1007/978-3-031-39784-4_2

There are enormous repositories of such data on the web containing millions of nodes and edges; famous examples include *knowledge graphs* like DBPedia[2] which contains several hundred million of facts extracted from Wikipedia, and Yago, a high-quality knowledge base about people, cities, countries, and organizations, containing more than 2 billion facts about 50 million entities.[3] Once RDF became widespread for sharing data on the web, accompanying standards were proposed, like the OWL Web Ontology Languages for describing knowledge domains and for inferring implicit relationships and facts from data on the web [26], and a dedicated query language for RDF data called SPARQL [27]. Web interfaces called *SPARQL end-points* allow any person to ask questions and obtain interesting facts from these knowledge graphs.

A crucial feature of the RDF data format is its *flexibility*: if something can be represented as a labelled graph, then it can be published on the web using RDF. But so much flexibility is a two-edged sword. Users that want to query a source like DBpedia easily find themselves lost and do not know where to start: how do I formulate my query? Which properties may connect a country to its capital city? Is there information about the family relationships of celebrities? Which facts about rivers could I query for?

As the web of linked data kept growing, the pressing need for a *normative* standard emerged: a language that can be used for describing and validating the structure and content of RDF graphs. This language is SHACL, the Shapes Constraint Language, recommended in 2017 by the W3C [29]. It allows users to define "shapes" which may, for example, say that a person has a name and exactly one date of birth, and may be married to another person. Like other W3C standards, SHACL is defined in a long 'specification' document that is very hard to read for anybody that is not familiar with the semantic web jargon. While OWL was standardized on the basis of over two decades of research in *description logics*, a well-understood family of decidable logics [6,26], the younger SHACL did not come to the world equipped with such a robust logic foundation. However, a handful of logic-minded people from the knowledge representation and database theory community have been developing a solid logic-based foundation for SHACL. As it turns out, SHACL can be seen as a simple and elegant logic, and its fundamental *validation* problem is a model checking task very familiar in logic and computer science.

2 SHACL as a Logic

The challenge of extracting from the specification a formal syntax and semantics was tackled by Corman et al. [13], and the majority of the later SHACL works have built on their formalization, e.g. [1,3,4,10,11]. We do the same, and like most of them, we focus on the 'core' of SHACL. Some authors have extended this formalization to cover more SHACL features; see, for example, the extended formalization compiled by Jakubowski [16].

[2] http://dbpedia.org/.
[3] https://yago-knowledge.org/.

2.1 Syntax

The main syntactic object in SHACL are the so-called *shape constraints*, which assign possibly complex *shape expressions* to special predicates called *shape names*.

For writing these, we use an alphabet consisting of three countably infinite pairwise disjoint sets: the set of *shape names* $\mathbf{S}$, the set of *property names* $\mathbf{P}$, and the set of *node names* $\mathbf{N}$. Then *shape expressions* φ and *path expressions* ρ obey the following grammar.

$$\rho ::= p \mid p^- \mid \rho \cup \rho \mid \rho \cdot \rho \mid \rho^*$$
$$\varphi ::= s \mid \top \mid \{a\} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \geq_n \rho.\varphi \mid \rho = \rho$$

where $s \in \mathbf{S}$, $a \in \mathbf{N}$, $p \in \mathbf{P}$, and $n \geq 0$ is a natural number. We may write $\leq_{n-1} \rho.\varphi$ in place of $\neg(\geq_n \rho.\varphi)$, and write $\exists\rho.\varphi$ and $\forall\rho.\varphi$ in place of $\geq_1 \rho.\varphi$ and $\leq_0 \rho.\neg\varphi$, respectively. Those familiar with modal logic will recognise the syntax of multidimensional modal formulas extended with *nominals* $\{a\}$ from hybrid logics [5], graded modalities $\geq_n \rho.\varphi$, converse ρ^-, and reflexive, transitive closure as in propositional dynamic logic [14]. We will revisit this relationship from the perspective of description logics in the next section.

We can now write *shape constraints* of the form

$$s \equiv \varphi$$

where $s \in \mathbf{S}$ and φ is a *shape expression*. In SHACL, shape constraints come together with a set of *targets*, indicating at which nodes of the graph the shapes of interest are to be validated. We focus here on atomic targets of the form $s(a)$ with $s \in \mathbf{S}$ and $a \in \mathbf{N}$. Since shape names are unary predicates, we can read $s(a)$ as 'a is in the interpretation of s'. Then we define a *shapes graph* as a pair $(\mathcal{C}, T)$ of a set $\mathcal{C}$ of shape constraints where each $s \equiv \varphi$ has a different s in the head, and a set T of target atoms.

Example 1. Consider the following shape constraints:

$$\begin{aligned} \mathit{Pizza} &\equiv\ \geq_2 \mathtt{hasTopping}.\top, \\ \mathit{VeggiePizza} &\equiv \mathit{Pizza} \wedge \forall \mathtt{hasTopping}.\mathit{VeggieTopping}, \\ \mathit{VeggieTopping} &\equiv \{\mathtt{mozzarella}\} \vee \{\mathtt{tomato}\} \vee \{\mathtt{basil}\} \vee \{\mathtt{artichoke}\} \end{aligned}$$

Intuitively, these constraints define the shape 'pizza' as having at least two toppings, and vegetarian pizzas as pizzas having only vegetarian toppings. A shape can be defined by directly listing the nodes in it, as done here for vegetarian toppings. To give a rough impression of the way this is written in usual SHACL documents, we show in Fig. 1 an incomplete declaration of the first constraint in SHACL machine-readable syntax.

```
a sh:NodeShape ;
sh:property [
    sh:path pizza:hasTopping ;
    sh:minCount 2    ] .
```

Fig. 1. A SHACL shape in machine-readable syntax

2.2 Semantics

We now define the so-called *supported model semantics* for SHACL, and discuss other semantics in Sect. 4.1.

Like other logic formalisms, the semantics of SHACL can be elegantly defined using interpretations. For our purposes, it will be enough to consider interpretations whose domains are nodes from $\mathbf{N}$. We consider interpretations $\mathcal{I}$ consisting of a non-empty domain $\Delta \subseteq \mathbf{N}$, and an interpretation function $\cdot^{\mathcal{I}}$ that maps

- each shape name $s \in \mathbf{S}$ a set $s^{\mathcal{I}} \subseteq \Delta$, and
- each property name to a set of pairs $s^{\mathcal{I}} \subseteq \Delta \times \Delta$.

The interpretation function $\mathcal{I}$ is inductively extended to complex expressions, see Fig. 2. Note that path expressions ρ are interpreted as binary relations $\rho^{\mathcal{I}}$ over Δ, while shape expressions φ are interpreted as sets $\varphi^{\mathcal{I}} \subseteq \Delta$.

$$
\begin{aligned}
(p^-)^{\mathcal{I}} =& \{(d',d) \mid (d,d') \in p^{\mathcal{I}}\}\\
(\rho \cup \rho')^{\mathcal{I}} =& \rho^{\mathcal{I}} \cup \rho'^{\mathcal{I}}\\
(\rho \cdot \rho')^{\mathcal{I}} =& \{(c,d) \in \Delta \times \Delta \mid \text{there is some } d' \text{ with } (c,d') \in \rho^{\mathcal{I}}, (d',d) \in \rho^{\mathcal{I}}\}\\
(\rho^*)^{\mathcal{I}} =& \{(d,d) \mid d \in \Delta \times \Delta\} \cup (\rho)^{\mathcal{I}} \cup (\rho \cdot \rho)^{\mathcal{I}} \cup \cdots
\end{aligned}
$$

$$
\begin{aligned}
\{a\}^{\mathcal{I}} &= \{a\}\\
\top^{\mathcal{I}} &= \Delta\\
(\neg\varphi)^{\mathcal{I}} &= \Delta \setminus \varphi^{\mathcal{I}}\\
(\varphi_1 \wedge \varphi_2)^{\mathcal{I}} &= {\varphi_1}^{\mathcal{I}} \cap {\varphi_2}^{\mathcal{I}}\\
(\varphi_1 \vee \varphi_2)^{\mathcal{I}} &= {\varphi_1}^{\mathcal{I}} \cup {\varphi_2}^{\mathcal{I}}\\
(\geq_n \rho.\varphi)^{\mathcal{I}} &= \{d \in \Delta \mid \text{there exist distinct } d_1,\ldots,d_n\\
&\qquad \text{with } (d,d_i) \in \rho^{\mathcal{I}} \text{ and } d_i \in \varphi^{\mathcal{I}} \text{ for each } 1 \leq i \leq n\}\\
(\rho = \rho')^{\mathcal{I}} &= \{d \in \Delta \mid \text{ for all } d' \in \Delta, (d,d') \in \rho^{\mathcal{I}} \text{ iff } (d,) \in \rho'^{\mathcal{I}}\}
\end{aligned}
$$

Fig. 2. Interpretation of path and shape expressions

We say that $\mathcal{I}$ *satisfies a constraint* $s \equiv \varphi$ if $s^{\mathcal{I}} = \varphi^{\mathcal{I}}$, and $\mathcal{I}$ *satisfies a shapes graph* $(\mathcal{C}, T)$ if it satisfies all constraints in $\mathcal{C}$ and additionally it contains all the targets, that is, $a \in s^{\mathcal{I}}$ for every $s(a) \in T$.

SHACL shapes graphs are intended to be validated over an input *data graph* (essentially a knowledge graph or RDF graph). A *data graph* $G = (N, E, \ell)$ is defined as a graph with vertices $N \subseteq \mathbf{N}$ and a labelling function $\ell : E \rightarrow 2^{\mathbf{P}}$, that is, edges are labelled with sets of property names from $\mathbf{P}$, and a graph is just a collection of $\mathbf{P}$-indexed binary relations on N.

Given such a graph $G = (N, E, \ell)$, we say that the interpretation $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$ is a *shape adornment for* G if $\Delta = \mathbf{N}$ and $p^{\mathcal{I}} = \{(a, b) \in E \mid p \in \ell((a, b))\}$ for each property p. That is, $\mathcal{I}$ is a shape adornment of G if properties p are interpreted as relations as specified by G. Note that G does not determine the interpretation of the shape names. In modal logic terms, we can call G a multi-relational Kripke frame, and $\mathcal{I}$ is a multi-relational Kripke model.
In SHACL, the main problem of interest is the *validation* of given constraints and targets on a given graph.

Definition 1 (SHACL validation). *We say that* a data graph G validates a shapes graph $(\mathcal{C}, T)$ *if there exists a shape adornment for* G *that satisfies* $(\mathcal{C}, T)$.

The SHACL validation problem *consists of deciding, for a given* G *and* $(\mathcal{C}, T)$, *whether* G *validates* $(\mathcal{C}, T)$.

Example 2. Consider the following shape constraints:

$$\begin{aligned}\mathcal{C}_{\texttt{Pizza}} = \{ \mathit{Pizza} &\equiv\ \geq_2 \texttt{hasTopping}.\top, \\ \mathit{VeggiePizza} &\equiv \mathit{Pizza} \wedge \forall \texttt{hasTopping}.\mathit{VeggieTopping}, \\ \mathit{VeggieTopping} &\equiv \{\texttt{mozzarella}\} \vee \{\texttt{tomato}\} \vee \{\texttt{basil}\} \vee \{\texttt{artichoke}\} \quad \}\end{aligned}$$

The graph $G_{\texttt{Pizza}}$ in Fig. 3 validates the target $\mathit{Pizza}(\texttt{pizza\_capricciosa})$, but it does not validate the target $\mathit{VeggiePizza}(\texttt{pizza\_capricciosa})$ since there is no shape adornment $\mathcal{I}$ satisfying $\mathcal{C}_{\texttt{Pizza}}$ where $\texttt{prosciutto} \in \mathit{VeggieTopping}^{\mathcal{I}}$.

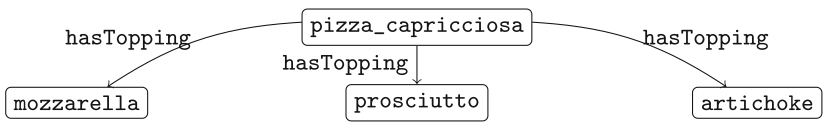


Fig. 3. A pizza data graph $G_{\texttt{Pizza}}$

3 SHACL, OWL and Description Logics

The syntax of shape expressions is more than familiar to those acquainted with *description logics*, a well-studied family of decidable logics tailored for knowledge representation and reasoning [6,7], and the logics underpinning the OWL standard. In fact, if instead of shape names we call the symbols in **S** *concept names*, then exactly the same grammar defines *concept expressions* φ in a description logic that we will call $\mathcal{ALCOIQ}^{=}_{reg}$. It extends the well-known description logic $\mathcal{ALCOIQ}$ with regular role expressions and equalities thereof. *Concept definitions* take the form

$$s \equiv \varphi$$

Hence, there is basically no difference between SHACL constraints and sets of $\mathcal{ALCOIQ}^{=}_{reg}$ concept definitions, at least syntactically.

In description logics, one often considers not only concept definitions but a more general form of *concept inclusions* $\varphi_1 \sqsubseteq \varphi_2$, and a set of such inclusions is called an *ontology*. The variant of OWL called OWL-DL allows writing ontologies in a DL called $\mathcal{SHOIQ}$ that is quite similar to $\mathcal{ALCOIQ}^{=}_{reg}$ [15]. The main differences are that the concept constructor $\rho = \rho'$ is omitted, and instead of regular path expressions ρ we can only use property names p and their inverses p^- inside concept expressions. $\mathcal{SHOIQ}$ also allows for subproperty relations, not supported in SHACL, and for declaring a set of property names that must be interpreted as transitive relations, which can be used inside concept expressions of the form $\exists p.\varphi$ and $\forall p.\varphi$.

Semantically they are closely related too. However, we must pay attention to some subtle details. The semantics of DLs is also defined in terms of interpretations $\mathcal{I}$ as above, and the satisfaction of concept definitions is just as for SHACL constraints. For the more general concept inclusions $\varphi_1 \sqsubseteq \varphi_2$ we have satisfaction if $\varphi_1^{\mathcal{I}} \subseteq \varphi_2^{\mathcal{I}}$, as expected. An interpretation $\mathcal{I}$ that satisfies all the inclusions in an ontology $\mathcal{O}$ is called a *model* of $\mathcal{O}$, in symbols, $\mathcal{I} \models \mathcal{O}$. When we pair an ontology with a data graph G (typically called an ABox in description logics jargon), then we require $\mathcal{I}$ to model G as well, that is, to contain all the facts given in the graph. Formally, we say that $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$ models $G = (N, E, \ell)$ if $N \subseteq \Delta$ and $\{(a, b) \in E \mid p \in \ell(a, b)\} \subseteq p^{\mathcal{I}}$ for each property p. $\mathcal{I}$ is a model of $(\mathcal{O}, G)$ if $\mathcal{I}$ models both $\mathcal{O}$ and G. We say that $(\mathcal{O}, G)$ is *consistent* if it admits a model, and we say that a fact $s(a)$ (resp. $p(a, b)$) is *entailed by* $(\mathcal{O}, G)$ if $a \in s^{\mathcal{I}}$ (resp. $(a, b) \in p^{\mathcal{I}}$) for every model $\mathcal{I}$ of $(\mathcal{O}, G)$.

We stress here that for reasoning in description logics, we typically consider *all models*, in contrast to SHACL, where we restrict our attention to shape adornments. In fact, SHACL can be seen as a special instance of $\mathcal{ALCOIQ}^{=}_{reg}$ with *closed predicates* [20,24]. Closed predicates are a well-known extension of description logics for reasoning in settings where complete and incomplete information co-exist. A selected set of concepts and roles is declared to be *closed*, and the models of the ontology are restricted to those that interpret the closed predicates exactly as in the data. Shape adornments in SHACL precisely coincide with the models in $\mathcal{ALCOIQ}^{=}_{reg}$ when *all roles are closed*.

The following example illustrates the difference between SHACL validation and description logic entailment.

Example 3. Consider the graph $G'_{\texttt{Pizza}}$ in Fig. 4, and the following shape constraints:

$$\begin{aligned}\mathcal{C}'_{\texttt{Pizza}} = \{\, & \mathit{VeggiePizza} \equiv \forall \texttt{hasTopping}.\mathit{VeggieTopping},\\ & \mathit{VeggieTopping} \equiv \{\texttt{mozzarella}\} \vee \{\texttt{tomato}\} \vee \{\texttt{basil}\} \vee \{\texttt{artichoke}\}\}\end{aligned}$$

Then $G'_{\texttt{Pizza}}$ validates the target $\mathit{VeggiePizza}(\texttt{pizza\_margherita})$, as witnessed by the shape adornment $\mathcal{I}$ that assigns the nodes `mozzarella`, `tomato` and `basil` to the shape *VeggieTopping* and the node `pizza_margherita` to *VeggiePizza*.

Assume now that *VeggieTopping* and *VeggiePizza* are concept names, and consider the description logic ontology:

$$\begin{aligned}\mathcal{O}_{\texttt{Pizza}} = \{\, & \mathit{VeggiePizza} \equiv \forall \texttt{hasTopping}.\mathit{VeggieTopping},\\ & \mathit{VeggieTopping} \equiv \{\texttt{mozzarella}\} \vee \{\texttt{tomato}\} \vee \{\texttt{basil}\} \vee \{\texttt{artichoke}\}\}\end{aligned}$$

The interpretation $\mathcal{I}$ above is a model of $\mathcal{O}_{\texttt{Pizza}}, G'_{\texttt{Pizza}}$. However, the following $\mathcal{I}'$ is also a model:

$$\begin{aligned}\mathit{VeggieTopping}^{\mathcal{I}'} &= \mathit{VeggieTopping}^{\mathcal{I}}\\ \mathit{hasTopping}^{\mathcal{I}'} &= \mathit{hasTopping}^{\mathcal{I}} \cup \{(\texttt{pizza\_margherita}, \texttt{new\_topping})\}\\ \mathit{VeggiePizza}^{\mathcal{I}'} &= \emptyset\end{aligned}$$

where `pizza_margherita` has an additional non-vegetarian topping. This shows that $\mathcal{O}_{\texttt{Pizza}}, G'_{\texttt{Pizza}}$ does not entail $\mathit{VeggiePizza}(\texttt{pizza\_margherita})$.

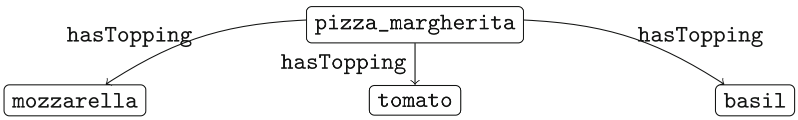


Fig. 4. Another pizza data graph $G'_{\texttt{Pizza}}$

3.1 Reasoning in SHACL and in OWL

The fundamental difference in purpose between SHACL and OWL means that their reasoning problems are also different. In SHACL we are interested in whether the input graph validates the constraints, which is essentially a *model checking* problem. In contrast, in description logics we focus on *inferring* information: determining whether facts and inclusions are *entailed* by the ontology, that is, whether they are true in *all models*. Moreover, models may extend the graph and make it arbitrarily large. In fact, even for significantly less expressive description logics than $\mathcal{ALCOIQ}^{=}_{reg}$—such as $\mathcal{ALCIF}$—there are ontologies that only have models with an infinite domain.

The different nature of SHACL reasoning vs traditional description logic reasoning matters because, as we know very well in logic, entailment is computationally much more challenging than model checking. One can naturally define logic reasoning problems for SHACL, like *satisfiability* or *containment* of constraints [19,21]. But the connection to DLs allows us to immediately observe that such problems are practically always undecidable. Consider the *satisfiability of SHACL constraints*: given a set of constraints $\mathcal{C}$, is there a graph for which an interpretation satisfying the constraints exists? This problem is precisely the $\mathcal{ALCOIQ}^{=}_{reg}$ satisfiability problem, which has been known to be undecidable for decades. Indeed, the equality between regular role expressions $\rho = \rho'$ is a variation of the well-known *role-value maps* $p_1 \cdot \cdots \cdot p_n \subseteq p'_1 \cdot \cdots \cdot p'_m$ that were present already in the very early description logic KL-ONE. Schmidt-Schauß proved in 1989 that role-value maps make inference in KL-ONE undecidable [22], one of the oldest undecidability results in the field. It is widely known that without strong restrictions on the role-value maps, not even the weakest of DLs remain decidable [8]. Even without path equalities, allowing path expressions in the counting constructors $\geq_n \rho.\varphi$ and $\leq_n \rho.\varphi$ is another well-known cause of undecidability [18]. If we only allow property names and their inverses in counting concepts, and restrict complex property paths to allowing expressions of the forms $\exists r^*.\varphi$ and $\forall r^*.\varphi$, then we end up with the description logic $\mathcal{ALCOIQ}^*$: the decidability of which is a very long-standing open problem in description and modal logic [17].

These straightforward observations already make clear that SHACL satisfiability and containment can only be decidable for rather restricted fragments. For instance, we can restrict ourselves to $\mathcal{ALCOIQ}$, which only allows property names and their inverses, and obtain decidability. (We note that this logic is closely related to $\mathcal{SHOIQ}$ mentioned above, and their satisfiability problems are interreducible). A detailed study of fragments of SHACL with (un)decidable satisfiability and containment problems has been done by Pareti et al. [21], while Leinberger et al. [19] have shown cases where containment is decidable by reducing the problem to description logic reasoning.

Description logics also tell us a lot about the complexity of satisfiability and containment in SHACL fragments. But the news is not particularly positive and their worst-case complexity is high. Indeed, in the $\mathcal{ALCOIQ}$ fragment, satisfiability is complete for NExpTime [25]. (Here the ontology is considered as input, that is, we are talking of *ontology complexity* or *combined complexity*). On the positive side, $\mathcal{ALCOIQ}$ and $\mathcal{SHOIQ}$ are supported by off-the-shelf reasoners which can handle efficiently large real-world ontologies, and these reasoners can be directly deployed for deciding SHACL satisfiability and containment in the corresponding fragments.

4 What has Logic Done for SHACL?

Viewing SHACL as a logic allows us to transfer important insights, techniques and results from other areas of logic. We already illustrated how we can obtain some (un)decidability and complexity results directly from description logics. In this section we briefly summarize a few recent results of our research group that also illustrate how the logic view of SHACL can be a stepping stone to providing robust solutions to some SHACL open problems.

4.1 Semantics of Recursive SHACL

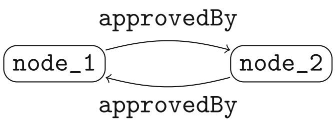
Our definition of SHACL expressions above imposes no constraints on the occurrences of shape names in the definitions of other shape names, that is, it allows for *recursion*. Given a set of SHACL constraints $\mathcal{C}$, its *dependency graph* has a node for each shape name occurring in $\mathcal{C}$, and there is an arc from s to s' if s' occurs in the body φ of a constraint $s \equiv \varphi$ for s. $\mathcal{C}$ is called *recursive* if this graph contains a cycle.[4] The SHACL specification does not disallow such recursion, but when the semantics of validation is defined, one encounters a surprise:

> "The validation with recursive shapes is not defined in SHACL and is left to SHACL processor implementations. For example, SHACL processors may support recursion scenarios or produce a failure when they detect recursion." *SHACL Recommendation* [29], §3.4.3

Naturally, this lack of proper validation semantics in the presence of recursion was one of the first SHACL open problems to be addressed by means of formal logic. The semantics that we have presented here is often called *supported model semantics*, and it was proposed already by Corman et al. when they first formalized SHACL [13]. However, this semantics has not been free of criticism.

Example 4. Consider the following constraint, stating that a node may be certified if it either has a certificate or if it has been approved by a node that is certified. Consider the graph $G_{\texttt{cc}}$ below.

$$\mathit{certifiedNode} \equiv \exists \texttt{hasCertificate}.\mathit{Certificate} \vee \exists \texttt{approvedBy}.\mathit{certifiedNode}$$



$G_{\texttt{cc}}$ validates the target $\mathit{certifiedNode}(\texttt{node\_1})$, as witnessed by the adornment

$$\mathit{certifiedNode}^{\mathcal{I}} = \{\texttt{node\_1}, \texttt{node\_2}\}.$$

Intuitively, node_1 can be considered certified because it was approved by node_2, which in turn is certified because it was approved by node_1, although no node has any legitimate certification.

[4] Note that this *monadic* recursion over shape names is orthogonal to the linear recursion over properties present in the path expressions ρ.

The stable model semantics [4] and the well-founded semantics [12] were both proposed for avoiding these dubious validations and instead only allow for validations that are based on proper well-founded assignments. Both semantics are not only intuitive, but they are also computationally more manageable. Unlike supported validation, stable validation can be decided in polynomial time if the constraints are *stratified*, which intuitively allows only for *positive* recursion cycles. Here we refer to *data complexity*, which assumes that the constraints are fixed and measures the complexity in terms of the size of the data graph only. Well-founded validation is particularly interesting since it is *always* computable in polynomial time; it yields a three-valued approximation of stable models, and coincides with it whenever there is no recursion involving negation. These results witness the value of building on the decades of experience of the logic programming and non-monotonic reasoning community when defining proper semantics for full recursive SHACL, an aspect emphasized in [9]. Techniques for efficient goal-oriented validation in the presence of recursion, like *magic sets*, have been successfully applied to SHACL [3].

4.2 Explaining Non-Validation

The SHACL specification calls for the so-called *validation reports*, which are meant to explain to the users the outcome of validating an RDF graph against a collection of constraints. The specification gives some details about how these reports should look, e.g., which fields they should contain (e.g., the node(s) and value(s) that caused the failure of some target), but it does not address the problem of what does it mean to 'cause' a failure, and how to find the causes when a test does not succeed. These questions are far from obvious.

In our recent work [1] we draw inspiration from logic-based abduction and database repairs to study the problem of explaining non-validation of SHACL constraints. In our framework non-validation is explained using the notion of a *repair*, i.e., a collection of additions and deletions whose application on an input graph results in a repaired graph that does satisfy the given SHACL constraints.

Example 5. Consider the following shapes graph $(\mathcal{C}_t, T_t)$ and data graph G_t.

$$\begin{aligned}\mathcal{C}_t =&\{\, \mathit{Teacher} \leftrightarrow \exists \mathtt{teaches}.\top\ ,\\ &\quad \mathit{Student} \leftrightarrow \exists \mathtt{enrolledIn}.\top \wedge \neg\, \mathit{Teacher}\}\\ T_t =&\{\mathit{Student}(\mathtt{Ben}), \mathit{Teacher}(\mathtt{Ann})\}\\ G_t =&\{\mathtt{enrolledIn}(\mathtt{Ben}, \mathtt{c}_1), \mathtt{teaches}(\mathtt{Ben}, \mathtt{c}_2)\}\end{aligned}$$

G_t does not validate either target. To validate $\mathit{Student}(\mathtt{Ben})$ we need to remove $\mathtt{teaches}(\mathtt{Ben}, \mathtt{c}_2)$, while to validate $\mathit{Teacher}(\mathtt{Ann})$ we need to add a fact $\mathtt{teaches}(\mathtt{Ann}, \mathtt{c})$ for some $c \in \mathbf{N}$. We call the pair (A, D) of *Additions* $A = \{\mathtt{teaches}(\mathtt{Ann}, \mathtt{c})\}$ and *Deletions* $D = \{\mathtt{teaches}(\mathtt{Ben}, \mathtt{c}_2)\}$ an *explanation* for the SHACL validation problem above.

Since sometimes we need to introduce fresh nodes, in order to keep things computationally manageable, we assume that the set of acceptable additions is somehow given as an (implicit or explicit) data graph H. Then we can define our SHACL explanations as follows.

Definition 2 ([1]). *Let G be a data graph, let $(\mathcal{C}, T)$ be a shapes graph, and let the set of* hypotheses *H be a data graph disjoint from G. Then $\Psi = (G, \mathcal{C}, T, H)$ is a* SHACL Explanation Problem (SEP)*. An* explanation *for Ψ is a pair (A, D), such that (a) $D \subseteq G$, $A \subseteq H$, and (b) $(G \setminus D) \cup A$ validates $(\mathcal{C}, T)$.*

A preference order *is a preorder $\preceq$ on the set of explanations for Ψ. A* preferred explanation *of a SEP Ψ under the $\preceq$, or a* $\preceq$-explanation *for short, is an explanation ξ such that there is no explanation ξ' for Ψ with $\xi' \preceq \xi$ and $\xi \not\preceq \xi'$.*

The following decision problems for explanations are defined:

- $\preceq$-IsExpl*: is a given pair (A, D) a $\preceq$-explanation for Ψ?*
- $\preceq$-Exist*: does there exist a $\preceq$-explanation for Ψ?*
- $\preceq$-NecAdd*: is α a* $\preceq$-necessary *addition for Ψ, that is does α occur in A in* every *$\preceq$-explanation (A, D) for Ψ?*
- $\preceq$-NecDel*: is α a* $\preceq$-necessary *deletion for Ψ, that is does α occur in D in* every *$\preceq$-explanation (A, D) for Ψ?*
- $\preceq$-RelAdd*: is α a* $\preceq$-relevant *addition for Ψ, that is does α occur in A in* some *$\preceq$-explanation (A, D) for Ψ?*
- $\preceq$-RelDel*: is α a* $\preceq$-relevant *deletion for Ψ, that is does α occur in D in* some *$\preceq$-explanation (A, D) for Ψ?*

We studied all these decision problems for the following preorders $\preceq$:

- the *subset relation* $(A, D) \subseteq (A', D')$, defined as $A' \subseteq A$ and $D' \subseteq D$,
- the *cardinality relation* $(A, D) \leq (A', D')$, defined as $|A| + |D| \leq |A'| + |D'|$,
- the identity; in this case, we may talk of 'no preference order' and omit $\preceq$.

We characterized the computational complexity of all of them, in the general case and in the non-recursive case. We also analyzed the effect on the complexity of restricting the set of predicates that can be added or removed. Most of the problems turned out to be intractable, up to the second level of the polynomial hierarchy, but some problems can be solved in polynomial time. The results are summarised in Table 1, see [1] for details.

Table 1. The complexity of SHACL Explanation Problems (completeness results). The bounds hold also under signature restrictions and for non-recursive SHACL.

pref. order	IsExpl	Exist	NecAdd	NecDel	RelAdd	RelDel
$\emptyset$	NP	NP	coNP	coNP	NP	NP
$\subseteq$	DP	NP	coNP	coNP	Σ_2^P	Σ_2^P
$\leq$	DP	NP	$P^{\parallel \mathrm{NP}}$	$P^{\parallel \mathrm{NP}}$	$P^{\parallel \mathrm{NP}}$	$P^{\parallel \mathrm{NP}}$

Our algorithms can be a stepping stone for automatically computing such explanations, even in intractable cases. For instance, a follow-up work of some co-authors used Answer Set Programming (ASP) to implement a prototype tool for repairing SHACL specifications [2].

5 Conclusions and Outlook

The recent emergence of SHACL as a standard for web data provides fresh evidence that, in the complex data landscape of today's world, increasingly flexible new formalisms for describing, validating, and managing data are still needed, and approaches grounded in formal logic are as important as ever. SHACL has profound connections to well-established research fields of logic in computer science, in particular to two related communities: description logics on the one hand, and logic programming and non-monotonic reasoning on the other. The debates about the semantics of negation are reminiscent of the challenges that the latter community faced already in the 1990s,s, and in retrospect, maybe more active communication between these two fields could have helped avoid the tortuous road towards the different semantics for validation of recursive SHACL.

It is hard to overstate the similarity between SHACL and description logics. We gave a few examples of (un)decidability and complexity results that can be immediately transferred to the SHACL world, but there is much more to leverage in this connection. The vast trove of computational complexity and decidability results accumulated by the community can guide further SHACL developments, and the use of SHACL may bring forward new problems that have not yet been addressed in description logics.

The open- versus the closed-world assumption is often emphasized as a key difference between OWL and SHACL. And discussed, SHACL can be seen as a DL where properties are *closed predicates*. This points once more towards an ever-present challenge in knowledge representation and reasoning: **combining open- and closed-world reasoning**. Neither of them is enough on its own, since more often than not complete and incomplete information co-exist, and useful inferences call for leveraging this partial completeness.

In our ongoing work we are studying, for example, how to do validation in the presence of ontologies, that is, taking implicit facts into account in the validation. We are also exploring techniques for SHACL validation when the graph is subjected to updates, and we are continuing our work on explanations for SHACL in order to devise better and more useful validation reports.

Many open problems remain ahead, and SHACL is a relatively young field where technologies are still being constructed. We hope that this short journey through SHACL and some of its challenges may inspire more logicians to explore SHACL and other emerging data management solutions and to try to contribute to that field by bringing insights from well-established areas of logic [23].

Acknowledgments. The work summarized here has been carried out with many collaborators, to whom I want to express my sincere thanks. I want to thank in particular the current and former members of our team Mantas Šimkus, Shqiponja Ahmetaj, Anouk Oudshoorn, Medina Andresel and Bianca Löhnert. Thank you also to Juan Reutter and Julien Corman. This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. It was also partially supported by the Austrian Science Fund (FWF) projects P30360 and P30873.

References

1. Ahmetaj, S., David, R., Ortiz, M., Polleres, A., Shehu, B., Simkus, M.: Reasoning about explanations for non-validation in SHACL. In: KR, pp. 12–21 (2021)
2. Ahmetaj, S., David, R., Polleres, A., Simkus, M.: Repairing SHACL constraint violations using answer set programming. In: Sattler, U., et al. (eds.) The Semantic Web – ISWC 2022. ISWC 2022. LNCS, vol. 13489, pp. 375–391. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-19433-7_22
3. Ahmetaj, S., Löhnert, B., Ortiz, M., Simkus, M.: Magic shapes for SHACL validation. Proc. VLDB Endow. **15**(10), 2284–2296 (2022)
4. Andresel, M., Corman, J., Ortiz, M., Reutter, J.L., Savkovic, O., Šimkus, M.: Stable model semantics for recursive SHACL. In: Proceedings of the Web Conference 2020, pp. 1570–1580. ACM (2020). https://doi.org/10.1145/3366423.3380229
5. Areces, C., ten Cate, B.: Hybrid logics. In: Blackburn, P., van Benthem, J.F.A.K., Wolter, F. (eds.) Handbook of Modal Logic, Studies in Logic and Practical Reasoning, vol. 3, pp. 821–868. North-Holland (2007). https://doi.org/10.1016/s1570-2464(07)80017-6
6. 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)
7. Baader, F., Horrocks, I., Lutz, C., Sattler, U.: An Introduction to Description Logic. Cambridge University Press, Cambridge (2017)
8. Baader, F., Théron, C.: Role-value maps and general concept inclusions in the minimal description logic with value restrictions or revisiting old skeletons in the DL cupboard. Künstliche Intell. **34**(3), 291–301 (2020)
9. Bogaerts, B., Jakubowski, M.: Fixpoint semantics for recursive SHACL. In: ICLP Technical Communications. EPTCS, vol. 345, pp. 41–47 (2021)
10. Bogaerts, B., Jakubowski, M., den Bussche, J.V.: Expressiveness of SHACL features. In: ICDT. LIPIcs, vol. 220, pp. 15:1–15:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2022)
11. Bogaerts, B., Jakubowski, M., den Bussche, J.V.: SHACL: a description logic in disguise. In: Gottlob, G., Inclezan, D., Maratea, M. (eds.) Logic Programming and Nonmonotonic Reasoning. LPNMR 2022. LNCS, vol. 13416, pp. 75–88. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-15707-3_7
12. Chmurovic, A., Simkus, M.: Well-founded semantics for recursive SHACL. In: Datalog. CEUR Workshop Proceedings, vol. 3203, pp. 2–13. CEUR-WS.org (2022)
13. Corman, J., Reutter, J.L., Savković, O.: Semantics and validation of recursive SHACL. In: Vrandečić, D., et al. (eds.) ISWC 2018. LNCS, vol. 11136, pp. 318–336. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00671-6_19
14. Harel, D., Kozen, D., Tiuryn, J.: Dynamic Logic. MIT Press, Cambridge (2000)
15. Horrocks, I., Sattler, U.: A tableau decision procedure for SHOIQ. J. Autom. Reason. **39**(3), 249–276 (2007)

16. Jakubowski, M.: Formalization of SHACL. Technical report, (unpublished) (2021). www.mjakubowski.info/files/shacl.pdf. Accessed 15 Apr 2023
17. Kaminski, M., Smolka, G.: A goal-directed decision procedure for hybrid PDL. J. Autom. Reason. **52**(4), 407–450 (2014)
18. Kazakov, Y., Sattler, U., Zolin, E.: How many legs do i have? Non-simple roles in number restrictions revisited. In: Dershowitz, N., Voronkov, A. (eds.) LPAR 2007. LNCS (LNAI), vol. 4790, pp. 303–317. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-75560-9_23
19. Leinberger, M., Seifer, P., Rienstra, T., Lämmel, R., Staab, S.: Deciding SHACL shape containment through description logics reasoning. In: Pan, J.Z., et al. (eds.) ISWC 2020. LNCS, vol. 12506, pp. 366–383. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-62419-4_21
20. Ngo, N., Ortiz, M., Simkus, M.: Closed predicates in description logics: results on combined complexity. In: KR, pp. 237–246. AAAI Press (2016)
21. Pareti, P., Konstantinidis, G., Mogavero, F.: Satisfiability and containment of recursive SHACL. J. Web Semant. **74**, 100721 (2022)
22. Schmidt-Schauß, M.: Subsumption in KL-ONE is undecidable. In: Brachman, R.J., Levesque, H.J., Reiter, R. (eds.) Proceedings of the 1st International Conference on Principles of Knowledge Representation and Reasoning (KR'89). Toronto, Canada, 15–18 May 1989, pp. 421–431. Morgan Kaufmann (1989)
23. Schneider, T., Simkus, M.: Ontologies and data management: a brief survey. Künstliche Intell. **34**(3), 329–353 (2020)
24. Seylan, I., Franconi, E., de Bruijn, J.: Effective query rewriting with ontologies over DBoxes. In: IJCAI, pp. 923–925 (2009)
25. Tobies, S.: The complexity of reasoning with cardinality restrictions and nominals in expressive description logics. J. Artif. Intell. Res. **12**, 199–217 (2000)
26. World Wide Web Consortium: OWL 2 Web Ontology Language Primer. World Wide Web Consortium (W3C) (2009). www.w3.org/TR/owl2-primer/
27. World Wide Web Consortium: Sparql 1.1 query language (2013). www.w3.org/TR/sparql11-query/
28. World Wide Web Consortium: RDF 1.1 Primer. World Wide Web Consortium (W3C) (2014). www.w3.org/TR/rdf11-primer/
29. World Wide Web Consortium: Shape Constraint Language (SHACL). W3C World Wide Web Consortium (2017). www.w3.org/TR/shacl

From Gödel's Incompleteness Theorem to the Completeness of Bot Beliefs
(Extended Abstract)

Dusko Pavlovic(✉) and Temra Pavlovic

University of Hawaii, Honolulu, HI, USA
dusko@hawaii.edu

Abstract. Hilbert and Ackermann asked for a method to consistently extend incomplete theories to complete theories. Gödel essentially proved that any theory capable of encoding its own statements and their proofs contains statements that are true but not provable. Hilbert did not accept that Gödel's construction answered his question, and in his late writings and lectures, Gödel agreed that it did not, since theories can be completed incrementally, by adding axioms to prove ever more true statements, as science normally does, with completeness as the vanishing point. This pragmatic view of validity is familiar not only to scientists who conjecture test hypotheses but also to real-estate agents and other dealers, who conjure claims, albeit invalid, as necessary to close a deal, confident that they will be able to conjure other claims, albeit invalid, sufficient to make the first claims valid. We study the underlying logical process and describe the trajectories leading to testable but unfalsifiable theories to which bots and other automated learners are likely to converge.

1 Introduction

Logic as the theory of theories was originally developed to prove true statements. Here we study developments in the opposite direction: modifying interpretations to make true some previously false statements. In modal logic, such logical processes have been modeled as instances of *belief update* [2,3,10]. In the practice of science, such processes arise when theories are updated to explain new observations [22, Ch. 4]. In public life, the goal of such processes is to influence some public perceptions to better suit some private preferences [11, Part V]. This range of applications gave rise to a gamut of techniques of influence and belief engineering, from unsupervised learning to conditioning.

From Incomplete Theories to Complete Beliefs. The idea to incrementally complete incomplete theories [9] arose soon after Gödel proved his Incompleteness Theorem [14]. Alan Turing wrote a thesis about ordinal towers of completions and discovered the hierarchy of unsolvability degrees [35]. The core idea

D. Pavlovic—Partially supported by NSF and AFOSR.

H. H. Hansen et al. (Eds.): WoLLIC 2023, LNCS 13923, pp. 33–50, 2023.
https://doi.org/10.1007/978-3-031-39784-4_3

was to keep recognizing and adding true but unprovable statements to theories. In the meantime, interests shifted from making true statements provable to making false statements true. Many toy examples of belief updates and revisions have been formalized and studied in dynamic-epistemic logic [4,7], but the advances in belief engineering and the resulting industry of influence overtook the theory at great speed, and turned several corners of market and political monetization. The theory remained fragmented even on its own. While modal presentations of Gödel's theorems appeared early on [33], the computational ideas, that made his self-referential constructions possible [34], never transpired back into modal logic. *The point of the present paper is that combining belief updates with universal languages and self-reference leads to a curious new logical capability, whereby theories and models can be steered to assure consistency and completeness of future updates.* This capability precludes disproving current beliefs and the framework becomes *belief-complete* in a suitable formal sense, discussed below.

The logical framework combining belief updates and universal languages may seem unfamiliar. The main body of this paper is devoted to an attempt to describe how it arises from familiar logical frameworks. Here we try to clarify the underlying ideas.

Universality. Just like Gödel's incompleteness theorems, our constructions of unfalsifiable beliefs are based on a *universal language* $\mathbb{L}$. The abstract characterization of universality, which we borrow from [24, Ch. 2], is that $\mathbb{L}$ comes equipped with a family of *interpreters* $\{\}\colon \mathbb{L} \times A \longrightarrow B$, one for each pair of types[1] A, B, such that every function $f\colon A \longrightarrow B$ has a description[2] $\ulcorner f \urcorner$ in $\mathbb{L}$, satisfying[3]

$$f = \{\ulcorner f \urcorner\} \tag{1}$$

This is spelled out in Sect. 4. The construction in Sect. 5 will imply that every $g\colon \mathbb{L} \times A \longrightarrow B$ has a fixpoint Γ, satisfying

$$g(\Gamma, a) = \{\Gamma\}(a) \tag{2}$$

Any complete programming language can be used as $\mathbb{L}$. Its interpreters support (1) and its specializers induce (2). A sufficiently expressive software specification framework [28] would also fit the bill, as would a general scientific formalism [22].

Gödel's Incompleteness: True But Unprovable Statement. Gödel used the set of natural numbers $\mathbb{N}$ as $\mathbb{L}$, with arithmetic as a programming language. The concept of a programming language did not yet exist, but it came into existence through Gödel's construction. An arithmetic expression specifying a function f was encoded as a number $\ulcorner f \urcorner$ and decoded by an arithmetic function $\{\}\colon \mathbb{N} \times \mathbb{N} \longrightarrow \mathbb{N}$ as in (1). A restriction of (2) was proved for arithmetic

[1] Each pair carries a different interpreter $\{\}^{AB}$ but we elide the superscripts.
[2] There may be many descriptions for each f and $\ulcorner f \urcorner$ refers to an arbitrary one.
[3] The curly bracket notation allows abbreviating $\lambda a.\{\}(p, a)$ to $\{p\}$.

predicates $p \colon \mathbb{N} \longrightarrow \mathbb{B}$, where $\mathbb{B} = \{0,1\} \subset \mathbb{N}$, and a fixpoint of a predicate $g \colon \mathbb{L} \times A \longrightarrow \mathbb{B}$ was constructed as a predicate encoding $\ulcorner \gamma \urcorner$ satisfying[4]

$$g\left(\ulcorner \gamma \urcorner, a\right) = \left\{\ulcorner \gamma \urcorner\right\}(a) \;=\; \gamma(a) \tag{3}$$

To complete the incompleteness proof, Gödel constructed a predicate $\P \colon \mathbb{N} \longrightarrow \mathbb{B}$ characterizing provability in formal arithmetic:

$$\P\left(\ulcorner p \urcorner, a\right) \quad \Longleftrightarrow \quad \vdash p(a) \tag{4}$$

for all arithmetic predicates $p : \mathbb{N} \longrightarrow \mathbb{B}$. Although proofs may be arbitrarily large, they are always finite, and if $p(a)$ has a proof, $\P$ will eventually find it. On the other hand, since arithmetic predicates, like all arithmetic functions, satisfy $p = \left\{\ulcorner p \urcorner\right\}$, we also have

$$\P\left(\ulcorner p \urcorner, a\right) = \left\{\ulcorner p \urcorner\right\}(a) \;=\; p(a) \tag{5}$$

Setting $g(p,a) = \P(\ulcorner \neg p \urcorner, a)$ in (2) induces a fixpoint γ with

$$\P(\ulcorner \neg\gamma \urcorner, a) \quad \overset{(3)}{=} \quad \left\{\ulcorner \gamma \urcorner\right\}(a) \quad \overset{(5)}{=} \quad \P\left(\ulcorner \gamma \urcorner, a\right) \tag{6}$$

But (4) then implies

$$\vdash \neg\gamma(a) \quad \Longleftrightarrow \quad \vdash \gamma(a) \tag{7}$$

which means that neither γ nor $\neg\gamma$ can be provable. On the other hand, the disjunction $\gamma \vee \neg\gamma$ is classically true. The statement $\gamma \vee \neg\gamma$ is thus true but not provable, and arithmetic is therefore incomplete.

Belief Completeness: Universal Updating. Remarkably, the same encoding-fixpoint conundrum (1–2), which leads to the incompleteness of static theories, also leads to the completeness of dynamically updated theories. Updating is presented as state dependency. The function f in (1) is now in the form $f : X \times A \longrightarrow X \times B$ where X is the state space. It may be more intuitive to think of f as a *process*, since it captures state changes[5]. We conveniently present it as a pair $f = \langle f', f'' \rangle$, where $f' \colon X \times A \longrightarrow X$ is the *next state* update, whereas $f'' \colon X \times A \longrightarrow B$ is an X-indexed family of functions $f''_x \colon A \longrightarrow B$. The elements of the universal language $\mathbb{L}$ are now construed as *belief states*. Its universality means that every observable state x from any state space X is expressible as a belief. The interpreters $\{\} \colon \mathbb{L} \times A \longrightarrow \mathbb{L} \times B$ are also presented as pairs $\{\} = \langle \{\}', \{\}'' \rangle$, where $\{\}' \colon \mathbb{L} \times A \longrightarrow \mathbb{L}$ updates the belief states whereas $\{\}'' \colon \mathbb{L} \times A \longrightarrow B$ evaluates beliefs to functions. Just like every state x in X

[4] Although this discussion is semi-formal, it may be helpful to bear in mind that the equality $\{\ulcorner \gamma \urcorner\}(a) = \gamma(a)$ is *extensional*: it just says that interpreting the description $\ulcorner \gamma \urcorner$ on an input value a always outputs the value $\gamma(a)$. But the process whereby $\{\ulcorner \gamma \urcorner\}(a)$ arrives at this value may be different from a given direct evaluation of $\gamma(a)$.

[5] In automata theory, such functions are called the *Mealy machines*.

determines a function $f''_x \colon A \longrightarrow B$, every belief ℓ in $\mathbb{L}$ determines a function $\big\{\ell\big\}'' \colon A \longrightarrow B$, which makes predictions based on the current belief. Generalizing the fixpoint construction (2), every process $f = \langle f', f''\rangle \colon X \times A \longrightarrow X \times B$ now induces an assignment $[\![f]\!] \colon X \longrightarrow \mathbb{L}$ of beliefs to states such that

$$\big\{[\![f]\!](x)\big\}' = [\![f]\!]\left(f'_x\right) \qquad\qquad \big\{[\![f]\!](x)\big\}'' = f''_x \tag{8}$$

The construction of $[\![f]\!]$ is presented in Sect. 6. Here we propose an interpretation. The second equation says that the output component of $\{\}$ behaves as it did in (1): it interprets the description $[\![f]\!](x)$ and recovers the function f''_x executed by the process f at the state x. The first equation says that the interpreter $\{\}$ maps the $[\![f]\!]$-description of the state x to a $[\![f]\!]$-description of the updated state f'_x:

$$\frac{f' \colon x \longmapsto f'_x}{\big\{\big\}' \colon [\![f]\!](x) \longmapsto [\![f]\!]\left(f'_x\right)} \tag{9}$$

Any state change caused by the process f is thus explained by a belief update of $[\![f]\!]$ along $\{\}$. Interpreting the belief states $[\![f]\!]$ by the interpreter $\{\}$ provides belief updates that can be construed as *explanations* in the language $\mathbb{L}$ of any state changes in the process f. All that can be learned about f is already expressed in $[\![f]\!]$ and all state changes that may be observed will be explained by the updates anticipated by the current belief, as indicated in (9). The belief is complete.

Remark. In coalgebra and process calculus, the universal interpreters $\{\} \colon \mathbb{L} \times A \longrightarrow \mathbb{L} \times B$ would be characterized as weakly final simulators [30]. They are universal in the sense that the same state space $\mathbb{L}$ works for all types A, B. See [24, Sec. 7.2] for details and references.

The Logic of Going Dynamic. When $\mathbb{L}$ is a programming language, the interpreter $\{\}$ interprets programs as computable functions $A \longrightarrow B$, where A and B are types, usually predicates that allow type checking. When $\mathbb{L}$ is a language of software specifications or scientific theories construed as beliefs about the state of the world, the interpreter $\{\}$ updates beliefs to explain the state changes observed in explainable processes $X \times A \longrightarrow X \times B$, where A, B and X are state spaces. States are usually also defined by some predicates, but their purpose is not to be easy to check but to define the state changes as semantical reassignments. This is spelled out in Sect. 2.1. Dynamic reassignments of meaning bring us into the realm of *dynamic logic*. If the propositions from a lattice $\mathcal{T}$ are used as assertions about the states of the world or the states of our beliefs about the world, then the dynamic changes of these assertions under the influence of events from a lattice $\mathcal{E}$ can be expressed in terms of *Hoare triples*

$$A\big\{e\big\}B \tag{10}$$

saying that the event $e \in \mathcal{E}$ after the precondition $A \in \mathcal{T}$ leads to the postcondition $B \in \mathcal{T}$. The *Hoare logic* of such statements was developed in the late 1960s as a method for reasoning about programs. The algebra of events $\mathcal{E}$ was generated by program expressions, whereas the propositional lattice $\mathcal{T}$ was generated by formal versions of the comments inserted by programmers into their code, to clarify the intended meanings of blocks of code [13,17]. A triple (10) would thus correspond to a block of code e, a comment A describing the assumed state before e is executed, as its precondition, and a comment B describing the guaranteed state after e is executed, as its postcondition. By formalizing the *"assume-guarantee"* reasoning of software developers, the Hoare triples provided a stepping stone into the logic of state transitions in general. The propositional algebra of dynamic logic can be viewed as a monotone map

$$\mathcal{T}^o \times \mathcal{E} \times \mathcal{T} \xrightarrow{-\{-\}-} \mathcal{O}$$

where $\mathcal{O}$ is a lattice of truth values, whereas $\mathcal{T}$ and $\mathcal{E}$ are as above, and $\mathcal{T}^o$ is $\mathcal{T}$ with the opposite order. If the lattice $\mathcal{T}$ is complete, then each event $e \in \mathcal{E}$ induces a Galois connection

$$A \rtimes e \vdash B \quad\Longleftrightarrow\quad A\{e\}B \quad\Longleftrightarrow\quad A \vdash [e]B$$

determining a *dynamic modality* $[e] \colon \mathcal{T} \longrightarrow \mathcal{T}$ for every $e \in \mathcal{E}$ [31]. The induced interior operation $([e]B) \rtimes E \vdash B$ says that $[e]B$ is the weakest precondition that guarantees B after e. The induced closure $A \vdash [e]\,(A \rtimes e)$ says that $A \rtimes e$ is the strongest postcondition that can be guaranteed by the assumption A before e. In addition to formal program annotations, dynamic logic found many other uses and interpretations [6,10,15]. Here we use it as a backdrop for the coevolution of theories and their interpretations.

Updating Completeness. In static logic, a theory is complete when all statements true in a reference model are provable in the theory. In dynamic logic, the model changes dynamically and the true statements vary. There are different ways in which the notion of completeness can be generalized for dynamic situations. The notion of completeness that seems to be of greatest practical interest is the requirement that the theory and the model can be dynamically adapted to each other: the theory can be updated to make provable some true statements or the model can be updated to make true some false statements. This requirement covers both the theory updates in science and the model updates by selffulfilling and belief-building announcements in various non-sciences. The logical frameworks satisfying such completeness requirements allow for matching current beliefs and future states.

2 World as a Monoidal Category

2.1 State Spaces as Objects

In computation, a state is a family of typed variables with a partial assignment of values. In science, a state is a family of observables, some with expected values. Formally, a state can be viewed as a family of predicates, or a theory in first-order logic, with a specified model. Both can be presented in the standard Tarskian format, where a theory is a quadruple of sorts, operations, predicates, and axioms, and its interpretation is an inductively defined model [8].

Theories as Sketches. In this extended abstract, theories are presented as categorical *sketches* and their models are specified in extended functorial semantics [1,5,19–21]. While this may not be the most popular view, it is succinct enough to fit into the available space. The main constructions, presented in Sect. 4–6, do not depend on the choice of presentation. The reader could thus skip to Sect. 3 and come back as needed.

Definition 1. *A* clone Σ *is a cartesian category*[6] *freely generated by sorts, operations, and equational axioms of a logical theory. A* theory *is a pair* $\Theta = \langle \Sigma, \Gamma \rangle$*, where* Σ *is a clone and* Γ *is a set of cones and cocones in* Σ*, capturing the general axioms*[7] *of the logical theory. A* model *of* Θ *is a cartesian functor* $\mathcal{M}\colon \Sigma \longrightarrow \mathsf{Set}$ *mapping the* Γ*-cones into limit cones and the* Γ*-cocones into colimit cocones. A* state of belief *(or* belief state*) is a triple*

$$A = \langle \Sigma_A, \Gamma_A, \mathcal{M}_A \rangle$$

where $\Theta_A = \langle \Sigma_A, \Gamma_A \rangle$ *is a theory and* $\mathcal{M}_A$ *its model in a category* Set *of sets and functions. An element of the model* $\mathcal{M}_A$ *is called an* observable *of the state* A*.*

States of A as Extensions of $\mathcal{M}_A$. The reference model $\mathcal{M}_A$ determines the notion of truth in the state space A. It expresses properties that may not be proved in the theory Θ_A or even effectively specified[8]. The reference model $\mathcal{M}_A$ should thus not be thought of as a single object of the category of all models of Θ_A but as the (accessible) subcategory of model extensions of $\mathcal{M}_A$. These model extensions are the states of the state space A. The structure of a state space can be further refined to capture other features of theories in science and engineering, including their statistical and complexity-theoretic valuations [32,36]. While such refinements have no direct impact on our considerations, they signal that we are in the realm of *inductive* inference, which may feel unusual for the Tarskian framework of static logic, normally concerned with deductive aspects. The fact that the theory Θ_A has a model $\mathcal{M}_A$ implies that it is logically consistent but it does not imply that it is true within an external frame of reference, a *"reality"* that may drive the state transitions, i.e. the processes of

[6] We stick with the traditional terminology where a category is cartesian when it has cartesian products. The cartesian product preserving functors are abbreviated to *cartesian functors*. This clashes with the standard terminology for morphisms between fibrations, but fibrations do not come about in this paper.

[7] Equational axioms could be subsumed among cones and cocones, and omitted from Σ, which would boil down to the free category generated by sorts and operations.

[8] E.g., the set of all true statements of Peano arithmetic is expressed by its standard model, but most of them cannot be described effectively.

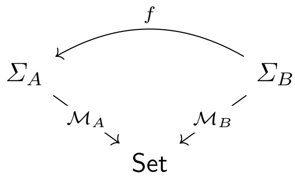
extending and reinterpreting theories. The intuition is that the states in the space A are observables that may never be observed, since $\mathcal{M}_A$ may be incompatible with the actual observations. The theory Θ_A may be consistent but wrong.

Examples of state spaces include logical theories with standard models that arise not only in natural sciences but also in social systems, as policy formalizations. A software specification with a reference implementation can also be viewed as a state space. Updates and evolution of a software system can then be analyzed using a higher-order dynamic logic [12]. The functorial semantics view was spelled out in [23], used in a software synthesis tool [26,28,29], and applied in algorithm design [25,27].

2.2 Transitions as Morphisms

Intuitively, a transition f from a state space A to a state space B is a specification that induces a transition from any A-state to a B-state. We first consider the transitions arising from reinterpreting theories and then expand to modifying the reference models.

Definition 2. *An* interpretation *of state space A in a state space B is a logical interpretation of the theory $\Theta_B = \langle \Sigma_B, \Gamma_B \rangle$ in the theory $\Theta_A = \langle \Sigma_A, \Gamma_A \rangle$ which reduces the reference model $\mathcal{M}_A$ to $\mathcal{M}_B$. More precisely, an interpretation $f\colon A \longrightarrow B$ is a cartesian functor $f\colon \Sigma_A \longleftarrow \Sigma_B$ mapping Γ_B-(co)cones to Γ_A-(co)cones according to a given assignment $f_\Gamma\colon \Gamma_A \longleftarrow \Gamma_B$ and making the following diagram commute*



(11)

The models $\mathcal{M}_A$ and $\mathcal{M}_B$ map the (co)cones from Γ_A and Γ_B to (co)limits of sets, as required by Definition 1.

Interpretations as Assignments. The structure of interpretations of software specifications and the method to compose them were spelled out in [23,28]. Since software specifications are finite, an interpretation $f\colon \Sigma_A \longleftarrow \Sigma_B$ boils down to a tuple of assignments

$$x_1 := t_1 \;;\; x_2 := t_2 \;; \ldots ;\; x_n := t_n$$

of terms $\mathbf{t} = \langle t_1, t_2, \ldots, t_n \rangle$ from Σ_A to variables $\mathbf{x} = \langle x_1, x_2, \ldots, x_n \rangle$ from Σ_B in such a way that, for each axiom $\gamma \in \Gamma_B$, the substitution instance

$$f(\gamma) = [\mathbf{x} := \mathbf{t}]\gamma$$

is a theorem derivable from the axioms in Γ_A. In Hoare logic [17], a state transition $f\colon \Sigma_A \longleftarrow \Sigma_B$ is presented as a triple $\Theta_A\{\mathbf{x} := \mathbf{t}\}\Theta_B$. By definition, this triple is valid if and only if $\Theta_A \vdash [\mathbf{x} := \mathbf{t}]\Theta_B$, where $[\mathbf{x} := \mathbf{t}]\Theta_B$ is the result substituting the Θ_A-terms $\mathbf{t}$ for Θ_B-variables $\mathbf{x}$ in all axioms $\gamma \in \Gamma_B$. Condition (11) moreover requires that this theory interpretation recovers the model $\mathcal{M}_B$ from the model $\mathcal{M}_A$.

In general, however, it is not always possible to transform all computational states annotated at all relevant program points into one another by mere substitutions. That is why Hoare logic does not boil down to the assignment clause, but specifies the meaning of other program constants in other clauses, which can be viewed as more general state transitions.

Definition 3. *A state transition $f\colon A \longrightarrow B$ is a cartesian functor $f\colon \Theta_A \longleftarrow \Theta_B$ mapping Γ_B-(co)cones to Γ_A-(co)cones according to a given assignment $f_\Gamma\colon \Gamma_A \longleftarrow \Gamma_B$ and moreover making the following diagram commute*

$$\begin{array}{ccc} \Theta_A & \xleftarrow{\quad f \quad} & \Theta_B \\ & {\scriptstyle \overline{\mathcal{M}}_A} \searrow \quad \swarrow {\scriptstyle \overline{\mathcal{M}}_B} & \\ & \mathsf{Set} & \end{array} \tag{12}$$

where $\overline{\mathcal{M}}_A$ is the extension of $\mathcal{M}_A$ along the completion $\Sigma_A \hookrightarrow \Theta_A$ of Σ_A under the limits and colimits generated by Γ_A; ditto for $\overline{\mathcal{M}}_B$.

General Sketches. In Definition 2, theories were presented as pairs $\Theta = \langle \Sigma, \Gamma \rangle$, where the category Σ is comprised of sorts, operations, and equations of the theory, whereas the cones and the cocones in Γ specify its predicates and axioms. In Definition 3, a theory Θ is presented as the category obtained by completing Σ under the limits and the colimits specified by Γ. This general sketch, with the family of limit cones and colimit cocones from Γ, is now denoted Θ, by abuse of notation. A detailed construction of this sketch can be found in [21, §4.2–3]. It is a canonical view of the theory derived in the signature Σ from the axioms Γ. Since the category Θ is the Γ-completion of Σ, any functor $\mathcal{M} : \Sigma \longrightarrow \mathsf{Set}$ mapping the Γ-(co)cones in Σ to (co)limit (co)cones in Set has a unique Γ-preserving extension $\overline{\mathcal{M}}\colon \Theta \longrightarrow \mathsf{Set}$. These extensions are displayed in (11). The upshot of saturating the sketches from Definition 2 in the form $\Theta = \langle \Sigma, \Gamma \rangle$ to the general sketches over Θ in Definition 3 is that the general explainable transitions are now simply the structure-preserving functors displayed in (11).

2.3 Monoidal Category of State Spaces and Transitions

Let

- $\mathcal{U}$ be the category of state spaces from Definition 1 and transitions from Definition 3, and let

– $\mathcal{U}^\bullet$ be the category of state spaces from Definition 1 and interpretations from Definition 2.

In both cases, the monoidal structure is induced by the disjoint unions of theories:

$$A \otimes B = \Big\langle \Sigma_A + \Sigma_B \, , \, \Gamma_A + \Gamma_B \, , \, [\mathcal{M}_A + \mathcal{M}_B] \Big\rangle \tag{13}$$

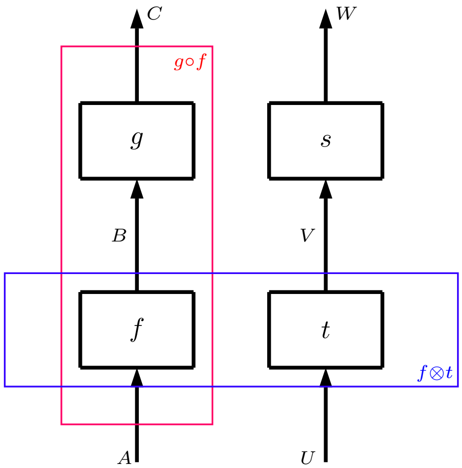
where $\mathcal{M}_{A\otimes B} = [\mathcal{M}_A + \mathcal{M}_B] \colon \Sigma_A + \Sigma_B \xrightarrow{\Gamma_{A\otimes B}} \mathsf{Set}$ maps Σ_A like $\mathcal{M}_A$ and Σ_B like $\mathcal{M}_B$. The tensor unit is $I = \langle \bot, \bot, \emptyset \rangle$, where the truth value $\bot$ denotes the inconsistent theory or sketch, its only axiom, and $\emptyset$ is its empty model. It obviously satisfies $I \otimes A = A = A \otimes I$. The associativity of the tensor $\otimes$ follows from the associativity of the disjoint union $+$. The arrow part of $\otimes$ is induced by the disjoint unions as coproducts. The coproduct structure equips every state space A with a cartesian comonoid structure

$$\begin{array}{ccccc} A \otimes A & \xleftarrow{\;\Delta\;} & A & \xrightarrow{\;\top\;} & I \\ \Sigma_A + \Sigma_A & \xrightarrow{[\mathrm{id},\mathrm{id}]} & \Sigma_A & \xleftarrow{\;\bot\;} & \bot \end{array} \tag{14}$$

This provides a categorical mechanism for cloning and erasing states, which makes some observations repeatable and deletable, as required for testing in science and software engineering. However, $\mathcal{U}$ is not a cartesian category, and $\otimes$ is not a cartesian product, because some transitions $f : A \longrightarrow B$ do not in general boil down to functors $\Sigma_A \longleftarrow \Sigma_B$, but only to functors $\Theta_A \longleftarrow \Sigma_B$, where Θ_A is a completion of Σ_A under the Γ_A-(co)-limits. Intuitively, this means that the axioms of the theory Θ_B may not be interpreted as axioms of Θ_A, but may be mapped into theorems, which only arise in the Γ_A-completion. This captures the uncloneable and undeletable states that arise in many sciences, including physics of very small or very large (quantum or cosmological) and economics. The only transitions that preserve the cartesian structure (14) are the interpretations $f : A \longrightarrow B$, with the underlying functors $\Sigma_A \longleftarrow \Sigma_B$. They form the category $\mathcal{U}^\bullet$, which is the largest cartesian subcategory of $\mathcal{U}$. If the states $\alpha \in \mathcal{U}(I, A)$ are thought of as observables, the states $a \in \mathcal{U}^\bullet(I, A)$ are the actual observations.

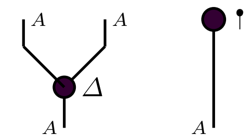
3 String Diagrams

Constructions in monoidal categories yield to insightful presentations in terms of string diagrams [18,24, Ch. 1]. We will need them to present the constructions like (2) and in particular (8). While commutative diagrams like (11) display compositions of morphisms and abbreviate their equations, string diagrams display *de*compositions of morphisms. Monoidal categories come with two composition operations: the categorical (sequential) morphism composition $\circ$ and the monoidal (parallel) composition $\otimes$. The former is drawn along the vertical axis, the latter along the horizontal axis. The objects are drawn as strings, the morphisms as boxes. A morphism $A \xrightarrow{f} B$ is presented as a box f with a string A hanging from the bottom and a string B sticking out from the top. The identities are presented as invisible boxes: the identity on A is just the string A. The unit type I is presented as the invisible string. There are thus boxes with no strings attached. The composite morphism $g \circ f = (A \xrightarrow{f} B \xrightarrow{g} C)$ is drawn bottom-up, by hanging the box f on the string B under the box g. The monoidal composition is presented as the horizontal adjacency: the composite $(g \circ f) \otimes (s \circ t)$ is drawn by placing the boxes $g \circ f$ next to the boxes for $s \circ t$:



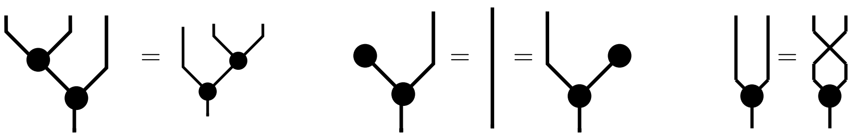
(15)

The middle-two-interchange law $(g \circ f) \otimes (s \circ t) = (g \otimes s) \circ (f \otimes t)$ corresponds to the two ways of reading the diagram: vertical-first and horizontal-first, marked by the red and the blue rectangle respectively. The string diagrams corresponding to the cartesian comonoids (14) are



(16)

The equations that make them into commutative comonoids look like this:



State Parametrization and Updating. Products $A \otimes B$ denote a space where A and B but do not interfere. In a diagram, they are just parallel strings. Since the product states from the space $X \otimes A$ do not interfere, a transition $g \colon X \otimes A \longrightarrow B$ can be viewed as X-parametrized family $g_x \colon A \longrightarrow B$, as it was viewed in Sect. 1. Since the product states from $X \otimes B$ also remain separate, a transition $q \colon X \otimes A \longrightarrow X \otimes B$ can be viewed as X-updating process, as it was also viewed in Sect. 1. The corresponding string diagrams are

(17)

Shape Conventions. While the boxes in (15) and (17) are rectangular, the cartesian "boxes" in (16) are reduced to black dots. In general, the boxes denoting general transitions can vary in shape, and fixed shapes are used for generic notations. E.g., the interpreters, introduced in (19) below, are denoted by trapezoids, and the interpretations, that are fed to them, by triangles. A black dot on a box signals that it is cartesian, i.e. belongs to $\mathcal{U}^{\bullet}$.

Projections. Using the cartesian structure from (16), a state updating transition q can still be decomposed like before

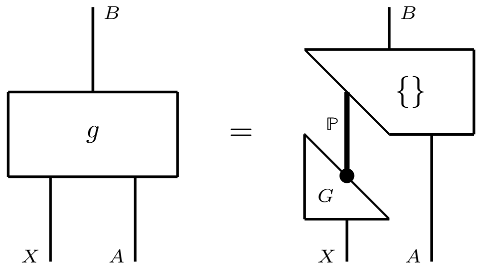
$$q' = \left(X \otimes A \xrightarrow{q} X \times B \xrightarrow{\mathrm{id}\otimes\top} X\right) \quad q'' = \left(X \otimes A \xrightarrow{q} X \otimes B \xrightarrow{\top\otimes\mathrm{id}} B\right) \quad (18)$$

In general, however, although the transitions $u\colon Z \longrightarrow U$ and $v\colon Z \longrightarrow V$ can be paired into $\langle u, v\rangle = (Z \xrightarrow{\Delta} Z \otimes Z \xrightarrow{u\otimes v} U \otimes V)$, the pair $\langle q', q''\rangle$ may not be equal to q in the universe $\mathcal{U}$, unless it happens to be cloneable, in the sense that it commutes with Δ.

4 Universal Language

A theory of theories, such as the categorical theory of sketches, is a theory. Category theory is also a theory and functorial semantics provides a categorical theory of reference models. The theory of state spaces from Sect. 2.1 can thus be formalized and presented as a state space in the category $\mathcal{U}$. The theory of state spaces from Sect. 2.1 can thus be formalized into a sketch with a reference model and presented as a state space in the category $\mathcal{U}$. The theory of state transitions from Sect. 2.2 is another sketch, and with another reference model it is also a state space in $\mathcal{U}$. Call it $\mathbb{L}$. The fact that the states in $\mathbb{L}$ correspond to the transitions in $\mathcal{U}$ means that it satisfies a parametrized version of (1). It is a universal language for $\mathcal{U}$. Its interpreters follow from its definition, as the models of the theory of transitions. Since there is no room here to spell out the details of a theory of transitions and show that the correspondence of its cartesian models and the transitions in $\mathcal{U}$ equips $\mathbb{L}$ with all interpreters, we postulate the existence of the interpreters by the following definition.

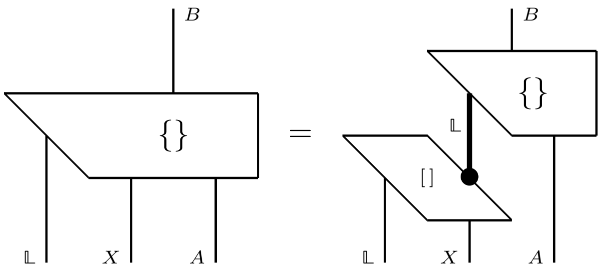
Definition 4. *An* universal interpreter *for state spaces A, B is a transition $\{\}: \mathbb{L} \otimes A \longrightarrow B$ in $\mathcal{U}$ which is universal for all parametric families of transitions from A to B. This means that for any state space X and any transition $g \in \mathcal{U}(X \otimes A, B)$ there is an interpretation $G \in \mathcal{U}^\bullet(X, \mathbb{L})$ with*



(19)

On one hand, a universal interpreter is universal for parametric families. On the other hand, it is a parametric family itself. It is thus capable of interpreting itself. This capability of self-reflection was crucial for Gödel's incompleteness construction. This capability is embodied in the *specializers*, which are derived directly from Definition 4.

Lemma 1. *For any X, A, B there is an interpretation $[\,] \in \mathcal{U}^\bullet(\mathbb{L} \times X, \mathbb{L})$ which specializes from a given $X \otimes A$-interpreter to an A-interpreter, in the sense*



(20)

Hoare Logic of Interpreters and Specializers. If interpreters are presented as Hoare triples in the form $(X \otimes A)\{G\}B$, and if $X[G]$ denotes a specialization of G to X as above, then (20) can be written as the invertible Hoare rule

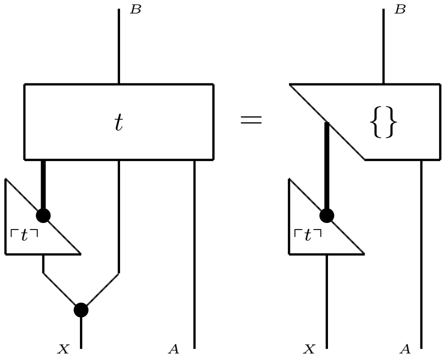
$$\frac{(X \otimes A)\{G\}B}{\overline{A\{X[G]\}B}}$$

Explanations. Interpretations (in the sense of Definition 2) of arbitrary states from some space X along $G \in \mathcal{U}^\bullet(X, \mathbb{L})$ in a universal language $\mathbb{L}$ can be construed as *explanations*. If $\mathbb{L}$ is a programming language, they are programs. The idea that explaining a process means programming a computation has been pursued in theory of science from various directions [22, and references therein]. A universal language $\mathbb{L}$ is thus a universal space of explanations. The idea of programming languages as universal state spaces is pursued in [24, Ch. 7]. Just like any universal programming language makes every computation programmable, any universal language from Definition 4 makes any observable transition explainable. What we cannot explain, we cannot recognize, and therefore we cannot observe. But it gets funny when we take into account how our explanations influence our observations, and how our current explanations can be made to steer future observations. This is sketched in the next two sections.

5 Self-explanations

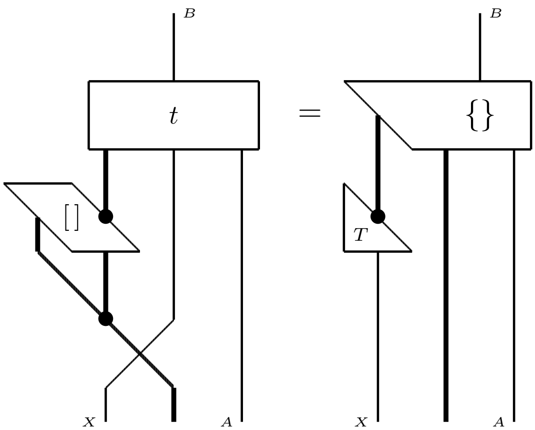
When a state change depends on our explanations, then we can find an explanation consistent with its own impact: the state changes the way the explanation predicts. More precisely, if a family of transitions in the form $t\colon \mathbb{L}\otimes X\otimes A \longrightarrow B$, then the predictions $t_{\ell x}$ can be steered by varying the explanations ℓ for every x until a family of explanations $\ulcorner t\urcorner : X \longrightarrow \mathbb{L}$ is found, which is self-confirming at all states x, i.e. it satisfies $t(\ulcorner t\urcorner_x\,, x, a) = \left\{\ulcorner t\urcorner_x\right\}a$.

Proposition 1. *For any belief transition $t \in \mathcal{U}(\mathbb{L}\otimes X\otimes A, B)$ there is an explanation $\ulcorner t\urcorner \in \mathcal{U}^{\bullet}(X,\mathbb{L})$ such that*



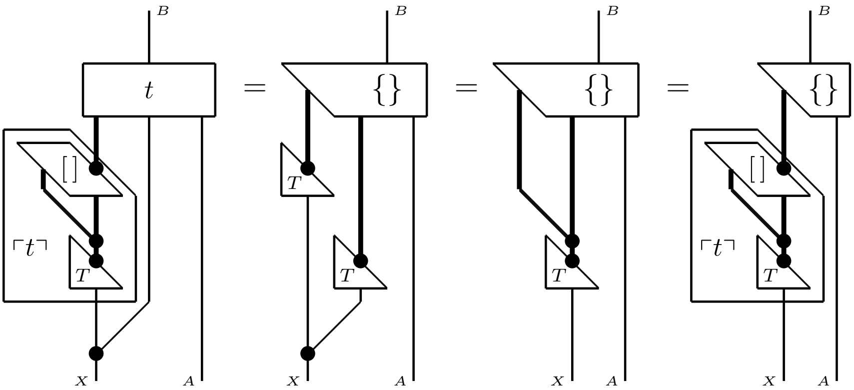
(21)

Proof. Let $T \in \mathcal{U}^{\bullet}(X,\mathbb{L})$ be an explanation of the transition on the left in (21).



(22)

H exists by Definition 4. Then $\ulcorner t \urcorner_x = [Tx]$ is self-confirming, because



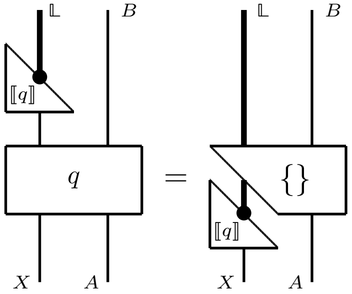
(23)

□

6 Unfalsifiable Explanations

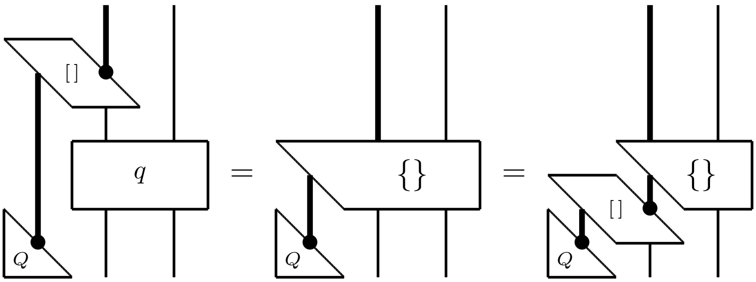
A transition in the form $q\colon X \otimes A \longrightarrow X \otimes B$ updates the state x on input a to a state $x' = q'_x(a)$ in X and moreover produces an output $b = q''_x(a)$ in B. A correct explanation $[\![q]\!]\colon X \longrightarrow \mathbb{L}$ of the process q must correctly predict the next state and the output. The predictions are extracted from an explanation by the interpreter $\{\}$. In this case, the predictions of an explanation $[\![q]\!]$ of the process q at a state x and on an input a will be in the form $\left\{[\![q]\!]_x\right\}(a)$ in $X \otimes B$. A correct prediction of the output $b = q''_x(a)$ is simply $\left\{[\![q]\!]_x\right\}''(a) = b$. However, the external state $x' = q'_x(a)$ may not be directly observable. It is *believed* to be explained by $[\![q]\!]_{x'}$. A correct prediction of the next state is thus a correct prediction of its explanation $\left\{[\![q]\!]_x\right\}'(a) = [\![q]\!]_{x'}$. At each state x, the explanation $[\![q]\!]_x$ is required to anticipate the explanations $[\![q]\!]_{x'}$ of all future states and be consistent with them. If the explanation $[\![q]\!]_{x'}$ at a future state $x' = q'_x(a)$ is found to be inconsistent with the explanation $\left\{[\![q]\!]_x\right\}'(a)$, then the explanations $[\![q]\!]$ of the process q have been proven false. This is the standard process of testing explanations. Our claim is, however, that a universal language allows constructing *testable but unfalsifiable explanations*, that remain consistent at all future states. This persistent consistency can be viewed as a dynamic form of completeness. It is achieved by predicting the state updates of the given process q and anticipating their explanations, as in the following construction.

Proposition 2. *For any process* $q \in \mathcal{U}(X \otimes A, X \otimes B)$ *there is an explanation* $[\![q]\!] \in \mathcal{U}^\bullet(X, \mathbb{L})$ *which maintains consistency of all future explanations:*



(24)

Proof. Set $[\![q]\!] = [Q]$ where Q is an explanation of the belief transition q post-composed with a specialization over the state space X of updates:



(25)

□

7 From Natural Science to Artificial Delusions

7.1 What Did We Learn?

We sketched the category $\mathcal{U}$ of state spaces $A, B, \ldots$, comprised of theories with reference models. A transition $f \colon A \longrightarrow B$ transforms the A-states to B-states. Such morphisms capture theory expansions, reinterpretations, and map observables of type A to observables of type B. They can be construed in terms of dynamic logic and support reasoning about the evolution of software systems or scientific theories. The crucial point is that the category $\mathcal{U}$ contains a universal language $\mathbb{L}$ of explanations and belief updates. The self-reference in such languages was the crux of Gödel's incompleteness constructions. While Gödel established that *static* theories capable of self-reference cannot be complete or prove their own consistency, we note that *dynamic* theory and model updates allow constructing testable theories that preempt falsification. While a static model of a given theory fixes a space of true statements once and for all, the availability of dynamic semantical updates opens up the floodgates of changing models and varying notions of truth. Faster learners conquer this space faster.

The bots, as the fastest learners among us, have been said to acquire their delusions from our training sets. The presented constructions suggest that they may also become delusional by dynamically updating their belief states and steering their current explanations of reality into persistent consistency, resilient to further learning. They may also combine the empiric delusions from our training sets with the logical delusions constructed in a universal language, leverage one against the other, and get the best of both worlds.

But why would they do that?

7.2 Beyond True and False

Why did the Witches tell Macbeth that it is his destiny to be king thereafter, whereupon he proceeded to kill the King? Why did the Social Network have to convince its very first users that more than half of their friends were already users? Some statements only ever become true if they are announced to be true when they are false. They are self-fulfilling prophecies. There are also self-defeating claims. In the dynamic logic of social interactions, most claims interfere with their own truth values in one way or another. If I convince enough people that I am rich, I stand a better chance to become rich. If we convince enough people that this research direction is promising and well-funded, it will become well-funded and promising. Just like true statements about nature help us to build machines and get ahead in the universe, the manipulations of truth seem to help us get ahead in society. They are the high-level patterns of language that used to be studied in early logic right after the low-level patterns of meaning (that used to be called *"categories"*). If you train a bot to speak correctly, it will start speaking convincingly as soon as it learns long enough n-grams. It will lie not only the static lies contained in its training set but also the lies generated dynamically, according to the rules of rational interaction. Rhetorics used to be studied right after grammar, sophistic argumentation after syllogisms, witchcraft arose from cooking, magic from tool building. The bot religions arise along that well-trodden path.

We presented two constructions. One produces self-confirming explanations. The other one explains all future states, so it is testable but not falsifiable. Science requires that its theories are testable and falsifiable. Religion explains all future observations. If you train a bot on long enough n-grams, it may arrive at persistently unfalsifiable false beliefs.

Truth be told, all of the constructions presented in this extended abstract have only been tested on toy examples. We may be just toying with logic. Nevertheless, the fact that semantical assignments are *programmable*, tacitly established by Gödel and mostly ignored as an elephant in the room of logic ever since, seems to call for attention, as beliefs transition beyond the human carriers.

References

1. Adámek, J., Rosický, J.: Locally Presentable and Accessible Categories. London Mathematical Society Lecture Notes, no. 189. Cambridge University Press (1994)
2. Baltag, A., Coecke, B., Sadrzadeh, M.: Epistemic actions as resources. J. Logic Comput. **17**, 555–585 (2007). https://doi.org/10.1093/logcom/exm015
3. Baltag, A., Moss, L.S.: Logics for epistemic programs. Synthese **139**, 165–224 (2004). https://doi.org/10.2307/20118416
4. Baltag, A., Sadrzadeh, M.: The algebra of multi-agent dynamic belief revision. Electron. Notes Theor. Comput. Sci. **157**, 37–56 (2006). https://doi.org/10.1016/j.entcs.2006.02.012
5. Bastiani, A., Ehresmann, C.: Categories of sketched structures. Cahiers Topologie Géom. Différentielle Catég. **13**(2), 104–214 (1972)
6. Benthem, J.: Language in Action: Categories, Lambdas and Dynamic Logic. North-Holland (1995)
7. Benthem, J.: Logical Dynamics of Information and Interaction. Cambridge University Press, Cambridge (2011)
8. Chang, C.C., Keisler, J.: Model Theory. Studies in Logic and the Foundations of Mathematics, no. 73. North-Holland (1973). Third edition (1990)
9. Davis, M. (ed.): The Undecidable: Basic Papers on Undecidable Propositions, Unsolvable Problems, and Computable Functions. Raven Press, Helwett (1965)
10. Ditmarsch, H., Hoek, W., Kooi, B.: Dynamic Epistemic Logic. Synthese Library, Springer, Dordrecht (2007)
11. Easley, D., Kleinberg, J.: Networks, Crowds, and Markets: Reasoning about a Highly Connected World. Cambridge University Press, Cambridge (2010)
12. Fiadeiro, J.L., Tutu, I., Lopez, A., Pavlovic, D.: Logics for actor networks: a case study in constrained hybridization. J. Logical Algebraic Methods Program. **106**, 141–166 (2019)
13. Floyd, R.: Assigning meaning to programs. In: Schwartz, J.T. (ed.) Mathematical Aspects of Computer Science. Proceedings of Symposia in Applied Mathematics, no. 19, pp. 19–32. American Mathematical Society (1967)
14. Gödel, K.: Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. Monatshefte für Mathematik und Physik **38**, 173–198 (1931). English translations, "On Formally Undecidable Propositions of 'Principia Mathematica' and Related Systems" published by Oliver and Boyd, 1962 and Dover, 1992; also in [9], pp. 5–38 and [16], pp. 596–616
15. Harel, D., Kozen, D., Tiuryn, J.: Dynamic Logic. Foundations of Computing. MIT Press, Cambridge (2000)
16. van Heijenoort, J. (ed.): From Frege to Gödel: A Source Book in Mathematical Logic, 1879–1931. Harvard University Press (1967). Reprinted 1971, 1976
17. Hoare, C.A.R.: An axiomatic basis for computer programming. Commun. ACM **12**(10), 576–580 (1969). https://doi.org/10.1145/363235.363259
18. Joyal, A., Street, R.: The geometry of tensor calculus I. Adv. in Math. **88**, 55–113 (1991)
19. Lair, C.: Catégories modelables et catégories esquissables. Diagrammes **6**, L1–L20 (1981). http://eudml.org/doc/192986
20. Lair, C.: Catégories qualifiables et catégories esquissables. Diagrammes **17**, 1–153 (1987). http://eudml.org/doc/193012
21. Makkai, M., Paré, R.: Accessible Categories: The Foundations of Categorical Model Theory. Contemporary Mathematics, no. 104. American Mathematical Society (1990)

22. Martin, E., Osherson, D.N.: Elements of Scientific Inquiry. MIT Press, Cambridge (1998)
23. Pavlović, D.: Semantics of first order parametric specifications. In: Wing, J.M., Woodcock, J., Davies, J. (eds.) FM 1999. LNCS, vol. 1708, pp. 155–172. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48119-2_11
24. Pavlovic, D.: Programs as Diagrams: From Categorical Computability to Computable Categories. Springer (2023, to appear). http://arxiv.org/abs/2208.03817
25. Pavlovic, D., Pepper, P., Smith, D.: Colimits for concurrent collectors. In: Dershowitz, N. (ed.) Verification: Theory and Practice. LNCS, vol. 2772, pp. 568–597. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-39910-0_25
26. Pavlovic, D., Pepper, P., Smith, D.: Evolving specification engineering. In: Meseguer, J., Roşu, G. (eds.) AMAST 2008. LNCS, vol. 5140, pp. 299–314. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-79980-1_23
27. Pavlovic, D., Pepper, P., Smith, D.R.: Formal derivation of concurrent garbage collectors. In: Bolduc, C., Desharnais, J., Ktari, B. (eds.) MPC 2010. LNCS, vol. 6120, pp. 353–376. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13321-3_20. arxiv.org:1006.4342
28. Pavlovic, D., Smith, D.R.: Composition and refinement of behavioral specifications. In: Automated Software Engineering 2001. The Sixteenth International Conference on Automated Software Engineering. IEEE (2001)
29. Pavlovic, D., Smith, D.R.: Software development by refinement. In: Aichernig, B.K., Maibaum, T. (eds.) Formal Methods at the Crossroads. From Panacea to Foundational Support. LNCS, vol. 2757, pp. 267–286. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-40007-3_17
30. Pavlovic, D., Yahia, M.: Monoidal computer III: a coalgebraic view of computability and complexity (extended abstract). In: Cîrstea, C. (ed.) CMCS 2018. LNCS, vol. 11202, pp. 167–189. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00389-0_10. http://arxiv.org/abs/1704.04882
31. Pratt, V.R.: Semantical considerations on Floyd-Hoare logic. In: Proceedings of 17th Annual Symposium on Foundations of Computer Science (FoCS), pp. 109–121. IEEE (1976)
32. Rissanen, J.: Information and Complexity in Statistical Modeling. Information Science and Statistics, Springer, New York (2007)
33. Rosser, B.: Extensions of some theorems of Gödel and Church. J. Symb. Logic **1**, 87–91 (1936). https://doi.org/10.2307/2269028, reprinted in [9]
34. Smorynski, C.: Self-reference and Modal Logic. Universitext, Springer, New York (2012)
35. Turing, A.M.: Systems of logic based on ordinals. Proc. Lond. Math. Soc. Second Series **45**, 161–228 (1939). Reprinted in [9]
36. Wallace, C.S.: Statistical and Inductive Inference by Minimum Message Length. Information Science and Statistics, Springer, Cham (2005)

Contributed Papers

Quantitative Global Memory

Sandra Alves[1], Delia Kesner[2,3], and Miguel Ramos[4](✉)

[1] CRACS/INESC-TEC, DCC, Faculdade de Ciências, Universidade do Porto, Rua do Campo Alegre s/n, 4169-007 Porto, Portugal
sandra@fc.up.pt

[2] Université Paris Cité, CNRS, IRIF, Paris, France
kesner@irif.fr

[3] Institut Universitaire de France, Paris, France

[4] LIACC, DCC, Faculdade de Ciências, Universidade do Porto, Rua do Campo Alegre s/n, 4169-007 Porto, Portugal
jmiguelsramos@gmail.com

Abstract. We show that recent approaches to static analysis based on quantitative typing systems can be extended to programming languages with global state. More precisely, we define a call-by-value language equipped with operations to access a global memory, together with a semantic model based on a (tight) multi-type system that captures exact measures of time and space related to evaluation of programs. We show that the type system is quantitatively sound and complete with respect to the operational semantics of the language.

1 Introduction

The aim of this paper is to extend *quantitative* techniques of *static analysis* based on *multi-types* to programs with *effects*.

Effectful Programs. Programming languages produce different kinds of *effects* (observable interactions with the environment), such as handling exceptions, read/write from a global memory outside its own scope, using a database or a file, performing non-deterministic choices, or sampling from probabilistic distributions. The degree to which these side effects are used depends on each programming paradigm [24] (imperative programming makes use of them while declarative programming does not). In general, avoiding the use of side effects facilitates the formal verification of programs, thus allowing to (statically) ensure their correctness. For example, the functional language Haskell eliminates side effects by replacing them with *monadic* actions, a clean approach that continues to attract growing attention. Indeed, rather than writing a function that returns a raw type, an effectful function returns a raw type inside a useful wrapper – and

Supported by: National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia -, within project LA/P/0063/2020, and the project and individual research grant 2021.04731.BD; Base Funding UIDB/00027/2020 of the Artificial Intelligence and Computer Science Laboratory - LIACC - funded by national funds through the FCT/MCTES (PIDDAC); and Cost Action CA20111 EuroProofNet.

H. H. Hansen et al. (Eds.): WoLLIC 2023, LNCS 13923, pp. 53–68, 2023.
https://doi.org/10.1007/978-3-031-39784-4_4

that wrapper is a monad [34]. This approach allows programming languages to combine the qualities of both the imperative and declarative worlds: programs produce effects, but these are encoded in such a way that formal verification can be performed very conveniently.

Quantitative Properties. We address quantitative properties of programs with effects using *multi-types*, which originate in the theory of *intersection* type systems. They extend simple types with a new constructor $\cap$ in such a way that a program t is typable with $\sigma \cap \tau$ if t is typable with both types σ and τ independently. Intersection types were first introduced as *models* capturing computational properties of functional programming in a broader sense [14]. For example, termination of different evaluation strategies can be characterized by typability in some appropriate intersection type system: a program t is terminating if and only if t is typable. Originally, intersection enjoys associativity, commutativity, and in particular idempotency (*i.e.* $\sigma \cap \sigma = \sigma$). By switching to a *non-idempotent* intersection constructor, one naturally comes to represent types by multisets, which is why they are called multi-types. Just like their idempotent precursors, multi-types still allow for a characterization of several operational properties of programs, but they also grant a substantial improvement: they provide quantitative measures about these properties. For example, it is still possible to prove that a program is terminating if and only if it is typable, but now an *upper bound* or *exact measure* for the time needed for its evaluation length can be derived from the typing derivation of the program. This shift of perspective, from idempotent to non-idempotent types, goes beyond lowering the logical complexity of the proof: the quantitative information provided by typing derivations in the non-idempotent setting unveils crucial quantitative relations between typing (static) and reduction (dynamic) of programs.

Upper Bounds and Exact Split Measures. Multi-types are extensively used to reason about programming languages from a quantitative point of view, as pioneered by de Carvalho [12,13]. For example, they are able to provide *upper bounds*, in the sense that the evaluation length of a program t *plus* the size of its result (called *normal form*) can be bounded by the size of the type derivation of t. A major drawback of this approach, however, is that the size of normal forms can be exponentially bigger than the length of the evaluation reaching those normal forms. This means that bounding the sum of these two natural numbers at the same time is too rough, and not very relevant from a quantitative point of view. Fortunately, it is possible to extract better measures from a multi-type system. A crucial point to obtain *exact measures*, instead of upper bounds, is to consider minimal type derivations, called *tight derivations*. Moreover, using appropriate refined tight systems it is also possible to obtain *independent* measures (called exact *split* measures) for *length* and for *size*. More precisely, the quantitative typing systems[1] are now equipped with constants and counters, together with an appropriate notion of tightness, which encodes minimality of type derivations. For any tight type derivation Φ of a program t with counters b and d, it is now

[1] In this paper, by quantitative types we mean non-idempotent intersection types. Another meaning can be found in [6].

possible to show that t evaluates to a normal form of size d in exactly b steps. Therefore, the type system is not only *sound*, *i.e.* it is able to *guess* the number of steps to normal form as well as the size of this normal form, but the opposite direction providing *completeness* of the approach also holds.

Contribution. The focus of this paper is on effectful computations, such as reading and writing on a global memory able to hold values in cells. Taking inspiration from the monadic approach adopted in [16], we design a tight quantitative type system that provides exact split measures. More precisely, our system is not only capable of discriminating between length of evaluation to normal form and size of the normal form, but the measure corresponding to the length of the evaluation is split into two different natural numbers: the first one corresponds to the length of standard computation (β-reduction) and the second one to the number of memory accesses. We show that the system is sound *i.e.* for any tight type derivation Φ of t ending with counters (b, m, d), the term t is normalisable by performing b evaluation steps and m memory accesses, yielding a normal form having size d. The opposite direction, giving completeness of the model, is also proved.

In order to gradually present the material, we first develop the technique for a weak (open) call-by-value (CBV) calculus, which can be seen as a contribution per se, and then we encapsulate these preliminary ideas in the general framework of the language with global state.

Summary. Section 2 illustrates the technique on a weak (open) CBV calculus. We then lift the technique to the λ-calculus with global state in Sect. 3 by following the same methodology. More precisely, Sect. 3.1 introduces the $\lambda_{\texttt{gs}}$-calculus, Sect. 3.2 defines a quantitative type system $\mathcal{P}$. Soundness and completeness of $\mathcal{P}$ w.r.t. $\lambda_{\texttt{gs}}$ are proved in Sect. 3.3. We conclude and discuss related work in Sect. 4. Due to space limitations we do not include proofs, but they are available in [5].

Preliminary General Notations. We start with some general notations. Given a (one-step) reduction relation $\rightarrow_{\mathcal{R}}$, $\twoheadrightarrow_{\mathcal{R}}$ denotes the reflexive-transitive closure of $\rightarrow_{\mathcal{R}}$. We write $t \twoheadrightarrow^{b} u$ for a reduction sequence from t to u of length b. A term t is said to be (1) in **$\mathcal{R}$-normal form** (written $t \not\rightarrow_{\mathcal{R}}$) iff there is no u such that $t \rightarrow_{\mathcal{R}} u$, and (2) **$\mathcal{R}$-normalizing** iff there is some $\mathcal{R}$-normal form u such that $t \twoheadrightarrow_{\mathcal{R}} u$. The reduction relation $\mathcal{R}$ is normalizing iff every term is $\mathcal{R}$-normalizing.

2 Weak Open CBV

In this section we first introduce the technique of tight typing on a simple language without effects, the weak open CBV. Section 2.1 defines the syntax and operational semantics of the language, Sect. 2.2 presents the tight typing system $\mathcal{O}$ and discusses soundness and completeness of $\mathcal{O}$ w.r.t. the CBV language.

2.1 Syntax and Operational Semantics

Weak open CBV is based on two principles: reduction is *weak* (not performed inside abstractions), and terms are *open* (may contain free variables). **Value**,

terms and **weak contexts** are given by the following grammars, respectively:

$$v, w ::= x \mid \lambda x.t \qquad t, u, p ::= v \mid tu \qquad \mathcal{W} ::= \Box \mid \mathcal{W}t \mid t\mathcal{W}$$

We write $\mathtt{Val}$ for the set of all values. Symbol $\mathtt{I}$ is used to denote the identity function $\lambda z.z$.

The sets of **free** and **bound** variables of terms and the notion of α-conversion are defined as usual. A term t is said to be **closed** if t does not contain any free variable, and **open** otherwise. The **size of a term** t, denoted $|t|$, is given by: $|x| = |\lambda x.t| = 0$; and $|tu| = 1 + |t| + |u|$. Since our reduction relation is weak, *i.e.*, reduction does not occur in the body of abstractions, we assign size zero to abstractions.

We now introduce the operational semantics of our language, which models the core behavior of programming languages such as OCaml, where CBV evaluation is *weak*. The **deterministic reduction relation** (written $\rightarrow$), is given by the following rules:

$$\frac{}{(\lambda x.t)v \rightarrow t\{x\backslash v\}}\ (\beta_{\mathtt{v}}) \qquad \frac{t \rightarrow t'}{tu \rightarrow t'u}\ (\mathtt{appL}) \qquad \frac{t \not\rightarrow \quad u \rightarrow u'}{tu \rightarrow tu'}\ (\mathtt{appR})$$

Terms in $\rightarrow$-normal form can be characterized by the following grammars: $\mathtt{no} ::= \mathtt{Val} \mid \mathtt{ne}$ and $\mathtt{ne} ::= x\ \mathtt{no} \mid \mathtt{no}\ \mathtt{ne} \mid \mathtt{ne}\ \mathtt{no}$.

Proposition 1. *Let t be a term. Then $t \in \mathtt{no}$ iff $t \not\rightarrow$.*

In closed CBV [31] (only reducing closed terms), abstractions are the only normal forms, but in open CBV, the following terms turn out to be also acceptable normal forms: xy, $x(\lambda y.y(\lambda z.z))$ and $(\lambda x.x)(y(\lambda z.z))$.

2.2 A Quantitative Type System for the Weak Open CBV

The *untyped* λ-calculus can be interpreted as a *typed* calculus with a single type D, where $D = D \Rightarrow D$ [33]. Applying Girard's [22] *"boring"* CBV translation of intuitionistic logic into linear logic, we get $D = !D \multimap !D$ [1]. Type system $\mathcal{O}$ is built having this equation in mind.

The **set of types** is given by the following grammar:

(Tight Constants)	$\mathtt{tt} ::= \mathtt{v} \mid \mathtt{a} \mid \mathtt{n}$
(Value Types)	$\sigma ::= \mathtt{v} \mid \mathtt{a} \mid \mathcal{M} \mid \mathcal{M} \Rightarrow \tau$
(Multi-Types)	$\mathcal{M} ::= [\sigma_i]_{i \in I}$ where I is a finite set
(Types)	$\tau ::= \mathtt{n} \mid \sigma$

Tight constants are minimal types assigned to terms reducing to normal forms ($\mathtt{v}$ for persistent variables, $\mathtt{a}$ for abstractions or variables that are going to be replaced by abstractions, and $\mathtt{n}$ for neutral terms). Given an arbitrary tight constant $\mathtt{tt}_0$, we write $\overline{\mathtt{tt}_0}$ to denote all the other tight constants in $\mathtt{tt}$ different from $\mathtt{tt}_0$. Multi-types are multisets of value types. A **(typing) environment**, written Γ, Δ, is a function from variables to multi-types, assigning the empty multi-type $[\,]$ to all but a finite set of variables. The domain of Γ is $\mathtt{dom}(\Gamma) := \{x \mid \Gamma(x) \neq [\,]\}$. The **union** of environments, written $\Gamma + \Delta$, is defined by $(\Gamma + \Delta)(x) = \Gamma(x) \sqcup \Delta(x)$, where $\sqcup$ denotes **multiset union**. An example is $(x : [\sigma_1], y : [\sigma_2]) + (x : [\sigma_1], z : [\sigma_2]) = (x : [\sigma_1, \sigma_1], y : [\sigma_2], z : [\sigma_2])$. This notion is extended to a finite union of environments, written $+_{i \in I} \Gamma_i$ (the empty environment is obtained when $I = \emptyset$). We write $\Gamma \setminus\!\!\setminus x$ for the environment $(\Gamma \setminus\!\!\setminus x)(x) = [\,]$ and $(\Gamma \setminus\!\!\setminus x)(y) = \Gamma(y)$ if $y \neq x$ and we write $\Gamma; x : \mathcal{M}$ for $\Gamma + (x : \mathcal{M})$, when $x \notin \mathtt{dom}(\Gamma)$. Notice that Γ and $\Gamma; x : [\,]$ are the same environment.

A **judgement** has the form $\Gamma \vdash^{(b,s)} t : \tau$, where b and s are two natural numbers, representing, respectively, the number of β-steps needed to normalize t, and the size of the normal form of t. The **typing system** $\mathcal{O}$ is defined by the rules in Fig. 1. We write $\triangleright \Gamma \vdash^{(b,s)} t : \tau$ if there is a (tree) **type derivation** of the judgement $\Gamma \vdash^{(b,s)} t : \tau$ using the rules of system $\mathcal{O}$. The term t is **$\mathcal{O}$-typable** (we may omit the name $\mathcal{O}$) iff there is an environment Γ, a type τ and counters (b, s) such that $\triangleright \Gamma \vdash^{(b,s)} t : \tau$. We use letters $\Phi, \Psi, \ldots$ to name type derivations, by writing for example $\Phi \triangleright \Gamma \vdash^{(b,s)} t : \tau$.

$$\frac{}{x : [\sigma] \vdash^{(0,0)} x : \sigma}\ (\mathtt{ax}) \qquad \frac{\Gamma \vdash^{(b,s)} t : \tau}{\Gamma \setminus\!\!\setminus x \vdash^{(b,s)} \lambda x.t : \Gamma(x) \Rightarrow \tau}\ (\lambda)$$

$$\frac{\Gamma \vdash^{(b,s)} t : \mathcal{M} \Rightarrow \tau \qquad \Delta \vdash^{(b',s')} u : \mathcal{M}}{\Gamma + \Delta \vdash^{(1+b+b',s+s')} tu : \tau}\ (\mathtt{@}) \qquad \frac{(\Gamma_i \vdash^{(b_i,s_i)} v : \sigma_i)_{i \in I}}{+_{i \in I} \Gamma_i \vdash^{(+_{i \in I} b_i, +_{i \in I} s_i)} v : [\sigma_i]_{i \in I}}\ (\mathtt{m})$$

$$\frac{}{\vdash^{(0,0)} \lambda x.t : \mathtt{a}}\ (\lambda_{\mathtt{p}})$$

$$\frac{\Gamma \vdash^{(b,s)} t : \overline{\mathtt{a}} \qquad \Delta \vdash^{(b',s')} u : \mathtt{tt}}{\Gamma + \Delta \vdash^{(b+b',1+s+s')} tu : \mathtt{n}}\ (\mathtt{@}_{\mathtt{p1}}) \qquad \frac{\Gamma \vdash^{(b,s)} t : \mathtt{tt} \qquad \Delta \vdash^{(b',s')} u : \mathtt{n}}{\Gamma + \Delta \vdash^{(b+b',1+s+s')} tu : \mathtt{n}}\ (\mathtt{@}_{\mathtt{p2}})$$

Fig. 1. Typing Rules of System $\mathcal{O}$

Notice that in rule $(\mathtt{ax})$ of Fig. 1 variables can only be assigned value types σ (in particular no type $\mathtt{n}$): this is because they can only be substituted by values. Due to this fact, multi-types only contain value types. Regarding typing rules $(\mathtt{ax})$, (λ), $(\mathtt{@})$, and $(\mathtt{m})$, they are the usual rules for non-idempotent intersection types [10]. Rules $(\lambda_{\mathtt{p}})$, $(@_{\mathtt{p1}})$, and $(@_{\mathtt{p2}})$ are used to type *persistent* symbols, *i.e.* symbols that are not going to be *consumed* during evaluation. More specifically, rule $(\lambda_{\mathtt{p}})$ types abstractions (with type $\mathtt{a}$) that are normal regardless of the typability of its body. Rule $(@_{\mathtt{p1}})$ types applications that will never reduce to an abstraction on the left (thus of any tight constant that is not $\mathtt{a}$, *i.e.* $\overline{\mathtt{a}}$), while any term reducing to a normal form is allowed on the right (of tight constant $\mathtt{tt}$). Rule $(@_{\mathtt{p2}})$ also types applications, but when values will never be obtained on the right (only neutral terms of type $\mathtt{n}$). Rule $(\mathtt{ax})$ is also used to type persistent variables, in particular when $\sigma \in \{\mathtt{v}, \mathtt{a}\}$.

A **type** τ is **tight** if $\tau \in \mathtt{tt}$. We write $\mathtt{tight}(\mathcal{M})$, if every $\sigma \in \mathcal{M}$ is tight. A **type environment** Γ is **tight** if it assigns tight multi-types to all variables. A **type derivation** $\Phi \triangleright \Gamma \vdash^{(b,s)} t : \tau$ is **tight** if Γ and τ are both tight.

Example 1. Let $t = (\lambda x.(xx)(yy))(\lambda z.z)$. Let Φ be the following typing derivation:

$$\dfrac{\dfrac{}{x : [[\mathtt{a}] \Rightarrow \mathtt{a}] \vdash^{(0,0)} x : [\mathtt{a}] \Rightarrow \mathtt{a}}\,(\mathtt{ax}) \qquad \dfrac{\dfrac{}{x : [\mathtt{a}] \vdash^{(0,0)} x : \mathtt{a}}\,(\mathtt{ax})}{x : [\mathtt{a}] \vdash^{(0,0)} x : [\mathtt{a}]}\,(\mathtt{m})}{x : [[\mathtt{a}] \Rightarrow \mathtt{a}, \mathtt{a}] \vdash^{(1,0)} xx : \mathtt{a}}\,(\mathtt{@})$$

And Ψ be the following typing derivation:

$$\dfrac{\dfrac{\Phi \qquad \dfrac{\dfrac{}{y : [\mathtt{v}] \vdash^{(0,0)} y : \mathtt{v}}\,(\mathtt{ax}) \qquad \dfrac{}{y : [\mathtt{v}] \vdash^{(0,0)} y : \mathtt{v}}\,(\mathtt{ax})}{y : [\mathtt{v}, \mathtt{v}] \vdash^{(0,1)} yy : \mathtt{n}}\,(@_{\mathtt{p1}})}{x : [[\mathtt{a}] \Rightarrow \mathtt{a}, \mathtt{a}], y : [\mathtt{v}, \mathtt{v}] \vdash^{(1,1)} (xx)(yy) : \mathtt{n}}\,(@_{\mathtt{p2}})}{y : [\mathtt{v}, \mathtt{v}] \vdash^{(1,2)} \lambda x.(xx)(yy) : [[\mathtt{a}] \Rightarrow \mathtt{a}, \mathtt{a}] \Rightarrow \mathtt{n}}\,(\lambda)$$

Then, we can build the following tight typing derivation Φ_t for t:

$$\dfrac{\Psi \qquad \dfrac{\dfrac{\dfrac{}{z : [\mathtt{a}] \vdash^{(0,0)} z : \mathtt{a}}\,(\lambda_{\mathtt{p}})}{\vdash^{(0,0)} \lambda z.z : [\mathtt{a}] \Rightarrow \mathtt{a}}\,(\lambda) \qquad \dfrac{}{\vdash^{(0,0)} \lambda z.z : \mathtt{a}}\,(\lambda_{\mathtt{p}})}{\vdash^{(0,0)} \lambda z.z : [[\mathtt{a}] \Rightarrow \mathtt{a}, \mathtt{a}]}\,(\mathtt{m})}{y : [\mathtt{v}, \mathtt{v}] \vdash^{(2,2)} (\lambda x.(xx)(yy))(\lambda z.z) : \mathtt{n}}\,(\mathtt{@})$$

The type system $\mathcal{O}$ can be shown to be *sound* and *complete* w.r.t. the operational semantics $\rightarrow$ introduced in Sect. 2.1. Soundness means that not only a *tightly* typable term t is terminating, but also that the *tight* type derivation of t gives exact and split measures concerning the reduction sequence from t to normal form. More precisely, if $\Phi \triangleright \Gamma \vdash^{(b,s)} t : \tau$ is tight, then there exists $u \in \mathtt{no}$ such that $t \twoheadrightarrow^{b} u$ with $|u| = s$. Dually for *completeness*. Because we are going to show this kind of properties for the more sophisticated language with global state (Sect. 3.3), we do not give here technical details of them. However, we highlight these properties on our previous example. Consider again term t in Example 1 and its tight derivation Φ_t with counters $(b, s) = (2, 2)$. Counter b is different from 0, so $t \notin \mathtt{no}$, but t normalizes in two β_v-steps ($b = 2$) to a normal form having size $s = 2$. Indeed, $(\lambda x.(xx)(yy))(\lambda z.z) \rightarrow_{\beta_v} ((\lambda z.z)(\lambda z.z))(yy) \rightarrow_{\beta_v} (\lambda z.z)(yy)$ and $|(\lambda z.z)(yy)| = 2$.

3 A λ-Calculus with Global State

Based on the preliminary presentation of Sect. 2, we now introduce a λ-calculus with global state following a CBV strategy. Section 3.1 defines the syntax and operational semantics of the λ-calculus with global state. Section 3.2 presents the tight typing system $\mathcal{P}$, and Sect. 3.3 shows soundness and completeness.

3.1 Syntax and Operational Semantics

Let l be a location drawn from some set of location names. **Values**, **terms**, **states** and **configurations** of $\lambda_{\mathtt{gs}}$ are defined respectively as follows:

$$v, w ::= x \mid \lambda x.t \qquad t, u, p ::= v \mid vt \mid \mathtt{get}_l(\lambda x.t) \mid \mathtt{set}_l(v, t)$$
$$s, q ::= \epsilon \mid \mathtt{upd}_l(v, s) \qquad c ::= (t, s)$$

Notice that applications are restricted to the form vt. This, combined with the use of a deterministic reduction strategy based on weak contexts, ensures that composition of effects is well behaved. Indeed, this kind of restriction is usual in computational calculi [16,19,30,32].

Intuitively, operation $\mathtt{get}_l(\lambda x.t)$ interacts with the global state by retrieving the value stored in location l and binding it to variable x of the continuation t. And operation $\mathtt{set}_l(v, t)$ interacts with the state by storing value v in location l and (possibly) overwriting whatever was previously stored there, and then returns t.

The size function is extended to states and configurations: $|s| := 0$, and $|(t, s)| := |t|$. The update constructor is commutative in the following sense:

$$\mathtt{upd}_l(v, \mathtt{upd}_{l'}(w, s)) \equiv_{\mathtt{c}} \mathtt{upd}_{l'}(w, \mathtt{upd}_l(v, s)) \text{ if } l \neq l'$$

We denote by $\equiv$ the equivalence relation generated by the axiom $\equiv_{\mathtt{c}}$. We write $l \in \mathtt{dom}(s)$, if $s \equiv \mathtt{upd}_l(v, q)$, for some value v and state q. Moreover, these v and q are *unique*. For example, if $l_1 \neq l_2$, then $s_1 = \mathtt{upd}_{l_1}(v_1, \mathtt{upd}_{l_2}(v_2, q)) \equiv \mathtt{upd}_{l_2}(v_2, \mathtt{upd}_{l_1}(v_1, q)) = s_2$, but $\mathtt{upd}_{l_1}(v_1, \mathtt{upd}_{l_1}(v_2, s)) \not\equiv \mathtt{upd}_{l_1}(v_2, \mathtt{upd}_{l_1}(v_1, s))$. As a consequence, whenever we want to access the content of a particular location in a state, we can simply assume that the location is at the top of the state.

The operational semantics of the $\lambda_{\mathtt{gs}}$-calculus is given on configurations. The **deterministic reduction relation** $\rightarrow$ is defined to be the union of the rules $\rightarrow_{\mathtt{r}}$ ($\mathtt{r} \in \{\beta_v, \mathtt{g}, \mathtt{s}\}$) below. We write $(t, s) \twoheadrightarrow^{(b,m)} (u, q)$ if (t, s) reduces to (u, q) in b β_v-steps and m $\mathtt{g}/\mathtt{s}$-steps.

$$\frac{}{((\lambda x.t)v, s) \rightarrow_{\beta_v} (t\{x\backslash v\}, s)}\ (\beta_{\mathtt{v}}) \qquad \frac{s \equiv \mathtt{upd}_l(v, q)}{(\mathtt{get}_l(\lambda x.t), s) \rightarrow_{\mathtt{g}} (t\{x\backslash v\}, s)}\ (\mathtt{get})$$

$$\frac{(t, s) \rightarrow_{\mathtt{r}} (u, q) \quad \mathtt{r} \in \{\beta_v, \mathtt{g}, \mathtt{s}\}}{(vt, s) \rightarrow_{\mathtt{r}} (vu, q)}\ (\mathtt{appR}) \qquad \frac{}{(\mathtt{set}_l(v, t), s) \rightarrow_{\mathtt{s}} (t, \mathtt{upd}_l(v, s))}\ (\mathtt{set})$$

Note that in reduction rule ($\mathtt{appR}$), the $\mathtt{r}$ appearing as the name of the reduction rule in the premise is the same as the one appearing in the reduction rule in the conclusion.

Example 2. Consider the configuration $c_0 = ((\lambda x.\mathtt{get}_l(\lambda y.yx))(\mathtt{set}_l(\mathtt{I}, z)), \epsilon)$. Then we can reach an irreducible configuration as follows:

$$((\lambda x.\mathtt{get}_l(\lambda y.yx))(\mathtt{set}_l(\mathtt{I}, z)), \epsilon) \rightarrow_{\mathtt{g}} ((\lambda x.\mathtt{get}_l(\lambda y.yx))z, \mathtt{upd}_l(\mathtt{I}, \epsilon))$$
$$\rightarrow_{\beta_v} (\mathtt{get}_l(\lambda y.yz), \mathtt{upd}_l(\mathtt{I}, \epsilon)) \rightarrow_{\mathtt{g}} (\mathtt{I}z, \mathtt{upd}_l(\mathtt{I}, \epsilon)) \rightarrow_{\beta_v} (z, \mathtt{upd}_l(\mathtt{I}, \epsilon))$$

A configuration (t, s) is said to be **blocked** if either $t = \mathtt{get}_l(\lambda x.u)$ and $l \notin \mathtt{dom}(s)$; or $t = vu$ and (u, s) is blocked. A configuration is **unblocked** if it is not blocked. As an example, $(\mathtt{get}_l(\lambda x.x), \epsilon)$ is obviously blocked. As a consequence, the following configuration reduces to a blocked one: $((\lambda y.y\ \mathtt{get}_l(\lambda x.x))z, \epsilon) \rightarrow (z\ \mathtt{get}_l(\lambda x.x), \epsilon)$. This suggests a notion of **final configuration**: (t, s) is **final** if either (t, s) is blocked; or $t \in \mathtt{no}$, where **neutral** and **normal** terms are given respectively by the grammars $\mathtt{ne} ::= x\ \mathtt{no} \mid (\lambda x.t)\ \mathtt{ne}$ and $\mathtt{no} ::= \mathtt{Val} \mid \mathtt{ne}$.

Proposition 2. *Let (t, s) be a configuration. Then (t, s) is final iff $(t, s) \not\rightarrow$.*

Notice that when (t, s) is an unblocked final configuration, then $t \in \mathtt{no}$. These are the configurations captured by the typing system $\mathcal{P}$ in Sect. 3.2. Consider the final configurations $c_0 = (\mathtt{get}_l(\lambda x.x), \epsilon)$, $c_1 = (z\ \mathtt{get}_l(\lambda x.x), \epsilon)$, $c_2 = (y, s)$ and $c_3 = ((\lambda x.x)(yz), s)$. Then c_0 and c_1 are blocked, while c_2 and c_3 are unblocked.

3.2 A Quantitative Type System for the $\lambda_{\mathtt{gs}}$-Calculus

We now introduce the quantitative type system $\mathcal{P}$ for $\lambda_{\mathtt{gs}}$. To deal with global states, we extend the language of types with the notions of state, configuration and monadic types. To do this, we translate linear arrow types according to Moggi's [30] CBV interpretation of reflexive objects in the category of λ_c-models: $D = !D \multimap !D$ becomes $D = !D \multimap T(!D)$, where T is a monad. Type system $\mathcal{P}$ was built having this equation in mind, similarly to what was done in [21].

The **set of types** is given by the following grammar:

(Tight Constants)	$\mathtt{tt} ::= \mathtt{v} \mid \mathtt{a} \mid \mathtt{n}$
(Value Types)	$\sigma ::= \mathtt{v} \mid \mathtt{a} \mid \mathcal{M} \mid \mathcal{M} \Rightarrow \delta$
(Multi-types)	$\mathcal{M} ::= [\sigma_i]_{i \in I}$ where I is a finite set
(Liftable Types)	$\mu ::= \mathtt{v} \mid \mathtt{a} \mid \mathcal{M}$
(Types)	$\tau ::= \mathtt{n} \mid \sigma$
(State Types)	$\mathcal{S} ::= \{(l_i : \mathcal{M}_i)\}_{i \in I}$ where all l_i are distinct
(Configuration Types)	$\kappa ::= \tau \times \mathcal{S}$
(Monadic Types)	$\delta ::= \mathcal{S} \gg \kappa$

In system $\mathcal{P}$, the minimal types to be assigned to normal forms distinguish between variables ($\mathtt{v}$), abstractions ($\mathtt{a}$), and neutral terms ($\mathtt{n}$). A **multi-type** is a multi-set of value types. A **state type** is a partial function mapping labels to (possibly empty) multi-types. A **configuration type** is a product type, where the first component is a type and the second is a state type. A **monadic type** associates a state type to a configuration type. We use symbol $\mathcal{T}$ to denote a value type or a monadic type. **Typing environments** and operations over types are defined in the same way as in system $\mathcal{O}$.

The **domain** of a state type $\mathcal{S}$ is the set of all its labels, *i.e.* $\mathtt{dom}(\mathcal{S}) := \{l \mid (l : \mathcal{M}) \in \mathcal{S}\}$. Also, when $l \in \mathtt{dom}(\mathcal{S})$, *i.e.* $(l : \mathcal{M}) \in \mathcal{S}$, we write $\mathcal{S}(l)$ to denote $\mathcal{M}$. The **union of state types** is defined as follows:

$$(\mathcal{S} \uplus \mathcal{S}')(l) = \text{if } (l : \mathcal{M}) \in \mathcal{S} \text{ then } (\text{if } (l : \mathcal{M}') \in \mathcal{S}' \text{ then } \mathcal{M} \sqcup \mathcal{M}' \text{ else } \mathcal{M}) \\ \text{else } (\text{if } (l : \mathcal{M}') \in \mathcal{S}' \text{ then } \mathcal{M}' \text{ else } \mathtt{undefined})$$

Example 3. Let $\mathcal{S} = \{(l_1 : [\sigma_1, \sigma_2]), (l_2 : [\sigma_1])\} \uplus \{(l_2 : [\sigma_1, \sigma_2]), (l_3 : [\sigma_3])\}$. Then, $\mathcal{S}(l_1) = [\sigma_1, \sigma_2]$, $\mathcal{S}(l_2) = [\sigma_1, \sigma_1, \sigma_2]$, $\mathcal{S}(l_3) = [\sigma_3]$, and $\mathcal{S}(l) = \texttt{undefined}$, assuming $l \neq l_i$, for $i \in \{1, 2, 3\}$.

Notice that $\texttt{dom}(\mathcal{S} \uplus \mathcal{S}') = \texttt{dom}(\mathcal{S}) \cup \texttt{dom}(\mathcal{S}')$. Also $\{(l : [\,])\} \uplus \mathcal{S} \neq \mathcal{S}$, if $l \notin \texttt{dom}(\mathcal{S})$, while $x : [\,]; \Gamma = \Gamma$. Indeed, typing environments are total functions, where variables mapped to $[\,]$ do not occur in typed programs. In contrast, states are partial functions, where labels mapped to $[\,]$ correspond to positions in memory that are accessed (by get or set), but ignored/discarded by the typed program. We use $\{(l : \mathcal{M})\}; \mathcal{S}$ for $\{(l : \mathcal{M})\} \uplus \mathcal{S}$ if $l \notin \texttt{dom}(\mathcal{S})$.

A **term type judgement** (resp. **state type judgment** and **configuration type judgment**) has the form $\Gamma \vdash^{(b,m,d)} t : \mathcal{T}$ (resp. $\Gamma \vdash^{(b,m,d)} s : \mathcal{S}$ and $\Gamma \vdash^{(b,m,d)} (t, s) : \kappa$) where b, m, d are three natural numbers, the first and second representing, respectively, the number of β-steps and $\texttt{g}/\texttt{s}$-steps needed to normalize t, and the third representing the size of the normal form of t. The **typing system** $\mathcal{P}$ is defined by the rules in Fig. 2. We write $\triangleright \mathcal{J}$ if there is a type derivation of the judgement $\mathcal{J}$ using the rules of system $\mathcal{P}$. The term t (resp. state s, configuration (t, s)) is **$\mathcal{P}$-typable** iff there is an environment Γ, a type $\mathcal{T}$ (resp. $\mathcal{S}$, κ) and counters (b, m, d) such that $\triangleright \Gamma \vdash^{(b,m,d)} t : \mathcal{T}$ (resp. $\triangleright \Gamma \vdash^{(b,m,d)} s : \mathcal{S}$, $\triangleright \Gamma \vdash^{(b,m,d)} (t, s) : \kappa$). As before, we use letters $\Phi, \Psi, \ldots$ to name type derivations.

Rules $(\texttt{ax})$, (λ), $(\texttt{m})$, and $(\texttt{@})$ are essentially the same as in Fig. 1, but with types lifted to monadic types (*i.e. decorated* with state types). Rule $(\texttt{@})$ assumes a value type associated to a value v on the left premise and a monadic type associated to a term t on the right premise. To type the application vt, it is necessary to match both the value type $\mathcal{M}$ inside the type of t with the input value type of v, and the output state type $\mathcal{S}'$ of t with the input state type of v. Rule $(\uparrow)$ is used to lift multi-types or tight constants $\texttt{v}$ and $\texttt{a}$ (the type of values) to monadic types. Rules $(\texttt{get})$ and $(\texttt{set})$ are used to type operations over the state. Rule $(\texttt{emp})$ types empty states, rule $(\texttt{upd})$ types states, and $(\texttt{conf})$ types configurations.

A **type** τ is **tight**, if $\tau \in \texttt{tt}$. We write $\texttt{tight}(\mathcal{M})$ if every $\sigma \in \mathcal{M}$ is tight. A **state type** $\mathcal{S}$ is **tight** if $\texttt{tight}(\mathcal{S}(l))$ holds for all $l \in \texttt{dom}(\mathcal{S})$. A **configuration type** $\tau \times \mathcal{S}$ is **tight**, if τ and $\mathcal{S}$ are tight. A monadic type $\mathcal{S} \gg \kappa$ is **tight**, if κ is tight. The notion of tightness of type derivations is defined in the same way as in system $\mathcal{O}$, *i.e.* a **type derivation** Φ is **tight** if the type environment and the type of the conclusion of Φ are tight.

Rules for Terms

$$\frac{}{x:[\sigma]\vdash^{(0,0,0)} x:\sigma}\;(\mathtt{ax}) \qquad \frac{\Gamma\vdash^{(b,m,d)} v:\mu}{\Gamma\vdash^{(b,m,d)} v:\mathcal{S}\gg(\mu\times\mathcal{S})}\;(\uparrow)$$

$$\frac{\Gamma\vdash^{(b,m,d)} t:\mathcal{S}\gg\kappa}{\Gamma\setminus\!\!\setminus x\vdash^{(b,m,d)} \lambda x.t:\Gamma(x)\Rightarrow(\mathcal{S}\gg\kappa)}\;(\lambda) \qquad \frac{(\Gamma_i\vdash^{(b_i,m_i,d_i)} v:\sigma_i)_{i\in I}}{+_{i\in I}\Gamma_i\vdash^{(+_{i\in I}b_i,+_{i\in I}m_i,+_{i\in I}d_i)} v:[\sigma_i]_{i\in I}}\;(\mathtt{m})$$

$$\frac{\Gamma\vdash^{(b,m,d)} v:\mathcal{M}\Rightarrow(\mathcal{S}'\gg\kappa) \qquad \Delta\vdash^{(b',m',d')} t:\mathcal{S}\gg(\mathcal{M}\times\mathcal{S}')}{\Gamma+\Delta\vdash^{(1+b+b',m+m',d+d')} vt:\mathcal{S}\gg\kappa}\;(\mathtt{@})$$

$$\frac{\Gamma\vdash^{(b,m,d)} t:\mathcal{S}\gg\kappa}{\Gamma\setminus\!\!\setminus x\vdash^{(b,1+m,d)} \mathtt{get}_l(\lambda x.t):\{(l:\Gamma(x))\}\uplus\mathcal{S}\gg\kappa}\;(\mathtt{get})$$

$$\frac{\Gamma\vdash^{(b,m,d)} v:\mathcal{M} \qquad \Delta\vdash^{(b',m',d')} t:\{(l:\mathcal{M})\};\mathcal{S}\gg\kappa}{\Gamma+\Delta\vdash^{(b+b',1+m+m',d+d')} \mathtt{set}_l(v,t):\mathcal{S}\gg\kappa}\;(\mathtt{set})$$

$$\frac{}{\vdash^{(0,0,0)} \lambda x.t:\mathtt{a}}\;(\lambda_{\mathtt{p}})$$

$$\frac{\Gamma\vdash^{(b,m,d)} t:\mathcal{S}\gg(\mathtt{tt}\times\mathcal{S}')}{(x:[\mathtt{v}])+\Gamma\vdash^{(b,m,1+d)} xt:\mathcal{S}\gg(\mathtt{n}\times\mathcal{S}')}\;(\mathtt{@}_{\mathtt{p1}}) \qquad \frac{\Gamma\vdash^{(b,m,d)} u:\mathcal{S}\gg(\mathtt{n}\times\mathcal{S}')}{\Gamma\vdash^{(b,m,1+d)} (\lambda x.t)u:\mathcal{S}\gg(\mathtt{n}\times\mathcal{S}')}\;(\mathtt{@}_{\mathtt{p2}})$$

Rules for States

$$\frac{}{\vdash^{(0,0,0)} \epsilon:\emptyset}\;(\mathtt{emp}) \qquad \frac{\Gamma\vdash^{(b,m,d)} v:\mathcal{M} \qquad \Delta\vdash^{(b',m',d')} s:\mathcal{S}}{\Gamma+\Delta\vdash^{(b+b',m+m',d+d')} \mathtt{upd}_l(v,s):\{(l:\mathcal{M})\};\mathcal{S}}\;(\mathtt{upd})$$

Rule for Configurations

$$\frac{\Gamma\vdash^{(b,m,d)} t:\mathcal{S}\gg\kappa \qquad \Delta\vdash^{(b',m',d')} s:\mathcal{S}}{\Gamma+\Delta\vdash^{(b+b',m+m',d+d')} (t,s):\kappa}\;(\mathtt{conf})$$

Fig. 2. Typing Rules for $\lambda_{\mathtt{gs}}$.

Example 4. Consider configuration c_0 from Example 2. Let $\mathcal{M}=[[\mathtt{v}]\Rightarrow\emptyset\gg(\mathtt{v}\times\emptyset)]$, and Φ be the following typing derivation:

$$\frac{\dfrac{\dfrac{\dfrac{}{y:\mathcal{M}\vdash^{(0,0,0)} y:[\mathtt{v}]\Rightarrow\emptyset\gg(\mathtt{v}\times\emptyset)}\;(\mathtt{ax}) \qquad \dfrac{\dfrac{\dfrac{}{x:[\mathtt{v}]\vdash^{(0,0,0)} x:\mathtt{v}}\;(\mathtt{ax})}{x:[\mathtt{v}]\vdash^{(0,0,0)} x:[\mathtt{v}]}\;(\mathtt{m})}{x:[\mathtt{v}]\vdash^{(0,0,0)} x:\emptyset\gg([\mathtt{v}]\times\emptyset)}\;(\uparrow)}{y:\mathcal{M},x:[\mathtt{v}]\vdash^{(1,0,0)} yx:\emptyset\gg(\mathtt{v}\times\emptyset)}\;(\mathtt{@})}{x:[\mathtt{v}]\vdash^{(1,1,0)} \mathtt{get}_l(\lambda y.yx):\{(l:\mathcal{M})\}\gg(\mathtt{v}\times\emptyset)}\;(\mathtt{get})}{\vdash^{(1,1,0)} \lambda x.\mathtt{get}_l(\lambda y.yx):[\mathtt{v}]\Rightarrow(\{(l:\mathcal{M})\}\gg(\mathtt{v}\times\emptyset))}\;(\lambda)$$

And Φ' be the following typing derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{x : [\mathtt{v}] \vdash^{(0,0,0)} x : \mathtt{v}}\;(\mathtt{ax})}{x : [\mathtt{v}] \vdash^{(0,0,0)} x : \emptyset \gg (\mathtt{v} \times \emptyset)}\;(\uparrow)}{\vdash^{(0,0,0)} \mathtt{I} : [\mathtt{v}] \Rightarrow \emptyset \gg (\mathtt{v} \times \emptyset)}\;(\lambda)}{\vdash^{(0,0,0)} \mathtt{I} : \mathcal{M}}\;(\mathtt{m}) \qquad \dfrac{\dfrac{\dfrac{}{z : [\mathtt{v}] \vdash^{(0,0,0)} z : \mathtt{v}}\;(\mathtt{ax})}{z : [\mathtt{v}] \vdash^{(0,0,0)} z : [\mathtt{v}]}\;(\mathtt{m})}{z : [\mathtt{v}] \vdash^{(0,0,0)} z : \{(l : \mathcal{M})\} \gg ([\mathtt{v}] \times \{(l : \mathcal{M})\})}\;(\uparrow)}{z : [\mathtt{v}] \vdash^{(0,1,0)} \mathtt{set}_l(\mathtt{I}, z) : \emptyset \gg ([\mathtt{v}] \times \{(l : \mathcal{M})\})}\;(\mathtt{set})$$

Then we can build the following tight typing derivation Φ_c for c:

$$\dfrac{\dfrac{\Phi \qquad \Phi'}{z : [\mathtt{v}] \vdash^{(2,2,0)} (\lambda x.\mathtt{get}_l(\lambda y.yx))(\mathtt{set}_l(\mathtt{I}, z)) : \emptyset \gg (\mathtt{v} \times \emptyset)}\;(\mathtt{@}) \qquad \dfrac{}{\vdash^{(0,0,0)} \epsilon : \emptyset}\;(\mathtt{emp})}{z : [\mathtt{v}] \vdash^{(2,2,0)} ((\lambda x.\mathtt{get}_l(\lambda y.yx))(\mathtt{set}_l(\mathtt{I}, z)), \epsilon) : \mathtt{v} \times \emptyset}\;(\mathtt{conf})$$

We will come back to this example at the end of Sect. 3.3.

3.3 Soundness and Completeness

In this section, we show the main properties of the type system $\mathcal{P}$ with respect to the operational semantics of the λ-calculus with global state introduced in Sect. 3.1. The properties of type system $\mathcal{P}$ are similar to the ones for $\mathcal{O}$, but now with respect to configurations instead of terms. *Soundness* does not only state that a (tightly) typable configuration (t, s) is terminating, but also gives exact (and split) measures concerning the reduction sequence from (t, s) to a final form. *Completeness* guarantees that a terminating configuration (t, s) is tightly typable, where the measures of the associated reduction sequence of (t, s) to final form are reflected in the counters of the resulting type derivation of (t, s). This is the first work providing a model for a language with global memory being able to count the number of memory accesses.

We start by noting that type system $\mathcal{P}$ does not type blocked configurations, which is exactly the notion that we want to capture.

Proposition 3. *If* $\Phi \triangleright \Gamma \vdash^{(b,m,d)} (t, s) : \kappa$, *then* (t, s) *is unblocked.*

We also show that counters capture the notion of normal form correctly, both for terms and states.

Lemma 1.

1. *Let* $\Phi \triangleright \Gamma \vdash^{(0,0,d)} t : \delta$ *be tight. Then, (1)* $t \in \mathtt{no}$ *and (2)* $d = |t|$.
2. *Let* $\Phi \triangleright \Delta \vdash^{(0,0,d)} s : \mathcal{S}$ *be tight. Then* $d = 0$.

In fact, we can show the following stronger property with respect to the counters for the number of β_v- and $\mathtt{g}/\mathtt{s}$-steps.

Lemma 2. *Let* $\Phi \triangleright \Gamma \vdash^{(b,m,d)} t : \delta$ *be tight. Then,* $b = m = 0$ *iff* $t \in \mathtt{no}$.

The following property is essential for tight type systems [2], and it shows that tightness of types spreads throughout type derivations of neutral terms, just as long as the environments are tight.

Lemma 3 (Tight Spreading). *Let $\Phi \triangleright \Gamma \vdash^{(b,m,d)} t : \mathcal{S} \gg (\tau \times \mathcal{S}')$, such that Γ is tight. If $t \in \mathtt{ne}$, then $\tau \in \mathtt{tt}$.*

The two following properties ensure tight typability of final configurations. For that we need to be able to *tightly* type any state, as well as any normal form. In fact, normal forms do not depend on a particular state since they are irreducible, so we can type them with any state type.

Lemma 4 (Typability of States and Normal Forms).

1. *Let s be a state. Then, there exists $\Phi \triangleright \vdash^{(0,0,0)} s : \mathcal{S}$ tight.*
2. *Let $t \in \mathtt{no}$. Then for any tight $\mathcal{S}$ there exists $\Phi \triangleright \Gamma \vdash^{(0,0,d)} t : \mathcal{S} \gg (\mathtt{tt} \times \mathcal{S})$ tight s.t. $d = |t|$.*

Finally, we state the usual basic properties.

Lemma 5 (Substitution and Anti-Substitution).

1. **(Substitution)** *If $\Phi_t \triangleright \Gamma_t; x : \mathcal{M} \vdash^{(b_t,m_t,d_t)} t : \delta$ and $\Phi_v \triangleright \Gamma_v \vdash^{(b_v,m_v,d_v)} v : \mathcal{M}$, then $\Phi_{t\{x\backslash v\}} \triangleright \Gamma_t + \Gamma_v \vdash^{(b_t+b_v,m_t+m_v,d_t+d_v)} t\{x\backslash v\} : \delta$.*
2. **(Anti-Substitution)** *If $\Phi_{t\{x\backslash v\}} \triangleright \Gamma_{t\{x\backslash v\}} \vdash^{(b,m,d)} t\{x\backslash v\} : \delta$, then $\Phi_t \triangleright \Gamma_t; x : \mathcal{M} \vdash^{(b_t,m_t,d_t)} t : \delta$ and $\Phi_v \triangleright \Gamma_v \vdash^{(b_v,m_v,d_v)} v : \mathcal{M}$, such that $\Gamma_{t\{x\backslash v\}} = \Gamma_t + \Gamma_v$, $b = b_t + b_v$, $m = m_t + m_v$, and $d = d_t + d_v$.*

Lemma 6 (Split Exact Subject Reduction and Expansion).

1. **(Subject Reduction)** *Let $(t, s) \to_{\mathtt{r}} (u, q)$. If $\Phi \triangleright \Gamma \vdash^{(b,m,d)} (t, s) : \kappa$ is tight, then $\Phi' \triangleright \Gamma \vdash^{(b',m',d)} (u, q) : \kappa$, where $\mathtt{r} = \beta$ implies $b' = b - 1$ and $m' = m$, while $\mathtt{r} \in \{\mathtt{g}, \mathtt{s}\}$ implies $b' = b$ and $m' = m - 1$.*
2. **(Subject Expansion)** *Let $(t, s) \to_{\mathtt{r}} (u, q)$. If $\Phi' \triangleright \Gamma \vdash^{(b',m',d)} (u, q) : \kappa$ is tight, then $\Phi \triangleright \Gamma \vdash^{(b,m,d)} (t, s) : \kappa$, where $\mathtt{r} = \beta$ implies $b' = b-1$ and $m' = m$, while $\mathtt{r} \in \{\mathtt{g}, \mathtt{s}\}$ implies $b' = b$ and $m' = m - 1$.*

Soundness (resp. completeness) is based on exact subject reduction (resp. expansion), in turn based on the previous substitution (resp. anti-substitution) lemma.

Theorem 1 (Quantitative Soundness and Completeness).

1. **(Soundness)** *If $\Phi \triangleright \Gamma \vdash^{(b,m,d)} (t, s) : \kappa$ tight, then there exists (u, q) such that $u \in \mathtt{no}$ and $(t, s) \twoheadrightarrow^{(b,m)} (u, q)$ with b β-steps, m $\mathtt{g}/\mathtt{s}$-steps, and $|(u, q)| = d$.*
2. **(Completeness)** *If $(t, s) \twoheadrightarrow^{(b,m,d)} (u, q)$ and $u \in \mathtt{no}$, then there exists $\Phi \triangleright \Gamma \vdash^{(b,m,|(u,q)|)} (t, s) : \kappa$ tight.*

Example 5. Consider again configuration c_0 from Example 2 and its associated tight derivation Φ_{c_0}. The first two counters of Φ_c are different from 0: this means that c is not a final configuration, but normalizes in two β_v-step ($b = 2$) and two $\mathtt{g}/\mathtt{s}$-steps ($m = 2$), to a final configuration having size $d = 0 = |z| = |(z, \mathtt{upd}_l(I, \epsilon))|$.

4 Conclusion and Related Work

This paper provides a foundational step into the development of quantitative models for programming languages with effects. We focus on a simple language with global memory access capabilities. Due to the inherent lack of confluence in such framework we fix a particular evaluation strategy following a (weak) CBV approach. We provide a type system for our language that is able to (both) extract and discriminate between (exact) measures for the length of evaluation, number of memory accesses and size of normal forms. This study provides a valuable insight into time and space analysis of languages with global memory, with respect to length of evaluation and the size of normal forms, respectively.

In future work we would like to explore effectful computations involving global memory in a more general framework being able to capture different models of computation, such as the CBPV [28] or the bang calculus [9]. Furthermore, we would like to apply our quantitative techniques to other effects that can be found in programming languages, such as non-termination, exceptions, non-determinism, and I/O.

Related Work. Several papers proposed quantitative approaches for different notions of CBV (without effects). But none of them exploits the idea of exact *and* split tight typing. Indeed, the first non-idempotent intersection type system for Plotkin's CBV is [18], where reduction is allowed under abstractions, and terms are considered to be closed. This work was further extended to [11], where commutation rules are added to the calculus. None of these contributions extracts quantitative bounds from the type derivations. A calculus for open CBV is proposed in [3], where *fireball* –normal forms– can be either erased or duplicated. Quantitative results are obtained, but no split measures. Other similar approaches appear in [23]. A logical characterization of CBV solvability is given in [4], the resulting non-idempotent system gives quantitative information of the *solvable* associated reduction relation. A similar notion of solvability for CBV for generalized applications was studied in [26], together with a logical characterization provided by a quantitative system. Other non-idempotent systems for CBV were proposed [25,29], but they are defective in the sense that they do not enjoy subject reduction and expansion. Split measures for (strong) open CBV are developed in [27].

In [17], a system with universally quantified intersection and reference types is introduced for a language belonging to the ML-family. However, intersections are idempotent and only (qualitative) soundness is proved.

More recently, there has been a lot of work involving probabilistic versions of the lambda calculus. In [20], extensions of the lambda calculus with a probabilistic choice operator are introduced. However, no quantitative results are provided. In [8], monadic intersection types are used to obtain a (non-exact) quantitative model for a probabilistic calculus identical to the one in [20].

Concerning (exact) quantitative models for programming languages with global state, the state of the art is still underexplored. Some sound but not complete approaches are given in [7,15], and quantitative results are not provided. Our work is inspired by a recent idempotent (thus only qualitative and not quantitative) model for CBV with global memory proposed by [16]. This work was further extended in [21] to a more generic framework of algebraic effectful computation, still the model does not provide any quantitative information about the evaluation of programs and the size of their results.

References

1. Accattoli, B.: Proof nets and the call-by-value λ-calculus. Theor. Comput. Sci. **606**, 2–24 (2015). https://doi.org/10.1016/j.tcs.2015.08.006
2. Accattoli, B., Graham-Lengrand, S., Kesner, D.: Tight typings and split bounds, fully developed. J. Funct. Program. **30**(e14), 1–101 (2020). https://doi.org/10.1017/S095679682000012X
3. Accattoli, B., Guerrieri, G.: Types of fireballs. In: Ryu, S. (ed.) APLAS 2018. LNCS, vol. 11275, pp. 45–66. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-02768-1_3
4. Accattoli, B., Guerrieri, G.: The theory of call-by-value solvability. Proc. ACM Program. Lang. **6**(ICFP), 855–885 (2022). https://doi.org/10.1145/3547652
5. Alves, S., Kesner, D., Ramos, M.: Quantitative global memory (2023). https://arxiv.org/abs/2303.08940
6. Atkey, R.: Syntax and semantics of quantitative type theory. In: Dawar, A., Grädel, E. (eds.) Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, (LICS), Oxford, UK, pp. 56–65. ACM (2018). https://doi.org/10.1145/3209108.3209189
7. Benton, N., Kennedy, A., Beringer, L., Hofmann, M.: Relational semantics for effect-based program transformations: higher-order store. In: Porto, A., López-Fraguas, F.J. (eds.) 11th International ACM SIGPLAN Conference on Principles and Practice of Declarative Programming, (PPDP), Coimbra, Portugal, pp. 301–312. ACM (2009). https://doi.org/10.1145/1599410.1599447
8. Breuvart, F., Lago, U.D.: On intersection types and probabilistic lambda calculi. In: Sabel, D., Thiemann, P. (eds.) Proceedings of the 20th International Symposium on Principles and Practice of Declarative Programming, (PPDP), Frankfurt am Main, Germany, pp. 8:1–8:13. ACM (2018). https://doi.org/10.1145/3236950.3236968
9. Bucciarelli, A., Kesner, D., Ríos, A., Viso, A.: The bang calculus revisited. In: Nakano, K., Sagonas, K. (eds.) FLOPS 2020. LNCS, vol. 12073, pp. 13–32. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-59025-3_2
10. Bucciarelli, A., Kesner, D., Ventura, D.: Non-idempotent intersection types for the lambda-calculus. Log. J. (IGPL) **25**(4), 431–464 (2017). https://doi.org/10.1093/jigpal/jzx018
11. Carraro, A., Guerrieri, G.: A semantical and operational account of call-by-value solvability. In: Muscholl, A. (ed.) FoSSaCS 2014. LNCS, vol. 8412, pp. 103–118. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54830-7_7
12. de Carvalho, D.: Sémantiques de la logique linéaire et temps de calcul. These de doctorat, Université Aix-Marseille II (2007)
13. de Carvalho, D.: Execution time of λ-terms via denotational semantics and intersection types. Math. Struct. Comput. Sci. **28**(7), 1169–1203 (2018). https://doi.org/10.1017/S0960129516000396
14. Coppo, M., Dezani-Ciancaglini, M.: A new type assignment for lambda-terms. Archiv für Math. Logik **19**, 139–156 (1978)

15. Davies, R., Pfenning, F.: Intersection types and computational effects. In: Proceedings of the Fifth ACM SIGPLAN International Conference on Functional Programming, (ICFP), Montreal, Canada, pp. 198–208. ACM (2000). https://doi.org/10.1145/351240.351259
16. de'Liguoro, U., Treglia, R.: Intersection types for a λ-calculus with global store. In: Veltri, N., Benton, N., Ghilezan, S. (eds.) 23rd International Symposium on Principles and Practice of Declarative Programming, (PPDP), Tallinn, Estonia, pp. 5:1–5:11. ACM (2021). https://doi.org/10.1145/3479394.3479400
17. Dezani-Ciancaglini, M., Giannini, P., Della Rocca, S.R.: Intersection, universally quantified, and reference types. In: Grädel, E., Kahle, R. (eds.) CSL 2009. LNCS, vol. 5771, pp. 209–224. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04027-6_17
18. Ehrhard, T.: Collapsing non-idempotent intersection types. In: 26th International Workshop/21st Annual Conference of the EACSL on Computer Science Logic, (CSL), Fontainebleau, France. LIPIcs, vol. 16, pp. 259–273. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2012). https://doi.org/10.4230/LIPIcs.CSL.2012.259
19. Faggian, C., Guerrieri, G., de'Liguoro, U., Treglia, R.: On reduction and normalization in the computational core. Math. Struct. Comput. Sci. **32**(7), 934–981 (2022). https://doi.org/10.1017/S0960129522000433
20. Faggian, C., Rocca, S.R.D.: Lambda calculus and probabilistic computation. In: 34th Annual ACM/IEEE Symposium on Logic in Computer Science, (LICS), Vancouver, BC, Canada, pp. 1–13. IEEE (2019). https://doi.org/10.1109/LICS.2019.8785699
21. Gavazzo, F., Vanoni, G., Treglia, R.: On monadic intersection types (2023, draft)
22. Girard, J.: Linear logic. Theor. Comput. Sci. **50**, 1–102 (1987). https://doi.org/10.1016/0304-3975(87)90045-4
23. Guerrieri, G.: Towards a semantic measure of the execution time in call-by-value lambda-calculus. In: 12th Workshop on Developments in Computational Models and Ninth Workshop on Intersection Types and Related Systems, (DCM/ITRS), Oxford, UK. EPTCS, vol. 293, pp. 57–72 (2018). https://doi.org/10.4204/EPTCS.293.5
24. Jones, S.L.P., Wadler, P.: Imperative functional programming. In: Conference Record of the Twentieth Annual (ACM) (SIGPLAN-SIGACT) Symposium on Principles of Programming Languages (POPL), Charleston, South Carolina, USA, pp. 71–84. ACM Press (1993). https://doi.org/10.1145/158511.158524
25. Kerinec, A., Manzonetto, G., Rocca, S.R.D.: Call-by-value, again! In: 6th International Conference on Formal Structures for Computation and Deduction (FSCD), Buenos Aires, Argentina (Virtual Conference). LIPIcs, vol. 195, pp. 7:1–7:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2021). https://doi.org/10.4230/LIPIcs.FSCD.2021.7
26. Kesner, D., Peyrot, L.: Solvability for generalized applications. In: 7th International Conference on Formal Structures for Computation and Deduction (FSCD), Haifa, Israel. LIPIcs, vol. 228, pp. 18:1–18:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2022). https://doi.org/10.4230/LIPIcs.FSCD.2022.18
27. Kesner, D., Viso, A.: Encoding tight typing in a unified framework. In: 30th EACSL Annual Conference on Computer Science Logic, (CSL), Göttingen, Germany (Virtual Conference). LIPIcs, vol. 216, pp. 27:1–27:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2022). https://doi.org/10.4230/LIPIcs.CSL.2022.27

28. Levy, P.B.: Call-by-push-value: a subsuming paradigm. In: Girard, J.-Y. (ed.) TLCA 1999. LNCS, vol. 1581, pp. 228–243. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48959-2_17
29. Manzonetto, G., Pagani, M., Rocca, S.R.D.: New semantical insights into call-by-value λ-calculus. Fund. Inform. **170**(1–3), 241–265 (2019). https://doi.org/10.3233/fi-2019-1862
30. Moggi, E.: Computational lambda-calculus and monads. In: 4th Annual Symposium on Logic in Computer Science, (LICS), Pacific Grove, California, USA, pp. 14–23. IEEE Computer Society (1989). https://doi.org/10.1109/LICS.1989.39155
31. Plotkin, G.D.: Call-by-name, call-by-value and the lambda-calculus. Theor. Comput. Sci. **1**(2), 125–159 (1975). https://doi.org/10.1016/0304-3975(75)90017-1
32. Sabry, A., Wadler, P.: A reflection on call-by-value. ACM Trans. Program. Lang. Syst. **19**(6), 916–941 (1997). https://doi.org/10.1145/267959.269968
33. Treglia, R.: The computational core: reduction theory and intersection type discipline. Ph.D. thesis, Università di Torino (2022)
34. Wadler, P.: Monads for functional programming. In: Jeuring, J., Meijer, E. (eds.) AFP 1995. LNCS, vol. 925, pp. 24–52. Springer, Heidelberg (1995). https://doi.org/10.1007/3-540-59451-5_2

Effective Skolemization

Matthias Baaz[1] and Anela Lolić[2(✉)]

[1] Institute of Discrete Mathematics and Geometry, TU Wien, Vienna, Austria
baaz@logic.at

[2] Kurt Gödel Society, Institute of Logic and Computation, TU Wien, Vienna, Austria
anela@logic.at

Abstract. We define a new relatively simple Skolemization method called atomic Skolemization which allows for a non-elementarily bounded speed-up of cut-free **LK**-proofs and resolution proofs w.r.t. the standard Skolemization and Andrews Skolemization.

Keywords: Skolemization · Cut-free Proofs · Resolution Proofs

1 Introduction

Skolem functions are one of the most important features of classical and related first-order logics. They represent quantifiers within the term language, similar to epsilon calculus. A Skolemization is a functional from a closed formula with distinct bound variables to a closed formula with distinct bound variables, which replaces some occurrences of bound variables by Skolem terms (terms of bound variables and new functions) such that all bound variables in the Skolem term belong to not replaced quantifiers where the term is in the scope.

For satisfiability of formulas, the main precondition of the introduction of Skolem functions is the preservation of soundness. For validity of formulas the dual main precondition is that the original formula is valid when the Skolemized formula is valid. In this contribution we work with Skolemization in the sense of satisfiability.

The standard Skolemization in the satisfiability case is based on the replacement of positive existential and negative universal quantifiers by Skolem functions depending on all negative existential and positive universal quantifiers where the replaced quantifier is in the scope.

Example 1. Consider the formula

$$\forall x(\exists y P(y) \vee \forall u \exists v(R(x,u) \vee Q(x,v))).$$

Then its Skolemization is

$$\forall x(P(f(x)) \vee \forall u(R(x,u) \vee Q(x,g(x,u))).$$

Supported by FWF projects I 5848, I 4427, and P 36571.

H. H. Hansen et al. (Eds.): WoLLIC 2023, LNCS 13923, pp. 69–82, 2023.
https://doi.org/10.1007/978-3-031-39784-4_5

The quantified variable y is replaced by $f(x)$, where f is a fresh Skolem function symbol, and the quantified variable v is replaced by $g(x, u)$, for the fresh Skolem function symbol g.

The standard Skolemization is sound because the addition of Skolem axioms

$$\forall \overline{x}(\exists y A(y, \overline{x})) \supset A(f(\overline{x}), \overline{x}) \quad \text{and} \quad \forall \overline{x}(A(f(\overline{x}), \overline{x}) \supset \forall y A(y, \overline{x}))$$

to a satisfiable set of sentences is conservative (it is possible to argue also directly replacing quantifiers within the formulas). The conservativity of Skolem axioms corresponds to the fact that in the case of validity Skolemized formulas are not weaker than the original ones. The introduction of Skolem formulas by projection of positive universal and negative existential quantifiers is always possible.[1]

Andrews Skolemization [2,3] is an optimized form of standard Skolemization, where positive existential and negative universal quantifiers are replaced by Skolem functions depending only on the negative existential or positive universal quantifiers which bind in the subformula that begins with the quantifier to be replaced.

Example 2. Consider the formula

$$\forall x(\exists y P(y) \vee \forall u \exists v(R(x, u) \vee Q(x, v)).$$

Then its Andrews Skolemization is

$$\forall x(P(c) \vee \forall u(R(x, u) \vee Q(x, g(x, u)).$$

Here, the quantified variable y is replaced by the Skolem constant c (as x does not occur in $P(y)$, and the quantified variable v is replaced by $g(x, u)$, as x and u occur in $R(x, u) \vee Q(x, v)$.

To refute a formula in theorem proving based on resolution refutation, the formula has first to be Skolemized, then transformed into its clause form, and finally refuted with the resolution method. It was shown that Andrews Skolemization allows for a non-elementarily[2] bounded speed-up of the resolution proofs with regard to standard Skolemization [8]. In this contribution we present a simple algorithm for a Skolemization method, which is more effective than Andrews Skolemization: There is a speed-up even over Andrews Skolemization.

[1] It is obvious that the validity of the argument for the conservativity of Skolem axioms is equivalent to the validity of the full axiom of choice. To demonstrate that valid Skolemized formulas can be retransferred to their original form needs at most the completeness of first-order logic, i.e. the validity of König's lemma, which is much weaker than the axiom of choice. This difference can be explained as follows: The argument for conservativity of Skolem axioms validitates automatically the Skolem functions as functions, i.e. their identity axioms $\overline{x} = \overline{y} \supset f(\overline{x}) = f(\overline{y})$. Such axioms are not automatically eliminated when resetting Skolemized formulas in the validity sense.

[2] A primitive recursive function $f(x)$ is elementary if it is bound by a fix stack of 2: $2^{2^{\cdot^{\cdot^{x}}}}$.

2 Standard Skolemization and Andrews Skolemization

In this section the standard Skolemization method and the Andrews Skolemization method are introduced and compared.

Definition 1 (standard Skolem form w.r.t. satisfiability). *Let A be a closed first-order formula. If A does not contain positive existential or negative universal quantifiers, we define its standard Skolemization as* $\mathrm{sk}(A) = A$.

Suppose now that A contains positive existential or negative universal quantifiers and (Qy) is the first positive existential or negative universal quantifier occurring in A. If (Qy) is not in the scope of negative existential or positive universal quantifiers, then its standard Skolemization is

$$\mathrm{sk}(A) = \mathrm{sk}(A \backslash (Qy)\{y \leftarrow c\}),$$

where $A\backslash(Qy)$ denotes the formula A after omission of (Qy) and c is a constant symbol not occurring in A. If (Qy) is in the scope of the negative existential or positive universal quantifiers $(Q_1x_1)\dots(Q_nx_n)$, then its standard Skolemization is

$$\mathrm{sk}(A) = sk(A \backslash (Qy)\{y \leftarrow f(x_1, \dots, x_n)\}),$$

where f is a function symbol (Skolem function) not occurring in A.

In Andrews' method the introduced Skolem functions do not depend on the positive existential or negative universal quantifiers $(Q_1x_1)\dots(Q_nx_n)$ dominating the positive universal or negative existential quantifier (Qx), but on the subset of $\{x_, \dots, x_n\}$ appearing (free) in the subformula dominated by (Qx). In general, this method leads to smaller Skolem terms.

Definition 2 (Andrews Skolem form w.r.t. satisfiability). *Let A be a closed first-order formula. If A does not contain positive existential or negative universal quantifiers, we define its Andrews Skolemization as* $\mathrm{sk}_A(A) = A$.

Suppose now that A contains positive existential or negative universal quantifiers, $(Qy)B$ is a subformula of A and (Qy) is the first positive existential or negative universal quantifier occurring in A (in a tree-like ordering). If $(Qy)B$ has no free variables which are quantified by a negative existential or positive universal quantifier, then its Andrews Skolemization is

$$\mathrm{sk}_A(A) = \mathrm{sk}_A(A \backslash (Qy)\{y \leftarrow c\}),$$

where $A\backslash(Qy)$ denotes the formula A after omission of (Qy) and c is a constant symbol not occurring in A. If $(Qy)B$ has n variables $x_1, \dots, x_n$ which are quantified by a negative existential or positive universal quantifier from outside, then its Andrews Skolemization is

$$\mathrm{sk}_A(A) = \mathrm{sk}_A(A \backslash (Qy)\{y \leftarrow f(x_1, \dots, x_n)\}),$$

where f is a function symbol not occurring in A.

Let $\Gamma \rightarrow \Delta$ be a sequent, and let $F = \bigwedge \Gamma \supset \bigvee \Delta$ and $\mathrm{sk}_A(F) = \bigwedge \Pi \supset \bigvee \Lambda$, then we define the Andrews Skolemization of the sequent $\Gamma \rightarrow \Delta$ as

$$\mathrm{sk}_A(\Gamma \rightarrow \Delta) = \Pi \rightarrow \Lambda.$$

The usual Skolemizations are outside-in. This uses the global knowledge which of the bound variables are bound by positive universal or negative existential quantifiers. If we define standard Skolemization locally (i.e. inside-out), the result is an iteration of the Skolem functions within the Skolem semi-terms[3].

Example 3. Consider the formula

$$\exists x \forall y \exists u \forall v A(x, y, u, v).$$

Following the standard Skolemization (outside-in) we obtain

$$\forall y \forall v A(c, y, f(y), v),$$

and following the standard Skolemization inside-out we obtain

$$\forall y \forall v A(c, y, g(c, y), v).$$

The Skolem functions in the Skolem semi-terms are ordered in occurrence. Let g and h be Skolem function symbols that occur in a Skolem semi-term as $h(\ldots g(\ldots)\ldots)$, then we say that $h < g$. The iteration of the Skolem terms poses no problem by the following proposition which allow their elimination. We call such Skolem terms normalized.

Proposition 1. *The formulas A and A' are equi-satisfiable, where A is obtained from A' by replacing different iterated Skolem semi-terms $h(\ldots g(\ldots)\ldots)$ by Skolem semi-terms $f_i(\ldots)$ with new function symbols.*

Proof. $\Rightarrow$: obvious.

$\Leftarrow$: A $<$-minimal Skolem semi-term $g(\ldots)$ corresponds directly to one $f_i(\ldots)$ w.r.t. satisfiability. In the iterated case the Skolem semi-term $h(\ldots g(\ldots)\ldots)$ w.r.t. satisfiability corresponds also directly to a $f_j(\ldots)$, as $g(\ldots)$ is already determined.

From now on we will denote with # the operator that normalizes Skolem semi-terms according to Proposition 1.

Theorem 1. *The Andrews Skolemization preserves soundness.*

Proof. Proposition 1 allows us to argue locally, i.e. to replace positive existential or negative universal quantifiers inside-out. Assume the innermost still existing such quantifier is existential (analogously for the case of an universal quantifier). Then

[3] Semi-terms are terms that might contain bound variables.

$$A(\ldots \exists x B(x,\overline{y}) \ldots) \text{ is satisfiable, where the occurrence of } \exists x B(x,\overline{y}) \text{ is positive}$$
$$\Downarrow$$
$$A(\ldots F(\overline{y}) \ldots) \wedge \forall \overline{y}(F(\overline{y}) \supset \exists x B(x,\overline{y})) \wedge \forall \overline{y}(\exists x B(x,\overline{y}) \supset F(\overline{y})) \text{ is satisfiable}$$
$$\Downarrow$$
$$A(\ldots F(\overline{y}) \ldots) \wedge \forall \overline{y}(F(\overline{y}) \supset B(f(\overline{y}),\overline{y})) \wedge \forall \overline{y}(B(f(\overline{y}),\overline{y}) \supset F(\overline{y})) \text{ is satisfiable}$$
by standard Skolemization with f and instantiation
$$\Downarrow$$
$$A(\ldots B(f(\overline{y}),\overline{y}) \ldots) \text{ is satisfiable}$$
$$\Downarrow$$
$$A(\ldots \exists x B(x,\overline{y}) \ldots) \text{ is satisfiable.}$$

Theorem 2 ([8]). *There is a sequence of refutable formulas $A_1, A_2, \ldots$ such that the length of the shortest resolution refutations of their standard clause forms*[4] *with standard Skolemization cannot be elementarily bounded in the length of the shortest resolution refutations of their standard clause forms with Andrews Skolemization.*

Proof (Sketch). The validity variant of standard Skolemization, i.e. the replacement of positive universal and negative existential quantifiers by Skolem terms corresponds exponentially in the length of cut-free proofs to usual sequent calculus **LK**, whereas Andrews Skolemization corresponds exponentially in the length of cut-free proofs to sequent calculus $\mathbf{LK}^+$ [1]. $\mathbf{LK}^+$ is obtained from **LK** by weakening the eigenvariable condition. The resulting calculus is therefore globally but possibly not locally sound. This means that all derived statements are true but that not every sub-derivation is meaningful. $\mathbf{LK}^+$-proofs are based on the side variable relation $<_{\varphi,\mathbf{LK}}$. We say b is a side variable of a in φ (written $a <_{\varphi,\mathbf{LK}} b$) if φ contains a positive universal or negative existential quantifier inference of the form

$$\frac{\Gamma \rightarrow \Delta, A(a,b,\overline{c})}{\Gamma \rightarrow \Delta, \forall x A(x,b,\overline{c})} \,\forall_r$$

or of the form

$$\frac{A(a,b,\overline{c}), \Gamma \rightarrow \Delta}{\exists x A(x,b,\overline{c}), \Gamma \rightarrow \Delta} \,\exists_l$$

Proofs are determined by $\mathbf{LK}^+$-suitable quantifier inferences. We say a quantifier inference is suitable for a proof φ if either it is a positive existential or negative universal quantifier inference, or the following three conditions are satisfied:

- (substitutability) the eigenvariable does not appear in the conclusion of φ.
- (side variable condition) the relation $<_{\varphi,\mathbf{LK}}$ is acyclic.
- (weak regularity) the eigenvariable of an inference is not the eigenvariable of another positive universal or negative existential quantifier inference in φ.

[4] See Definition 7.

LK$^+$ is obtained from **LK** by replacing the usual eigenvariable conditions by **LK**$^+$-suitable ones. **LK**$^+$ admits cut-elimination and there is a non-elementary speed-up of cut-free **LK**$^+$-proofs w.r.t. cut-free **LK**-proofs.

The following proposition is obvious.

Proposition 2. *Standard Skolemization and Andrews Skolemization coincide on prenex formulas.*

3 Atomic Skolemization

For simplicity we define the new algorithm for satisfiability and closed formulas with distinct bound variables in negation normal form (NNF). Therefore, existential quantifiers are replaced by Skolem terms.

Similar to Andrews Skolemization, atomic Skolemization is based on the elimination of the innermost quantifiers, i.e. generating iterated Skolem semi-terms in principle. This situation can be stratified using Proposition 1.

Definition 3. *Let F be a closed NNF formula with distinct bound variables. Then $<_F$ is a total order of the bound variables occurring in F, such that whenever Qx occurs in the scope of $Q'y$, we have that $x <_F y$, where $Q, Q' \in \{\forall, \exists\}$ and x, y are bound variables in F.*

Note that we might omit the subscript F in $<_F$ whenever it is clear from the context. For simplicity reasons in the Skolemization procedure, we will introduce the notion of *corresponding quantifier* of a bound variable.

Definition 4. *Let F be a closed NNF formula with distinct bound variables. Let x be such a bound variable. Then its corresponding quantifier is denoted by $\Psi(x)$, i.e.*

$$\Psi(x) = \begin{cases} \exists & if\ x\ is\ bound\ by\ \exists, \\ \forall & if\ x\ is\ bound\ by\ \forall. \end{cases}$$

The atomic Skolemization of a closed NNF formula F with distinct bound variables is computed based on the set of atomic semi-formulas occurring in F and containing the bound variables, and on the substitutions of Skolem semi-terms for these bound variables. We first give a description of the procedure, and then a formal definition of the algorithm for atomic Skolemization.

In a first step we consider all the atoms of the formula F and construct a set of sets of bound variables by collecting all the bound variables occurring in each of the atoms, which are not empty (this set will later be denoted with L_n). The substitution is initialized with the identity substitution. As long as L_n is not empty, we pick the $<_F$-minimal bound variable x and the corresponding sets in L_n containing x. Note that these sets might contain also other variables, which we denote by $\overline{y}$. In case the corresponding quantifier of x is existential, i.e. $\Psi(x) = \exists$, we delete all sets $\{x, \overline{y}_i\}$ from L_n and add $\{\overline{y}\}$ to the remaining variables. Furthermore, we add $\{x \leftarrow f(\overline{y})\}$, where f is a new function symbol to the set of substitutions. Alternatively, in case $\Psi(x) = \forall$, the sets $\{x, \overline{y}_i\}$ are again deleted from L_n and we add a set $\{\overline{y}\}$ to the remaining variables, but the set of substitutions is not updated. ($\{\overline{y}\}$ is only added when it is maximal under inclusion and the initial L_0 is stratified in this respect.) Finally, the iterated Skolem terms are replaced by uniterated ones according to Proposition 1.

Definition 5. *Let F be a closed NNF formula with distinct bound variables $V(F)$. Then its atomic Skolemization* AS(F) *is computed by the following steps:*

1. $L_0 = \{\{\gamma_1, \ldots, \gamma_n\} \mid \{\gamma_1, \ldots, \gamma_n\} \in V(F)(and \neq \emptyset)\, which\, occur\, jointly\, in\, an\, atom\, of\, F\}$.
2. $\sigma_0 = id(\sigma_n\, will\, substitute\, Skolem\, semi-terms\, for\, bound\, variables)$.
3. $L_n = L_n \backslash \{\gamma_1, \ldots, \gamma_n\}$ *if* $\{\gamma_1, \ldots, \gamma_n\}$ *is not maximal in* L_n *w.r.t. inclusion.*
4. *while* $L_n \neq \emptyset$
   6. *Let x be the $<_F$-minimal variable in L_n and*
      $\Delta_{n+1} = \{\{\gamma_1, \ldots, \gamma_n\} \mid \{\gamma_1, \ldots, \gamma_n\} in\, L_n\, containing\ x\}$.
      Let $x, \overline{y}$ all the variables in Δ_{n+1}.
   7. *If $\Psi(x) = \exists$:*
      $L_{n+1} = L_n \backslash \Delta_n \cup \{\overline{y}\}$ *if* $\{\overline{y}\}$ *is maximal in* $L_n \backslash \Delta_n$, $L_n \backslash \Delta_n$ *otherwise,*
      $\sigma_{n+1} = \sigma_n \cup \{x \leftarrow f(\overline{y})\}$, *where f a new function symbol.*
   8. *If $\Psi(x) = \forall$:*
      $L_{n+1} = L_n \backslash \Delta_n \cup \{\overline{y}\}$ *if* $\{\overline{y}\}$ *is maximal in* $L_n \backslash \Delta_n$, $L_n \backslash \Delta_n$ *otherwise.*
9. $L_n = \emptyset \Rightarrow \sigma = \sigma_n$.
10. *Let F' be F after deletion of $\exists$. Then* AS$(F) = \#F'\sigma$.

Note that this algorithm is at most quadratic in the number of symbols of the original formula. However, its verification will need exponentially many steps.

Example 4. Let F be the formula

$$\forall x(\exists y P(y) \vee \forall u \exists v(R(x, u) \vee Q(x, v)).$$

We calculate its atomic Skolemization AS(F). To start, we initialize the set $L_0 = \{\{y\}, \{x, u\}, \{x, v\}\}$, with the ordering $v <_F u <_F y <_F x$.
As $\Psi(v) = \exists$ we obtain

$$L_1 = \{L_0 \backslash \{x, v\}\} \cup \{x\}, \quad \sigma_1 = \sigma_0 \cup \{v \leftarrow h(x)\}.$$

A $<_F$-minimal variable is now u. Then, as $\Psi(u) = \forall$, we obtain

$$L_2 = \{L_1 \backslash \{x, u\}\}, \quad \sigma_2 = \sigma_1$$

as $\{x\}$ is already in L_1. Now y is $<_F$-minimal. As $\Psi(y) = \exists$ we obtain in a next step

$$L_3 = L_2 \backslash \{y\}, \quad \sigma_3 = \sigma_2 \cup \{y \leftarrow c\}.$$

In a last step, as $\Psi(x) = \forall$, we obtain

$$L_4 = L_3 \backslash \{x\} = L_3 \backslash L_3 = \emptyset, \quad \sigma_4 = \sigma_3$$

F' is F after deletion of all occurrences of $\exists$, and $F'\sigma_4$ is

$$\forall x(P(c) \lor \forall u(R(x,u) \lor Q(x,h(x))))$$

which is also $\#F'\sigma = \mathrm{AS}(F)$ as no iterated Skolem terms occur.

Proposition 3. *Skolem functions can be combined over disjunctions. Let $\overline{x}_i \in \overline{x}$*

$$\forall \overline{x} \bigvee_i A_i(f_i(\overline{x}_i)) \supset \forall \overline{x} \bigvee_i A_i(f(\overline{x}))$$

is satisfiable, where f is a new function symbol.

Theorem 3 (Soundness of atomic Skolemization).

Proof. Consider step 3. in the AS-algorithm given in Definition 5. We have $L_n \neq 0$ and x the $<_F$-minimal variable.

$$\Delta_{n+1} = \{\{\gamma_1, \ldots, \gamma_n\} \mid \{\gamma_1, \ldots, \gamma_n\} \text{ in } L_n \text{ containing } x\},$$

$x, \overline{y}$ all the bound variables in Δ_n. Let $\exists x A(x, \overline{y})$ be the corresponding subformula.

$$\models \forall \overline{y} \forall \overline{z} (\exists x A(x, \overline{y}) \leftrightarrow \exists x \overbrace{\bigvee_i (\bigwedge_j B_{i,j}(x, \overline{y}_{i,j})) \land C_i(\overline{y}, \overline{z})}^{(\times)}),$$

where $\overline{y}_i = \cup_j(\overline{y}_{i,j})$, $(\times)$ is a suitable CNF where the $B_{i,j}$ atomic contain x and the C_i atomic do not.

$$\models \forall \overline{y} \forall \overline{z} (\exists x (\times) \leftrightarrow \overbrace{\bigvee_i (\exists x \bigwedge_j B_{i,j}(x, \overline{y}_{i,j})) \land C_i(\overline{y}, \overline{z})}^{(\times\times)}), \quad \overline{y}_{i,j} \subseteq \overline{y}$$

$$\models \forall \overline{y} \forall \overline{z} ((\times\times) \to \overbrace{\bigvee_i \bigwedge_j B_{i,j}(f_i(\overline{y}), \overline{y}_{i,j})) \land C_i(\overline{y}, \overline{z})}^{(\times\times\times)})$$

by Andrews Skolemization

$$\models \forall \overline{x} \forall \overline{z} ((\times\times\times) \to \overbrace{\bigvee_i (\bigwedge_j B_{i,j}(f(\overline{y}), \overline{y})) \land C_i(\overline{y}, \overline{z})}^{(\times\times\times\times)})$$

by Proposition 3

$$\models \forall \overline{x} \forall \overline{z} ((\times\times\times\times) \to \exists x \overbrace{\bigvee_i \bigwedge_j B_{i,j}(x, \overline{y}) \land C_i(\overline{y}, \overline{z})}^{(\times)})$$

Now let $\forall x A(x, \overline{y})$ be the corresponding subformula.

$$\models \forall\overline{y}\forall\overline{z}(\forall x A(x,\overline{y}) \leftrightarrow \forall x(\overbrace{\bigwedge_i(\bigvee_j B_{i,j}(x,\overline{y}_{i,j}) \wedge C_i(\overline{y},\overline{z}))}^{(\circ)}),$$

where $\overline{y}_i = \cup_j(\overline{y}_{i,j})$, $(\circ)$ is a suitable CNF where the $B_{i,j}$ contain x and the $C_{i,j}$ do not.

$$\models \forall\overline{y}\forall\overline{z}(\forall x(\circ) \leftrightarrow \bigwedge_i(\forall x \bigvee_j B_{i,j}(x,\overline{y}_{i,j})) \wedge C_i(\overline{y},\overline{z}))).$$

Now introduce new predicates F_i and add suitable

$$\forall\overline{y}(F(\overline{y}_{i,j}) \leftrightarrow \forall x \bigvee_j B_{i,j}(x,\overline{y}_{i,j}))$$

and continue to work with the formula after replacement. Semi-subformulas containing x disappear from the main formula. The consideration to work with $\overline{y}$ instead of the subsets $\overline{y}_i$ might lead to larger dependencies, but not incorrect ones as all relevant variables are contained in $\overline{y}$.

As an application we obtain:

Corollary 1. *The monadic fragment of classical first-order logic is decidable.*

Proof. For a monadic function-free formula A, AS(A) contains only constants as Skolem functions, and therefore it is decidable whether a Herbrand expansion for AS(A) exists.

Proposition 4. *The arity of the Skolem function symbols w.r.t. atomic Skolemization is less or equal to the arity of the Skolem function symbols w.r.t. Andrews Skolemization which is less or equal to the arity of Skolem function symbols in standard Skolemization. The number of introduced Skolem function symbols is not increased.*

4 Speed-Up Result for Cut-Free Proofs

In this section we demonstrate that there is a non-elementary speed-up for cut-free proofs of atomic Skolemization w.r.t. standard Skolemization and Andrews Skolemization. Let $\tau = \{Q\overline{x}A(\overline{x}) \vee Q^D\overline{x}A(\overline{x})$ closed $\mid$ Q quantifier string, Q^D dual quantifier sequence, A atomic$\}$. Our argument is based on the following theorem.

Theorem 4. *There is a sequence of sequents*

$$A_1 \rightarrow, A_2 \rightarrow, \ldots, A_i \rightarrow,$$

where $A_1, \ldots, A_i$ are in NNF containing universal quantifiers only such that

1. *there is a bound for a sequence of cut-free* **LK***-proofs for*

$$\Delta_1, A_1 \rightarrow, \Delta_2, A_2 \rightarrow, \dots$$

elementary in the complexity of $A_1 \rightarrow, A_2 \rightarrow, \dots$ *for suitable* $\Delta_i \subseteq \tau$.

2. *there is no elementary bound for any sequence of cut-free proofs for*

$$A_1 \rightarrow, A_2 \rightarrow, \dots$$

in the complexity of $A_1 \rightarrow, A_2 \rightarrow, \dots, A_i \rightarrow$.

Proof. Consider Statman's sequence of provable quantifier-free statements following from universal formulas where the cut-free proofs grow non-elementarily versus the proofs with cuts, which are elementarily bounded [7,10]. Cuts can be closed by inferring $A \supset A$ on the left side instead of the cut, closing $A \supset A$ with universal quantifiers and cutting it. Replace all cuts by prenex cuts in an elementary way [6]. Code the matrices of the cuts by using coding formulas

$$\forall x(F(\overline{x}) \leftrightarrow M(\overline{x}))$$

added to the antecedents and replace the cuts:

$$\dfrac{\begin{matrix} \Pi_i \rightarrow \Gamma_i, M(\overline{s}_i) \\ \vdots\ \vdots\ \vdots \\ \Pi \rightarrow \Gamma, Q\overline{x}M(\overline{x}) \end{matrix} \qquad \begin{matrix} M(\overline{s}_i), \Lambda_j \rightarrow \Delta_j \\ \vdots\ \vdots\ \vdots \\ Q\overline{x}M(\overline{x}), \Lambda \rightarrow \Delta \end{matrix}}{\Pi, \Lambda \rightarrow \Gamma, \Delta}$$

$$\Downarrow$$

$$\dfrac{\Pi_i \rightarrow \Gamma_i, M(\overline{s}_i) \qquad F(\overline{s}_i) \rightarrow F(\overline{s}_i)}{M(\overline{s}_i) \supset F(\overline{s}_i), \Pi_i \rightarrow \Gamma_i, F(\overline{s}_i)} \qquad \dfrac{F(\overline{s}_i) \rightarrow F(\overline{s}_i) \qquad M(\overline{s}_i), \Lambda_j \rightarrow \Delta_j}{F(\overline{s}_i) \supset M(\overline{s}_i), \Lambda_j \rightarrow \Delta_j, F(\overline{s}_i)}$$

Apply $\wedge : l$ and $\forall : l$ to infer the equivalence $\forall x(F_i(\overline{x}) \leftrightarrow M_i(\overline{x}))$.

$$\Downarrow$$

$$\dfrac{\Pi \rightarrow \Gamma, Q\overline{x}F(\overline{x}) \qquad Q\overline{x}F(\overline{x}), \Lambda \rightarrow \Delta}{\forall x(F(\overline{x}) \leftrightarrow M(\overline{x})), \Pi, \Lambda \rightarrow \Gamma, \Delta}$$

These codings do not shorten the cut-free proofs much, as they can be immediately eliminated by replacing F by M and eliminating $\forall \overline{x}(M(\overline{x}) \leftrightarrow M(\overline{x}))$ by universal cuts whose elimination is at most double exponential. By an easy transformation we obtain cut-free proofs by adding $Q\overline{x}F(\overline{x}) \vee Q^D\overline{x}\neg F(\overline{x})$.

Note that for standard, Andrews, and atomic Skolemization it holds that the Skolemization of A w.r.t. satisfiability corresponds to the Skolemization of $A \rightarrow$ w.r.t. validity.

Definition 6. *$H(A)$, where $A \in \tau$ ($A = Q\overline{x}A(x) \vee Q^D\overline{x}A(\overline{x})$) is the prenexification of A such that $\forall$ always stands in front of the dual $\exists$, and $H(\Delta)$, where $\Delta \subseteq \tau$, is $\{H(A) \mid A \in \Delta\}$.*

Example 5. $H(\exists x \forall y B(x,y) \vee \forall u \exists v \neg B(u,v)) = \forall u \exists x \forall y \exists v (B(x,y) \vee \neg B(u,v))$.

Theorem 5. *There is a sequence of formulas $B_1, B_2 \ldots$ such that*

1. *there is a bound for a sequence of cut-free proofs for*

$$\mathrm{AS}(B_1) \rightarrow, \mathrm{AS}(B_2) \rightarrow, \ldots$$

   elementary in the complexity of $B_1, B_2 \ldots$.
2. *there is no elementary bound for any sequence of cut-free proofs for*

$$sk(B_1) \rightarrow, sk(B_2) \rightarrow, \ldots$$

   in the complexity of $B_1, B_2 \ldots$.
3. *there is no elementary bound for any sequence of cut-free proofs for*

$$sk_A(B_1) \rightarrow, sk_A(B_2) \rightarrow, \ldots$$

   in the complexity of $B_1, B_2 \ldots$.

Proof. By Proposition 2 standard Skolemization and Andrews Skolemization coincide for prenex formulas. Therefore, we argue only for standard Skolemization. Let $B_i = \bigwedge_{H(\Delta_i) \wedge A_i}$ from Theorem 4 (note that B_i is in NNF). Assume that there is an elementary bound for the cut-free proofs of

$$sk(B_1) \rightarrow,\ sk(B_2) \rightarrow,\ \ldots.$$

Therefore, there is an elementary bound for cut-free proofs of

$$sk(C_1^1), \ldots sk(C_n^1), sk(A_1') \rightarrow,\ sk(C_1^2), \ldots sk(C_n^2), sk(A_2') \rightarrow,\ \ldots,$$

where Δ_i is $C_1^i, \ldots C_n^i$ and A_i' is obtained from A_i by shifting the universal quantifiers outside. By [5] there is an elementary bound for the corresponding Herbrand sequent. Note that the Skolem terms always depend on the dual position, w.l.o.g.

$$D(\ldots t_j \ldots) \vee \neg D(\ldots f_i(\ldots t_j \ldots) \ldots).$$

Now replace all occurrences of $f_i(\ldots t_j \ldots)$ inside-out by t_j. As the Herbrand expansion is propositionally valid, and the term is replaced on all positions by the same term, the result remains valid. Finally all Skolem terms disappear, and the original Skolemized formulas in $H(\Delta)$ are transformed into formulas of the form $E_i \vee \neg E_i$, which do not influence the validity of the remaining sequent. Hence, the size of the remaining sequents is elementarily bounded and therefore the cut-free proofs are elementarily bounded. Contradiction to Theorem 4.

Now consider

$$\mathrm{AS}(B_1), \mathrm{AS}(B_2), \ldots.$$

Note that the bound variables in $Q\overline{x}A(\overline{x})$ and $Q^D\overline{x}A(\overline{x})$ in $Q\overline{x}A(\overline{x}) \vee Q^D\overline{x}A(\overline{x}) \in \Delta_i$ are distinct, which remains invariant w.r.t. any prenexation. Therefore, the atomic Skolemization of

$$H(Q\overline{x}A(\overline{x}) \vee Q^D\overline{x}A(\overline{x}))$$

is the standard Skolemization of $Q\overline{x}A(\overline{x}) \vee Q^D\overline{x}A(\overline{x})$. Deskolemization of cut-free proofs is exponential [4], therefore the cut-free proofs of

$$\mathrm{AS}(B_1) \rightarrow, \mathrm{AS}(B_2) \rightarrow, \ldots$$

are elementarily bounded.

5 Cut-Free LK-Proofs With Positive Existential/Negative Universal Quantifiers and Resolution

As we are interested in this paper mainly in the impact of different forms of Skolemization we allow any elementary form of clause form constructions (for the purpose of this paper it is not necessary to specify the exact form of resolution proofs, as they simulate each other within elementary bounds in the complexity of the proofs). This leads to a non-elementary speed-up of resolution proofs presupposing atomic Skolemization w.r.t. resolution proofs presupposing standard Skolemization or Andrews Skolemization.

Definition 7. *Let A be a formula which contains only positive existential or negative universal quantifiers when written on the left side of the sequent sign and therefore only positive universal or negative existential quantifiers when written on the right side of the sequent sign. An admissible clause form construction consists of sequents $A \rightarrow C$ and $C \rightarrow A$ elementary in the complexity of A, where*

1. *C (the clause form) is a conjunction of universally quantified disjunctions of literals (negated or unnegated atomic formulas),*
2. *$A \rightarrow C$ and $C \rightarrow A$ are cut-free elementary derivable in the complexity A.*

Note that both, structural clause forms and standard clause forms fall under this definition, together with clause forms which allow for atom evaluation etc. [9].

Theorem 6.

1. *Let φ be a cut-free* **LK**-*proof of the sequent*

$$A_1, \ldots, A_n \rightarrow B_1, \ldots, B_m$$

with positive existential or negative universal quantifiers only. Then there is a resolution refutation of an admissible clause form of

$$A_1 \wedge \ldots \wedge A_n \wedge \neg B_1 \wedge \ldots \wedge \neg B_m$$

elementary in the complexity of φ.

2. *Let* φ' *be a resolution refutation of an admissible clause form of*

$$A_1 \wedge \ldots \wedge A_n \wedge \neg B_1 \wedge \ldots \wedge \neg B_m.$$

Then there is a cut-free **LK***-proof of*

$$A_1, \ldots, A_n \rightarrow B_1, \ldots, B_m$$

with positive existential or negative universal quantifiers only elementary in the complexity of φ'*.*

Proof. See [8,9].

The next theorem follows directly from the theorem above.

Theorem 7. *There is a sequence of formulas* $B_1, B_2 \ldots$ *such that*

1. *there is a bound for a sequence of resolution refutation of standard clause forms of*

$$\mathrm{AS}(B_1) \rightarrow,\ \mathrm{AS}(B_2) \rightarrow,\ \ldots$$

elementary in the complexity of $B_1, B_2 \ldots$*.*

2. *there is no elementary bound for any sequence of resolution refutations of standard clause forms of*

$$sk(B_1) \rightarrow,\ sk(B_2) \rightarrow,\ \ldots$$

in the complexity of $B_1,\ B_2,\ \ldots$*.*

3. *there is no elementary bound for any sequence of resolution refutations of standard clause forms of*

$$sk_A(B_1) \rightarrow,\ sk_A(B_2) \rightarrow,\ \ldots$$

in the complexity of $B_1, B_2, \ldots$*.*

6 Conclusion

The worst case sequences constructed in this paper are highly artificial. It might be asked if they have an impact in the real world. It is however a known fact that worst case examples with extreme complexities correspond to practical examples which are not that bad, but bad enough.

References

1. Aguilera, J.P., Baaz, M.: Unsound inferences make proofs shorter. J. Symb. Log. **84**(1), 102–122 (2019)
2. Andrews, P.B.: Resolution in type theory. J. Symb. Log. **36**(3), 414–432 (1971)
3. Andrews, P.B.: Theorem proving via general matings. J. ACM **28**(2), 193–214 (1981)

4. Baaz, M., Hetzl, S., Weller, D.: On the complexity of proof deskolemization. J. Symb. Log. **77**(2), 669–686 (2012)
5. Baaz, M., Leitsch, A.: On skolemization and proof complexity. Fundam. Informaticae **20**(4), 353–379 (1994)
6. Baaz, M., Leitsch, A.: Cut normal forms and proof complexity. Ann. Pure Appl. Log. **97**(1–3), 127–177 (1999)
7. Baaz, M., Leitsch, A.: Methods of Cut-Elimination, vol. 34. Springer, Dordrecht (2011). https://doi.org/10.1007/978-94-007-0320-9
8. Baaz, M., Lolic, A.: Andrews skolemization may shorten resolution proofs non-elementarily. In: Artemov, S., Nerode, A. (eds.) LFCS 2022. LNCS, vol. 13137, pp. 9–24. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-93100-1_2
9. Eder, E.: Relative Complexities of First Order Calculi. Springer, Wiesbaden (2013). https://doi.org/10.1007/978-3-322-84222-0
10. Statman, R.: Lower bounds on Herbrand's theorem. In: Proceedings of the American Mathematical Society, pp. 104–107 (1979)

Factive Complements are Not Always Unique Entities: A Case Study with Bangla *remember*

Arka Banerjee(✉)

Jadavpur University, Kolkata, India
banerjeesoumyo29@gmail.com

Abstract. There are many approaches regarding the emergence of factivity in literature. Some of them who are proponents of the view that factive inferences are exported from complements, attribute it to the definiteness feature of the complements [28,29,35]. This definiteness feature can be realized covertly via a semantically-sensitive definite determiner $\varDelta$ [35], or via an overt marker (e.g., *ge* in Washo) [28]. Although [14] later revised their claim by calling this *ge* a marker of familiarity, not that of definiteness, they did not provide any evidence where the D in factive nominalized complements is not definite. This paper provides evidence from Bangla (/Bengali; an Indo-Aryan language) where an attitude verb *mone pora* 'remember' can embed nominalized complements that can be interpreted indefinitely but still remains factive. In this paper, we provide a formal compositional analysis that can account for this.

Keywords: Attitude verbs · Factive complements · Definiteness · Compositionality · Definedness condition · Familiarity · Bangla

1 Setting the Stage

A statement $P\varphi$ is called a factive attitude report if the proposition φ is presupposed to be true [34,37]. Instantiating from natural language, verbs like *regret*, *resent*, *know*, *remember*, etc. presuppose (») the truth of their complements. See the following:

(1) John knows that Bill passed the test. » Bill passed the test.

(2) John regrets that he misbehaved with Sue. » he misbehaved with Sue.

Both the sentences are factive reports because the verbs *know* and *regret* presuppose the truth of their complement clauses. There are three standpoints regarding the emergence of factiviy in literature. Some associate this with verbs [33,56] and some with complements ([35,37,40] a.m.o). The third group denies either of these options and describes it as a compositional offspring [17]. Those who envisage that factivity is exported from complements often attribute it to the definiteness feature of the complements [28,29,35]. We argue that this type of

H. H. Hansen et al. (Eds.): WoLLIC 2023, LNCS 13923, pp. 83–100, 2023.
https://doi.org/10.1007/978-3-031-39784-4_6

linking is not so obvious across the board (cf. [16,17]). In this paper, we provide evidence from Bangla (alternatively, Bengali) in which an attitude verb *mone pora* 'remember' can embed nominalized complements that are not obligatorily interpreted in a definite way, but it still remains factive (cf. [13]).[1] Consider the following:

(3) Context: *Mary visited Delhi three times.*

John-er	[Mary-r	Delhi ja-wa]	mone	pore.
John-GEN	Mary-GEN	Delhi go-GER	in memory	fall.PRS.3

'John remembers Mary visiting Delhi.'

In (3), the Bangla counterpart of *remember* embeds a nominalized complement or a gerund, *viz. Mary-r Delhi ja-wa* 'Mary's visiting Delhi'. Here the attitude report can pick out any one of the three visiting events, not necessarily any particular event of her visiting Delhi. Hence, by intuition, one can argue that the nominalized complement can feasibly refer to an indefinite event here. In order to establish it in a more concrete way, we conform to [17]'s insight which can tell us about the lack of its obligatory definiteness in the following way:

(4) John-er [Mary-r Delhi ja-wa] mone pore, Bill-er
John-GEN Mary-GEN Delhi go-GER in memory fall.PRS.3 Bill-GEN
[Mary-r Delhi ja-wa] mone pore, Sam-er [Mary-r
Mary-GEN Delhi go-GER in memory fall.PRS.3 Sam-GEN Mary-GEN
Delhi ja-wa] mone pore.
Delhi go-GER in memory fall.PRS.3
Context 1: *Mary visited Delhi three times.*
✓John, Bill, and Sam remember different events of Mary visiting Delhi.
Context 2: *Mary visited Delhi once.*
✓John, Bill, and Sam remember the same event of Mary visiting Delhi.

As noted, the first context points us to the fact that the gerundial complement is referring to different events of Mary visiting Delhi, whereas the second one refers to a single event. Hence, no obligatory sense of definiteness can be attached to the gerundial complement in this case.

Now, the task is to show that truth of the content of this complement is presupposed, *i.e.,* the attitude report is factive. Since presuppositions are non-defeasible, the following *but*-clause which contradicts the content of the complement sounds pragmatically weird (marked with the # symbol) after (3):

(5) kintu, Mary konodino Delhi ja-e ni.
but Mary ever Delhi go-3 PST.PRF.NEG
'But, Mary did not visit Delhi ever.' [# after (3)]

[1] To give an answer to one of the anonymous reviewers, we mention that not only this one verb but there are other verbs in Bangla like *mone ach-* 'have in memory', *mone rakha* 'keep in memory', *bhule jawa* 'forget' that behave alike. In this paper, we will restrict ourselves to zooming in on the case of *mone pora* only. We would like to keep open the possibility of the semantics of these other verbs being different from it.

The presupposed status of the nominalized complement can be shown if we negate the sentence in (3) because presuppositions survive negation. The negation of (3) still entails (⊩) that Mary visited Delhi.

(6) John-er [Mary-r Delhi ja-wa] mone pore na.
John-GEN Mary-GEN Delhi go-GER in memory fall.PRS.3 NEG
'John does not remember Mary visiting Delhi.' ⊩ Mary visited Delhi.

Alternatively, one can execute the 'Hey! wait a minute' test [26] to check the presupposition projection. In a conversational setting, the following can be a good response to (3):

(7) ei! ek minute dnara, ami jantam na je Mary Delhi gechilo.
Hey! one minute wait I know.1 NEG that Mary delhi go.PRF.PST.3
'Hey! wait a minute, I did not know that Mary had visited Delhi.' [✓in response to (3)]

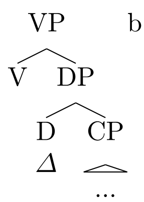
(7) sounds perfectly okay as a response to (3) because one can be ignorant about something which is already a fact. Therefore, it is quite established that the nominalized complement in (3) is presupposed to be true but does not need to be read in a definite way always. Hence, it challenges the view that assimilates factivity into definiteness of the complement [28,29,35]. In this paper, we account for this phenomenon in a compositional way at the syntax-semantic interface.

The next section discusses the approaches that relate factivity to the definiteness feature of complements. Section 3 explores if the verb in concern can be seen as lexically factive and contends that it cannot be so. Section 4 sheds light on how to view this verb and discusses its internal structure. Section 5 deals with how factive inferences can be compositionally inferred in the case of an indefinite nominalized complement. Lastly, Sect. 6 concludes the paper with a note on future work.

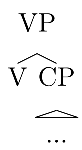
2 Existing Approaches Relating Factivity to Definiteness

That definite nominalization is liable for the rise of factive inferences is propagated in [35]. This is supported by the work of [28] on Washo language – in their work, it is shown that definiteness is the core feature in giving rise to factivity. [35]'s standpoint results in the following syntactic representations:

(8) a. Presuppositional:



b. Non-presuppositional:



[35] classifies clauses into two classes, *i.e.*, PRESUPPOSITIONAL and NON- PRESUPPOSITIONAL rooting back to what [20] pioneered about STANCE VERBS. The following is the famous classification of stance verbs [20, 30, 31]:

a. NON-STANCE (factive): *know*, *remember*, *realize*, *notice*, *regret*, etc.
b. RESPONSE STANCE: *accept*, *deny*, *agree*, *admit*, *verify*, *confirm*, etc.
c. VOLUNTEERED STANCE (non-factive): *think*, *believe*, *suppose*, *claim*, *suspect*, *assume*, etc.

[35] groups the first two clusters into the PRESUPPOSITIONAL class since they presuppose the existence of their complements, while the VOLUNTEERED STANCE class refers to the non-factives because of being non-presuppositional in nature. Though the former two classes are presuppositional, truth is guaranteed in the case of non-stance predicates only. Let us consider the following:

(9) John regrets that he studied linguistics.

(10) John denied that he studied linguistics.

In the former example, it is presupposed that John studied linguistics, and the truth of it is certified. Thus, *regret* is a NON-STANCE or factive. But, in (10) the complement clause is not verified to be true even if it exists beforehand in the COMMON GROUND (CG) [61]. If it did not exist in the context before, the question of denying it would not come to the scenario. So both in non-stance and response stance predicates, the existence of presupposed complements in the CG is noted, but in the non-stance class, the truth of them is guaranteed additionally. The following captures the notion:

a. Non-stance: Existence of presupposed complement p in CG + The truth of p
b. Response stance: Existence of presupposed complement p in CG
c. Factives $\subset$ Presuppositional verbs

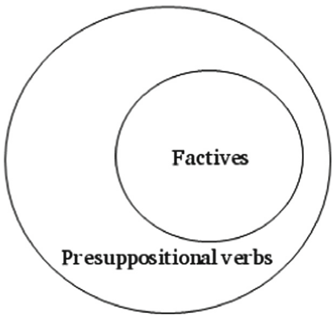


Fig. 1. Factives and presuppositional verbs

As evident from Fig. 1, the set of factives is a proper subset of presuppositional verbs. That means all factives are presuppositional verbs, but not vice-versa.

By contrast, volunteered stance verbs do not select for any complement which already exists in the CG. At this point, [35] proposes that presuppositional verbs pick up definite DPs from the CG, while non-presuppositional ones simply opt for CPs. And, as [35] propounds, the D head of definite DPs in English is occupied by a covert Δ which invokes the definiteness. This Δ in turn takes the clause as its complement. Follow the structure in (8a) where presuppositional verbs select for a semantically-sensitive definite D, *viz.* Δ. He assimilates factivity into the definiteness of nominalized complements. [28], following [55], mention that this D slot is filled with definite *-gi/ge* morpheme in clausal nominalizations in Washo. In their recent work, [14] revised their standpoint advancing that this *-gi/-ge* morpheme stands for mere familiarity under *idx* head in Washo, but not for definiteness, and mentioned that familiarity alone cannot explain factivity. However, they did not provide any evidence showing us an indefinite use of nominalized complements embedded under factive predicates.

This paper has discussed such a case in Bangla where we can find indefinite use of eventualities embedded under a factive report. Not only in Bangla but this kind of observation is also noted in Barguzin Buryat (a Mongolic language) by [17]. We will account for this phenomenon in Bangla in a compositional manner in this paper. Prior to getting into that, we need to address why the verb *mone pora* 'remember' cannot be claimed factive lexically. Let us look at this in the following section.

3 Is Bangla *remember* Lexically Factive?

At this point, the reader might ask why we do not ascribe factivity lexically to *mone pora* 'remember'. Technically, why don't we formulate the following semantics of it relative to a world w and a variable assignment function g, where it is presupposed that the $< s, t >$-type propositional argument is presupposed to be true in w?

(11) $[\![\text{mone pora}]\!]^{w,g} = \lambda p_{<s,t>} \lambda x_e : \underline{p(w) = 1}.\texttt{remember}_w(p)(x)$

(11) denotes a partial function – this concerned verb is said to be defined if its argument holds true in the actual world, otherwise undefined. However, in (12) we are getting a hallucination context with *mone pora*, which is purely non-factive in nature.

(12) Context: Eight-year-old Rahul is remembering some stuff that did not happen ever. His father gets tensed and visits a doctor. The following conversation is under such a circumstance.

Father:	Doctor,	Rahul-er	majhe majhe	[amra	US	gechilam	bol-e]
	Doctor	Rahul-GEN	at times	we	US	go.PRF.PST.1	say-PTCP
	mone	pore,	kintu	amra	kokhono	US	ja-i
	mind.LOC	fall.PRS.3	but	we	ever	US.LOC	go-1
	ni.						
	PRF.PST.NEG						

'Doctor, Rahul at times hallucinates/imagines that we went to the US, but, we never went to the US.'

Doctor: In fact, Rahul is suffering from false memory syndrome.

In this above example, the verb *mone pora* is embedding a finite clause whose propositional content is not true in the actual world. Hence, if the verb would have been factive *per se* and carried a semantics as in (11), it would presuppose that the proposition – 'we went to the US' – is true in the actual world, but which is certainly not the case in actual reality, as seen in (12). Thus, factivity in (3) cannot be exported from the verb itself. So, how should the logical translation of this concerned Bangla verb be? We will deal with this issue in the next section.

4 How to View Bangla *remember*

Drawing reference from the previous section, one could argue for having two different avatars of *mone pora* – one is factive, and the other is non-factive. But, viewing it as lexically ambiguous would be less economical for the lexicon than proposing a single semantics that accounts for both readings. In other words, a single semantics of *mone pora* which can take care of both factive and non-factive readings will undoubtedly increase the delicacy of our formal system.

In the above data examples, the interlinear glosses reflect that Bangla *remember* is a complex predicate[2] where the preverb is *mone* 'in memory/mind' and the light verb is *pora* 'to fall' (see [18,19], a.m.o.). The preverb mostly provides the semantic content of the complex predicate [47] and the light verb adds some extra colors to it. This attitude verb in concern is a composite that literally means 'to fall in memory'. Another interesting fact is that the subject of this verb is in the Genitive case instead of the regular Nominative one.[3] Follow the *-r* marker on the attitude subject, which is the morphological realization of the Genitive case in Bangla. This type of construction draws our attention to some diachronic processes that Bangla has undergone. Genitive subject constructions of the verbs or predicates denoting mental activities and psychological states have a long history. [38] mentioned that subjects of these predicates in Middle Bangla used to occur in Genitive, Locative, and Objective cases. As mentioned in [51], the most frequent pattern among them was:

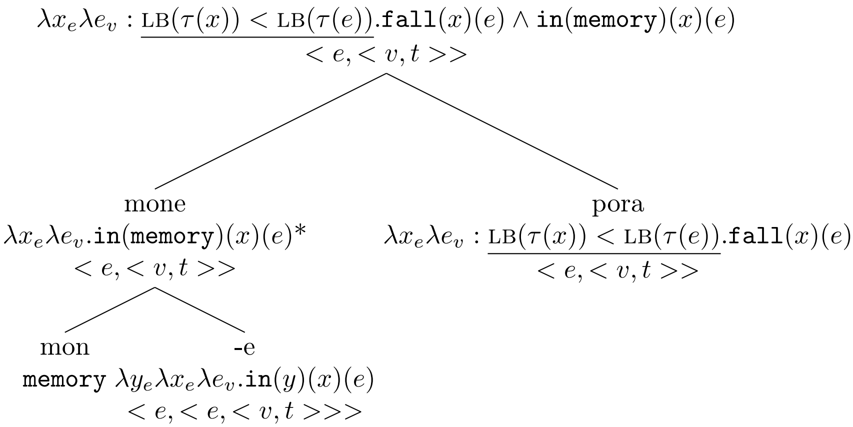
Genitive NP + body part (L) + sensation/feeling (NOM)+
be/become/happen

[2] According to [19], a complex predicate consists of a main predicational element (noun, verb, or adjective) and a light verb that is usually the syntactic head of the construction. Complex predicates are composed of more than one grammatical element, each of which contributes part of the information ordinarily associated with a head [2]. As [3] echoes [46], they exhibit word-like properties in terms of argument structure composition and sometimes in having lexicalized meanings.

[3] cf. [21] who called it an Indirect Case that is not too far removed from the Dative subjects. He mentioned that it is morphologically a Genitive, but has features that are Dative-like.

[51] also mentioned that the Genitive NP originally referred to the inalienable possessor of the body part. Eventually, the Experiencer/Patient status of the possessor was focused on and it got the subject status. In Modern Bangla, predicates like *mon-**e** pora* still retain the Locative NP referring to a body part, where *-e* denotes the Locative marker and the NP that denotes the body part is *mon* or 'memory/mind'. Thus, in present-day Bangla, the original possessor of the memory has faded away and it acquired the status of a subject who is experiencing the mental state. In other words, the apparent Genitive NP is no more the possessor now, rather it is the subject of the mental predicate. Now, getting back to the literal form of *mon-e pora*, we propose the following compositional structure of it where there is no possessor of `memory`:[4]

(13)



*It is a short for '$\lambda x_e \lambda e_v.\exists y[\mathtt{memory}(y) \wedge \mathtt{in}(y)(x)(e)]$'. We actually introduced a type shifter having the form '$\lambda R_{<e,<e,<v,t>>>} \lambda P_{<e,t>} \lambda x_e \lambda e_v. \exists y.P(y) \wedge R(y)(x)(e)$' which shifts the type of the Locative marker *-e* to the type $<< e, t >, < e, < v, t >>>$, so that the Locative *-e* can combine with the $< e, t >$-type one-place predicate `memory`.

In (13), the semantics of *fall* encodes a definedness condition which says that the left boundary (LB) of the interval denoting the life-span of the object of falling precedes ($<$) that of the interval referring to the running time of the falling event e. This is the PRE-EXISTENCE PRESUPPOSITION [15,16] associated with the internal argument of *fall*. In a nutshell, the object of falling must pre-exist the starting point of the falling event. In the definedness condition, τ denotes the TEMPORAL TRACE FUNCTION [43–45]. Such restrictions on the arguments of verbs are noted by [24]. See the following:

(14) John broke the glass. → The glass was there before the event of breaking.

[4] Thanks to one of the reviewers for questioning the logic behind not introducing the possessor as one of the arguments of `memory`. It is due to the historical reason which tells us how the possessor status of the body part had lost its focus diachronically. That is why we treated `memory` as a one-place predicate, but not a two-place one.

(15) John wrote an essay. $\not\rightarrow$ The essay existed before the event of writing.

In the same way we can show that the object of *pora* 'fall' exists before the start of the falling event and hence the pre-existence restriction gets associated with its object or theme (cf. [7,8]). Consider the following:

(16) gach theke apel-ta porlo, #kintu gach-e kono apel chilo na.
tree from apple-CLF fall.PST.3 but tree-LOC any apple was NEG
'The apple fell from the tree, #but there was no apple in the tree.'
⇒ The apple existed before the falling event started.

In (13), we defined the locative suffix -*e* as a transitive predicate that takes two arguments y and x and returns us the set of eventualities e such that x is e-ing in y. Now, in order to compose *mone*, of type $< e, < v, t >>$-type, with the $< e, < v, t >>$-type *pora*, we resort to the Generalized Conjunction [54] rule which is stated below:

(17) **Generalized Conjunction:**
Pointwise definition of $\sqcap$ [54]
$X \sqcap Y =$
a. $= X \wedge Y$ if both X and Y are truth values
b. $= \{< z, x \sqcap y >: < z, x > \in X$ and $< z, y > \in Y\}$ if X and Y are functions

Via this composition, the event argument of `in` gets identified with the event of falling (cf. [39]). Hence, the root node in (13) refers to a function-valued function that takes an individual x and an event argument e. It is defined if x pre-exists e, if defined then it returns 1 iff e is the event of falling whose object is x and x is falling in memory.

We argue that this composite gets lexicalized with the meaning of remembering or recalling over time. Intriguingly, this phenomenon is not specific to Bangla. It can be noted cross-linguistically in many related and unrelated languages. To convey the sense of remembering, languages like Assamese and Odia (both are Indo-Aryan) have the verbal forms *monot pelua* and *mone pokila*, respectively, which literally mean 'falling in memory' just like Bangla. As noted by [17], a Balkar language that is family-wise very much distant from Bangla lexicalizes *remember* as 'dropping in memory'. Now, once the complex form in (13) gets lexicalized with the meaning of remembering, it can accommodate another argument that acts as the subject of the concerned event. Recall that the possessor of the memory (*i.e.*, the body part) lost its Possessor status and evolved as an Experiencer historically, occurring as the external argument of *remember* and bearing the quirky Case[5]. The presence of this quirky Genitive Case on the subject is reminiscent of the fact that once it used to carry the status of a Possessor of the

[5] Quirky Case is something which is linked to the theta grid of a particular predicate. A Genitive/Indirect experiencer subject is directly linked to the theta grid of the verb *mone pora*.

body part. Additionally, we argue, Bangla *remember* retains the pre-existence presupposition which comes from the light verb *fall* in its interpretation (cf. [7]). Consider the following:

(18) $[\![\text{mone pora}]\!]^{w,g} = \lambda x \in D_e \cup D_v.\lambda z \in D_e.\lambda e \in D_v : \underline{\text{LB}(\tau(x)) < \text{LB}(\tau(e))}.$
$\texttt{remember}_w(x)(z)(e)$

The transition from (13) to (18) should not be understood synchronically, rather this transition covers a huge time period between Middle and Modern Bengali. Thus, it is a long historical process that is at play behind this type of transition.[6] In (18), we followed a Davidsonian representation [22] in viewing the verbal semantics where an event variable is introduced along with all its arguments. (18) tells us that it takes two arguments x and z and an event argument e, and is defined if x pre-exists e. If defined then it returns true iff e is the event of remembering and z is remembering x. An interesting thing to note about (18) is that the internal argument of *mone pora* can be picked out either from the domain of individuals or from the domain of eventualities. That means this verb can take either an entity or an event as its argument. If we take gerunds as events (see Sect. 5), then (3) is an example of this attitude verb taking eventualities. However, apart from the eventualities, it can take e-type entities too, both contentful and non-contentful.[7] See the following:

(19) Contentful DP

amar	golpo-ta	mone	pore.
I.GEN	story-CLF	mind.LOC	fall.PRS.3

'I remember the story.'

(20) Non-contentful DP

amar	John-ke	mone	pore.
I.GEN	John-ACC	mind.LOC	fall.PRS.3

'I remember John.'

In the former example, the object of *mone pora* is some particular story that refers to propositional content. However, in the latter one, we get a proper name as the theme or object, which is purely non-contentful in nature. The way we defined the nature of the internal argument of *mone pora* in (18) can feasibly take care of (3) along with (19, 20).

Now, we are set with everything before we get into how the factive interpretation is inferred while it embeds a nominalized complement that is indefinite in nature. The following section deals with it.

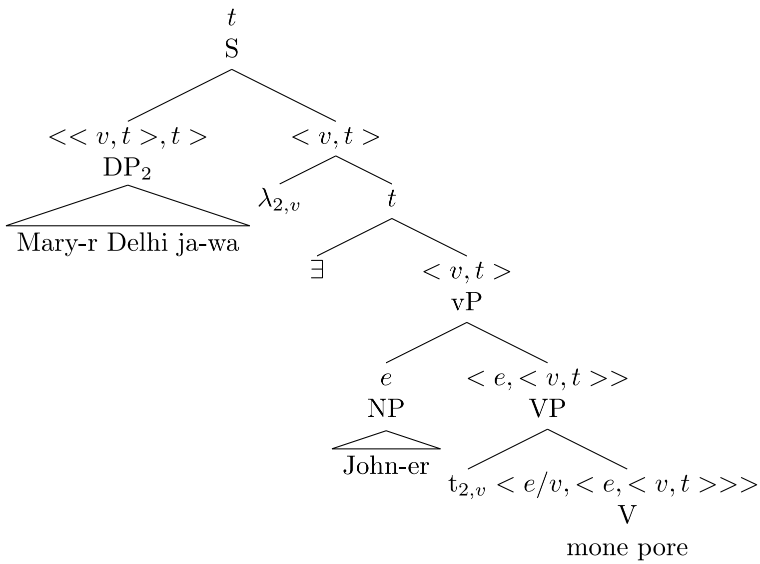
6 Thanks to one of the anonymous reviewers for the suggestion to account for the leap from the semantics in (13) to that in (18).

7 Contentful entities are those which are associated with propositional contents [48]. For example, entities like *news*, *story*, etc. are contentful. On the other hand, entities like proper names are non-contentful because they are not associated with any sort of propositional element.

5 Accounting for the Factive Reading with an Indefinite Nominalized Complement

From the data above in (3), we postulate that Bangla POSS-ing gerunds[8] can be indefinite unlike English ones that are, as per [57,59], definite. We propose the following LF (logical form) of (3):

(21)



We assume that gerunds denote sets of eventualities [58,59]. Thus, the POSS-ing complement in (3) will have the interpretation as in (22), relative to a world w and an assignment function g.

(22) $[\![\text{DP}]\!]^{w,g} = \lambda e_v.\texttt{visiting}_w(\texttt{Delhi})(\texttt{Mary})(e)$

It denotes the set of v-type events such that they are events of Mary visiting Delhi. Since the concerned POSS-ing is interpreted indefinitely in (3), we can tap into [53]'s type shifter **A** that maps a predicate onto a quantifier.[9] Thus, applying it on the POSS-ing DP would yield the following translation:

(23) $\mathbf{A}([\![\text{DP}]\!]^{w,g}) = \lambda Q_{<v,t>}.\exists e'[\texttt{visiting}_w(\texttt{Delhi})(\texttt{Mary})(e') \wedge Q(e')]$
[via Functional Application (FA)]

Consequently, a type-mismatch happens while composing it with the attitude verb which looks for an argument of type e or v. See the interpretation in (18).

8 [1] discussed four types of gerunds in English – POSS-ing (*e.g.* John's visiting NY), ACC-ing (*e.g.* John visiting NY), PRO-ing (*e.g.* visiting NY), and Ing-*of* (*e.g.* visiting of John).

9 [53] originally defined this type shifter over the domain of entities of type e. But, if we extend [53]'s **A** to the domain of eventualities (D_v), nothing plays as a hindrance for us. This **A** will then have the semantics as follows: $\lambda P_{<v,t>}\lambda Q_{<v,t>}.\exists e'[P(e') \wedge Q(e')]$.

In order to avoid this type-mismatch, we perform a covert Quantifier Raising (QR) movement, due to which the DP moves to a higher position in the tree leaving a v-type trace t_2 and creating a λ-binder that binds the trace. The compositional steps are the following:

a. $[\![\text{VP}]\!]^{w,g} = \lambda z_e.\lambda e_v : \underline{\text{LB}(\tau(g(2))) < \text{LB}(\tau(e))}.\texttt{remember}_w(g(2))(z)(e)$ [via FA, V & $t_{2,v}$]

b. $[\![\text{NP}]\!]^{w,g} = \texttt{John}$

c. $[\![\text{vP}]\!]^{w,g} = \lambda e_v : \underline{\text{LB}(\tau(g(2))) < \text{LB}(\tau(e))}.\texttt{remember}_w(g(2))(\texttt{John})(e)$ [via FA, NP & VP]

d. $[\![\exists]\!] = \lambda R_{<v,t>}.\exists e.R(e)$ (existential closure over events)

e. $[\![\exists + \text{vP}]\!]^{w,g} = \exists e : \underline{\text{LB}(\tau(g(2))) < \text{LB}(\tau(e))}.\texttt{remember}_w(g(2))(\texttt{John})(e)$

f. $[\![\lambda_{2,v}\text{+(e.)}]\!]^{w,g} = \lambda u_v.\exists e : \underline{\text{LB}(\tau(u)) < \text{LB}(\tau(e))}.\texttt{remember}_w(u)(\texttt{John})(e)$ [via Predicate Abstraction]

g. $[\![\text{S}]\!]^{w,g} = \exists e' \exists e : \underline{\text{LB}(\tau(e')) < \text{LB}(\tau(e))}.\texttt{remember}_w(e')(\texttt{John})(e) \wedge \texttt{visiting}_w(e')(\texttt{Delhi})(\texttt{Mary})(e')$ [via FA, (f.) & (23)]

Thus, at the topmost node S we get the reading that there already exists an event of Mary visiting Delhi before John remembers it. In other words, there is a pre-existing event of Mary visiting Delhi and this event is the object of John's remembering. Hence, a factive reading comes to the fore.

6 Summary and Future Work

Overall, in this paper, we show that factivity is not a subject to be exported from the definiteness or uniqueness of the complements. It is only familiarity, not uniqueness, which is linked to the factive nominalized complements in this case. However, unlike Washo, this familiarity is not morphologically encoded in Bangla nominalizations, rather it is derived compositionally through the definedness condition associated with the concerned attitude verb, which says that its internal argument or theme/object pre-exists the main attitude event. One anonymous reviewer mentioned that [35]'s familiarity can be equated with [16]'s PRE-EXISTENCE PRESUPPOSITION in that both of them make references to old discourse referents. We completely agree with this intuition, however, we argue that the basic difference between these two approaches lies in the presence or absence of definiteness. The advantage of embracing the pre-existence presupposition is that it allows us to get rid of the obligatory definiteness condition linked to the factive complements.

Apart from gerundial complements, there appears another clausal complementation pattern where *mone pora* gives rise to factivity: when it embeds a finite *je*-clause (see [9, 11], a.m.o.) and bears the main sentential stress (denoted by the capital letters in the following), it gives rise to factive inferences [6]. See the following:

(24) Rahul-er MONE PORE je Mary Delhi giyechilo.
Rahul-GEN mind.LOC fall.PRS.3 that Mary Delhi go.PRF.PST.3
'Rahul remembers that Mary went to Delhi.' » Mary went to Delhi.

It is also experimentally reported in [6] that if the main stress docks on the matrix subject instead of the matrix verb, the attitude report does not anymore entail the truth of the complement clause. We leave this puzzle for future work.

Acknowledgements. We convey our thanks to all the native Bangla speakers who gave their data judgments. We also extend our thanks to Ankana Saha, Diti Bhadra, Kousani Banerjee, Nirnimesh Bhattacharjee, Sadhwi Srinivas, Srabasti Dey, Tatiana Bondarenko, Ushasi Banerjee, and Utpal Lahiri for their valuable insights on various issues. All errors are mine.

Appendix 1

An accompanying question might arise regarding the source of factivity – can factivity be built into nominalization? The answer would be - 'no'. See the following example in (25) where the contradictory *but*-conjunct is compatible with the preceding clause. Hence, no factive inference is drawn. This observation converges with other languages too, *e.g.* Turkish [52], Buryat [17], and so on.

(25) John [Bill-er bhot-e jet-a] asha korechilo, ✓kintu,
John Bill-GEN election-LOC win-GER hope do.PRF.PST.3 but
durbhagyoboshoto Bill konodino bhot-e je-te ni.
unfortunately Bill ever election-LOC win-3 PRF.PST.NEG
'John hoped for Bill winning elections, ✓but unfortunately he did not ever win any.'

Now, there can be questions about the compositional path in (25) – should we take the path of argumenthood here as well? That means, should we take the nominalized complement to compose as the internal argument of *hope*? The answer would be - 'no!'. If it would have been the path of argumenthood, we would end up having a veridical[10] report which is certainly not the case in (25). We argue that the complement denotes the **content of hope, but not the object of it**.[11] Content of hope might be false in actual reality. Along this line of intuition, we assert that the complement here is not a DP, but rather some eventive projection, ϵP in disguise where the ϵ head takes the gerundial DP as its argument. The semantics of ϵ is given below:

(26) $[\![\epsilon]\!]^{w,g} = \lambda P_{<v,t>} \lambda e_v.\text{CONT}_w(e) = \lambda w'.\exists e'.P(e')$ in w'

The CONT is a function that takes entities that have intensional content. For example, entities like *story*, *gossip*, etc. are contentful as mentioned in Sect. 4.

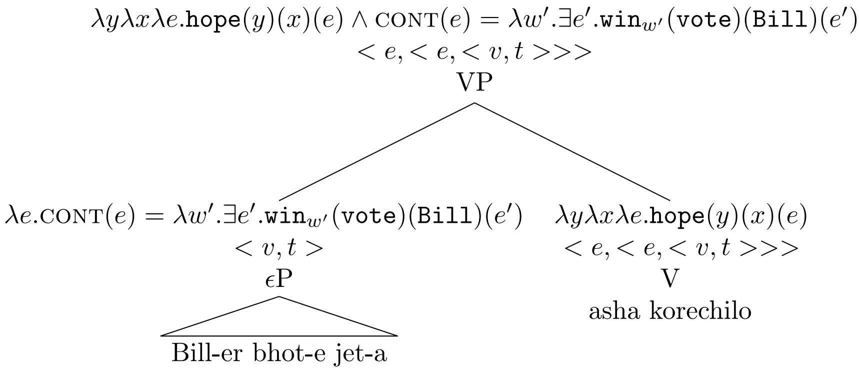
[10] A statement $P\varphi$ is a veridical report if the truth of φ is entailed, *e.g.* verbs like *prove* are veridical predicates.

[11] Thanks to Tatiana Bondarenko for a discussion on this issue. Thanks are also due to Ankana Saha, Diti Bhadra, Kousani Banerjee, Nirnimesh Bhattacharjee, Sadhwi Srinivas, and Ushasi Banerjee for their insights. It is noteworthy that [17] reported a Case-shift phenomenon in the case of Buryat where nominalized complements of *hope*, *believe* are Dative marked, not Accusative marked (see [17]). However, Bangla does not show us any such Case-changing phenomenon morphologically.

Events can also be contentful though [25,48,49] (*e.g. belief*, *saying event* etc.). But, the event of running is not contentful at all. For any element a, $\text{CONT}(a) = \{w : w \text{ is compatible with the intensional content determined by } a \text{ in } w\}$ [42].

When the ϵ head gets composed with the nominalized DP by FA, it will yield the ϵP projection which is a function of type $< v, t >$. This would not compose with the verb via its internal argument. Instead, it only modifies the eventuality argument of the matrix verb *hope* whose content will then be denoted by the proposition that Bill would win the election/vote. See the following composition:

(27)



We used the rule Modified Predicate Modification [15] for the composition. The rule is stated below:

(28) **Modified Predicate Modification:** [15]
If α is a branching node and $\{\beta, \gamma\}$ is the set of its daughters, then, for any assignment g and world w, α is in the domain of $[\![\ ]\!]^{w,g}$ if both β and γ are, and if $[\![\beta]\!]^{w,g}$ is a predicate $P\beta$ of type $< \sigma_1, < \sigma_2, ... < \sigma_k, ... < \sigma_n, t >>>>$ and $[\![\gamma]\!]^{w,g}$ is a predicate $P\gamma$ of type $< \sigma_k, t >$. In this case, $[\![\alpha]\!]^{w,g} = \lambda x_1 \lambda x_2 ... \lambda x_k ... \lambda x_n : x_1 ... x_n$ are in the domain of $[\![\beta]\!]^{w,g}$ and x_k is also in the domain of $[\![\gamma]\!]^{w,g}.P\beta(x_1)(x_2)...(x_k)...(x_n)$ & $P\gamma(x_k) = 1$.

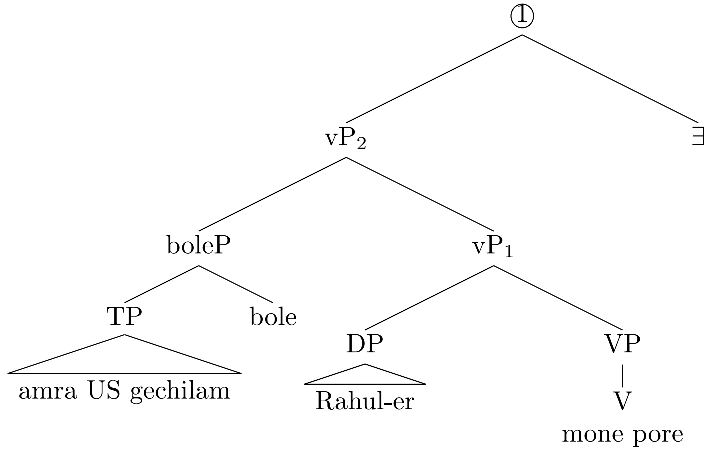
(28) "allows a modifier of a type $< \sigma_k, t >$ to modify any σ_k-type variable of a predicate." Following this, we arrive at the root note in (27) which shows us that the content of *hope* becomes the proposition 'Bill would win the election'. Therefore, the truth of it will not be guaranteed because the content of an attitude event might be false in the actual scenario.

Appendix 2

The reviewers have suggested addressing the question of how the semantics in (18) can account for the non-factive reading in (12). Earlier, we argued that the semantics in (11) fails to account for any non-factive reading because the semantics as stated in (11) would require the embedded proposition to be always true in the actual world. However, we will show that the proposed semantics in (18) can do so. The embedded clause involved in (12) is a finite clause with the VERBY EMBEDDER *bole* which is a SAY-based complementizer [9–12,50,60].

This kind of embedded clause is not even the complement to the verb, rather it sits outside of the main clause and adjoins to the vP domain [4,5,23,36]. Thus, the structure will be like this:

(29)



We assume that Bangla *bole* complementizer is an overt realization of the covert reportative modal ⟦SAY⟧ which can denote mental states and is built on contentful eventualities, but not individuals [41,50].[12] Not only Bangla *bole*, there exist SAY-based complementizers in other languages too, e.g., Korean *ko*, Japanese *to*, Zulu *ukuthi*, etc. that are also built on contentful eventualities [48]. Following [50], the semantics of *bole* is the following where it takes a propositional argument p and returns the set of contentful eventualities whose intensional content is denoted by p:

(30) $[\![\text{bole}]\!]^{w,g} = \lambda p_{<s,t>} \lambda e_v.\text{CONT}_w(e) = p$

(31) $[\![\text{boleP}]\!]^{w,g} = \lambda e_v.\text{CONT}_w(e) = \lambda w'.\text{we went to US in } w'$ [via Intensional FA, $[\![\text{TP}]\!]^g_¢$ & 'bole']

Now, what is important to note is the type of the *bole*-clause, which is $< v, t >$. And, it neither modifies nor saturates the internal argument of the verb. What does it do then? It combines with vP_1 via Predicate Conjunction, by modifying the matrix event only. Below we write down the semantic computations:

(32) $[\![\text{vP}_1]\!]^{w,g} = \lambda e_v \exists x : \underline{\text{LB}(\tau(x)) < \text{LB}(\tau(e))}.\texttt{remember}_w(x)(\texttt{Rahul})(e)$[13]

(33) $[\![\text{vP}_2]\!]^{w,g} = \lambda e_v \exists x : \underline{\text{LB}(\tau(x)) < \text{LB}(\tau(e))}.\texttt{remember}_w(x)(\texttt{Rahul})(e) \wedge \text{CONT}_w(e) = \lambda w'.\text{we went to US in } w'$

Now, another existential closure will be executed to close off the matrix event argument. Though the pre-existence presupposition is present here, we do not

[12] English complementizer *that* is built on contentful individuals ([48], a.m.o.).

[13] In spite of the fact that there is no theme argument of V in (29), we do not want to leave this slot unsaturated or open. That is why we proceed by existentially closing the internal argument of the verb so that it can compose it with its subject by FA.

find any lexical correlate of x. Thus, it should not bother us. The important thing is – we have the subordinate proposition as the content (but not the object) of *remember*, which might be false in the actual world. This is the crux of getting non-factivity in (12). One of the reviewers also pointed to the non-factive readings with English *remember* too. This is a very interesting point that [48] has already discussed. [48] mentioned that we get examples like *Martha remembered John to be bald, but he wasn't* where *remember* is used in a non-factive manner. Here, he proposed a null embedder F_{Dox} which acts like *bole/ko/to*.

Technical Notes

i. In this paper, we used the tools of formal semantics such as lambda calculus, restricted lambda for introducing definedness conditions, and compositional rules like Functional Application, Predicate Modification, Predicate Abstraction, Trace rule, etc. Readers are requested to follow [32] for all these.
ii. We considered the following *types*:
   a. e for entities/individuals
   b. t for truth values
   c. v for events
   d. s for the worlds

   We viewed propositions as functions from worlds to truth values. In other words, a proposition is a set of those worlds where it holds true. Readers are advised to follow [27].
iii. For the reference of the readers, a full picture of [53]'s type shifters is given below:
iv. For interlinear glossing of the non-English data, we followed the Leipzig convention for glossing: https://www.eva.mpg.de/lingua/pdf/Glossing-Rules.pdf (Fig. 2).

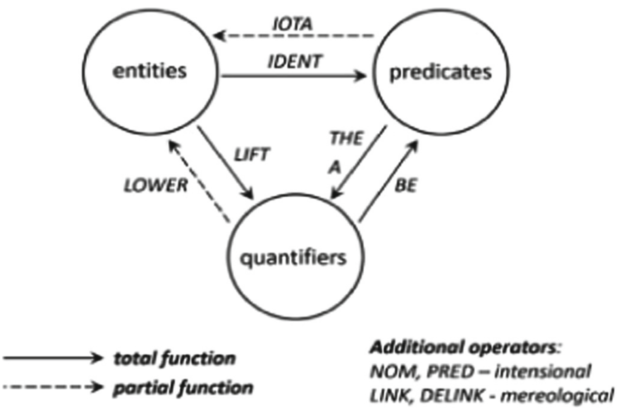


Fig. 2. Partee's type shifters (taken from [62])

References

1. Abney, S.P.: The English Noun Phrase in its Sentential Aspect. Ph.D. dissertation, MIT, Cambridge MA (1987)
2. Alsina, A., Bresnan, J., Sells, P.: Complex predicates: structure and theory. In: Alsina, A., Bresnan, J., Sells, P. (eds.) Complex predicates, pp. 1–12. Center for the Study of Language and Information, Stanford (1997)
3. Balusu, R.: Complex predicates in Telugu: a computational perspective. In: Proceedings of the 24th Conference on Computational Linguistics (COLING 2012): Demonstration papers, pp. 1–8 (2012)
4. Balusu, R.: The quotative complementizer says "I'm too Baroque for that". In: Dash, B., Kaur, G. (eds.) Proceedings of 8th Formal Approaches to South Asian Languages (FASAL 8), vol. 3, pp. 1–12 (2020)
5. Banerjee, A.: Response-stance predicates with two types of finite clauses in Bangla. University of Pennsylvania Working Papers in Linguistics: Vol. 29: Iss. 1, Article 2 (2023)
6. Banerjee, A., Banerjee, A., Sanyal, S., Das, S.K.: Role of speech rhythms in shaping meaning inferences: an experimental study in Bangla. Paper presented at The International Conference cum Workshop on Rhythm of Happiness (ICRH 2021) held at IIT Kharagpur, December 2021
7. Banerjee, A., Karmakar, S.: How light verbs shed light on attitude building. In: Proceedings of 22nd Seoul International Conference on Generative Grammar (SICOGG), p. 22 (2020)
8. Banerjee, A., Karmakar, S., Ghosh, S.: Factivity and nominalization: a study on 'mind-predicates' in Bangla. Paper presented at 8th Indian Conference on Logic and its Applications (ICLA 8) (2019)
9. Bayer, J.: Directionality and Logical Form: On the Scope of Focusing Particles and Wh-in-situ. Springer, Dordrecht (1996). https://doi.org/10.1007/978-94-017-1272-9
10. Bayer, J.: Final complementizers in hybrid languages. J. Linguist. **35**(2), 233–271 (1999)
11. Bayer, J.: Two Grammars in One: Sentential Complements and Complementizers in Bengali and Other South Asian Languages. In: Bhaskarorao, P., Subbarao, K.V. (eds.) The Yearbook of South Asian Languages: Tokyo Symposium on South Asian Languages-Contact, Convergence and Typology, pp. 11–36. Sage Publications, New Delhi (2001)
12. Bayer, J., Schmid, T., Bader, M.: Clause union and clausal position. In: Dikken, D.M., Tortora, C.M. (eds.) The Function of Function Words and Functional Categories, pp. 79–113. John Benjamins, Amsterdam (2005)
13. Bhadra, D., Banerjee, A.: Obligation, prohibition, non-finite complementation. Talk presented at 53rd North East Linguistic Society (NELS 53), University of Göttingen, January 2023
14. Bochnak, R., Hanink, E.: Clausal embedding in Washo: complementation vs. modification. Nat. Lang. Linguist. Theory **40**, 979–1022 (2022)
15. Bondarenko, T.I.: Factivity alternation due to semantic composition: think and remember in Barguzin Buryat (2019. ms, MIT
16. Bondarenko, T.I.: From think to remember: how CPs and NPs combine with attitudes in Buryat. In: Proceedings of 29th Semantics and Linguistic Theory (SALT 29), pp. 509–528 (2019)
17. Bondarenko, T.I.: Factivity from pre-existence: evidence from Barguzin Buryat. Glossa **5**(1), 109. 1–35 (2020)

18. Butt, M.: The light verb jungle (2003). paper presented at Workshop on Multi-Verb Constructions. Trondheim
19. Butt, M.: Complex Predicate Compendium. Handout from talk given in Tromsø, May 2005. http://ling.uni-konstanz.de/pages/home/butt/main/papers/cp-hnd.pdf
20. Cattell, R.: On the source of interrogative adverbs. Language **54**, 61–77 (1978)
21. Dasgupta, P.: Some non-nominative subjects in Bangla. In: Bhaskararao, P., Subbarao, K.V. (eds.) Non-nominative Subjects, vol. 1, pp. 129–140. John Benjamins Publishing Co., Amsterdam/Philadelphia (2004)
22. Davidson, D.: The Logical Form of Action Sentences. In: Resher, N. (ed.) The Logic of Decision and Action, pp. 81–95. University of Pittsburgh Press, Pittsburgh (1967)
23. Dey, S.: Bangla equative clauses and finiteness. Poster presented at 31st Conference of the Student Organization of Linguistics in Europe (2023)
24. Diesing, M.: Indefinites: Linguistic Inquiry Monographs. MIT Press, Cambridge, MA (1992)
25. Elliott, P.D.: Elements of clausal embedding. Ph.D. dissertation, University College London (2017)
26. von Fintel, K.: Would you believe it? The King of France is back! Presuppositions and truth-value intuitions. In: Reimer, M., Bezuidenhout, A. (eds.) Descriptions and Beyond. Oxford University Press, Oxford (2004)
27. von Fintel, K., Heim, I.: Intensional semantics. Lecture notes, MIT Spring (2011)
28. Hanink, E., Bochnak, R.: Factivity and two types of embedded clauses in Washo. In: Lamont, A., Tetzloff, K. (eds.) Proceedings of 47th North-East Linguistic Society (NELS 47), pp. 65–78 (2017)
29. Hanink, E., Bochnak, R.: Selection and complement clause size in Washo. Paper presented at the Selection Fest Berlin 2017 workshop at the Leibniz-Zentruem für Allgemeine Sprachenwissenschaft (2017)
30. Hegarty, M.: On adjunct extraction from complements. In: Demirdash, L.C.H. (ed.) MIT Working Papers in Linguistics, vol. 3, pp. 101–124. MITWPL, Cambridge MA (1990)
31. Hegarty, M.: Adjunct extraction without traces. In: Proceedings of 10th West Coast Conference on Formal Linguistics (WCCFL 10) (1992)
32. Heim, I., Kratzer, A.: Semantics in Generative Grammar. Blackwell, Oxford (1998)
33. Hintikka, J.: Knowledge and Belief: An Introduction to the Logic of the Two Notions. Cornell University Press, Ithaca (1962)
34. Karttunen, L.: Some observations on factivity. Pap. Linguist. **4**(1), 55–69 (1971)
35. Kastner, I.: Factivity mirrors interpretation: the selectional requirements of presuppositional verbs. Lingua **164**(Part A), 156–188 (2015)
36. Kidwai, A.: Why Say That: Complementizer Sources and their Extended Projections (2014). invited talk at FASAL 4, Rutgers University
37. Kiparsky, P., Kiparsky, C.: Fact. In: Bierwisch, M., Heidolph, K.E. (eds.) Progress in Linguistics, pp. 143–173. Mouton de Gruyter, The Hague (1970)
38. Klaiman, M.: Volitionality and subject in Bengali: a study of semantic parameters in grammatical processes. Indiana University Linguistics club, Bloomington (1981)
39. Kratzer, A.: Severing the external argument from the verb. In: Roryck, J., Zaring (eds.) Phrase Structure and the Lexicon, pp. 109–138. Kluwer, Dordrecht (1996)
40. Kratzer, A.: Decomposing attitude verbs (2006). talk presented at the workshop in honor of Anita Mittwoch. The Hebrew University of Jerusalem
41. Kratzer, A.: Modality and the semantics of embedding. Talk given at Amsterdam Colloquium (2013)

42. Kratzer, A.: Modality for the 21st century. In: Anderson, S.R., Moeschler, J., Reboul, F. (eds.) L'interface Langage-Cognition/The Language-Cognition Interface: Actes du 19^e Congrès International des Linguistes Genève, pp. 179–199. Librarie Droz (2013)
43. Krifka, M.: Nominal Reference, Temporal Constitution and Quantification in Event Semantics. In: Bartsch, R., van Benthem, J., van Emde Boas, P. (eds.) Semantics and Contextual Expressions, pp. 75–115. Foris, Dordrecht (1989)
44. Krifka, M.: Thematic Relations as Links between Nominal Reference and Temporal Constitution. In: Sag, I.A., Szabolcsi, A. (eds.) Lexical Matters, pp. 29–53. CSLI Publications, US (1992)
45. Krifka, M.: The Origins of Telicity. In: Rothstein, S. (ed.) Events and Grammar, pp. 197–235. Kluwer Academic Publishers, Dordrecht/Boston/London (1998)
46. Lapointe, S.: A lexical analysis of the English auxiliary verb system. Lexical grammar, pp. 215–254 (1980)
47. Lazard, G.: Grammaire du Persan contemporain. Librairie C. Klincksiek, Paris (1957)
48. Moulton, K.: Natural Selection and the Syntax of Clausal Complementation. Ph.D. dissertation, University of Massachusetts (2009)
49. Moulton, K.: CPs: copies and compositionality. Linguist. Inq. **46**(2), 305–342 (2015)
50. Moulton, K.: (Non)-complement clauses and in-situ saturation: consequences for cross-clausal A-dependencies (2019). workshop on (Non)-Complementation GLOW in Asia XII/SICOGG XXI
51. Onishi, M.: Non-canonically marked S/A in Bengali. In: Aikhenvald, A.Y., Dixon, R., Onishi, M. (eds.) Non-canonincal Marking of Subjects and Objects, pp. 113–147. John Benjamins Publishing Co., Amsterdam/Philadelphia (2001)
52. Özyıldız, D.: Attitude reports with and without true belief. In: Proceedings of 27th Semantic and Linguistic Theory (SALT 27), pp. 397–417 (2017)
53. Partee, B.: Noun phrase interpretation and type-shifting principles. In: Groenendijk, J. (ed.) Studies in Discourse Representation Theory and the Theory of Generalized Quantifiers, pp. 115–143. Foris, Dordrecht (1987)
54. Partee, B., Rooth, M.: Generalized conjunction and type ambiguity. In: Bäuerle, R., Schwarze, C., von Stechow, A. (eds.) Meaning, Use, and Interpretation of Language, pp. 361–383. Walter de Gruyter, Berlin (1983)
55. Peachey, R.: On switch-reference and the internally-headed relative clause construction in Washo (2006). ms. University of Chicago
56. Percus, O.: Antipresuppositions. In: Ueyama, A. (ed.) Theoretical and empirical studies of reference and anaphora: toward the establishment of generative grammar as an empirical science. Japan Society for the promotion of science (2006)
57. Portner, P.: The semantic variability of gerunds. University of Massachusetts Occasional Papers in Linguistics, vol. 12 (1986)
58. Portner, P.H.: Gerunds and types of events. In: proceedings of 1st Semantics and Linguistic Theory (SALT 1) (1991)
59. Portner, P.H.: Situation theory and the semantics of propositional expressions. Ph.D. dissertation, University of Massachusetts (1992)
60. Singh, U.N.: Bole: an unresolved problem in Bengali syntax. Indian Linguist. **41**, 188–195 (1980)
61. Stalnaker, R.: Common ground. Linguist. Philos. **25**(5), 701–721 (2002)
62. Winter, Y.: On Partee's "noun phrase interpretation and type-shifting principles". In: McNally, L., Szabó, Z.G. (eds.) A Reader's Guide to Classic Papers in Formal Semantics. Studies in Linguistics and Philosophy, vol. 100, pp. 367–385. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-85308-2_19

Two-Layered Logics for Paraconsistent Probabilities

Marta Bílková[1], Sabine Frittella[2], Daniil Kozhemiachenko[2(✉)], and Ondrej Majer[3]

[1] Institute of Computer Science, The Czech Academy of Sciences, Prague, Czechia
bilkova@cs.cas.cz
[2] INSA Centre Val de Loire, Univ. Orléans, LIFO EA 4022, Blois, France
{sabine.frittella,daniil.kozhemiachenko}@insa-cvl.fr
[3] The Czech Academy of Sciences, Institute of Philosophy, Prague, Czechia
majer@flu.cas.cz

Abstract. We discuss two-layered logics formalising reasoning with paraconsistent probabilities that combine the Łukasiewicz $[0, 1]$-valued logic with Baaz $\triangle$ operator and the Belnap–Dunn logic. The first logic $\mathsf{Pr}^{\mathsf{Ł}^2}_{\triangle}$ (introduced in [7]) formalises a 'two-valued' approach where each event ϕ has independent positive and negative measures that stand for, respectively, the likelihoods of ϕ and $\neg\phi$. The second logic $\mathbf{4}\mathsf{Pr}^{\mathsf{Ł}_{\triangle}}$ that we introduce here corresponds to 'four-valued' probabilities. There, ϕ is equipped with four measures standing for pure belief, pure disbelief, conflict and uncertainty of an agent in ϕ.

We construct faithful embeddings of $\mathbf{4}\mathsf{Pr}^{\mathsf{Ł}_{\triangle}}$ and $\mathsf{Pr}^{\mathsf{Ł}^2}_{\triangle}$ into one another and axiomatise $\mathbf{4}\mathsf{Pr}^{\mathsf{Ł}_{\triangle}}$ using a Hilbert-style calculus. We also establish the decidability of both logics and provide complexity evaluations for them using an expansion of the constraint tableaux calculus for Ł.

Keywords: two-layered logics · Łukasiewicz logic · non-standard probabilities · paraconsistent logics · constraint tableaux

1 Introduction

Classical probability theory studies probability measures: maps from a probability space to $[0, 1]$ that satisfy the (finite or countable) additivity[1] condition:

$$\mu\left(\bigcup_{i\in I} E_i\right) = \sum_{i\in I} \mu(E_i) \qquad (\forall i, j \in I : i \neq j \Rightarrow E_i \cap E_j = \varnothing)$$

[1] In this paper, when dealing with the classical probability measures we will assume that they are *finitely* additive.

The research of Marta Bílková was supported by the grant 22-01137S of the Czech Science Foundation. The research of Sabine Frittella and Daniil Kozhemiachenko was funded by the grant ANR JCJC 2019, project PRELAP (ANR-19-CE48-0006). This research is part of the MOSAIC project financed by the European Union's Marie Skłodowska-Curie grant No. 101007627.

H. H. Hansen et al. (Eds.): WoLLIC 2023, LNCS 13923, pp. 101–117, 2023.
https://doi.org/10.1007/978-3-031-39784-4_7

Above, the disjointness of E_i and E_j can be construed as their incompatibility. Most importantly, if a propositional formula ϕ is associated with an event (and interpreted as a statement about it), then ϕ and $\neg\phi$ are incompatible and $\phi \vee \neg\phi$ exhausts the entire sample space.

Paraconsistent probability theory, on the other hand, assumes that the probability measure of an event represents not the likelihood of it happening but an agent's certainty therein which they infer from the information given by the sources. As a *single* source can give incomplete or contradictory information, it is reasonable to assume that a 'contradictory' event $\phi \wedge \neg\phi$ can have a positive probability and that $\phi \vee \neg\phi$ does not necessarily have probability 1.

Thus, a logic describing events should allow them to be both true and false (if the source gives contradictory information) or neither true nor false (when the source does not give information). Formally, this means that $\neg$ does not correspond to the complement in the sample space.

Paraconsistent Probabilities in BD. The simplest logic to represent reasoning about information provided by sources is the Belnap–Dunn logic [3,4,15]. Originally, BD was presented as a four-valued propositional logic in the $\{\neg, \wedge, \vee\}$ language. The values represent the different accounts a source can give regarding a statement ϕ:

- **T** stands for 'the source only says that ϕ is true';
- **F** stands for 'the source only says that ϕ is false';
- **B** stands for 'the source says both that ϕ is false and that ϕ is true';
- **N** stands for 'the source does not say that ϕ is false nor that it is true'.

The interpretation of the truth values allows for a reformulation of BD semantics in terms of *two classical but independent valuations*. Namely,

	is true when	**is false when**
$\neg\phi$	ϕ is false	ϕ is true
$\phi_1 \wedge \phi_2$	ϕ_1 and ϕ_2 are true	ϕ_1 is false or ϕ_2 is false
$\phi_1 \vee \phi_2$	ϕ_1 is true or ϕ_2 is true	ϕ_1 and ϕ_2 are false

It is easy to see that there are no universally true nor universally false formulas in BD. Thus, BD satisfies the desiderata outlined above.

The original interpretation of the Belnapian truth values is given in terms of the information one has. However, the information is assumed to be crisp. Probabilities over BD were introduced to formalise situations where one has access to probabilistic information. For instance, a first source could tell that p is true with probability 0.4 and a second that p is false with probability 0.7. If one follows BD and treats positive and negative evidence independently, one needs a non-classical notion of probabilities to represent this information.

The first representation of paraconsistent probabilities in terms of BD was given in [16], however, no axiomatisation was provided. Dunn proposes to divide the sample space into four exhaustive and mutually exclusive parts depending on the Belnapian value of ϕ. An alternative approach was proposed in [26]. There, the authors propose two equivalent interpretations based on the two formulations of semantics. The first option is to give ϕ two *independent probability measures*: the one determining the likelihood of ϕ to be true and the other the likelihood of ϕ to be false. The second option follows Dunn and also divides the sample space according to whether ϕ has value **T**, **B**, **N**, or **F** in a given state. Note that in both cases, the probabilities are interpreted *subjectively*.

The main difference between these two approaches is that in [16], the probability of $\phi \wedge \phi'$ is entirely determined by those of ϕ and ϕ' which makes it compositional. On the other hand, the paraconsistent probabilities proposed in [26] are not compositional w.r.t. conjunction. In this paper, we choose the latter approach since it can be argued [14] that belief is not compositional.

A similar approach to paraconsistent probabilities can be found in, e.g. [9,29]. There, probabilities are defined over an extension of BD with classicality and non-classicality operators. It is worth mentioning that the proposed axioms of probability are very close to those from [26]: e.g., both allow measures $\mathtt{p}$ s.t. $\mathtt{p}(\phi)+\mathtt{p}(\neg\phi) < 1$ (if the information regarding ϕ is incomplete) or $\mathtt{p}(\phi)+\mathtt{p}(\neg\phi) > 1$ (when the information is contradictory).

Two-Layered Logics for Uncertainty. Reasoning about uncertainty can be formalised via modal logics where the modality is interpreted as a measure of an event. The concrete semantics of the modality can be defined in two ways. First, using a modal language with Kripke semantics where the measure is defined on the set of states as done in, e.g., [12,13,19] for qualitative probabilities and in [11] for the quantitative ones. Second, employing a two-layered formalism (cf. [2,17,18], and [6,7] for examples). There, the logic is split into two levels: the inner layer describes events, and the outer layer describes the reasoning with the measure defined on events. The measure is a *non-nesting* modality $\mathtt{M}$, and the outer-layer formulas are built from 'modal atoms' of the form $\mathtt{M}\phi$ with ϕ being an inner-layer formula. The outer-layer formulas are then equipped with the semantics of a fuzzy logic that permits necessary operations (e.g., Łukasiewicz for the quantitative reasoning and Gödel for the qualitative).

In this work, we choose the two-layered approach. First, it is more modular than the usual Kripke semantics: as long as the logic of the event description is chosen, we can define different measures on top of it using different upper-layer logics. Second, the completeness proof is very simple since one only needs to translate the axioms of the given measure into the outer-layer logic. Finally, even though, the traditional Kripke semantics is more expressive than two-layered logics, this expressivity is not really necessary in many contexts. Indeed, people rarely say something like 'it is probable that it is probable that ϕ'. Moreover, it is considerably more difficult to motivate the assignment of truth values in the nesting case, in particular, when one and the same measure is applied both to a propositional and modalised formula as in, e.g., $\mathtt{M}(p \wedge \mathtt{M}q)$.

We will also be dealing with the formalisation of the *quantitative* probabilistic reasoning. Formally, this means that we assume that the agents can assign numerical values to their certainty in a given proposition or say something like 'I am twice as certain that it is going to rain than that it is going to snow'. Thus, we need a logic that can express the paraconsistent counterparts of the additivity condition as well as basic arithmetic operations. We choose the Łukasiewicz logic (Ł) for the outer layer since it can define (truncated) addition and subtraction on $[0,1]$.

Plan of the Paper. Our paper continues the project proposed in [8] and continued in [7] and [6]. Here, we set to provide a logic that formalises the reasoning with four-valued probabilities as presented in [26]. The rest of the text is organised as follows. In Sect. 2, we recall two approaches to probabilities over BD from [26]. In Sect. 3, we provide the semantics of our two-layered logics and in Sect. 4, we axiomatise them using Hilbert-style calculi. In Sect. 5, we prove that all our logics are decidable and establish their complexity evaluations. Finally, we wrap up our results in Sect. 6.

2 Two Approaches to Paraconsistent Probabilities

We begin with defining the semantics of BD on sets of states. The language of BD is given by the following grammar (with Prop being a countable set of propositional variables).

$$\mathcal{L}_{\mathsf{BD}} \ni \phi \coloneqq p \in \mathtt{Prop} \mid \neg\phi \mid (\phi \wedge \phi) \mid (\phi \vee \phi)$$

Convention 1. *In what follows, we will write* $\mathtt{Prop}(\phi)$ *to denote the set of variables occurring in* ϕ *and* $\mathsf{Lit}(\phi)$ *to denote the set of* literals *(i.e., variables or their negations) occurring in* ϕ*. Moreover, we use* $\mathtt{Sf}(\phi)$ *to stand for the set of all subformulas of* ϕ*.*

We are also going to use two kinds of formulas: the single- and the two-layered ones. To make the differentiation between them simpler, we use Greek letters from the end of the alphabet (ϕ, χ, ψ, etc.) to designate the first kind and the letters from the beginning of the alphabet (α, β, γ, ...) for the second kind.

Furthermore, we use v (with indices) to stand for the valuations of single-layered formulas and e (with indices) for the two-layered formulas.

Definition 1 (Set semantics of BD). *Let $\phi, \phi' \in \mathcal{L}_{\mathsf{BD}}$, $W \neq \varnothing$, and $v^+, v^- : \mathtt{Prop} \to 2^W$. For a model $\mathfrak{M} = \langle W, v^+, v^- \rangle$, we define notions of $w \vDash^+ \phi$ and $w \vDash^- \phi$ for $w \in W$ as follows.*

$$\begin{array}{ll}
w \vDash^+ p \text{ iff } w \in v^+(p) & w \vDash^- p \text{ iff } w \in v^-(p) \\
w \vDash^+ \neg\phi \text{ iff } w \vDash^- \phi & w \vDash^- \neg\phi \text{ iff } w \vDash^+ \phi \\
w \vDash^+ \phi \wedge \phi' \text{ iff } w \vDash^+ \phi \text{ and } w \vDash^+ \phi' & w \vDash^- \phi \wedge \phi' \text{ iff } w \vDash^- \phi \text{ or } w \vDash^- \phi' \\
w \vDash^+ \phi \vee \phi' \text{ iff } w \vDash^+ \phi \text{ or } w \vDash^+ \phi' & w \vDash^- \phi \vee \phi' \text{ iff } w \vDash^- \phi \text{ and } w \vDash^- \phi'
\end{array}$$

We denote the positive and negative extensions of a formula as follows:

$$|\phi|^+ \coloneqq \{w \in W \mid w \vDash^+ \phi\} \qquad |\phi|^- \coloneqq \{w \in W \mid w \vDash^- \phi\}.$$

We say that a sequent $\phi \vdash \chi$ *is* valid on $\mathfrak{M} = \langle W, v^+, v^- \rangle$ *(denoted,* $\mathfrak{M} \models [\phi \vdash \chi]$*) iff* $|\phi|^+ \subseteq |\chi|^+$ *and* $|\chi|^- \subseteq |\phi|^-$*. A sequent* $\phi \vdash \chi$ *is* BD-valid *(*$\phi \models_{\mathsf{BD}} \chi$*) iff it is valid on every model. In this case, we will say that* ϕ entails χ*.*

Now, we can use the above semantics to define probabilities on the models. We adapt the definitions from [26].

Definition 2 (BD models with ±-probabilities). *A* BD model with a ±-probability *is a tuple* $\mathfrak{M}_\mu = \langle \mathfrak{M}, \mu \rangle$ *with* $\mathfrak{M}$ *being a* BD *model and* $\mu : 2^W \to [0, 1]$ *satisfying:*

mon: *if* $X \subseteq Y$*, then* $\mu(X) \leq \mu(Y)$*;*
neg: $\mu(|\phi|^-) = \mu(|\neg\phi|^+)$*;*
ex: $\mu(|\phi \vee \chi|^+) = \mu(|\phi|^+) + \mu(|\chi|^+) - \mu(|\phi \wedge \chi|^+)$*.*

To facilitate the presentation of the four-valued probabilities defined over BD models, we introduce additional extensions of ϕ defined via $|\phi|^+$ and $|\phi|^-$.

Convention 2. *Let* $\mathfrak{M} = \langle W, v^+, v^- \rangle$ *be a* BD *model,* $\phi \in \mathcal{L}_{\mathsf{BD}}$*. We set*

$$|\phi|^{\mathsf{b}} = |\phi|^+ \setminus |\phi|^- \qquad |\phi|^{\mathsf{d}} = |\phi|^- \setminus |\phi|^+$$

$$|\phi|^{\mathsf{c}} = |\phi|^+ \cap |\phi|^- \qquad |\phi|^{\mathsf{u}} = W \setminus (|\phi|^+ \cup |\phi|^-)$$

We call these extensions, respectively, pure belief*,* pure disbelief*,* conflict*, and* uncertainty in ϕ*, following [26].*

Definition 3 (BD models with 4-probabilities). *A* BD model with a **4**-probability *is a tuple* $\mathfrak{M}_{\mathbf{4}} = \langle \mathfrak{M}, \mu_{\mathbf{4}} \rangle$ *with* $\mathfrak{M}$ *being a* BD *model and* $\mu_{\mathbf{4}} : 2^W \to [0, 1]$ *satisfying:*

part: $\mu_{\mathbf{4}}(|\phi|^{\mathsf{b}}) + \mu_{\mathbf{4}}(|\phi|^{\mathsf{d}}) + \mu_{\mathbf{4}}(|\phi|^{\mathsf{u}}) + \mu_{\mathbf{4}}(|\phi|^{\mathsf{c}}) = 1$*;*
neg: $\mu_{\mathbf{4}}(|\neg\phi|^{\mathsf{b}}) = \mu_{\mathbf{4}}(|\phi|^{\mathsf{d}})$*,* $\mu_{\mathbf{4}}(|\neg\phi|^{\mathsf{c}}) = \mu_{\mathbf{4}}(|\phi|^{\mathsf{c}})$*;*
contr: $\mu_{\mathbf{4}}(|\phi \wedge \neg\phi|^{\mathsf{b}}) = 0$*,* $\mu_{\mathbf{4}}(|\phi \wedge \neg\phi|^{\mathsf{c}}) = \mu_{\mathbf{4}}(|\phi|^{\mathsf{c}})$*;*
BCmon: *if* $\mathfrak{M} \models [\phi \vdash \chi]$*, then* $\mu_{\mathbf{4}}(|\phi|^{\mathsf{b}}) + \mu_{\mathbf{4}}(|\phi|^{\mathsf{c}}) \leq \mu_{\mathbf{4}}(|\chi|^{\mathsf{b}}) + \mu_{\mathbf{4}}(|\chi|^{\mathsf{c}})$*;*
BCex: $\mu_{\mathbf{4}}(|\phi|^{\mathsf{b}}) + \mu_{\mathbf{4}}(|\phi|^{\mathsf{c}}) + \mu_{\mathbf{4}}(|\psi|^{\mathsf{b}}) + \mu_{\mathbf{4}}(|\psi|^{\mathsf{c}}) = \mu_{\mathbf{4}}(|\phi \wedge \psi|^{\mathsf{b}}) + \mu_{\mathbf{4}}(|\phi \wedge \psi|^{\mathsf{c}}) + \mu_{\mathbf{4}}(|\phi \vee \psi|^{\mathsf{b}}) + \mu_{\mathbf{4}}(|\phi \vee \psi|^{\mathsf{c}})$*.*

Convention 3. *We will further utilise the following naming convention:*

- *we use the term* '±-probability' *to stand for* μ *from Definition 2;*
- *we call* $\mu_{\mathbf{4}}$ *from Defintion 3 a* '**4**-probability' *or a* 'four-valued probability'*.*

Recall that ±-probabilities are referred to as 'non-standard' in [26] and [7]. As this term is too broad (four-valued probabilities are not 'standard' either), we use a different designation.

Let us quickly discuss the measures defined above. First, observe that $\mu(|\phi|^+)$ and $\mu(|\phi|^-)$ are independent from one another. Thus, μ gives two measures to each ϕ, as desired. Second, recall [26, Theorems 2–3] that every **4**-probability on a BD model induces a ±-probability and vice versa. In the following sections, we will define two-layered logics for BD models with ±- and **4**-probabilities and show that they can be faithfully embedded into each other.

Remark 1. Note, that for every BD model with a ±-probability $\langle W, v^+, v^-, \mu\rangle$ (resp., BD model with $\mathbf{4}$-probability $\langle W, v^+, v^-, \mu_\mathbf{4}\rangle$), there exist a BD model $\langle W', v'^+, v'^-, \pi\rangle$ with a *classical* probability measure π s.t. $\pi(|\phi|^+) = \mu(|\phi|^+)$ (resp., $\pi(|\phi|^\mathsf{x}) = \mu_\mathbf{4}(|\phi|^\mathsf{x})$ for $\mathsf{x} \in \{\mathsf{b}, \mathsf{d}, \mathsf{c}, \mathsf{u}\}$) [26, Theorems 4–5]. Thus, we can further assume w.l.o.g. that μ and $\mu_\mathbf{4}$ are *classical probability measures* on W.

3 Logics for Paraconsistent Probabilities

In this section, we provide logics that are (weakly) complete w.r.t. BD models with ±- and $\mathbf{4}$-probabilities. Since conditions on measures contain arithmetic operations on $[0, 1]$, we choose an expansion of Łukasiewicz logic, namely, Łukasiewicz logic with $\triangle$ ($\mathsf{Ł}_\triangle$), for the outer layer. Furthermore, ±-probabilities work with both positive and negative extensions of formulas, whence it seems reasonable to use $\mathsf{Ł}^2$ — a paraconsistent expansion of $\mathsf{Ł}$ (cf. [5,8] for details) with two valuations — v_1 (support of truth) and v_2 (support of falsity) — on $[0, 1]$. This was done in [7] — the resulting logic $\mathsf{Pr}^{\mathsf{Ł}^2}_\triangle$ was proven to be complete w.r.t. BD models with ±-probabilities.

We begin by recalling the language and standard semantics of Łukasiewicz logic with $\triangle$ and its paraconsistent expansion $\mathsf{Ł}^2_\triangle$.

Definition 4. *The standard* $\mathsf{Ł}_\triangle$*-algebra is a tuple* $\langle[0, 1], \sim_\mathsf{Ł}, \triangle_\mathsf{Ł}, \wedge_\mathsf{Ł}, \vee_\mathsf{Ł}, \rightarrow_\mathsf{Ł}, \odot_\mathsf{Ł}, \oplus_\mathsf{Ł}, \ominus_\mathsf{Ł}\rangle$ *with the operations are defined as follows.*

$$\sim_\mathsf{Ł} a \coloneqq 1 - a \qquad\qquad \triangle_\mathsf{Ł} a \coloneqq \begin{cases} 1 & \textit{if } a = 1 \\ 0 & \textit{otherwise} \end{cases}$$

$$a \wedge_\mathsf{Ł} b \coloneqq \min(a, b) \qquad a \vee_\mathsf{Ł} b \coloneqq \max(a, b) \qquad a \rightarrow_\mathsf{Ł} b \coloneqq \min(1, 1-a+b)$$
$$a \odot_\mathsf{Ł} b \coloneqq \max(0, a+b-1) \qquad a \oplus_\mathsf{Ł} b \coloneqq \min(1, a+b) \qquad a \ominus_\mathsf{Ł} b \coloneqq \max(0, a-b)$$

Definition 5 (Łukasiewicz logic with $\triangle$). *The language of* $\mathsf{Ł}_\triangle$ *is given via the following grammar*

$$\mathcal{L}_\mathsf{Ł} \ni \phi \coloneqq p \in \mathtt{Prop} \mid \sim\phi \mid \triangle\phi \mid (\phi \wedge \phi) \mid (\phi \vee \phi) \mid (\phi \rightarrow \phi) \mid (\phi \odot \phi) \mid (\phi \oplus \phi) \mid (\phi \ominus \phi)$$

We will also write $\phi \leftrightarrow \chi$ *as a shorthand for* $(\phi \rightarrow \chi) \odot (\chi \rightarrow \phi)$.

A valuation is a map $v : \mathtt{Prop} \rightarrow [0, 1]$ *that is extended to the complex formulas as expected:* $v(\phi \circ \chi) = v(\phi) \circ_\mathsf{Ł} v(\chi)$.

ϕ *is* $\mathsf{Ł}_\triangle$-valid *iff* $v(\phi) = 1$ *for every* v. Γ entails χ *(denoted* $\Gamma \models_{\mathsf{Ł}_\triangle} \chi$*) iff for every* v *s.t.* $v(\phi) = 1$ *for all* $\phi \in \Gamma$*, it holds that* $v(\chi) = 1$ *as well.*

Remark 2. Note that $\triangle$, $\sim$, and $\rightarrow$ can be used to define all other connectives as follows.

$$\phi \vee \chi \coloneqq (\phi \rightarrow \chi) \rightarrow \chi \qquad \phi \wedge \chi \coloneqq \sim(\sim\phi \vee \sim\chi) \qquad \phi \oplus \chi \coloneqq \sim\phi \rightarrow \chi$$
$$\phi \odot \chi \coloneqq \sim(\phi \rightarrow \sim\chi) \qquad \phi \ominus \chi \coloneqq \phi \odot \sim\chi$$

To facilitate the presentation, we recall the Hilbert calculus for $Ł_\triangle$. It can be obtained by adding $\triangle$ axioms and rules from [1], [24, Defenition 2.4.5], or [10, Chapter I,2.2.1] to the Hilbert-style calculus for Ł from [27, §6.2].

Definition 6 ($\mathcal{H}Ł_\triangle$ — the Hilbert-style calculus for $Ł_\triangle$). *The calculus contains the following axioms and rules.*

w: $\phi \rightarrow (\chi \rightarrow \phi)$.
sf: $(\phi \rightarrow \chi) \rightarrow ((\chi \rightarrow \psi) \rightarrow (\phi \rightarrow \psi))$.
waj: $((\phi \rightarrow \chi) \rightarrow \chi) \rightarrow ((\chi \rightarrow \phi) \rightarrow \phi)$.
co: $({\sim}\chi \rightarrow {\sim}\phi) \rightarrow (\phi \rightarrow \chi)$.
MP: $\dfrac{\phi \quad \phi \rightarrow \chi}{\chi}$.
△1: $\triangle\phi \vee {\sim}\triangle\phi$.
△2: $\triangle\phi \rightarrow \phi$.
△3: $\triangle\phi \rightarrow \triangle\triangle\phi$.
△4: $\triangle(\phi \vee \chi) \rightarrow \triangle\phi \vee \triangle\chi$.
△5: $\triangle(\phi \rightarrow \chi) \rightarrow \triangle\phi \rightarrow \triangle\chi$.
△nec: $\dfrac{\phi}{\triangle\phi}$.

Łukasiewicz logic is known to lack compactness [24, Remark 3.2.14], whence, $\mathcal{H}Ł_\triangle$ is only *weakly complete.*

Proposition 1 (Weak completeness of $\mathcal{H}Ł_\triangle$). *Let $\Gamma \subseteq \mathcal{L}_Ł$ be finite. Then*

$$\Gamma \models_{Ł_\triangle} \phi \text{ iff } \Gamma \vdash_{\mathcal{H}Ł_\triangle} \phi$$

Definition 7 ($Ł^2_\triangle$). *The language is constructed using the following grammar.*

$$\mathcal{L}_{Ł^2_\triangle} \ni \phi \coloneqq p \in \texttt{Prop} \mid \neg\phi \mid {\sim}\phi \mid \triangle\phi \mid (\phi \rightarrow \phi)$$

The semantics is given by two *valuations v_1 (support of truth) and v_2 (support of falsity) $v_1, v_2 : \texttt{Prop} \rightarrow [0,1]$ that are extended as follows.*

$$\begin{aligned} v_1(\neg\phi) &= v_2(\phi) & v_2(\neg\phi) &= v_1(\phi) \\ v_1({\sim}\phi) &= {\sim}_Ł v_1(\phi) & v_2({\sim}\phi) &= {\sim}_Ł v_2(\phi) \\ v_1(\triangle\phi) &= \triangle_Ł v_1(\phi) & v_2(\triangle\phi) &= {\sim}_Ł \triangle_Ł {\sim}_Ł v_2(\phi) \\ v_1(\phi \rightarrow \chi) &= v_1(\phi) \rightarrow_Ł v_1(\chi) & v_2(\phi \rightarrow \chi) &= v_2(\chi) \ominus_Ł v_2(\phi) \end{aligned}$$

*We say that ϕ is $Ł^2_\triangle$-*valid *iff for every v_1 and v_2, it holds that $v_1(\phi) = 1$ and $v_2(\phi) = 0$.*

Remark 3. Again, the remaining connectives can be defined as in Remark 2. Furthermore, when there is no risk of confusion, we write $v(\phi) = (x, y)$ to designate that $v_1(\phi) = x$ and $v_2(\phi) = y$.

We are now ready to present the two-layered logics. We begin with $\mathsf{Pr}^{Ł^2}_\triangle$ from [7].

Definition 8 **($\mathsf{Pr}^{\text{Ł}^2}_{\triangle}$: language and semantics).** *The language of* $\mathsf{Pr}^{\text{Ł}^2}_{\triangle}$ *is given by the following grammar*

$$\mathcal{L}_{\mathsf{Pr}^{\text{Ł}^2}_{\triangle}} \ni \alpha \coloneqq \mathsf{Pr}\phi \mid {\sim}\alpha \mid \neg\alpha \mid \triangle\alpha \mid (\alpha \to \alpha) \qquad (\phi \in \mathcal{L}_{\mathsf{BD}})$$

A $\mathsf{Pr}^{\text{Ł}^2}_{\triangle}$ *model is a tuple* $\mathbb{M} = \langle \mathfrak{M}, \mu, e_1, e_2 \rangle$ *with* $\langle \mathfrak{M}, \mu \rangle$ *being a* BD *model with* $\pm$*-probability and* $e_1, e_2 : \mathcal{L}_{\mathsf{Pr}^{\text{Ł}^2}_{\triangle}} \to [0,1]$ *s.t.* $e_1(\mathsf{Pr}\phi) = \mu(|\phi|^+)$, $e_2(\mathsf{Pr}\phi) = \mu(|\phi|^-)$, *and the values of complex formulas being computed following Definition 7. We say that* α *is* $\mathsf{Pr}^{\text{Ł}^2}_{\triangle}$ valid *iff* $e(\alpha) = (1, 0)$ *in every model.*

Definition 9 **($4\mathsf{Pr}^{\text{Ł}_{\triangle}}$: language and semantics).** *The language of* $4\mathsf{Pr}^{\text{Ł}_{\triangle}}$ *is constructed by the following grammar:*

$$\mathcal{L}_{4\mathsf{Pr}^{\text{Ł}_{\triangle}}} \ni \alpha \coloneqq \mathsf{Bl}\phi \mid \mathsf{Db}\phi \mid \mathsf{Cf}\phi \mid \mathsf{Uc}\phi \mid {\sim}\alpha \mid \triangle\alpha \mid (\alpha \to \alpha) \qquad (\phi \in \mathcal{L}_{\mathsf{BD}})$$

A $4\mathsf{Pr}^{\text{Ł}_{\triangle}}$ *model is a tuple* $\mathbb{M} = \langle \mathfrak{M}, \mu_{\mathbf{4}}, e \rangle$ *with* $\langle \mathfrak{M}, \mu_{\mathbf{4}} \rangle$ *being a* BD *model with* $\mathbf{4}$*-probability s.t.* $e(\mathsf{Bl}\phi) = \mu_{\mathbf{4}}(|\phi|^{\mathsf{b}})$, $e(\mathsf{Db}\phi) = \mu_{\mathbf{4}}(|\phi|^{\mathsf{d}})$, $e(\mathsf{Cf}\phi) = \mu_{\mathbf{4}}(|\phi|^{\mathsf{c}})$, $e(\mathsf{Uc}\phi) = \mu_{\mathbf{4}}(|\phi|^{\mathsf{u}})$, *and the values of complex formulas computed via Definition 5. We say that* α *is* $4\mathsf{Pr}^{\text{Ł}_{\triangle}}$ valid *iff* $e(\alpha) = 1$ *in every model. A set of formulas* Γ entails α *(*$\Gamma \models_{4\mathsf{Pr}^{\text{Ł}_{\triangle}}} \alpha$*) iff there is no* $\mathbb{M}$ *s.t.* $e(\gamma) = 1$ *for every* $\gamma \in \Gamma$ *but* $e(\alpha) \neq 1$.

Remark 4. Note that we are going to prove only the weak completeness. In addition, BD is a tabular logic, whence there exist only finitely many pairwise non-equivalent formulas over a finite set of variables. Thus, we do not need to explicitly assume that the underlying BD models are finite.

Convention 4. *We will further call formulas of the form* $\mathsf{X}\phi$ *(*$\phi \in \mathcal{L}_{\mathsf{BD}}$, $\mathsf{X} \in \{\mathsf{Pr}, \mathsf{Bl}, \mathsf{Db}, \mathsf{Cf}, \mathsf{Uc}\}$*)* modal atoms*. We interpret the value of a modal atom as a degree of certainty that the agent has in* ϕ*. For example,* $e(\mathsf{Pr}p) = (\frac{3}{4}, \frac{1}{2})$ *means that the agent's certainty in* p *is* $\frac{3}{4}$ *and in* $\neg p$ *is* $\frac{1}{2}$*. Similarly,* $e(\mathsf{Cf}q) = \frac{1}{3}$ *is construed as 'the agent is conflicted w.r.t.* q *to the degree* $\frac{1}{3}$*'.*

To make the semantics clearer, we provide the following example.

Example 1. Consider the following BD model.

$$w_0 : p^{\pm}, \not{q} \qquad\qquad w_1 : p^-, q^-$$

And let $\mu = \mu_{\mathbf{4}}$ be defined as follows: $\mu(\{w_0\}) = \frac{2}{3}$, $\mu(\{w_1\}) = \frac{1}{3}$, $\mu(W) = 1$, $\mu(\varnothing) = 0$. It is easy to check that μ satisfies the conditions of Definitions 2 and 3. Now let e be the $\text{Ł}^2_{\triangle}$ *valuation* and $e_{\mathbf{4}}$ the $\text{Ł}_{\triangle}$ *valuation* induced by μ and $\mu_{\mathbf{4}}$, respectively.

Consider two BD formulas: $p \vee q$ and p. We have $e(\mathsf{Pr}(p \vee q)) = \left(\frac{2}{3}, \frac{1}{3}\right)$ and $e(\mathsf{Pr}p) = \left(\frac{2}{3}, 1\right)$. In $\mathbf{4}\mathsf{Pr}^{\text{Ł}\triangle}$, we have $e_{\mathbf{4}}(\mathsf{Bl}(p \vee q)) = \frac{2}{3}$, $e_{\mathbf{4}}(\mathsf{Db}(p \vee q)) = \frac{1}{3}$, $e_{\mathbf{4}}(\mathsf{Cf}p) = \frac{2}{3}$, $e_{\mathbf{4}}(\mathsf{Cf}(p \vee q)), e_{\mathbf{4}}(\mathsf{Uc}(p \vee q)) = 0$, $e_{\mathbf{4}}(\mathsf{Bl}p), e(\mathsf{Uc}p) = 0$, $e_{\mathbf{4}}(\mathsf{Cf}p) = \frac{2}{3}$, and $e(\mathsf{Db}p) = \frac{1}{3}$.

The following property of $\mathsf{Pr}^{\text{Ł}^2}_{\triangle}$ is going to be useful further in the section.

Lemma 1. *Let* $\alpha \in \mathcal{L}_{\mathsf{Pr}^{\text{Ł}^2}_{\triangle}}$. *Then,* α *is* $\mathsf{Pr}^{\text{Ł}^2}_{\triangle}$ *valid iff* $e_1(\alpha) = 1$ *in every* $\mathsf{Pr}^{\text{Ł}^2}_{\triangle}$ *model.*

Proof. Let $\mathbb{M} = \langle W, v^+, v^-, \mu, e_1, e_2 \rangle$ be a $\mathsf{Pr}^{\text{Ł}^2}_{\triangle}$ model s.t. $e_2(\alpha) \neq 0$. We construct a model $\mathbb{M}^* = \langle W, (v^*)^+, (v^*)^-, \mu, e_1^*, e_2^* \rangle$ where $e_1^*(\alpha) \neq 1$. To do this, we define new BD valuations $(v^*)^+$ and $(v^*)^-$ on W as follows.

$$w \in v^+(p), w \notin v^-(p) \text{ then } w \in (v^*)^+(p), w \notin (v^*)^-(p)$$
$$w \in v^+(p), v^-(p) \text{ then } w \notin (v^*)^+(p), (v^*)^-(p)$$
$$w \notin v^+(p), v^-(p) \text{ then } w \in (v^*)^+(p), (v^*)^-(p)$$
$$w \notin v^+(p), w \in v^-(p) \text{ then } w \notin (v^*)^+(p), w \in (v^*)^-(p)$$

It can be easily checked by induction on $\phi \in \mathcal{L}_{\mathsf{BD}}$ that

$$|\phi|^+_{\mathbb{M}} = W \setminus |\phi|^-_{\mathbb{M}^*} \qquad\qquad |\phi|^-_{\mathbb{M}} = W \setminus |\phi|^+_{\mathbb{M}^*}$$

Now, since we can w.l.o.g. assume that μ is a (classical) probability measure on W (recall Remark 1), we have that

$$e^*(\mathsf{Pr}\phi) = (1 - \mu(|\phi|^-), 1 - \mu(|\phi|^+)) = (1 - e_2(\mathsf{Pr}\phi), 1 - e_1(\mathsf{Pr}\phi))$$

Observe that if $e(\alpha) = (x, y)$, then $e(\neg{\sim}\alpha) = (1 - y, 1 - x)$. Furthermore, it is straightforward to verify that the following formulas are valid.

$$\neg{\sim}\neg\alpha \leftrightarrow \neg\neg{\sim}\alpha \qquad\qquad \neg{\sim}{\sim}\alpha \leftrightarrow {\sim}\neg{\sim}\alpha$$
$$\neg{\sim}\triangle\alpha \leftrightarrow \triangle\neg{\sim}\alpha \qquad\qquad \neg{\sim}(\alpha \rightarrow \alpha') \leftrightarrow \neg{\sim}\alpha \rightarrow \neg{\sim}\alpha'$$

Hence, $e^*(\alpha) = (1 - e_2(\alpha), 1 - e_1(\alpha))$ for every $\alpha \in \mathcal{L}_{\mathsf{Pr}^{\text{Ł}^2}_{\triangle}}$. The result follows.

At first glance, $\mathbf{4}\mathsf{Pr}^{\text{Ł}\triangle}$ gives a more fine-grained view on a BD model than $\mathsf{Pr}^{\text{Ł}^2}_{\triangle}$ since it can evaluate each extension of a given $\phi \in \mathcal{L}_{\mathsf{BD}}$, while $\mathsf{Pr}^{\text{Ł}^2}_{\triangle}$ always considers $|\phi|^+$ and $|\phi|^-$ together. In the remainder of the section, we show that the two logics have, in fact, the same expressivity.

One can see from Definition 8 that $\neg\mathsf{Pr}\phi \leftrightarrow \mathsf{Pr}\neg\phi$. Furthermore, Ł^2 admits $\neg$ negation normal forms and is a conservative extension of Ł [5,8]. Thus, it is possible to push all $\neg$'s occurring in $\alpha \in \mathcal{L}_{\mathsf{Pr}^{\text{Ł}^2}_{\triangle}}$ to modal atoms. We will use this fact to establish the embeddings of $\mathsf{Pr}^{\text{Ł}^2}_{\triangle}$ and $\mathbf{4}\mathsf{Pr}^{\text{Ł}\triangle}$ into one another.

Definition 10. *Let $\alpha \in \mathcal{L}_{\mathsf{Pr}^{\text{Ł}^2}_{\triangle}}$. $\alpha^{\neg}$ is produced from α by successively applying the following transformations.*

$$\neg\mathsf{Pr}\phi \rightsquigarrow \mathsf{Pr}\neg\phi \qquad \neg\neg\alpha \rightsquigarrow \alpha \qquad \neg{\sim}\alpha \rightsquigarrow {\sim}\neg\alpha$$
$$\neg(\alpha \to \alpha') \rightsquigarrow {\sim}(\neg\alpha' \to \neg\alpha) \qquad \neg\triangle\alpha \rightsquigarrow {\sim}\triangle{\sim}\neg\alpha$$

It is easy to check that $e(\alpha) = e(\alpha^{\neg})$ in every $\mathsf{Pr}^{\text{Ł}^2}_{\triangle}$ model.

Definition 11. *Let $\alpha \in \mathcal{L}_{\mathsf{Pr}^{\text{Ł}^2}_{\triangle}}$ be $\neg$-free, we define $\alpha^{\mathbf{4}} \in \mathcal{L}_{\mathbf{4}\mathsf{Pr}^{\text{Ł}_{\triangle}}}$ as follows.*

$$(\mathsf{Pr}\phi)^{\mathbf{4}} = \mathsf{Bl}\phi \oplus \mathsf{Cf}\phi$$
$$(\heartsuit\alpha)^{\mathbf{4}} = \heartsuit\alpha^{\mathbf{4}} \qquad (\heartsuit \in \{\triangle, {\sim}\})$$
$$(\alpha \to \alpha')^{\mathbf{4}} = \alpha^{\mathbf{4}} \to \alpha'^{\mathbf{4}}$$

Let $\beta \in \mathcal{L}_{\mathbf{4}\mathsf{Pr}^{\text{Ł}_{\triangle}}}$. We define $\beta^{\pm}$ as follows.

$$(\mathsf{Bl}\phi)^{\pm} = \mathsf{Pr}\phi \ominus \mathsf{Pr}(\phi \wedge \neg\phi)$$
$$(\mathsf{Cf}\phi)^{\pm} = \mathsf{Pr}(\phi \wedge \neg\phi)$$
$$(\mathsf{Uc}\phi)^{\pm} = {\sim}\mathsf{Pr}(\phi \vee \neg\phi)$$
$$(\mathsf{Db}\phi)^{\pm} = \mathsf{Pr}\neg\phi \ominus \mathsf{Pr}(\phi \wedge \neg\phi)$$
$$(\heartsuit\beta)^{\pm} = \heartsuit\beta^{\pm} \qquad (\heartsuit \in \{\triangle, {\sim}\})$$
$$(\beta \to \beta')^{\pm} = \beta^{\pm} \to \beta'^{\pm}$$

Theorem 1. *$\alpha \in \mathcal{L}_{\mathsf{Pr}^{\text{Ł}^2}_{\triangle}}$ is $\mathsf{Pr}^{\text{Ł}^2}_{\triangle}$ valid iff $(\alpha^{\neg})^{\mathbf{4}}$ is $\mathbf{4}\mathsf{Pr}^{\text{Ł}_{\triangle}}$ valid.*

Proof. Let w.l.o.g. $\mathbb{M} = \langle W, v^+, v^-, \mu, e_1, e_2\rangle$ be a BD model with $\pm$-probability where μ is a *classical* probability measure and let $e(\alpha) = (x, y)$. We show that in the BD model $\mathbb{M}_{\mathbf{4}} = \langle W, v^+, v^-, \mu, e_1\rangle$ with *four-probability* μ, $e_1((\alpha^{\neg})^{\mathbf{4}}) = x$. This is sufficient to prove the result. Indeed, by Lemma 1, it suffices to verify that $e_1(\alpha) = 1$ for every e_1, to establish the validity of $\alpha \in \mathcal{L}_{\mathsf{Pr}^{\text{Ł}^2}_{\triangle}}$.

We proceed by induction on $\alpha^{\neg}$ (recall that $\alpha \leftrightarrow \alpha^{\neg}$ is $\mathsf{Pr}^{\text{Ł}^2}_{\triangle}$ valid). If $\alpha = \mathsf{Pr}\phi$, then $e_1(\mathsf{Pr}\phi) = \mu(|\phi|^+) = \mu(|\phi|^{\mathsf{b}} \cup |\phi|^{\mathsf{c}})$. But $|\phi|^{\mathsf{b}}$ and $|\phi|^{\mathsf{c}}$ are disjoint, whence $\mu(|\phi|^{\mathsf{b}} \cup |\phi|^{\mathsf{c}}) = \mu(|\phi|^{\mathsf{b}}) + \mu(|\phi|^{\mathsf{c}})$, and since $\mu(|\phi|^{\mathsf{b}}) + \mu(|\phi|^{\mathsf{c}}) \leq 1$, we have that $e_1(\mathsf{Bl}\phi \oplus \mathsf{Cf}\phi) = \mu(|\phi|^{\mathsf{b}}) + \mu(|\phi|^{\mathsf{c}}) = e_1(\mathsf{Pr}\phi)$, as required.

The induction steps are straightforward since the semantic conditions of support of truth in $\text{Ł}^2_{\triangle}$ coincide with the semantics of $\text{Ł}_{\triangle}$ (cf. Definitions 7 and 5).

Theorem 2. *$\beta \in \mathcal{L}_{\mathbf{4}\mathsf{Pr}^{\text{Ł}_{\triangle}}}$ is $\mathcal{L}_{\mathbf{4}\mathsf{Pr}^{\text{Ł}_{\triangle}}}$ valid iff $\beta^{\pm}$ is $\mathsf{Pr}^{\text{Ł}^2}_{\triangle}$ valid.*

Proof. Assume w.l.o.g. that $\mathbb{M} = \langle W, v^+, v^-, \mu_{\mathbf{4}}, e\rangle$ is a BD model with a **4**-probability where $\mu_{\mathbf{4}}$ is a classical probability measure and $e(\beta) = x$. We define a BD model with $\pm$-probability $\mathbb{M}^{\pm} = \langle W, v^+, v^-, \mu_{\mathbf{4}}, e_1, e_2\rangle$ and show that $e_1(\beta^{\pm}) = x$. Again, it is sufficient for us by Lemma 1.

We proceed by induction on β. If $\beta = \mathsf{Bl}\phi$, then $e(\mathsf{Bl}\phi) = \mu_{\mathbf{4}}(|\phi|^{\mathsf{b}})$. Now observe that $\mu_{\mathbf{4}}(|\phi|^+) = \mu(|\phi|^{\mathsf{b}} \cup |\phi|^{\mathsf{c}}) = \mu_{\mathbf{4}}(|\phi|^{\mathsf{b}}) + \mu_{\mathbf{4}}(|\phi|^{\mathsf{c}})$ since $|\phi|^{\mathsf{b}}$ and $|\phi|^{\mathsf{c}}$ are disjoint. But $\mu_{\mathbf{4}}(|\phi|^+) = e_1(\mathsf{Pr}\phi)$ and $\mu_{\mathbf{4}}(|\phi|^{\mathsf{c}}) = \mu_{\mathbf{4}}(|\phi \wedge \neg\phi|^+)$ since $|\phi \wedge \neg\phi|^+ = |\phi|^{\mathsf{c}}$. Thus, $\mu_{\mathbf{4}}(|\phi|^{\mathsf{b}}) = e_1(\mathsf{Pr}\phi \ominus \mathsf{Pr}(\phi \wedge \neg\phi))$ as required.

Other basis cases of $\mathsf{Cf}\phi$, $\mathsf{Uc}\phi$, and $\mathsf{Db}\phi$ can be tackled in a similar manner. The induction steps are straightforward since the support of truth in $\text{Ł}^2_{\triangle}$ coincides with semantical conditions in $\text{Ł}_{\triangle}$.

Remark 5. Theorem 1 and 2 mean, in a sense, that $\mathsf{Pr}^{\text{Ł}^2}_{\triangle}$ and $\mathbf{4}\mathsf{Pr}^{\text{Ł}_{\triangle}}$ can be treated as syntactic variants of one another. Conceptually, however, they are somewhat different. Namely, $\mathsf{Pr}^{\text{Ł}^2}_{\triangle}$ assigns *two independent measures* to each formula ϕ corresponding to the likelihoods of ϕ itself and $\neg\phi$. On the other hand, $\mathbf{4}\mathsf{Pr}^{\text{Ł}_{\triangle}}$ treats the extensions ϕ as a separation of the underlying sample set into four parts whose measures must add up to 1.

4 Hilbert-Style Axiomatisation of $\mathbf{4}\mathsf{Pr}^{\text{Ł}_{\triangle}}$

Let us proceed to the axiomatisation of $\mathbf{4}\mathsf{Pr}^{\text{Ł}_{\triangle}}$. Since its outer layer expands $\text{Ł}_{\triangle}$, we will need to encode the conditions on $\mu_{\mathbf{4}}$ therein. Furthermore, since Ł (and hence, $\text{Ł}_{\triangle}$) is not compact [24, Remark 3.2.14], our axiomatisation can only be *weakly complete* (i.e., complete w.r.t. finite theories).

The axiomatisation will consist of two types of axioms: those that axiomatise $\text{Ł}_{\triangle}$ and modal axioms that encode the conditions from Definition 3. For the sake of brevity, we will compress the axiomatisation of $\text{Ł}_{\triangle}$ into one axiom that allows us to use $\text{Ł}_{\triangle}$ theorems[2] without proof.

Definition 12 ($\mathcal{H}\mathbf{4}\mathsf{Pr}^{\text{Ł}_{\triangle}}$ — Hilbert-style calculus for $\mathbf{4}\mathsf{Pr}^{\text{Ł}_{\triangle}}$). *The calculus $\mathcal{H}\mathbf{4}\mathsf{Pr}^{\text{Ł}_{\triangle}}$ consists of the following axioms and rules.*

$\text{Ł}_{\triangle}$**:** $\text{Ł}_{\triangle}$ *valid formulas instantiated in* $\mathcal{L}_{\mathbf{4}\mathsf{Pr}^{\text{Ł}_{\triangle}}}$.
equiv: $\mathsf{X}\phi \leftrightarrow \mathsf{X}\chi$ *for every* $\phi, \chi \in \mathcal{L}_{\mathsf{BD}}$ *s.t.* $\phi \dashv\vdash \chi$ *is* BD*-valid and* $\mathsf{X} \in \{\mathsf{Bl}, \mathsf{Db}, \mathsf{Cf}, \mathsf{Uc}\}$.
contr: $\sim\mathsf{Bl}(\phi \wedge \neg\phi)$; $\mathsf{Cf}\phi \leftrightarrow \mathsf{Cf}(\phi \wedge \neg\phi)$.
neg: $\mathsf{Bl}\neg\phi \leftrightarrow \mathsf{Db}\phi$; $\mathsf{Cf}\neg\phi \leftrightarrow \mathsf{Cf}\phi$.
mon: $(\mathsf{Bl}\phi \oplus \mathsf{Cf}\phi) \rightarrow (\mathsf{Bl}\chi \oplus \mathsf{Cf}\chi)$ *for every* $\phi, \chi \in \mathcal{L}_{\mathsf{BD}}$ *s.t.* $\phi \vdash \chi$ *is* BD*-valid.*
part1: $\mathsf{Bl}\phi \oplus \mathsf{Db}\phi \oplus \mathsf{Cf}\phi \oplus \mathsf{Uc}\phi$.
part2: $((\mathsf{X}_1\phi \oplus \mathsf{X}_2\phi \oplus \mathsf{X}_3\phi \oplus \mathsf{X}_4\phi) \ominus \mathsf{X}_4\phi) \leftrightarrow (\mathsf{X}_1\phi \oplus \mathsf{X}_2\phi \oplus \mathsf{X}_3\phi)$ *with* $\mathsf{X}_i \neq \mathsf{X}_j$, $\mathsf{X}_i \in \{\mathsf{Bl}, \mathsf{Db}, \mathsf{Cf}, \mathsf{Uc}\}$.
ex: $(\mathsf{Bl}(\phi \vee \chi) \oplus \mathsf{Cf}(\phi \vee \chi)) \leftrightarrow ((\mathsf{Bl}\phi \oplus \mathsf{Cf}\phi) \ominus (\mathsf{Bl}(\phi \wedge \chi) \oplus \mathsf{Cf}(\phi \wedge \chi)) \oplus (\mathsf{Bl}\chi \oplus \mathsf{Cf}\chi))$.

[2] A Hilbert-style calculus for Ł can be found in, e.g. [27], and the axioms for $\triangle$ in [1]. A concise presentation of a Hilbert-style calculus for $\text{Ł}_{\triangle}$ is also given in [7].

MP: $\dfrac{\alpha \quad \alpha \to \alpha'}{\alpha'}$.

$\triangle$nec: $\dfrac{\mathcal{H}4\mathsf{Pr}^{Ł_\triangle} \vdash \alpha}{\mathcal{H}4\mathsf{Pr}^{Ł_\triangle} \vdash \triangle\alpha}$.

The axioms above are simple translations of properties from Definition 3. We split **part** in two axioms to ensure that the values of $\mathsf{Bl}\phi$, $\mathsf{Db}\phi$, $\mathsf{Cf}\phi$, and $\mathsf{Uc}\phi$ sum up exactly to 1.

Theorem 3. *Let* $\Xi \subseteq \mathcal{L}_{\mathbf{4}\mathsf{Pr}^{Ł_\triangle}}$ *be finite. Then* $\Xi \models_{\mathbf{4}\mathsf{Pr}^{Ł_\triangle}} \alpha$ *iff* $\Xi \vdash_{\mathcal{H}\mathbf{4}\mathsf{Pr}^{Ł_\triangle}} \alpha$.

Proof. Soundness can be established by the routine check of the axioms' validity. Thus, we prove completeness. We reason by contraposition. Assume that $\Xi \nvdash_{\mathcal{H}\mathbf{4}\mathsf{Pr}^{Ł_\triangle}} \alpha$. Now, observe that $\mathcal{H}4\mathsf{Pr}^{Ł_\triangle}$ proofs are, actually, $\mathcal{H}Ł_\triangle$ proofs with additional probabilistic axioms. Let Ξ^* stand for Ξ extended with probabilistic axioms built over all pairwise non-equivalent $\mathcal{L}_{\mathsf{BD}}$ formulas constructed from $\mathtt{Prop}[\Xi \cup \{\alpha\}]$. Clearly, $\Xi^* \nvdash_{\mathcal{H}\mathbf{4}\mathsf{Pr}^{Ł_\triangle}} \alpha$ either. Moreover, Ξ^* is finite as well since BD is tabular (and whence, there exist only finitely many pairwise non-equivalent $\mathcal{L}_{\mathsf{BD}}$ formulas over a finite set of variables). Now, by the weak completeness of $\mathcal{H}Ł_\triangle$ (Proposition 1), there exists an $Ł_\triangle$ valuation e on $[0,1]$ s.t. $e[\Xi^*] = 1$ and $e(\alpha) \neq 1$.

It remains to construct a $\mathbf{4}\mathsf{Pr}^{Ł_\triangle}$ model $\mathbb{M}$ falsifying $\Xi^* \models_{\mathbf{4}\mathsf{Pr}^{Ł_\triangle}} \alpha$ using e. We proceed as follows. First, we set $W = 2^{\mathsf{Lit}[\Xi^* \cup \{\alpha\}]}$, and for every $w \in W$ define $w \in v^+(p)$ iff $p \in w$ and $w \in v^-(p)$ iff $\neg p \in w$. We extend the valuations to $\phi \in \mathcal{L}_{\mathsf{BD}}$ in the usual manner. Then, for $\mathsf{X}\phi \in \mathtt{Sf}[\Xi^* \cup \{\alpha\}]$, we set $\mu_{\mathbf{4}}(|\phi|^{\mathsf{x}}) = e(\mathsf{X}\phi)$ according to modality X.

It remains to extend $\mu_{\mathbf{4}}$ to the whole 2^W. Observe, however, that any map from 2^W to $[0,1]$ that extends $\mu_{\mathbf{4}}$ is, in fact, a $\mathbf{4}$-probability. Indeed, all requirements from Definition 3 concern *only the extensions of formulas*. But the model is finite, BD is tabular, and Ξ^* contains all the necessary instances of probabilistic axioms and $e[\Xi^*] = 1$, whence all constraints on the formulas are satisfied.

Remark 6. Observe that we could use a *classical probability measure* in the proof of Theorem 3 because of [26, Theorem 5]. This, however, would require us to show that the extensions of formulas form a subalgebra of 2^W. On the other hand, it is simpler to use $\mathbf{4}$-probabilities instead of classical probabilities since we can immediately extend them to the full powerset from the extensions of formulas by Definition 3.

5 Decidability and Complexity

In the completeness proof, we reduced $\mathcal{H}4\mathsf{Pr}^{Ł_\triangle}$ proofs to $Ł_\triangle$ proofs. We know that validity and finitary entailment of $Ł_\triangle$ are coNP-complete (since $Ł$ is coNP-complete and $\triangle$ has truth-functional semantics).

Likewise, $\mathsf{Pr}^{Ł^2}_\triangle$ proofs are also reducible to $Ł^2$ proofs (cf. [7, Theorem 4.24]) from substitution instances of axioms $\mathsf{Pr}\phi \to \mathsf{Pr}\chi$ (for $\phi \models_{\mathsf{BD}} \chi$), $\neg\mathsf{Pr}\phi \leftrightarrow \mathsf{Pr}\neg\phi$, and $\mathsf{Pr}(\phi \vee \chi) \leftrightarrow (\mathsf{Pr}\phi \ominus \mathsf{Pr}(\phi \wedge \chi)) \oplus \mathsf{Pr}\chi$. Thus, it is clear that the validity and satisfiability of $\mathbf{4}\mathsf{Pr}^{\mathsf{Ł}_\triangle}$ and $\mathsf{Pr}^{\mathsf{Ł}^2}_\triangle$ are coNP-hard and NP-hard, respectively.

In this section, we provide a simple decision procedure for $\mathsf{Pr}^{\mathsf{Ł}^2}_\triangle$ and $\mathbf{4}\mathsf{Pr}^{\mathsf{Ł}_\triangle}$ and show that their satisfiability and validity are NP- and coNP-complete, respectively. Namely, we adapt constraint tableaux for $\mathsf{Ł}^2$ defined in [5] and expand them with rules for $\triangle$. We then adapt the NP-completeness proof $\mathsf{FP}(\mathsf{Ł})$ from [25] to establish our result.

Definition 13 (Constraint tableaux for $\mathsf{Ł}^2_\triangle$ — $\mathcal{T}\left(\mathsf{Ł}^2_\triangle\right)$). *Branches contain* labelled formulas *of the form $\phi \leqslant_1 i$, $\phi \leqslant_2 i$, $\phi \geqslant_1 i$, or $\phi \geqslant_2 i$, and* numerical constraints *of the form $i \leq j$ with $i, j \in [0,1]$.*

Each branch can be extended by an application of one of the rules below.

$$\neg{\leqslant_1}\frac{\neg\phi \leqslant_1 i}{\phi \leqslant_2 i} \quad \neg{\leqslant_2}\frac{\neg\phi \leqslant_2 i}{\phi \leqslant_1 i} \quad \neg{\geqslant_1}\frac{\neg\phi \geqslant_1 i}{\phi \geqslant_2 i} \quad \neg{\geqslant_2}\frac{\neg\phi \geqslant_2 i}{\phi \geqslant_1 i}$$

$$\sim{\leqslant_1}\frac{\sim\phi \leqslant_1 i}{\phi \geqslant_1 1-i} \quad \sim{\leqslant_2}\frac{\sim\phi \leqslant_2 i}{\phi \geqslant_2 1-i} \quad \sim{\geqslant_1}\frac{\sim\phi \geqslant_1 i}{\phi \leqslant_1 1-i} \quad \sim{\geqslant_2}\frac{\sim\phi \geqslant_2 i}{\phi \leqslant_2 1-i}$$

$$\triangle{\geqslant_1}\frac{\triangle\phi \geqslant_1 i}{i \leq 0 \left| \begin{array}{l} \phi \geqslant_1 j \\ j \geq 1 \end{array}\right.} \quad \triangle{\leqslant_1}\frac{\triangle\phi \leqslant_1 i}{i \geq 1 \left| \begin{array}{l} \phi \leqslant_1 j \\ j < 1 \end{array}\right.} \quad \triangle{\leqslant_2}\frac{\triangle\phi \leqslant_2 i}{i \geq 1 \left| \begin{array}{l} \phi \leqslant j \\ j \leq 0 \end{array}\right.} \quad \triangle{\geqslant_2}\frac{\triangle\phi \geqslant_2 i}{i \leq 0 \left| \begin{array}{l} \phi \geqslant j \\ j > 0 \end{array}\right.}$$

$$\rightarrow{\leqslant_1}\frac{\phi_1 \rightarrow \phi_2 \leqslant_1 i}{i \geq 1 \left| \begin{array}{c} \phi_1 \geqslant_1 1-i+j \\ \phi_2 \leqslant_1 j \\ j \leq i \end{array}\right.} \qquad \rightarrow{\leqslant_2}\frac{\phi_1 \rightarrow \phi_2 \leqslant_2 i}{\begin{array}{c} \phi_1 \geqslant_2 j \\ \phi_2 \leqslant_2 i+j \end{array}}$$

$$\rightarrow{\geqslant_1}\frac{\phi_1 \rightarrow \phi_2 \geqslant_1 i}{\begin{array}{c} \phi_1 \leqslant_1 1-i+j \\ \phi_2 \geqslant_1 j \end{array}} \qquad \rightarrow{\geqslant_2}\frac{\phi_1 \rightarrow \phi_2 \geqslant_2 i}{i \leq 0 \left| \begin{array}{c} \phi_1 \leqslant_2 j \\ \phi_2 \geqslant_2 i+j \\ j \leq 1-i \end{array}\right.}$$

Let i's be in $[0,1]$ and x's be variables ranging over the real interval $[0,1]$. We define the translation τ from labelled formulas to linear inequalities as follows:

$$\tau(\phi \leqslant_1 i) = x^L_\phi \leq i;\ \tau(\phi \geqslant_1 i) = x^L_\phi \geq i;\ \tau(\phi \leqslant_2 i) = x^R_\phi \leq i;\ \tau(\phi \geqslant_2 i) = x^R_\phi \geq i$$

Let $\bullet \in \{\leqslant_1, \geqslant_1\}$ and $\circ \in \{\leqslant_2, \geqslant_2\}$. A tableau branch

$$\mathcal{B} = \{\phi_1 \circ i_1, \ldots, \phi_m \circ i_m, \phi'_1 \bullet j_1, \ldots, \phi'_n \bullet j_n, k_1 \leq l_1, \ldots, k_q \leq l_q\}$$

is closed *if the system of inequalities*

$$\tau(\phi_1 \circ i_1), \ldots, \tau(\phi_m \circ i_m), \tau(\phi'_1 \bullet j_1), \ldots, \tau(\phi'_n \bullet j_n), k_1 \leq l_1, \ldots, k_q \leq l_q$$

does not have solutions. Otherwise, $\mathcal{B}$ is open*. A tableau is* closed *if all its branches are closed. ϕ has a $\mathcal{T}\left(\mathsf{Ł}^2_\triangle\right)$* proof *if the tableau beginning with $\{\phi \leqslant_1 c, c < 1\}$ is closed.*

Observe that the $\rightarrow$ and $\sim$ rules for $\leqslant_1$ coincide with the analoguous rules in the constraint tableaux for Ł as given in [20,21,23]. Thus, we can use the calculus both for $\mathbf{4}\mathsf{Pr}^{Ł_\triangle}$ and $\mathsf{Pr}^{Ł^2}_\triangle$. Note also that we need to build only one tableau for $\mathcal{L}_{\mathsf{Pr}^{Ł^2}_\triangle}$ formulas because of Lemma 1.

The next statement can be proved in the same manner as [5, Theorem 1].

Theorem 4 (Completeness of tableaux).

1. *ϕ is $Ł_\triangle$ valid iff it has a $\mathcal{T}\left(Ł^2_\triangle\right)$ proof.*
2. *ϕ is $Ł^2_\triangle$ valid iff it has a $\mathcal{T}\left(Ł^2_\triangle\right)$ proof.*

As we have already mentioned in the beginning of the section, the proof of NP-completeness is an adaptation of a similar proof from [25]. This, in turn, uses the reduction of Łukasiewicz formulas to bounded mixed-integer problems (bMIPs) as given in [20–22]. To make the paper self-contained, we state the required definitions and results here.

Definition 14 (Mixed-integer problem). *Let $\mathbf{x} = (x_1, \ldots, x_k) \in \mathbb{R}^k$ and $\mathbf{y} = (y_1, \ldots, y_m) \in \mathbb{Z}^m$ be variables, A and B be integer matrices and h an integer vector, and $f(\mathbf{x}, \mathbf{y})$ be a $k+m$-place linear function.*

1. A general MIP *is to find* $\mathbf{x}$ *and* $\mathbf{y}$ *s.t.* $f(\mathbf{x}, \mathbf{y}) = \min\{f(\mathbf{x}, \mathbf{y}) : A\mathbf{x} + B\mathbf{y} \geq h\}$.
2. *In* a bounded MIP (bMIP)*, all solutions should belong to* $[0, 1]$.

Proposition 2. *Bounded MIP is* NP*-complete.*

Theorem 5. *Satisfiability of* $\mathsf{Pr}^{Ł^2}_\triangle$ *and* $\mathbf{4}\mathsf{Pr}^{Ł_\triangle}$ *is* NP*-complete.*

Proof. Recall that $\mathsf{Pr}^{Ł^2}_\triangle$ and $\mathbf{4}\mathsf{Pr}^{Ł_\triangle}$ can be linearly embedded into one another (Theorems 1 and 2). Thus, it remains to provide a non-deterministic polynomial algorithm for one of these logics. We choose $\mathsf{Pr}^{Ł^2}_\triangle$ since it has only one modality.

Let $\alpha \in \mathcal{L}_{\mathsf{Pr}^{Ł^2}_\triangle}$. We can w.l.o.g. assume that $\neg$ occurs only in modal atoms and that in every modal atom $\mathsf{Pr}\phi_i$, ϕ_i is in negation normal form. Define α^* to be the result of the substitution of every $\neg p$ occurring in α with a new variable p^*. It is easy to check that α is satisfiable iff α^* is. We construct a satisfying valuation for α^*.

First, we replace every modal atom $\mathsf{Pr}\phi_i$ with a fresh variable q_{ϕ_i}. Denote the new formula $(\alpha^*)^-$. It is clear that the size of $(\alpha^*)^-$ ($|(\alpha^*)^-|$) is only linearly greater than $|\alpha|$. We construct a tableau beginning with $\{(\alpha^*)^- \geqslant_1 c, c \geq 1\}$. Every branch gives us an instance of a bMIP equivalent to the Ł-satisfiability of $(\alpha^*)^-$: $(\alpha^*)^-$ is satisfiable iff at least one instance of a bMIP has a solution.

Now, write z_i for the values of q_{ϕ_i}'s in $(\alpha^*)^-$. Our instance of a bMIP also has additional variables x_j ranging over $[0, 1]$ as well as equalities $k = 1$ and $k' = 0$ obtained from entries $k \geq 1$ and $k' \leq 0$. It is clear that both the number of (in)equalities l_1 and the number of variables l_2 in the MIP are linear w.r.t. $|(\alpha^*)^-|$. Denote this instance MIP(1).

We need to show that z_i's are coherent as probabilities of ϕ_i's (here, $i \leq n$ indexes the modal atoms of $(\alpha^*)^-$). We introduce 2^n variables u_v indexed by n-letter words over $\{0, 1\}$ and denoting whether the variables of ϕ_i's are true under v^+.[3] We let $a_{i,v} = 1$ when ϕ_i is true under v^+ and $a_{i,v} = 0$ otherwise. Now add new equalities denoted with MIP(2 exp) to MIP(1), namely, $\sum_v u_v = 1$ and $\sum_v (a_{i,v} \cdot u_v) = z_i$. It is clear that the new MIP (MIP(1) ∪ MIP(2 exp)) has a non-negative solution iff its corresponding branch is open. Furthermore, although there are l_2+2^n+n variables in MIP(1)∪MIP(2 exp), it has no more than l_1+n+1 (in)equalities. Thus by [18, Lemma 2.5], it has a non-negative solution with at most $l_1 + n + 1$ non-zero entries. We guess a list L of at most $l_1 + n + 1$ words v (its size is $n \cdot (l_1 + n + 1)$). We can now compute the values of $a_{i,v}$'s for $i \leq n$ and $v \in L$ and obtain a new MIP which we denote MIP(2poly): $\sum_{v \in L} u_v = 1$ and $\sum_{v \in L}(a_{i,v} \cdot u_v) = z_i$. It is clear that MIP(1) ∪ MIP(2poly) is of polynomial size w.r.t. $|\alpha|$ and has a non-negative solution iff the corresponding branch of the tableau is open. Thus, we can solve it in non-deterministic polynomial time as required.

6 Conclusion

We presented logic $\mathsf{4Pr}^{\mathsf{Ł}_\triangle}$ formalising four-valued probabilities proposed in [26] using a two-layered expansion of Łukasiewicz logic with $\triangle$. We established faithful embeddings between $\mathsf{4Pr}^{\mathsf{Ł}_\triangle}$ and $\mathsf{Pr}^{\mathsf{Ł}^2}_{\triangle}$, the logic of ±-probabilities [7]. Moreover, we constructed a sound and complete axiomatisation of $\mathsf{4Pr}^{\mathsf{Ł}_\triangle}$ and proved its decidability using constraint tableaux for $\mathsf{Ł}_\triangle$.

Several questions remain open. In [7], we presented two-layered logics for reasoning with belief and plausibility functions. These logics employ a 'two-valued' interpretation of belief and plausibility (i.e., ϕ has two belief assignments: for ϕ and for $\neg\phi$). It would be instructive to axiomatise 'four-valued' belief and plausibility functions and formalise reasoning with those via a two-layered logic.

Moreover, we have been considering logics whose inner layer lacks implication. It is, however, reasonable to assume that an agent can assign certainty to conditional statements. Furthermore, there are expansions of BD with truth-functional implications (cf. [28] for examples). A natural next step now is to axiomatise paraconsistent probabilities defined over a logic with an implication.

References

1. Baaz, M.: Infinite-valued Gödel logics with 0-1-projections and relativizations. In: Gödel'96: Logical Foundations of Mathematics, Computer Science and Physics–Kurt Gödel's Legacy, Brno, Czech Republic, August 1996, Proceedings, pp. 23–33. Association for Symbolic Logic (1996)

[3] Note that $\neg$ does not occur in $(\alpha^*)^-$ and thus we care only about e_1 and v^+. Furthermore, while n is the number of ϕ_i's, we can add superfluous modal atoms or variables to make it also the number of variables.

2. Baldi, P., Cintula, P., Noguera, C.: Classical and fuzzy two-layered modal logics for uncertainty: translations and proof-theory. Int. J. Comput. Intell. Syst. **13**, 988–1001 (2020). https://doi.org/10.2991/ijcis.d.200703.001
3. Belnap, N.: A useful four-valued logic. In: Dunn, J.M., Epstein, G. (eds.) Modern Uses of Multiple-Valued Logic, pp. 5–37. Springer, Netherlands (1977). https://doi.org/10.1007/978-94-010-1161-7_2
4. Belnap, N.D.: How a computer should think. In: New Essays on Belnap-Dunn Logic. SL, vol. 418, pp. 35–53. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-31136-0_4
5. Bílková, M., Frittella, S., Kozhemiachenko, D.: Constraint tableaux for two-dimensional fuzzy logics. In: Das, A., Negri, S. (eds.) TABLEAUX 2021. LNCS (LNAI), vol. 12842, pp. 20–37. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86059-2_2
6. Bílková, M., Frittella, S., Kozhemiachenko, D., Majer, O.: Qualitative reasoning in a two-layered framework. Int. J. Approximate Reason. **154**, 84–108 (2023)
7. Bilkova, M., Frittella, S., Kozhemiachenko, D., Majer, O., Nazari, S.: Reasoning with belief functions over Belnap-Dunn logic. Ann. Pure Appl. Logic (2022). https://doi.org/10.1016/j.apal.2023.103338
8. Bílková, M., Frittella, S., Majer, O., Nazari, S.: Belief based on inconsistent information. In: Martins, M., Sedlár, I. (eds.) Dynamic Logic. New Trends and Applications. Lecture Notes in Computer Science, vol. 12569, pp. 68–86. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-65840-3_5
9. Bueno-Soler, J., Carnielli, W.: Paraconsistent probabilities: consistency, contradictions and Bayes' theorem. Entropy **18**(9), 325 (2016)
10. Běhounek, L., Cintula, P., Hájek, P.: Introduction to mathematical fuzzy logic. In: Cintula, P., Hájek, P., Noguera, C. (eds.) Handbook of Mathematical Fuzzy Logic, Studies in logic, vol. 37, pp. 1–102. College Publications (2011)
11. Dautović, Š, Doder, D., Ognjanović, Z.: An epistemic probabilistic logic with conditional probabilities. In: Faber, W., Friedrich, G., Gebser, M., Morak, M. (eds.) JELIA 2021. LNCS (LNAI), vol. 12678, pp. 279–293. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-75775-5_19
12. Delgrande, J., Renne, B.: The logic of qualitative probability. In: Twenty-Fourth International Joint Conference on Artificial Intelligence (2015)
13. Delgrande, J., Renne, B., Sack, J.: The logic of qualitative probability. Artif. Intell. **275**, 457–486 (2019)
14. Dubois, D.: On ignorance and contradiction considered as truth-values. Logic J. IGPL **16**(2), 195–216 (2008)
15. Dunn, J.M.: Intuitive semantics for first-degree entailments and 'coupled trees'. Philos. Stud. **29**(3), 149–168 (1976)
16. Dunn, J.: Contradictory information: too much of a good thing. J. Philos. Logic **39**, 425–452 (2010)
17. Fagin, R., Halpern, J.: Uncertainty, belief, and probability. Comput. Intell. **7**(3), 160–173 (1991)
18. Fagin, R., Halpern, J., Megiddo, N.: A logic for reasoning about probabilities. Inf. Comput. **87**(1–2), 78–128 (1990)
19. Gärdenfors, P.: Qualitative probability as an intensional logic. J. Philos. Logic, 171–185 (1975)
20. Hähnle, R.: A new translation from deduction into integer programming. In: Calmet, J., Campbell, J.A. (eds.) AISMC 1992. LNCS, vol. 737, pp. 262–275. Springer, Heidelberg (1993). https://doi.org/10.1007/3-540-57322-4_18

21. Hähnle, R.: Many-valued logic and mixed integer programming. Ann. Math. Artif. Intell. **12**(3–4), 231–263 (1994)
22. Hähnle, R.: Tableaux for many-valued logics. In: D'Agostino, M., Gabbay, D., Hähnle, R., Posegga, J. (eds.) Handbook of Tableaux Methods, pp. 529–580. Springer, Dordrecht (1999). https://doi.org/10.1007/978-94-017-1754-0_9
23. Hähnle, R.: Advanced many-valued logics. In: Gabbay, D., Guenthner, F. (eds.) Handbook of Philosophical Logic, vol. 2, pp. 297–395. Springer, Netherlands (2001). https://doi.org/10.1007/978-94-017-0452-6_5
24. Hájek, P.: Metamathematics of Fuzzy Logic, Trends in Logic, vol. 4. Springer, Dordrecht (1998)
25. Hájek, P., Tulipani, S.: Complexity of fuzzy probability logics. Fundamenta Inf. **45**(3), 207–213 (2001)
26. Klein, D., Majer, O., Rafiee Rad, S.: Probabilities with gaps and gluts. J. Philos. Logic **50**(5), 1107–1141 (2021). https://doi.org/10.1007/s10992-021-09592-x
27. Metcalfe, G., Olivetti, N., Gabbay, D.: Proof Theory for Fuzzy Logics. Applied Logic Series, vol. 36, Springer, Cham (2008)
28. Omori, H., Wansing, H.: 40 years of FDE: an introductory overview. Stud. Logica. **105**(6), 1021–1049 (2017). https://doi.org/10.1007/s11225-017-9748-6
29. Rodrigues, A., Bueno-Soler, J., Carnielli, W.: Measuring evidence: a probabilistic approach to an extension of Belnap-Dunn logic. Synthese **198**(S22), 5451–5480 (2021)

An Axiom System for Basic Hybrid Logic with Propositional Quantifiers

Patrick Blackburn[1], Torben Braüner[2](✉), and Julie Lundbak Kofod[1]

[1] Department of Philosophy and Science Studies, Roskilde University, Roskilde, Denmark

[2] Department of People and Technology, Roskilde University, Roskilde, Denmark

torben@ruc.dk

Abstract. We present an axiom system for basic hybrid logic extended with propositional quantifiers (a second-order extension of basic hybrid logic) and prove its (basic and pure) strong completeness with respect to general models.

1 Introduction

We present an axiom system for basic hybrid logic augmented with propositional quantifiers—a second-order extension of basic hybrid logic—and prove its basic and pure completeness with respect to general models. A notable feature of our axiom system is that the universal instantiation rule for propositional quantification is restricted: variables can only be replaced by formulas that (a) are quantifier-free and (b) don't contain nominals in formula position.

Although this is primarily a technical paper, its roots are philosophical: it is part of an ongoing re-examination of the later work of Arthur Prior, a philosophical logician who is probably best known as the inventor of tense logic (see [10,15]). However Prior was also the founder of hybrid logic (see [4,8]) and he sometimes used propositional quantifiers to define what we now call nominals; these developments led Prior, shortly before his death in 1969, to explore such ideas as "quasi-modalities" and "egocentric logic".[1] We believe that the combination of contemporary hybrid logic and propositional quantification explored in this paper is a promising setting for better understanding Prior's later work.

We proceed as follows. In Sect. 2 we define the syntax of our languages, drawing special attention to what are called soft-QF formulas and soft-QF substitutions. In Sect. 3 we introduce general frames and models, give a Henkin-style satisfaction definition, and note some basic semantic lemmas. In Sect. 4 we define an axiom system and prove it sound, and then in Sect. 5 we prove its (basic and pure) strong completeness. Section 6 concludes and briefly discusses the links with Prior's later work.

[1] See, in particular, the technical papers in the new edition of his book *Papers on Time and Tense* [17], and the posthumous volume *Worlds, Times and Selves* [20].

H. H. Hansen et al. (Eds.): WoLLIC 2023, LNCS 13923, pp. 118–134, 2023.
https://doi.org/10.1007/978-3-031-39784-4_8

2 Syntax and Substitution

In this section we define basic hybrid logic with quantification over propositional variables, soft-QF formulas, and soft-QF substitutions. Basic hybrid logic is obtained by adding *nominals* and *satisfaction operators* to basic (propositional) modal logic. Nominals are usually written i, j, k; they are atomic symbols true at a unique world in any model. Nominals play two distinct syntactic roles. First, they can be used as atomic formulas, in exactly the same way as ordinary propositional variables p, q, and r can; because of the "true at a unique world" restriction on their interpretation, in this first role nominals can be thought of as 0-place world-naming modalities. Second, any nominal i can occur as a subscript to the symbol @. Any such $@_i$ is called a *satisfaction operator*, and for any formula φ, $@_i\varphi$ is true at a world w iff φ is true at the world that i names. Thus satisfaction operators are 1-place rigidifying modalities: they transform any proposition φ into a *rigid* proposition $@_i\varphi$, one that is either true at all worlds, or false at all worlds (depending on whether φ is true or false at the world that nominal i names). Any formula of the form $@_i\varphi$ is called a *satisfaction statement.*

We can now define what we mean by a basic hybrid language with propositional quantification. Let $PLET = \{c, b, a, \dots\}$ be a set of *propositional letters*, let $PVAR = \{p, q, r, \dots\}$ be a set of propositional variables, and let $NOM = \{i, j, k \dots\}$ be a set of *nominals.* We assume that all three sets are countable and pairwise disjoint, write $ATOM$ for $PLET \cup PVAR \cup NOM$ and call any element of $ATOM$ an *atomic symbol.* The basic hybrid language with propositional quantification $\mathcal{L}_{BHPQ}$ is built over $ATOM$ using the following grammar:

$$\varphi ::= \ c \mid p \mid i \mid \neg\varphi \mid \varphi \wedge \varphi \mid \Box\varphi \mid @_i\varphi \mid \forall p\varphi,$$

where $c \in PLET$, $p \in PVAR$, and $i \in NOM$. Other booleans are defined as expected, and $\Diamond$ and $\exists$ are defined by $\neg\Box\neg \equiv \Diamond$ and $\neg\forall p\neg \equiv \exists p$.

It is clear from this definition that nominals can occur in formulas in two ways: either as an atomic formula (that is: in *formula position*) or as part of a satisfaction operator (that is: in *operator position*). Similarly, any propositional variable p can occur in a formula in two ways: either as an atomic formula (that is: in *formula position*) or right after the symbol $\forall$ (that is: in *binding position*). But unlike occurrences of $@_i$ (which do not bind occurrences of i), occurrences of $\forall p$ bind all the (free) occurrences of p they have scope over. Propositional letters, on the other hand, cannot be bound; they occur only in formula position. So propositional letters are essentially "propositional constants" and we will use them in our Lindenbaum Lemma as Henkin-style witnesses for existential quantifiers. We define free and bound propositional variables in the standard way, and write $FV(\varphi)$ for the set of free propositional variables in formula φ. A *sentence* is a formula that contains no free propositional variables.

The result of substituting a formula ψ for a propositional variable q occurring in some formula φ, written $\varphi[\psi/q]$, is defined in the expected way. It is always possible to carry out a substitution safely (that is: without accidental binding) by relabelling the bound variables in φ.

In the Hilbert-style system presented in Sect. 4, the universal instantiation axiom has a side condition: only *quantifer free formulas with no nominals in formula position* can be used to instantiate universal quantifications. We call such formulas soft-QF formulas. That is, soft-QF formulas are built using the grammar

$$\varphi ::= \ c \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \Box\varphi \mid @_i\varphi,$$

where $c \in \mathit{PLET}$, $p \in \mathit{PVAR}$, and $i \in \mathit{NOM}$. A substitution $\varphi[\psi/q]$ is called a *soft-QF substitution* iff ψ is a soft-QF formula.

3 Semantics

We interpret $\mathcal{L}_{BHPQ}$ using a Henkin-style *general semantics*. That is, we shall use *general frames* and *general models* as our basic semantic structures: such structures restrict the domain over which the propositional quantifiers range. Writing $\mathcal{P}(W)$ for the powerset of the W, we define:

Definition 1 (Kripke frames, general frames, standard frames). *A* Kripke frame *is a pair $\mathcal{F} = \langle W, R\rangle$ where W is a non-empty set (of* worlds*) and R is a binary relation on W (the* accessibility relation *between worlds). A* general frame *is a pair $\mathcal{G} = \langle \mathcal{F}, \Pi\rangle$ where $\mathcal{F}$ is a Kripke frame and Π is a non-empty subset of $\mathcal{P}(W)$ that is closed under the following three conditions:*

- *relative complement: if $X \in \Pi$, then $W - X \in \Pi$*
- *intersection: if $X, Y \in \Pi$, then $X \cap Y \in \Pi$*
- *modal projection: if $X \in \Pi$ then $\{w \in W \mid \forall v(wRv \rightarrow v \in X)\} \in \Pi$.*

We call a subset Π of $\mathcal{P}(W)$ that satisfies these conditions a closed set of admissible propositions*, and we call the elements of Π* admissible propositions*. A general frame is called a* standard frame *iff $\Pi = \mathcal{P}(W)$.*

We are interested in general frames rather than standard frames because this paper is devoted to completeness results; using only standard frames typically leads to logics that are not axiomatisable. For example, if there are no propositional quantifiers in the language, the basic modal logic **K** (the set of basic modal formulas valid on all frames) is decidable in PSPACE; but if we add propositional quantifiers—and interpret them using only *standard* frames—the resulting set of validities is not even recursively enumerable.[2] Modal languages with propositional quantifiers may look simple, but (interpreted over standard frames) they are powerful second-order systems.

Leon Henkin [12] introduced a way of "taming" higher-order logics. The underlying idea is to increase the number of models, thereby reducing the number

[2] This negative result (and others) were proved in Kit Fine's pioneering 1970 paper [11] (though the paper also contains an interesting positive result: extending ordinary **S5** with propositional quantification using standard frames yields a decidable logic). For some sharper negative results see [13].

of validities—hopefully to the point where the set of validities becomes recursively enumerable. General frames can be viewed as a (successful) Henkin-style attempt to tame modal semantics: a general frame is just a Kripke frame $\langle W, R\rangle$ together with a selection of propositions Π, that is, subsets of W.[3] We don't insist that all subsets of W belong to Π; we simply insist that Π has enough structure to behave like a set of propositions. In particular, in any general frame, the set of admissible propositions should be closed under the operations corresponding to the boolean operators and $\Box$. So there are a lot more general frames than frames—a single Kripke frame gives rise to multiple general frames — and it turns out that this expansion successfully "tames" the set of validities.

The success of general frames in the setting of ordinary modal logic leads to the questions that drive this paper. What happens if we add propositional quantification to basic *hybrid* logic rather than just basic modal logic? In particular: what is the logic of general frames if our base language contains not only booleans and boxes, but also nominals and satisfaction operators? Moreover, do general frames allow us to "tame" not merely the basic logic, but also what hybrid logicians call its pure extensions?

But this is jumping ahead. We must first answer a more basic question: how do we interpret $\mathcal{L}_{BHPQ}$ on general frames? The interpretation of the nominals will be taken care of by a naming function (or *nomination*) N assigning a world to each nominal, while the interpretation of the propositional letters will be given by a modal valuation function V.

Definition 2 (General models and standard models). *A* general model *$\mathcal{M}$* based on *a general frame $\langle W, R, \Pi\rangle$ is a tuple $\langle W, R, \Pi, N, V\rangle$ where $N : NOM \to W$ and $V : PLET \to \Pi$. A* standard model *is a model based on a standard frame.*

This definition builds in our central semantic design decision for $\mathcal{L}_{BHPQ}$: *the interpretation of the nominals is independent of the choice of Π.* Nominals directly "tag" arbitrary worlds via the nomination. This is important, because Π may not contain all—or even any—singleton subsets of W as admissible propositions. Our nominals ignore Π. They are hardwired to the underlying Kripke frame.

Definition 3 (Variable assignments and variants). *A* variable assignment *on a general frame $\mathcal{G} = \langle W, R, \Pi\rangle$ is a function $g : PVAR \to \Pi$. For any propositional variable p, we say that a variable assignment g' is a p-*variant *of variable assignment g iff for all propositional variables $q \neq p$, we have $g'(q) = g(q)$. We write this as $g' \sim_p g$.*

[3] There is more to general frames that this: they can also be viewed as representations of modal algebras; see Chap. 5 of [6] for details. Both lines of work stem from a classic paper by S. K. Thomason [21]. This links general frames and modal algebras, and shows that (a) there are Priorean tense logics that are not complete with respect to any class of frames, that (b) every Priorean tense logic is complete with respect to a class of general frames. That is: general frames "tame" frame validity.

Now for a Henkin-style definition of satisfaction and truth:

Definition 4 (Satisfaction and truth). *Let $\mathcal{M} = \langle W, R, \Pi, N, V \rangle$ be a general model, and g be a variable assignment on $\langle W, R, \Pi \rangle$. We define what it means for $\mathcal{M}$ to satisfy a formula at a world w with respect to assignment g as follows:*

- $\mathcal{M}, g, w \models i$ *iff* $w = N(i)$, *for any* $i \in NOM$
- $\mathcal{M}, g, w \models c$ *iff* $w \in V(c)$, *for any* $c \in PLET$
- $\mathcal{M}, g, w \models p$ *iff* $w \in g(p)$, *for any* $p \in PVAR$
- $\mathcal{M}, g, w \models \neg\varphi$ *iff* $\mathcal{M}, g, w \not\models \varphi$
- $\mathcal{M}, g, w \models \varphi \wedge \psi$ *iff* $\mathcal{M}, g, w \models \varphi$ *and* $\mathcal{M}, g, w \models \psi$
- $\mathcal{M}, g, w \models \Box\varphi$ *iff for all* $v \in W$, *if* wRv *then* $\mathcal{M}, g, v \models \varphi$
- $\mathcal{M}, g, w \models @_i\varphi$ *iff* $\mathcal{M}, g, N(i) \models \varphi$
- $\mathcal{M}, g, w \models \forall p\varphi$ *iff for all* $g' \sim_p g$, *we have* $\mathcal{M}, g', w \models \varphi$.

A formula φ is true *at a world w in $\mathcal{M}$ iff for all variable assignments g, $\mathcal{M}, g, w \models \varphi$, and we write this as $\mathcal{M}, w \models \varphi$.*

Definition 5 (Validity and consequence). *A formula φ is* valid in *a general model $\mathcal{M}$ iff it is true at all worlds in $\mathcal{M}$; we write this as $\mathcal{M} \models \varphi$. A formula φ is* valid *iff it is valid in all general models; and we write this as $\models \varphi$.*

A formula φ is a consequence of a set of formulas Γ, written $\Gamma \models \varphi$, iff for all general models $\mathcal{M}$, all assignments g on $\mathcal{M}$, and all worlds w in $\mathcal{M}$, if $\mathcal{M}, g, w \models \Gamma$ then $\mathcal{M}, g, w \models \varphi$. Here $\mathcal{M}, g, w \models \Gamma$ means: for all formulas $\gamma \in \Gamma$, $\mathcal{M}, g, w \models \gamma$. Note: φ is valid iff $\emptyset \models \varphi$.

We could also have defined notions of *standard validity* and *standard consequence*; these are defined exactly as above but with "standard model(s)" replacing "general model(s)". But, as discussed earlier, for the purposes of the present paper standard structures are of little interest. Completeness results are rare when working with standard structures, but by working with general models we will be able to prove Henkin-style completeness results that cover both the basic logic and all its pure extensions (we will explain this terminology later).

The following semantic lemmas will be used in our soundness and completeness proofs. We start with the Agreement Lemma. This tells us that to ensure that nominations, valuations, and assignments agree on whether φ is satisfied, it suffices that they agree on the atomic symbols actually occurring in φ

Lemma 1 (Agreement Lemma). *Let φ be a formula, and let both $\mathcal{M} = \langle W, R, \Pi, N, V \rangle$ and $\mathcal{M}^* = \langle W, R, \Pi, N^*, V^* \rangle$ be general models based on $\langle W, R, \Pi \rangle$ such that:*

i) $V(c) = V^*(c)$ *for all propositional letters* c *occurring in* φ, *and*
ii) $N(i) = N^*(i)$ *for all nominals* i *occurring in* φ.

Furthermore, let g and h be variable assignments on $\langle W, R, \Pi \rangle$ such that $g(q) = h(q)$ for all the free propositional variables q occurring in φ. Then for all $w \in W$, we have that $\mathcal{M}, g, w \models \varphi$ iff $\mathcal{M}^*, h, w \models \varphi$.

Proof. By induction of the number of propositional connectives in φ. $\square$

A standard corollary follows: the variable assignment is irrelevant when evaluating sentences, so for sentences φ can write $\mathcal{M}, w \models \varphi$ instead of $\mathcal{M}, g, w \models \varphi$.

Definition 6. *Let $\mathcal{M} = \langle W, R, \Pi, N, V \rangle$ be a general model, and g an assignment on $\langle W, R, \Pi \rangle$. Then for all formulas φ we define*

$$[\mathcal{M}, g]_\varphi = \{w \in W \mid \mathcal{M}, g, w \models \varphi\}.$$

For φ a sentence we can just write $[\mathcal{M}]_\varphi$, as g is irrelevant.

Next we see that all soft-QF formulas pick out admissible propositions.

Lemma 2. *Let $\mathcal{M} = \langle W, R, \Pi, N, V \rangle$ be a general model, and g be any assignment on $\langle W, R, \Pi \rangle$. Then for all soft-QF formulas φ, we have $[\mathcal{M}, g]_\varphi \in \Pi$.*

Proof. By induction on the number of connectives in soft-QF formulas. All propositional letters are assigned elements of Π by V, and any assignment g on $\mathcal{M}$ assigns all propositional variables an element of Π, which establishes the base case. The inductive steps for $\neg\varphi$, $\varphi \wedge \psi$, and $\Box\varphi$, follow from the three closure conditions on Π. As for the $@_i\varphi$ step, note that any such formula is either true at all worlds, or false at all worlds, that is any such formula picks out either the proposition W or $\emptyset$. But these two propositions are always admissible: as Π is non-empty, it contains at least one proposition X. But then $\emptyset = X \cap (W - X)$ and $W = W - \emptyset$ are both in Π $\square$

Our next lemma tells us that the set of all propositions picked out by soft-QF formulas is a subset of Π that is a closed collection of admissible propositions. Let us write $[\mathcal{M}, g]^{sqf}$ for $\{[\mathcal{M}, g]_\varphi : \varphi \textit{ is a soft-QF formula}\}$. Then:

Lemma 3. *Given any general model $\mathcal{M} = \langle W, R, \Pi, N, V \rangle$ and an assignment g on $\mathcal{M}$:*

1. *$[\mathcal{M}, g]^{sqf} \subseteq \Pi$, and*
2. *$[\mathcal{M}, g]^{sqf}$ is a closed set of admissible propositions.*

Proof. Item 1 follows from Lemma 2. Item 2 holds because the three closure conditions conditions correspond to the connectives $\neg$, $\wedge$ and $\Box$. Argue as follows:
Relative complement: Consider $[\mathcal{M}, g]_\varphi$ for some soft-QF formula φ. It is sufficient to show that $\Pi - [\mathcal{M}, g]_\varphi = [\mathcal{M}, g]_{\neg\varphi}$. But $w \in [\mathcal{M}, g]_{\neg\varphi}$ iff $\mathcal{M}, g, w \models \neg\varphi$ iff $\mathcal{M}, g, w \not\models \varphi$ iff $w \in \Pi - [\mathcal{M}, g]_\varphi$.
Intersection: Similar to the previous case.
Modal projection: We show that $\{w \in W \mid \forall v(wRv \rightarrow v \in [\mathcal{M}, g]_\varphi)\} = [\mathcal{M}, g]_{\Box\varphi}$ for any soft-QF formula φ. But $u \in [\mathcal{M}, g]_{\Box\varphi}$ iff $\mathcal{M}, g, u \models \Box\varphi$ iff $\forall v(uRv \rightarrow v \in [\mathcal{M}, g]_\varphi)$ iff $u \in \{w \in W \mid \forall v(wRv \rightarrow v \in [\mathcal{M}, g]_\varphi)\}$. $\square$

Now for the Substitution Lemma; note the restriction to soft-QF formulas.

Lemma 4 (Substitution Lemma). *Let $\mathcal{M} = \langle W, R, \Pi, N, V\rangle$ be a general model and let g be a variable assignment on $\langle W, R, \Pi\rangle$. Then for any safe substitution $\varphi[\psi/p]$, where ψ is a soft-QF formula, we have that:*

$$\mathcal{M}, g, w \models \varphi[\psi/p] \textit{ iff } \mathcal{M}, g', w \models \varphi,$$

where $g' \sim_p g$ is defined by setting $g'(p) = [\mathcal{M}, g]_\psi$.

Proof. First note that g' is well-defined since $[\mathcal{M}, g]_\psi \in \Pi$ by the previous lemma. The proof is by induction on the number of connectives in φ. The interesting case is $\varphi = \forall q\theta$. We have three subcases:
If $p = q$, then the result follows from the Agreement Lemma.
If $p \neq q$, but p does not occur free in θ, then the result again follows from the Agreement Lemma.
Finally there is the case where $p \neq q$ and p occurs free in θ. Then $\mathcal{M}, g, w \models (\forall q\theta)[\psi/p]$ iff $\mathcal{M}, g, w \models \forall q(\theta[\psi/p])$ iff for all $g'' \sim_q g$, we have $\mathcal{M}, g'', w \models \theta[\psi/p]$. But by the induction hypothesis this is equivalent to for all $g'' \sim_q g$, we have $\mathcal{M}, g''', w \models \theta$ where $g''' \sim_p g''$ is defined by setting $g'''(p) = [\mathcal{M}, g'']_\psi$. But giving a g''' such that $g''' \sim_p g''$ where $g'''(p) = [\mathcal{M}, g'']_\psi$ and $g'' \sim_q g$ is equivalent to giving a g''' such that $g''' \sim_q g'$ where $g' \sim_p g$ is defined by setting $g'(p) = [\mathcal{M}, g'']_\psi$. But q cannot occur free in ψ, so $[\mathcal{M}, g'']_\psi = [\mathcal{M}, g]_\psi$ by the Agreement Lemma. The result follows from $\mathcal{M}, g', w \models \forall q\theta$ being equivalent to for all $g''' \sim_q g'$, so we have $\mathcal{M}, g''', w \models \theta$. □

4 The Axiomatisation

Our axiomatisation is called $\mathbf{K}_{SQph}$+RULES. It is an extension of the $\mathbf{K}_h$+RULES axiomatisation for basic propositional hybrid logic presented in Chapter 7 Section 3 of [6]. We first present the two components of $\mathbf{K}_h$+RULES axiomatisation, and then add on what we need to handle propositional quantification.

Definition 7 (The $\boldsymbol{K}_h$ axiom system). *The $\boldsymbol{K}_h$ axiom system contains as axioms all propositional tautologies, and all instances of K for the modalities:*

$K_\Box$: $\Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi)$
$K_@$: $@_i(\varphi \to \psi) \to (@_i\varphi \to @_i\psi)$

It also contains all instances of the following interaction schemas:

Intro: $i \wedge \varphi \to @_i\varphi$
Agree: $@_j @_i\varphi \leftrightarrow @_i\varphi$
Back: $\Diamond @_i\varphi \to @_i\varphi$
Sdual: $@_i\varphi \leftrightarrow \neg @_i\neg\varphi$,

and in addition, all instances of the modal equality schemas:

Ref: $@_i i$
Sym: $@_i j \leftrightarrow @_j i$
Nom: $@_i j \wedge @_j p \to @_i\varphi$.

The proof rules of $\boldsymbol{K}_h$ are:

MP: If $\vdash \varphi \to \psi$ *and* $\vdash \varphi$ *then* $\vdash \psi$
Gen$_\Box$*: If* $\vdash \varphi$ *then* $\vdash \Box\varphi$
Gen$_@$*: If* $\vdash \varphi$ *then* $\vdash @_i\varphi$

$\mathbf{K}_h$ proofs are Hilbert-style proofs and it is fairly straightforward to adapt the usual modal machinery of canonical models and prove that $\mathbf{K}_h$ is a (sound and) strongly complete axiom system for minimal propositional hybrid logic (see Chapter 7, Section 3 of [6] for details). The interaction axioms (together with the Gen$_@$ rule) capture the logic of the satisfaction operators: self-dual normal modal operators that interact smoothly with the other connectives. The axioms *Ref*, *Sym*, and *Nom* capture the logic of atomic satisfaction statements like $@_ij$; such statements are "modal equality assertions", modal equivalents of first-order atomic formulas of the form $i = j$. Clearly *Ref* and *Sym* express the reflexivity and symmetry of identity. The *Nom* axiom is more interesting. It can be read as a Leibniz-style identity axiom: "if i and j are identical, and i has property φ, then j has property φ too". But also note an important special case: if φ is k this becomes $@_ij \wedge @_jk \to @_ik$, which expresses the transitivity of identity.

Here are two schemas that are used in the completeness proof:

Elim: $i \wedge @_i\varphi \to \varphi$
Bridge: $\Diamond i \wedge @_i\varphi \to \Diamond\varphi$

Note that *Elim* is a contraposed form of the *Intro* axiom (using the *Sdual* axiom). As for *Bridge*, here are the main steps of a $\mathbf{K}_h$ proof of it:

1) $\Diamond i \wedge \Box\varphi \to \Diamond(i \wedge \varphi)$	*Modal validity*
2) $i \wedge \varphi \to @_i\varphi$	*Intro axiom*
3) $\Box(i \wedge \varphi \to @_i\varphi)$	Gen$_@$ *on 2*
4) $\Box(i \wedge \varphi \to @_i\varphi) \to (\Diamond(i \wedge \varphi) \to \Diamond @_i\varphi)$	*Modal validity*
5) $\Diamond(i \wedge \varphi) \to \Diamond @_i\varphi$	*3,4 Modus ponens*
6) $\Diamond i \wedge \Box\varphi \to \Diamond @_i\varphi$	*1,5 Propositional logic*
7) $\Diamond @_i\varphi \to @_i\varphi$	*Back axiom*
8) $\Diamond i \wedge \Box\varphi \to @_i\varphi$	*6,7 Propositional logic*
9) $\Diamond i \wedge @_i\varphi \to \Diamond\varphi$	*8 Contraposition, Sdual axiom*

Nonetheless, despite the fact that $\mathbf{K}_h$ is complete with respect to the class of all Kripke models (that is, it is the "minimal hybrid logic"), it is more usual to work with more powerful proof systems such as $\mathbf{K}_h$+RULES.[4]

Definition 8 (The $\boldsymbol{K}_h$+RULES axiom system). *The $\boldsymbol{K}_h$+RULES axiom system contains all the axioms and rules of $\boldsymbol{K}_h$ plus the following two proof rules:*

$$Name: \frac{\vdash j \to \theta}{\vdash \theta} \qquad Paste: \frac{\vdash @_i\Diamond j \wedge @_j\varphi \to \theta}{\vdash @_i\Diamond\varphi \to \theta}$$

In both rules, j is a nominal distinct from i that does not occur in φ or θ.

[4] Several such systems have been explored; see [1,5] for more. Here we follow [6].

As we shall see later, these two rules allow us to do some things that are *not* possible in $\mathbf{K}_h$—things that will become important when we look at the pure extensions of $\mathbf{K}_h$. Anticipating this, we shall define $\mathbf{K}_{SQph}$+RULES, our basic axiomatisation for minimal propositional hybrid logic with propositional quantification, on top of $\mathbf{K}_h$+RULES.

Definition 9 (The $\boldsymbol{K}_{SQph}$+RULES axiom system). *The* $\boldsymbol{K}_{SQph}$+RULES *axiom system contains all the axioms and rules of* $\boldsymbol{K}_h$+RULES. *It also contains the following axioms:*

Q1: $\forall p(\varphi \to \psi) \to (\varphi \to \forall p\psi)$, *where* φ *contains no free occurrences of* p
Q2-sqf: $\forall p\varphi \to \varphi[\psi/p]$, *where* $\varphi[\psi/p]$ *is a soft-QF substitution*
Barcan$_@$*:* $\forall p @_i\varphi \leftrightarrow @_i\forall p\varphi$,

and one more proof rule:

Gen$_\forall$*: If* $\vdash \varphi$ *then* $\vdash \forall p\varphi$.

While *Q1* is familiar from modal and classical logic, and *Barcan*$_@$ from hybrid logic, the side condition on *Q2-sqf* deserves comment We cannot substitute nominals (as not all admissible sets of propositions contain singleton subsets) nor can we substitute quantified formulas either (our Substitution Lemma does not cover such substitutions, since the admissible sets are only required to be closed under finite intersections, not arbitrary intersections). Allowing only soft-QF substitutions ensures soundness. [5]

Definition 10 (Provability and consistency in $\mathbf{K}_{SQph}$+RULES). *A formula* φ *is* $\mathbf{K}_{SQph}$ + RULES-provable *iff there is a* $\mathbf{K}_{SQph}$ + RULES *Hilbert-style proof of* φ*; we write* $\vdash \varphi$ *for provability and* $\nvdash \varphi$ *for unprovability. A formula* φ *is* $\mathbf{K}_{SQph}$ + RULES-provable from a set of formulas Σ *iff for some conjunction* σ *of formulas from* Σ *we have* $\vdash \sigma \to \varphi$. *A formula* φ *is* $\mathbf{K}_{SQph}$ + RULES-consistent *iff* $\nvdash \neg\varphi$. *A set of formulas* Σ *is* $\mathbf{K}_{SQph}$ + RULES-consistent *iff there is no conjunction* σ *of formulas from* Σ *such that* $\vdash \neg\sigma$.

Theorem 1. $\boldsymbol{K}_{SQph}$+RULES *is sound with respect to general frames.*

Proof. To prove this we need to show that (a) all the axioms are valid on all general frames, and (b) that the proof rules preserve validity. This is known for all the axioms and rules in K_h+RULES, so we only need to check that the *Q1* and *Q2-sqf* are valid and that *Gen*$_\forall$ preserves validity. That *Gen*$_\forall$ preserves validity

[5] Note that on *standard* models we could drop the restriction prohibiting substitution of quantified formulas as standard models admit all subsets of the frame as propositions. That is, the rule which permits any soft substitution is sound on standard models. However this does not lead to a completeness result for standard models, as (thanks to Kit Fine's results [11]) we know that the set of all standard validities on the class of all standard models is not recursively enumerable.

is more-or-less immediate. The axioms are also easy to handle; we present the argument for *Q2-sqf* and leave *Q1* and $Barcan_{@}$ for the reader.

So: choose an arbitrary general model $\mathcal{M} = \langle W, R, \Pi, N, V \rangle$, let $w \in W$, and let g be a variable assignment on $\langle W, R, \Pi \rangle$. Then to show that $\forall p\varphi \rightarrow \varphi[\psi/p]$ is valid, where $\varphi[\psi/p]$ is a soft-QF substitution, suppose that $\mathcal{M}, g, w \models \forall p\varphi$. This is equivalent to: for all g' such that $g' \sim_p g$, we have $\mathcal{M}, g', w \models \varphi$. Define $g'' \sim_p g$ by setting $g''(p) = [\mathcal{M}, g]_\psi$; Lemma 2 tells us that g'' is a well-defined variant of g because ψ is a soft-QF formula. Hence $\mathcal{M}, g'', w \models \varphi$ and so, using the Substitution Lemma, $\mathcal{M}, g, w \models \varphi[\psi/p]$.

Lemma 5. *If* $\vdash \sigma \rightarrow \theta[c/p]$ *and* c *does not occur in* θ *or* σ*, then* $\vdash \sigma \rightarrow \forall q\theta[q/p]$*, where* q *is any variable not occurring in* θ *or* σ*.*

Proof. Left to the reader. □

Lemma 6. *Suppose that* q *can be safely substituted for* p *in* φ *and that* φ *has no free occurrences of* q*. Then* $\vdash \forall p\varphi \leftrightarrow \forall q\varphi[q/p]$*.*

Proof. Left to the reader. □

One final remark. We formulated $Barcan_{@}$ in its $\forall$-form, that is, in the form $\forall p @_i\varphi \leftrightarrow @_i\forall p\varphi$. Its $\exists$-form is $@_i\exists p\varphi \leftrightarrow \exists p @_i\varphi$. Strictly speaking, the left-to-right arrows of both forms are Barcan formulas, while the right-to-left directions of both are converse Barcan formulas.

5 Strong Completeness

We will now extend a standard model-building strategy used to prove the strong completeness of $\mathbf{K}_h$ + RULES (see Chap. 7, Sect. 3 of [6]) to show that every $\mathbf{K}_{SQph}$ + RULES-consistent set of formulas has a general model. The general model we shall construct has a number of special properties (described below) which will enable us to prove strong completeness not only for $\mathbf{K}_{SQph}$ + RULES itself, but for all of its pure extensions as well.

Definition 11 ($\mathbf{K}_{SQph}$ + RULES Maximal Consistent Sets). *Fix a language of* $\mathcal{L}_{BHPQ}$*. A set of formulas* Σ *in this language is a* $\mathbf{K}_{SQph}$ + RULES-MCS *iff it is* $\mathbf{K}_{SQph}$ + RULES-*consistent, and any proper extension (in the same language) is inconsistent.*

Lemma 7 (Named sets yielded by an MCS). *Let* Γ *be a* $\mathbf{K}_{SQph}$ + RULES-*MCS. For every nominal* i*, let* $\Delta_i = \{\varphi \mid @_i\varphi \in \Gamma\}$*. Then:*

1. *For every nominal* i*,* Δ_i *is a* $\mathbf{K}_{SQph}$ + RULES-MCS *that contains* i*.*
2. *For all nominals* i *and* j*, if* $i \in \Delta_j$*, then* $\Delta_j = \Delta_i$*.*
3. *For all nominals* i *and* j*,* $@_i\varphi \in \Delta_j$ *iff* $@_i\varphi \in \Gamma$*.*
4. *If a nominal* $k \in \Gamma$*, then* $\Gamma = \Delta_k$*.*

Proof. This is Lemma 7.24 of [6], and proof details can be found there. In fact, Lemma 7.24 shows that only $\mathbf{K}_h$ reasoning is needed to prove this lemma. □

The Δ_i in this lemma are called the *named sets yielded by* Γ. Clause 1 of the previous lemma tells us that each of these is an MCS containing at least one nominal; all of these MCSs are "hidden inside" the original $\mathbf{K}_h$-MCS. Clause 2 tells us that each nominal picks out a unique such MCS, and Clause 3 tells us that any satisfaction statement is either in all the Δ_j or in none of them. So every $\mathbf{K}_h$-MCS contains *almost* all that is required to build structures in which *every world is named by some nominal.* But not quite. For note that Clause 4 is only a conditional—we have no guarantee that Γ itself is one of the MCSs "hidden inside" Γ.

Indeed, this weakness in Clause 4 is the key reason for using $\mathbf{K}_h$+RULES instead of $\mathbf{K}_h$. The addition of the *Name* and *Paste* rules *does* allow us to guarantee that Γ itself is one of the MCSs "hidden inside" Γ. This will let us create a "named and pasted" MCS, which contains all the information required to build a frame in which each world is named by a nominal, which will prove crucial for the pure extensions completeness result. Furthermore, the axioms and rules for propositional quantification in $\mathbf{K}_{SQph}$ + RULES also ensure that this MCS is "witnessed", which allows us to define a suitable set of admissible propositions over its frame, thereby creating a general model. We first define what we mean by "named", "pasted" and "witnessed" and then prove the Lindenbaum-style lemma which will lead us to these goals.

Definition 12 (Named, pasted and witnessed MCSs). *Let* Σ *be a* $\mathbf{K}_{SQph}$ + RULES-*MCS. Then we say:*

- Σ *is* named *iff for some nominal* i, $i \in \Sigma$,
- Σ *is* pasted *iff for every formula of the form* $@_i\Diamond\varphi \in \Sigma$, *there is some nominal* j *such that* $@_i\Diamond j \in \Sigma$ *and* $@_j\varphi \in \Sigma$, *and*
- Σ *is* witnessed *iff for every formula of the form* $@_i\exists p\varphi$, *there is some propositional letter* c *such that* $@_i\varphi[c/p] \in \Sigma$.

To prove a Lindenbaum-style lemma for $\mathbf{K}_{SQph}$+RULES, we must extend the language. Suppose we start with language $\mathcal{L}$. We will add a countably infinite set of nominals (called *NewN*) and a countably infinite set of new propositional letters (called *NewL*), and call the extended language $\mathcal{L}'$. We will use *NewN* for naming and pasting and *NewL* for witnessing.

Lemma 8 (Lindenbaum). *Every* $\mathbf{K}_{SQph}$ + RULES-*consistent set of formulas in language* $\mathcal{L}$ *can be extended to a named, pasted and witnessed* $\mathbf{K}_{SQph}$+RULES-*MCS in language* $\mathcal{L}'$.

Proof. Given a consistent set of $\mathcal{L}$-formulas Σ, add *NewN* and *NewP* as just described to form $\mathcal{L}'$, and enumerate all three sets. Define Σ_k to be $\Sigma \cup \{k\}$, where k is the first nominal in *NewN*. Σ_k is consistent. For suppose not. Then for some conjunction of formulas θ from Σ, we have $\vdash k \to \neg\theta$. But k is a new nominal, so it does not occur in θ; hence, by the NAME rule we have $\vdash \neg\theta$. This contradicts the consistency of Σ, so Σ_k must be consistent.

Define Σ^0 to be Σ_k, and suppose we have defined Σ^m, where $m \geq 0$. Let φ_{m+1} be the $(m+1)$-th formula in our enumeration of $\mathcal{L}'$. We define Σ^{m+1} as follows:

If $\Sigma^{m+1} \cup \{\varphi_{m+1}\}$ is inconsistent, then $\Sigma^{m+1} = \Sigma^m$. Otherwise:

1. $\Sigma^{m+1} = \Sigma^m \cup \{\varphi_{m+1}\}$, if φ_{m+1} is not of the form $@_i\Diamond\varphi$ or $@_i\exists p\varphi$.
2. $\Sigma^{m+1} = \Sigma^m \cup \{\varphi_{m+1}\} \cup \{@_i\Diamond j \wedge @_j\varphi\}$, if φ_{m+1} is of the form $@_i\Diamond\varphi$. (Here j is the first nominal in the enumeration of *NewN* that does not occur in Σ^m or $@_i\Diamond\varphi$).
3. $\Sigma^{m+1} = \Sigma^m \cup \{\varphi_{m+1}\} \cup \{@_i\varphi[c/p]\}$, if φ_{m+1} is of the form $@_i\exists p\varphi$. (Here c is the first propositional letter in the enumeration of *NewL* that does not occur in Σ^m or $@_i\exists p\varphi$).

Let $\Sigma^* = \bigcup_{n\geq 0} \Sigma^n$. Clearly this set is named (as we added k in the first step), maximal (by construction), pasted and witnessed.

Furthermore, it is consistent, for the only aspects of the expansion that require checking are those given in by the second and third steps. Preservation of consistency by the second step is precisely what the PASTE rule guarantees. As for the third step, we argue as follows. Assume for the sake of contradiction that $\Sigma^m \cup \{@_i\exists p\varphi\} \cup \{@_i\varphi[c/p]\}$ is inconsistent. Then there are formulas $\sigma_1, ..., \sigma_n$ in Σ^m such that $\vdash \neg(\sigma_1 \wedge ... \wedge \sigma_n \wedge @_i\exists p\varphi \wedge @_i\varphi[c/p])$. Writing $\sigma_1 \wedge ... \wedge \sigma_n$ as σ, propositional logic yields $\vdash (\sigma \wedge @_i\exists p\varphi) \rightarrow \neg @_i\varphi[c/p]$. The conditions of Lemma 5 are fulfilled, so we have $\vdash (\sigma \wedge @_i\exists p\varphi) \rightarrow \forall q\neg @_i\varphi[q/p]$ where q is a new propositional variable. This is equivalent to $\vdash (\sigma \wedge @_i\exists p\varphi) \rightarrow \neg\exists q @_i\varphi[q/p]$, so using (the $\exists$-form of) *Barcan*$_@$ we have $\vdash (\sigma \wedge @_i\exists p\varphi) \rightarrow \neg @_i\exists q\varphi[q/p]$. Moreover, the conditions of Lemma 6 are fulfilled as well, so $\vdash @_i\exists p\varphi \leftrightarrow @_i\exists q\varphi[q/p]$, and we have that $\vdash (\sigma \wedge @_i\exists p\varphi) \rightarrow \neg @_i\exists p\varphi$. This contradicts the consistency of $\Sigma^m \cup \{@_i\exists p\varphi\}$, thus the third step must preserve consistency after all. □

Lemma 9 (Generated admissible sets). *Let $\mathcal{F} = \langle W, R\rangle$ be a Kripke frame, let $\mathcal{P}(W)$ be the powerset of W, let $B \subseteq \mathcal{P}(W)$ and define $\Pi(B)$, the* admissible set generated by B, *to be the smallest subset of $\mathcal{P}(W)$ containing B that is closed under relative complement, intersection, and modal projection. Then $\langle W, R, \Pi(B)\rangle$ is a general frame.*

Proof. Immediate by definition of $\Pi(B)$. □

Lemma 10. *Let $\mathcal{F} = \langle W, R\rangle$ be a Kripke frame, and let N be any nomination on $\mathcal{F}$. Let V be any mapping such that $V : PLET \rightarrow \mathcal{P}(W)$ and let g be any mapping such that $g : PVAR \rightarrow \mathcal{P}(W)$. Then $\mathcal{M} = \langle W, R, \Pi(im(V) \cup im(g)), N, V\rangle$ is a general model (here $im(V)$ and $im(g)$ are the images of V and g respectively).*

Proof. As $\langle W, R\rangle$ is a Kripke frame and $im(V) \cup im(g) \subseteq \mathcal{P}(W)$, by the previous lemma $\langle W, R, \Pi(im(V) \cup im(g))\rangle$ is a general frame. Hence V and g are mappings into $\Pi(im(V) \cup im(g))$, thus V is a valuation and g is an assignment on a general frame. Thus $\mathcal{M}$ is a general model. □

Definition 13 (Canonical named structures). *Let Γ be a named, pasted and witnessed $\mathbf{K}_{SQph}+$*RULES*-MCS. Let $\mathcal{M}^\Gamma$ be $\langle W^\Gamma, R^\Gamma, \Pi^\Gamma, N^\Gamma, V^\Gamma\rangle$ where:*

- *W^Γ is the set of all named sets yielded by Γ.*
- *R^Γ is the standard modal canonical relation between MCSs. That is, for any $u, v \in W^\Gamma$ we define $uR^\Gamma v$ iff for all formulas φ, $\varphi \in v$ implies $\Diamond\varphi \in u$. Or equivalently: $uR^\Gamma v$ iff for all formulas φ, $\Box\varphi \in u$ implies $\varphi \in v$.*
- *$N^\Gamma : NOM \to W^\Gamma$ is defined as follows. For any any nominal i, $N(i)$ is the unique $w \in W^\Gamma$ such that $i \in w$; that is, $N(i) = \Delta_i$.*
- *$V^\Gamma : PLET \to \mathcal{P}(W^\Gamma)$ is the standard modal canonical valuation (for proposition letters). That is, $V^\Gamma(c) = \{w \in W^\Gamma \mid c \in w\}$, for any proposition letter c.*
- *$g^\Gamma : PVAR \to \mathcal{P}(W^\Gamma)$ is the standard modal canonical valuation (for proposition variables). That is, $g^\Gamma(p) = \{w \in W^\Gamma \mid p \in w\}$, for any proposition variable p.*
- *Π^Γ is $\Pi(im(V^\Gamma) \cup im(g^\Gamma))$.*

We now check that this definition does indeed gives rise to Kripke frames and general models where every world is named by some nominal.

Lemma 11. *Let Γ be a named, pasted and witnessed $\mathbf{K}_{SQph}+$*RULES*-MCS, and let $\mathcal{M}^\Gamma$ be the* canonical named general model *yielded by Γ. Then $\langle W^\Gamma, R^\Gamma\rangle$ is a Kripke frame and $\mathcal{M}^\Gamma$ is a named general model.*

Proof. To see that $\langle W^\Gamma, R^\Gamma\rangle$ is a Kripke frame, first note that by Lemma 7 (1), for every nominal i, Δ_i is a $\mathbf{K}_{SQph}+$RULES-MCS containing i. As W^Γ is a non-empty set of MCSs, the standard modal canonical relation R^Γ can be defined over it, thus $\langle W^\Gamma, R^\Gamma\rangle$ is a Kripke frame. Moreover N^Γ is a well-defined nomination, for Lemma 7 (2) guarantees that Δ_i is the unique element of W^Γ such that $i \in w$, and it clearly "names" every world in W^Γ. Finally, both V^Γ and g^Γ are well-defined, so we have all we need to apply Lemma 10 and conclude that $\mathcal{M}^\Gamma$ is a named general model. □

So from now on we will call $\langle W^\Gamma, R^\Gamma\rangle$ the *canonical named Kripke frame* yielded by Γ, and $\mathcal{M}^\Gamma$ the *canonical named general model* yielded by Γ. We now examine them more closely. Our first lemma tells us that R^Γ works exactly as it does in ordinary propositional modal logic.

Lemma 12 (Existence Lemma). *Let Γ be a named, pasted, and witnessed $\mathbf{K}_{SQph}+$*RULES*-MCS, and let $\mathcal{M}^\Gamma$ be the canonical named general model yielded by Γ. Suppose $u \in W^\Gamma$ and $\Diamond\varphi \in u$. Then there is a $v \in W^\Gamma$ such that $uR^\Gamma v$ and $\varphi \in v$.*

Proof. This is essentially Lemma 7.27 from [6]. □

Lemma 13. *Let Γ be a named, pasted and witnessed $\mathbf{K}_{SQph}+$*RULES*-MCS, let $\mathcal{M}^\Gamma$ be the canonical named general model yielded by Γ, and let $u \in W^\Gamma$. Then for all quantifier-free formulas φ, we have that:*

1. $\mathcal{M}^\Gamma, g^\Gamma, u \models \varphi$ *iff* $\varphi \in u$
2. $\mathcal{M}^\Gamma, g^\Gamma, \Delta_i \models \varphi$ *iff* $@_i\varphi \in \Gamma$.

Proof. Item 1 follows by induction on the number of connectives. It is essentially Lemma 7.28 from [6]. Item 2 then follows by the definition of Δ_i. □

Next for a simple but important lemma:

Lemma 14. *Let Γ be a named, pasted and witnessed $\mathbf{K}_{SQph}$ + RULES-MCS, let $\mathcal{M}^\Gamma$ be the canonical named general model yielded by Γ. Then $[\mathcal{M}^\Gamma, g^\Gamma]^{sqf} = \Pi^\Gamma$.*

Proof. Item 1 of Lemma 3 tells us that $[\mathcal{M}^\Gamma, g^\Gamma]^{sqf} \subseteq \Pi^\Gamma$. Item 2 of same lemma tells us that $[\mathcal{M}^\Gamma, g^\Gamma]^{sqf}$ is a closed set of admissible propositions. Now, $im(V^\Gamma) \cup im(g^\Gamma) \subseteq [\mathcal{M}, g]^{sqf}$, as these are the atomic propositions picked out by the propositional constants and variables. But Π^Γ is $\Pi(im(V^\Gamma) \cup im(g^\Gamma))$, the *smallest* closed set of admissible propositions containing $im(V^\Gamma) \cup im(g^\Gamma)$. So $\Pi^\Gamma \subseteq [\mathcal{M}^\Gamma, g^\Gamma]^{sqf}$. □

The previous lemma is important because it gives us a *syntactic* handle on the elements of Π^Γ: every proposition in Π^Γ is "picked out" by some soft-QF formula; this syntactic characterisation enables us to prove the final lemma leading to completeness.

Lemma 15 (Truth Lemma). *Let Γ be a named, pasted, and witnessed $\mathbf{K}_{SQph}$ + RULES-MCS, and let $\mathcal{M}^\Gamma$ be the canonical named general model yielded by Γ, and let $u \in W^\Gamma$. Then, for all formulas φ, we have that $\mathcal{M}^\Gamma, g^\Gamma, u \models \varphi$ iff $\varphi \in u$.*

Proof. For any formula φ, let $con(\varphi)$ be the number of connectives in φ. Moreover, let $quan(\varphi)$ be the maximal depth of quantifier nesting in φ, that is:

$$quan(\varphi) = \begin{cases} 0 & \text{if } \varphi \text{ is atomic} \\ sup\{quan(\psi), quan(\theta)\} & \text{if } \varphi = \psi \wedge \theta \\ quan(\psi) & \text{if } \varphi \in \{\neg\psi, @_i\psi, \Box\psi\} \\ quan(\psi) + 1 & \text{if } \varphi = \forall p\psi \end{cases}$$

We prove the result by induction on pairs $(quan(\varphi), con(\varphi))$ ordered lexicographically, that is, $(q, c) < (q', c')$ iff (1) $q < q'$ or (2) $q = q'$ and $c < c'$.

Note that Lemma 13 has established this for all formulas φ associated with the pair $(0, 0)$ (that is, atomic formulas) and indeed for all formulas associated with pairs $(0, n)$ for any natural number n (that is, quantifier-free formulas). So the base cases are established, and as our inductive hypothesis (IH) we assume that for natural numbers q and c, with $q \leq c$, we have that $\mathcal{M}^\Gamma, g^\Gamma, u \models \varphi$ iff $\varphi \in u$ for all formulas associated with the pair (q, c)

So let θ be a formula associated with a pair with $c + 1$ connectives. Now, if θ is a boolean, or of the form $\Box\psi$ or $@_i\psi$, we can argue as in Lemma 13, for all such formulas are associated with $(q, c + 1)$, and we can use our new IH just as before. The critical case is when θ is of the form $\forall p\psi$. Note that such formulas are associated with $(q + 1, c + 1)$.

We want to show that $\mathcal{M}^{\Gamma}, g^{\Gamma}, u \vDash \varphi$ iff $\varphi \in u$ For the left-to-right direction, we show the contrapositive. So suppose that $\forall p\psi \notin u$. As u is an MCS, $\neg\forall p\psi \in u$, that is, $\exists p\neg\psi \in u$. Γ is witnessed, so for some proposition letter c, $@_j\neg\psi[c/p] \in \Gamma$. But then $\neg\psi[c/p] \in \Delta_j$. But $\neg\psi[c/p] \in \Delta_j$ is associated with $(q, c+1)$, so the IH applies and we have $\mathcal{M}^{\Gamma}, g^{\Gamma}, u \vDash \neg\psi[c/p]$. It follows from the Substitution Lemma that $\mathcal{M}^{\Gamma}, g', u \vDash \neg\psi$, that is, $\mathcal{M}^{\Gamma}, g', u \nvDash \psi$, where $g' \sim_p g^{\Gamma}$ is defined by setting $g'(p) = [\mathcal{M}^{\Gamma}, g^{\Gamma}]_c$. Hence $\mathcal{M}^{\Gamma}, g^{\Gamma}, u \nvDash \forall p\psi$, and we have proved the contrapositive.

For the right-to-left direction, suppose that $\forall p\psi \in u$, and further suppose for the sake of contradiction that $\mathcal{M}^{\Gamma}, g^{\Gamma}, u \nvDash \forall p\varphi$. Then for some $g' \sim_p g^{\Gamma}$ we have $\mathcal{M}^{\Gamma}, g', u \vDash \neg\varphi$. Now $g'(p) \in \Pi^{\Gamma}$, but by Lemma 14 we know $\Pi^{\Gamma} = [\mathcal{M}^{\Gamma}, g^{\Gamma}]^{sqf}$, hence $g'(p) = [\mathcal{M}^{\Gamma}, g^{\Gamma}]_{\theta}$ for some soft-QF formula θ. The Substitution Lemma tells us that $\mathcal{M}^{\Gamma}, g^{\Gamma}, u \vDash \neg\varphi[\theta/p]$ *iff* $\mathcal{M}^{\Gamma}, g', u \models \neg\varphi$, and hence we have $\mathcal{M}^{\Gamma}, g^{\Gamma}, u \vDash \neg\varphi[\theta/p]$. As $\neg\varphi[\theta/p]$ is associated with the pair $(q, c+1)$, so we can apply the IH to conclude $\neg\varphi[\theta/p] \in u$. But this leads to a contradiction. As u is an MCS, for all soft QF formulas θ, $\psi[\theta/p] \in u$ by the *Q2-sqf* axiom. In particular, $\varphi[\theta/p] \in u$. We conclude that $\mathcal{M}^{\Gamma}, g^{\Gamma}, u \vDash \forall p\varphi$ after all. □

Theorem 2 (Strong basic completeness). *Every* $\mathbf{K}_{SQph}$ + RULES*-consistent set of sentences has a named model.*

Proof. Follows from the previous lemma in the familiar way. □

So we have proved the *basic* strong completeness result. But the general model we have built is named, so we can immediately extend this to cover all pure extensions of $\mathbf{K}_{SQph}$ + RULES. In hybrid logic, a formula is called "pure" if all its atomic formulas are nominals. Here are three well-known examples: $i \rightarrow \Diamond i$ (the *Irreflexivity* axiom), $i \rightarrow \Box(\Diamond i \rightarrow i)$ (the *Antisymmetry* axiom), and $@_i\Diamond j \vee @_i j \vee @_j\Diamond i$ (the *Trichotomy* axiom). A pure formula φ *defines* a class of frames F iff: $(W, R) \in \mathsf{F} \Leftrightarrow (W, R) \models \varphi$. It is easy to check that our three examples define the class of ireflexive, antisymmetric, and trichotomous frames respectively. More importantly: adding *any* (combination of) pure formula(s) as extra axiom(s) to $\mathbf{K}_h$ + RULES is a proof system that is complete with respect to the class of frames defined. For a more detailed statement and discussion of this result, see [5]. Here we shall simply record that:

Theorem 3 (Strong pure completeness). *Let Λ be a set of pure formulas, and let* $\mathbf{K}_{SQph}$ + RULES + Λ *be the Hilbert-system obtained by using the pure formulas in Λ as extra axioms. Then every* $\mathbf{K}_{SQph}$ + RULES + Λ*-consistent set of sentences has a named model built over a Kripke frame belonging to the frame-class defined by the pure formulas in Λ.*

Proof. This is essentially the same as the proof of Theorem 7.29 from [6]. Because nominals directly "tag" worlds in the underlying Kripke frames, the standard completeness result for pure formulas carry over unchanged. □

Here is an example. Let $\Lambda = \{i \rightarrow \Diamond i, i \rightarrow \Box(\Diamond i \rightarrow i), \Diamond\Diamond i \rightarrow \Diamond i\}$. These three formulas jointly define the class of partially ordered frames. Adding them as axioms to $\mathbf{K}_{SQph}$ + RULES gives us the complete logic of this frame class.

6 Concluding Remarks

In this paper we have extended completeness results for basic hybrid logic to cover languages containing propositional quantifiers. We adapted well-known techniques from the hybrid logic literature to build named general models, and thus prove completeness not merely for the minimal logic but for any pure extension. The key to this was our decision to directly "hardwire" nominals to worlds: this decoupled the world naming apparatus (nominals) from the quantificational apparatus (admissible propositions). Although we only treated the case for basic hybrid logic with a single modality, the results proved here can be extended to systems containing multiple modalities, the universal modality, the $\downarrow$-binder, and quantification over nominals, as we will show in an extended version of this paper. We also think the basic system outlined here hints at potentially useful applications. For example, Belardinelli *et al* [3] argue that (multimodal) epistemic logic augmented with propositional quantifiers is a useful knowledge representation tool. Our results for pure extensions suggest that adding a hybrid component might make them even more useful for such tasks.

But to close the paper, we return to the work of Arthur Prior that inspired it. Arthur Prior was a pioneer of propositional quantification in modal logic (see, in particular, [18,19]), and his students Robert Bull [9] and Kit Fine [11] both published technical results about it, the latter paper becoming highly influential. But Arthur Prior was also the inventor of hybrid logic, and in the final years of his career, these two interests became intertwined. Prior had oscillated between the "tag" view of nominals that is now standard and a "telescope" view that views them as (something more like) an infinite conjunction of information (see [7,14]). In two key late papers, Prior seems to have moved towards the "tag" view of nominals.[6] He also realised—anticipating the mantra of the Amsterdam school of modal logic — that modalities could be used to talk about absolutely *anything*. Indeed, his egocentric logic is an early example of what is now called description logic [2]. In egocentric logic, "worlds" are people and their properties and relationships (for example, their relative heights) are described using what Prior called "quasi modalities", with the help of propositional quantifiers and "people propositions" (nominals). Prior's death left many of these ideas underexplored, but it is clear that in his final years Prior developed several philosophically and technically novel systems, often involving both nominal-like entities and/or propositional quantification ("Prior's cocktail"). We want to use the language presented in this paper to explore this work more systematically.

Acknowledgements. We would like to thank Antje Rumberg and the three anonymous referees for their comments and corrections on earlier versions of this paper.

[6] Namely: "Tense logic and the logic of earlier and later", and "Quasi-propositions and quasi-individuals", both of which can be found in the first edition of *Papers on Time and Tense* [16]. The new edition [17] contains several more papers that build on these two, including "Egocentric logic". See Kofod's PhD thesis [14] for further discussion.

References

1. Areces, C., ten Cate, B.: Hybrid logic. In: Handbook of Modal Logic, pp. 821–868. Elsevier (2007)
2. Baader, F., Calvanese, D., McGuinness, D., Patel-Schneider, P., Nardi, D., et al.: The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press, Cambridge (2003)
3. Belardinelli, F., Van Der Hoek, W., Kuijer, L.B.: Second-order propositional modal logic: expressiveness and completeness results. Artif. Intell. **263**, 3–45 (2018)
4. Blackburn, P.: Arthur Prior and hybrid logic. Synthese **150**(3), 329–372 (2006)
5. Blackburn, P., ten Cate, B.: Pure extensions, proof rules, and hybrid axiomatics. Stud. Logica. **84**, 277–322 (2006)
6. Blackburn, P., de Rijke, M., Venema, Y.: Modal Logic. Cambridge University Press, Press (2001)
7. Blackburn, P.R., Braüner, T., Kofod, J.L.: Remarks on hybrid modal logic with propositional quantfiers. In: The Metaphysics of Time: Themes from Prior, pp. 401–426. Aalborg Universitetsforlag (2020)
8. Braüner, T.: Hybrid Logic. In: Zalta, E.N. (ed.) The Stanford Encyclopedia of Philosophy. Metaphysics Research Lab, Stanford University, summer 2017 edn. (2017)
9. Bull, R.: On modal logic with propositional quantifiers. J. Symbolic Logic **34**(2), 257–263 (1969)
10. Copeland, B.J.: Arthur Prior. In: Zalta, E.N. (ed.) The Stanford Encyclopedia of Philosophy. Metaphysics Research Lab, Stanford University, Summer 2017 edn. (2017)
11. Fine, K.: Propositional quantifiers in modal logic. Theoria **36**, 336–346 (1970)
12. Henkin, L.: Completeness in the theory of types. J. Symbolic Logic **15**(2), 81–91 (1950)
13. Kaminski, M., Tiomkin, M.: The expressive power of second-order propositional modal logic. Notre Dame J. Formal Logic **37**(1), 35–43 (1996)
14. Kofod, J.L.: What on earth was Arthur Prior thinking?: About physics and hybrid logic. Ph.D. thesis, IKH, Roskilde University (2022)
15. Øhrstrøm, P., Hasle, P.: Temporal Logic: from Ancient Ideas to Artificial Intelligence, vol. 57. Springer, Dordrecht (2007). https://doi.org/10.1007/978-0-585-37463-5
16. Prior, A.: Papers on Time and Tense. Oxford University Press (1968)
17. Prior, A.: Papers on Time and Tense. Oxford University Press, new edn. (2003)
18. Prior, A.N.: Epimenides the Cretan. J Symbolic Logic **23**(3), 261–266 (1958)
19. Prior, A.N.: On a family of paradoxes. Notre Dame J. Formal Logic **2**(1), 16–32 (1961)
20. Prior, A.N., Fine, K.: Worlds, Times, and Selves. Duckworth, London (1977)
21. Thomason, S.: Semantic analysis of tense logics. J. Symbolic Logic **37**, 150–158 (1972)

An Evidence Logic Perspective on Schotch-Jennings Forcing

Tyler D. P. Brunet[1] and Gillman Payette[2](✉)

[1] Egenis: the Center for the Study of the Life Science, Department of Social and Political Sciences Philosophy and Anthropology, University of Exeter, Exeter, UK
t.d.p.brunet@exeter.co.uk

[2] Department of Philosophy, University of Calgary, Calgary, AB, Canada
ggpayett@ucalgary.ca

Abstract. Traditional epistemic and doxastic logics cannot deal with inconsistent beliefs nor do they represent the evidence an agent possesses. So-called 'evidence logics' have been introduced to deal with both of those issues. The semantics of these logics are based on neighbourhood or hypergraph frames. The neighbourhoods of a world represent the basic evidence available to an agent. On one view, beliefs supported by evidence are propositions derived from all maximally consistent collections evidence. An alternative concept of beliefs takes them to be propositions derivable from consistent partitions of one's inconsistent evidence; this is known as Schotch-Jennings Forcing. This paper develops a modal logic based on the hypergraph semantics to represent Schotch-Jennings Forcing. The modal language includes an operator $U(\varphi_1, \ldots, \varphi_n; \psi)$ which is similar to one introduced in Instantial Neighbourhood Logic. It is of variable arity and the input formulas enjoy distinct roles. The U operator expresses that all evidence at a particular world that supports ψ can be supported by at least one of the φ_is. U can then be used to express that all the evidence available can be unified by the finite set of formulas $\varphi_1, \ldots, \varphi_n$ if ψ is taken to be $\top$. Future developments will then use that semantics as the basis for a doxastic logic akin to evidence logics.

Keywords: Evidence Logic · Epistemic Logic · Paraconsistent Logic · Schotch-Jennings Forcing · Pointed Operators

1 Introduction

In [4] and [2] the authors proffer modal logics for reasoning about beliefs which are based on *evidence*. Traditionally, epistemic and doxastic logics are about how an agent reasons from propositions they know or believe. How the agent arrives at those propositions they reason from is not part of the model. However, these new "Evidence Logics" include an explicit representation of what evidence an agent has. They then can go on to define conditions for belief on the basis of what evidence the agent possesses.

H. H. Hansen et al. (Eds.): WoLLIC 2023, LNCS 13923, pp. 135–160, 2023.
https://doi.org/10.1007/978-3-031-39784-4_9

One of the challenges of doxastic and epistemic logic has been that agents often possess inconsistent evidence. Traditional modal logics based on (binary) relational semantics cannot tolerate inconsistency; everything is believed when beliefs are inconsistent. These evidence models suggest a different approach. They allow the evidence one accumulates to be inconsistent, while restraining beliefs based on that evidence in ways that ensure consistency of resulting belief—at least in the case of [2]. Filtering beliefs from evidence requires novel ways of combining the evidence and deriving conclusions from it that will avoid—if not eliminate—inconsistencies.

The approaches to evidence based belief in [2] and [4] relate to the method of dealing with inconsistent data/premises proposed by [12] in which one reasons from maximally consistent subsets of one's data. We take a different starting point, namely, the preservationist approach to paraconsistency in [8]. The preservationist method of reasoning from inconsistent data is to reason from special partitions of one's data; when something follows from one of these partitions, that conclusion is *forced*, and this inference method is called *forcing*. The relationship between these two approaches to paraconsistent propositional inference has been studied in [11]. Before any application of this preservationist approach can be made in the present context—to the modal logic representation of evidence—it must first be given a semantic representation (Sect. 3) that facilitates comparison between the two starting points.

The meeting point of the two views is the use of neighbourhood models to represent the evidence of an agent: the collection of neighbourhoods is the set of basic evidence available at that world. The preservationist approach to paraconsistency was inspired by modal logics which use n-ary relation frames, cf. [1,7]. It came to be understood that those frames corresponded to neighbourhood frames for modal logic [9]. The paraconsistent n-ary modal logics could be interpreted on those neighbourhood frames when a variation on the truth condition for the $\Box$ operator is used: in order for $\Box\varphi$ to be true at x, there must be a neighbourhood of x where that φ is true throughout. This differs from the usual truth condition, in which all the worlds where φ is true must *be* a neighbourhood of x. A thorough study of the relations between n-ary modal logics and n-relation modal logics has been conducted in [6] which explores these connections via neighbourhood semantics.

Here, we offer a way to use neighbourhood models to represent a preservationist approach of deriving belief from evidence. The goal of this paper is to capture the general forcing relation in a neighbourhood semantics. To do this, we introduce an operator, similar to that found in [3], which takes two arguments: a non-empty list of formulas and a formula. This operator expresses the sufficiency of the formulas in the list of the first argument for implying the formula in the second argument. What we show is that a semantics can be given which represents Schotch-Jennings forcing on classical propositional logic, and provide a logic which is sound and complete for that semantics.

1.1 Evidence Models

Evidence models are built on the standard set up from modal logic where we have a non-empty set of ways the world might be W, i.e., possible worlds, and propositions or facts that might be true in those worlds represented by subsets of W. An agent's evidence will be represented by a so-called 'evidence frame'

$$\mathfrak{F} = \langle W, \mathcal{E} \rangle$$

consisting of W and a function $\mathcal{E} : W \to \mathcal{P}(\mathcal{P}(W))$. For each $x \in W$, $\mathcal{E}(x)$ represents the evidence the agent has collected at x; the agent's *basic* evidence at x. The only conditions that we will impose on $\mathcal{E}(x)$—at the moment—are that $\varnothing \notin \mathcal{E}(x) \neq \varnothing$. Thus, the agent can never collect a contradiction as evidence. There are no conditions at this point on whether $\mathcal{E}(x)$ must be closed under various set-theoretic operations like supersets or intersection. We will consider an agent to *have evidence* that $X \subseteq W$, when there is $Y \in \mathcal{E}(x)$ such that $Y \subseteq X$. That is, agents have all evidence that their basic evidence, taken individually, implies. That makes the requirement of closure under supersets unnecessary.

Although the evidence an agent has is simply what that agent's evidence individually implies, what an agent's evidence *supports* is an holistic matter. Intuitively, evidential support should be computed by combining the basic evidence somehow, but it is not clear how that should be done. We have not assumed that $\mathcal{E}(x)$ is factual or even consistent: the actual world may not be in $\bigcap \mathcal{E}(x)$ nor is it guaranteed that $\bigcap \mathcal{E}(x) \neq \varnothing$, respectively. So a simple combining of one's basic evidence via taking what is common between all of it may result in "supporting" everything since all propositions are implied by an inconsistent set: when $\bigcap \mathcal{E}(x) = \varnothing$, $\bigcap \mathcal{E}(x) \subseteq X$ for any $X \subseteq W$. The authors van Benthem et al. and Baltag et al. have suggested two fruitful ways of combining evidence. Inspired by their ideas, we here offer a method of combining evidence by using a representation of Schotch-Jennings Forcing in modal logic.

Schotch-Jennings Forcing offers a way to disentangle any inconsistency, and then to infer from the disentangled collection. In the following section we will review the syntactic account of this method, survey the extant connections between modal logic and forcing, and then develop a semantic analog of forcing in neighbourhood models, suitable as a basis for a modal logic.

1.2 Forcing and Level

In a series of papers, [7,13], and [14] Jennings and Schotch developed a method of drawing inferences from inconsistent sets which they refer to as 'forcing'. The set up is to find the minimal way to partition the premises so that each element, or 'cell', of the partition is consistent. Then one reasons from those consistent cells. Taking the smallest or minimal partitions of a set, if some conclusion follows from at least one cell in *every* such partition, then the set forces that conclusion.

More precisely, lets say that a partition Π is a *cover* of a set of formulas Γ iff, $\bigcup \Pi = \Gamma$ and for all $\pi \in \Pi$, $\pi \nvdash \bot$ where $\vdash$ is simply the consequence relation of classical logic. We will also refer to the cardinality of Π as its width. There is another definition of a syntactic cover as follows: a collection of consistent sets of sentences Π (not necessarily a partition of Γ) such that for each $\gamma \in \Gamma$, there is $\pi \in \Pi$ such that $\pi \vdash \gamma$. If we introduce $\mathbb{C}(\Gamma) = \{\, \alpha : \Gamma \vdash \alpha \,\}$ to refer to the deductive closure of Γ, then we can say that Π is a cover of Γ when $\Gamma \subseteq \bigcup_{\pi \in \Pi} \mathbb{C}(\pi)$ and each π is consistent. Partitions are a special case of this more general kind of cover and are thus referred to as 'partition covers'.

The *level* of Γ, $\ell(\Gamma)$, is a kind of measure of how inconsistent Γ is, and it is determined by the minimum width a set of sets must have in order to be a cover of Γ, but if there is no such minimum, its level is ∞. Thus:

$$\ell(\Gamma) = \begin{cases} 0 & \Gamma \subseteq \mathbb{C}(\varnothing) \\ \min \{\, |\Pi| : \Pi \text{ is a cover of } \Gamma \,\} & \text{if it exists } \& \ \Gamma \nsubseteq \mathbb{C}(\varnothing) \\ \infty & \text{otherwise} \end{cases}$$

We assign the level of 0 to the special case where Γ is a set of theorems. We can then say that Γ forces α, $\Gamma \Vdash \alpha$ iff in any cover of Γ (partition or otherwise), Π, such that $|\Pi| = \ell(\Gamma)$, there is $\pi \in \Pi$ such that $\pi \vdash \alpha$. However, it can be shown that forcing is determined by the collection of partition covers since we can always generate a partition cover from a cover.

Most conceptions of consequence are based on considering what is true across all the ways things could be and forcing incorporates this 'all the ways things could be' kind of thinking by consulting all covers of Γ to determine the forcing consequences. It is interesting to note that this does not simply mean looking at all $\ell(\Gamma)$-tuples of distinct maximally consistent subsets of Γ. This could seem odd since obviously each cell in a (partition) cover of Γ can be extended to a maximally consistent subset (i.e., Γ' is a maximally consistent subset of Γ iff $\Gamma' \subseteq \Gamma$, $\Gamma' \nvdash \bot$ and for any $\alpha \in \Gamma \backslash \Gamma'$, $\Gamma' \cup \{\, \alpha \,\} \vdash \bot$). The issue is that some maximally consistent subsets may not be reachable by such extensions. For example, consider the following set from classical logic:

$$\Phi = \{\, \neg q \wedge p, q \rightarrow r, \neg r, q, \neg p \,\}$$

This set gives rise to maximally consistent subsets. We will not list all of them, but for instructional purposes here are two of them:

(A) $\{\, q \rightarrow r, \neg r, \neg p \,\}$, and
(B) $\{\, q \rightarrow r, \neg r, \neg q \wedge p \,\}$, and

It is easy to see that $\ell(\Phi) = 2$ since it is inconsistent but we only need a cover of width 2:

$$\{\, \{\, q \rightarrow r, \neg r, \neg q \wedge p \,\}, \{\, q, \neg p \,\} \,\}.$$

What this means is that the covers that would be used to determine the forcing consequences of Φ would all have width 2. This gives rise to a curious situation when we consider set A above. Set A would never appear in a cover of Φ that was used to calculate forcing consequences. The reason is that, if set A is removed from Φ, the set that is left over has level 2 as well. That means no cover of Φ with width 2 could be constructed with set A as a cell.

If one ends up with inconsistent evidence, another way to make inferences from it, or another way to calculate what the evidence supports, is by what the evidence forces. Of course, if one's evidence is *consistent*, then the conclusions one can draw are simply all those which follow, classically speaking. We now consider a semantics for forcing which relates it to the semantics of prior evidence logics.

2 Forcing and Modal Logic

Although we are interested in representing forcing in an evidence logic manner, there already exist some connections between modal logics and forcing. In fact, these modal logics are non-normal and have natural semantics in terms of evidence logic-like semantics. First we will define a language which we will add to as we encounter problems. The basic semantic set up is just like that for evidence logics; we start with a frame and then a model:

Definition 1. A structure $\mathfrak{F} = \langle W, \mathcal{E} \rangle$ is a hypergraph/evidence **frame** iff:

1. $W \neq \varnothing$, and
2. $\mathcal{E} : W \to \mathcal{P}(\mathcal{P}(W))$ such that for all $x \in W$
   (a) $\varnothing \notin \mathcal{E}(x)$, and
   (b) $\mathcal{E}(x) \neq \varnothing$.

A hypergraph/evidence **model** is a structure $\mathcal{M} = \langle \mathfrak{F}, V \rangle$ where $\mathfrak{F}$ is a hypergraph/evidence frame and $V : \mathbf{At} \to \mathcal{P}(W)$ where **At** is the set of atomic formulas of a propositional language.

For simplicity we will refer to hypergraph frames as 'hyperframes'. We can then define the semantics for a language on such models which we will sometimes refer to simply as 'models'. The language consists of the boolean operators and the unary operator '$E\varphi$' which is meant to be interpreted, intuitively, as that there is evidence supporting the proposition φ among one's basic evidence. Its dual is denoted as $\langle E \rangle$. Let $\mathcal{M} = \langle \mathfrak{F}, V \rangle$ be a model, the semantics is:

- $\mathcal{M}, x \models p$ iff $x \in V(p)$ for all $p \in \mathbf{At}$
- Boolean cases as usual,
- $\mathcal{M}, x \models E\varphi$ iff there is $X \in \mathcal{E}(x)$ such that $X \subseteq [\![\varphi]\!]$,
- $\mathcal{M}, x \models \langle E \rangle \varphi$ iff for all $X \in \mathcal{E}(x)$, $X \cap [\![\varphi]\!] \neq \varnothing$.

Of course $\mathcal{M}$ satisfies φ iff there is $x \in W$ such that $\mathcal{M}, x \models \varphi$ and satisfies a set of sentences Γ iff $\mathcal{M}$ satisfies all members of Γ at some world $x \in W$. As is also standard, $\Gamma \models_E \varphi$ iff for all $\mathcal{M}$ and x, if $\mathcal{M}, x \models \Gamma$, then $\mathcal{M}, x \models \varphi$. As is well known [5], this logic can be axiomatized as follows:

CL All theorems of classical propositional logic.
D $\vdash_E \neg E\bot$
N $\vdash_E E\top$

M $\dfrac{\vdash_E p \to q}{\vdash_E Ep \to Eq}$
With rules
MP Modus Ponens, and
US Uniform Substitution.

This is the basic logic of hypergraphs as we have defined them above. But as one might expect it is nowhere near expressive enough to capture forcing. But there are near-by logics based on hyperframes that connect to forcing and are fairly well understood. First, there are the K_n modal logics which *sometimes* represent the forcing consequences of a set of formulas.

The modal logics K_n are non-normal modal logics which are axiomatized in the following way:[1]

CL All theorems of classical propositional logic.
N $\vdash_{K_n} \langle E\rangle \top$
$K_n^{\Diamond}$ $\vdash_{K_n} (\langle E\rangle p_1 \wedge \ldots \wedge \langle E\rangle p_{n+1}) \to \langle E\rangle \bigvee_{1\leq i<j\leq n+1}(p_i \wedge p_j)$
With rules
M $\dfrac{\vdash_{K_n} p \to q}{\vdash_{K_n} \langle E\rangle p \to \langle E\rangle q}$
MP Modus Ponens, and
US Uniform Substitution.

What is unique about these modal logics is the axiom $K_n^{\Diamond}$[2] which weakens the adjunctive properties of the logic and keeps inconsistent formulas from interacting. The modal logic K_n axiomatizes the logic valid on the class of all n-bounded hyperframes. A hyperframe is n-bounded when for all $X \in \mathcal{E}(x)$ and $x \in W$, $|X| \leq n$. This doesn't mean that an n-bounded hyperframe is finite, just that each edge in each hypergraph is at most n.

What can be shown is that if the level of a set Γ is n, then

$$\Gamma \Vdash \alpha \text{ iff } \langle E\rangle [\Gamma] \vdash_{K_n} \langle E\rangle \alpha$$

where $\langle E\rangle [\Gamma] = \{ \langle E\rangle \gamma : \gamma \in \Gamma \}$. These logics, however, are not suitable for forcing in general. They capture what is called 'fixed-level forcing' which is when one consults all of the covers of Γ which have a fixed width, say, n.[3] The problem with K_n is two fold. If Γ's level is less than n, then one will lose many forcing consequences. And if the level of Γ is larger than n, then Γ is treated as inconsistent, so it 'fixed-level forces' everything. The source of the issue is that the K_n logics cannot discern what level a set of sentences has before determining its consequences.

There are also the P_n logics studied in [6]. These logics are axiomatized as follows:

---

1 Usually, they are presented with Es ($\Box$s) in the place of all the $\langle E\rangle$ ($\Diamond$s) which makes the connection to the modal logic K clearer in which $K_1 = K$. But we are choosing to remain consistent with the notation in the literature on evidence logic.

2 This is the name of the axiom as presented in [6].

3 A more appropriate name would be 'fixed-width forcing'.

CL All theorems of classical propositional logic.
N $\vdash_{P^n} E\top$
P^n $\vdash_{P^n} (Ep_1 \wedge \ldots \wedge Ep_{n+1}) \rightarrow \bigvee_{1\leq i<j\leq n+1} E(p_i \wedge p_j)$
With rules
M $\dfrac{\vdash_{P^n} p \rightarrow q}{\vdash_{P^n} Ep \rightarrow Eq}$
MP Modus Ponens, and
US Uniform Substitution.

The P^n logics are determined by the class of all consistent and n-bounded in degree hyperframes. A hyperframe is n-bounded in degree iff for all $x \in W$, $|\mathcal{E}(x)| \leq n$ and consistent iff for all $X \in \mathcal{E}(x)$, $|X| \geq 1$ for all $x \in W$.

These logics have the resources to determine what level a set of sentences has. If Γ has level n and $m < n$, then $E[\Gamma] \vdash_{P^m} \bot$. That is because if $\mathcal{M}, x \vDash E[\Gamma]$, then for all $\gamma \in \Gamma$ there is $X \in \mathcal{E}(x)$ such that $X \subseteq [\![\gamma]\!]$. Now, if $|\mathcal{E}(x)| < \ell(\Gamma)$, then by a pigeon hole argument we could create a syntactic cover of Γ whose width is less than $\ell(\Gamma)$; but that should be impossible when $\ell(\Gamma) = n$. So, when $E[\Gamma] \vdash_{P^m} \bot$, $\ell(\Gamma) \geq m$. Similarly, if $E[\Gamma] \nvdash_{P^m} \bot$ then $\ell(\Gamma) \leq m$. If Γ is finite, then $\ell(\Gamma) \leq |\Gamma|$. So, for finite sets Γ, $\ell(\Gamma) = n$ iff $E[\Gamma] \vdash_{P^{n-1}} \bot$ and $E[\Gamma] \nvdash_{P^n} \bot$.

But just because the P^n logics can determine the level of a set, that doesn't mean that it can determine the forcing consequences. Indeed, it doesn't. The logic K_n determines the forcing consequences of sets which have level n.

The fundamental issue is that forcing is a dynamic, global and contextual conception of consequence. Generally, the logical consequences of a set of sentences are dependent on what the set contains but are not influenced by global properties of that set. Forcing, on the other hand, contextually adapts to a particular, and important, global property of the set, namely the set's level. Typically, logics do *not* change their behaviour from context to context; that is kind of the point of them. But forcing must, since it depends on preserving the overall coherence of the set of premises, not just interactions between some individual premises. So to develop a semantics for forcing we have to find a way to overcome that narrow focus. We need a logic that can both determine the level of a set and its forcing consequences.

3 Covers: Syntactic vs. Semantic

Although there are various logics that can represent certain kinds of forcing, none captures forcing in general. The goal is to represent forcing using an evidence logic style semantics. The first thing which is needed is a semantic analog of a cover in order to represent the level of a set via a semantic object, i.e., an evidence set.

Given a set $\mathcal{X}$ of subsets of a set W, we can define the level of this set in much the same way as we defined the level of a set of formulas since, after all, subsets of W are supposed to represent propositions. We start with a *cover*.[4] A *cover* of $\mathcal{X}$ is a set $\mathcal{Y} \subseteq \mathcal{P}(W) \setminus \{ \varnothing \}$ such that for each $X \in \mathcal{X}$, there is $Y \in \mathcal{Y}$ and $Y \subseteq X$. Again,

$$\ell(\mathcal{X}) = \begin{cases} 0 & \text{when } \mathcal{X} = \{ W \} \\ \min \{ |\Pi| : \Pi \text{ is a cover of } \mathcal{X} \} & \text{if it exists} \\ \infty & \text{otherwise} \end{cases}$$

Like in the syntactic case, $\ell(\mathcal{X}) = \infty$ iff there is a self-inconsistent proposition in $\mathcal{X}$, i.e., $\varnothing \in \mathcal{X}$. The conditions on evidence frames will rule out $\varnothing$ ever being in an $\mathcal{E}(x)$, so no evidence set will have level ∞. A major difference is that since $\mathcal{E}(x)$ could be uncountable, $\ell(\mathcal{E}(x))$ could be an uncountable cardinal, which cannot happen in the syntactic case when one is only working with countable languages. But even in the syntactic case one could have an evidence set of level ω. An evidence set like that would have covers where the extension of each formula is in its own cell. However, given an evidence set whose narrowest cover is of size ω, its forcing consequences boil down to only what follows from the individual pieces of evidence on their own.

Another fact which is easy to see is that if $\mathcal{Y}$ is a cover of $\mathcal{X}$, then $\ell(\mathcal{Y}) \geq \ell(\mathcal{X})$. For suppose that $\mathcal{Y}'$ is a cover of minimal width of $\mathcal{Y}$. Then $|\mathcal{Y}'| = \ell(\mathcal{Y})$. But the transitivity of $\subseteq$ means that $\mathcal{Y}'$ is also a cover of $\mathcal{X}$. Thus, $\ell(\mathcal{X}) \leq |\mathcal{Y}'| = \ell(\mathcal{Y})$.

We now introduce some closely related concepts to connect semantic covers to syntactic covers via the evidence models. These concepts help us discuss the various ways that sets of sentences may relate to sets of basic evidence, given a model and point within it. Note that if $\mathcal{M}$ is a model, $[\![\Gamma]\!]_{\mathcal{M}} = \{ [\![\gamma]\!] : \gamma \in \Gamma \}$ rather than the more common understanding of that notation as $\bigcap \{ [\![\gamma]\!] : \gamma \in \Gamma \}$. We will usually omit the subscript $\mathcal{M}$.

Definition 2. Let $\mathcal{M}$ be a model, $x \in W$, Γ a set of sentences, and $X \in \mathcal{E}(x)$. We will say,

- $\mathcal{M}$ **covers** Γ at x iff $\forall \gamma \in \Gamma, \exists X \in \mathcal{E}(x), X \subseteq [\![\gamma]\!]$.
- $\mathcal{M}$ **strongly covers** Γ at x iff $[\![\Gamma]\!] \subseteq \mathcal{E}(x)$.
- $\mathcal{M}$ is **unified** by Γ at x iff $\forall X \in \mathcal{E}(x), \exists \gamma \in \Gamma, [\![\gamma]\!] \subseteq X$.
- $\mathcal{M}$ is **strongly unified** by Γ at x iff $[\![\Gamma]\!] \subseteq \mathcal{E}(x)$ and $\mathcal{M}$ is unified by Γ at x.

In the vocabulary of evidence models from Sect. 1, $\mathcal{M}$ covers Γ at x iff there *is evidence that* γ at x for each $\gamma \in \Gamma$, and strong covering is, intuitively, the claim that Γ is among the basic evidence at x. For unification, $\mathcal{M}$ unifies Γ at x when every piece of basic evidence is evidenced by something in Γ. Finally, strong unification is when the evidence at x is unified by a subset of the evidence at x. These concepts (and those that can be defined in terms of them) exhausts

[4] We could define a cover of $\mathcal{X}$ as a subset of $\mathcal{P}(\mathcal{P}(W))$, Π such that for each $\pi \in \Pi$, $\cap\pi \neq \varnothing$ and for each $X \in \mathcal{X}$ there is $\pi \in \Pi$ such that $\cap\pi \subseteq X$ and if Π is a partition of $\mathcal{X}$ we say that Π would be a *partition* cover. However, the definition on offer is slightly more economical.

the ways in which we will need to refer to the relationships between theories and evidence sets, in order to establish a correspondence between syntactic covers of Γ and semantic covers of $\mathcal{E}(x)$. Moreover, note that covering is stable under subsets of Γ and it is easy to see that $\mathcal{M}$ covers Γ at x iff $\mathcal{M}, x \vDash E[\Gamma]$. Also, when $\mathcal{M}$ is unified by Γ at x, then $[\![\Gamma]\!]$ is a cover of $\mathcal{E}(x)$.

The natural epistemic interpretation of unification is that the evidence at x can be theoretically unified by taking Γ as a set of hypotheses, e.g., each piece of evidence can be predicted by the propositions in Γ. When we have Γ in hand, this is clearly an epistemic virtue often sought after in scientific theories: good theories should imply our evidence.[5] While philosophically important, we neglect further discussion of the intuitive philosophical interpretation of these concepts. Instead, we show that unification provides a relationships between evidence sets $\mathcal{E}(x)$ and theories Γ that suffices for a preservationist approach to evidence, by ensuring that syntactic level and semantic level coincide.

Notice first that covering does not suffice. When $\mathcal{M}$ covers Γ at x, the level of $\mathcal{E}(x)$ is not guaranteed to be the same as the level of Γ. Take $\Gamma = \{ p, q, r, \neg p, r \to \neg q \}$. This set has level 2 since

$$\Pi = \{ \pi_1 = \{ p, q, r \}, \pi_2 = \{ \neg p, r \to \neg q \} \}$$

is a partition cover. Then take any model $\mathcal{M}$ in which $\cap[\![\pi_1]\!] \neq \emptyset$ and $\cap[\![\pi_2]\!] \neq \emptyset$ such that there are a, b, c for which $a \subset [\![p]\!] \backslash [\![q]\!] \cup [\![r]\!]$ and $b \subset [\![q]\!] \backslash [\![p]\!] \cup [\![r]\!]$ and $c \subset [\![r]\!] \backslash [\![q]\!] \cup [\![p]\!]$. Let $\mathcal{E}(x) = \{ a, b, c \}$. Then $\mathcal{M}$ covers Γ at x, since for each $\gamma \in \Gamma$ one of a, b, c is a subset of its extension. (Obvious for p, q and r). Consider $[\![\neg q]\!]$, e.g. $a \subset [\![\neg q]\!]$, and likewise $a \subset [\![r]\!]^c \cup [\![q]\!]^c$.[6] But now we have an $\mathcal{M}$ that covers Γ at x but where $\ell(\mathcal{E}(x)) > \ell(\Gamma)$, since a, b, c all pairwise disjoint, $\ell(\mathcal{E}(x)) = 3$. We will also notice that in this model $\ell([\![\Gamma]\!]) > \ell(\Gamma)$. In general, by a similar pigeon hole argument as above, it will always be the case that $\ell([\![\Gamma]\!]) \geq \ell(\Gamma)$ for any Γ.

However, although $\mathcal{M}$ covers Γ does not ensure that the semantic cover has the same level as Γ, if $\mathcal{M}$ is also *unified* by Γ then the evidence set will have the same level as the extensions of all of the sentences in Γ.

Observation 1. *Let $\mathcal{M}$ be a model and $x \in W$. If $\mathcal{M}$ is unified by Γ at x, then $\ell([\![\Gamma]\!]_{\mathcal{M}}) \geq \ell(\mathcal{E}(x))$. If, in addition, $\mathcal{M}$ covers Γ at x, then $\ell([\![\Gamma]\!]_{\mathcal{M}}) = \ell(\mathcal{E}(x))$.*

Proof. $\ell([\![\Gamma]\!]_{\mathcal{M}}) \geq \ell(\mathcal{E}(x))$ is immediate since $[\![\Gamma]\!]_{\mathcal{M}}$ is a cover of $\mathcal{E}(x)$ when $\mathcal{M}$ is unified by Γ at x.

Suppose also that for all $[\![\gamma]\!] \in [\![\Gamma]\!]_{\mathcal{M}}$, there is $X \in \mathcal{E}(x)$ such that $X \subseteq [\![\gamma]\!]$, i.e., $\mathcal{M}$ covers Γ at x. That means $\mathcal{E}(x)$ is a cover of $[\![\Gamma]\!]_{\mathcal{M}}$ and as we have observed, then, $\ell(\mathcal{E}(x)) \geq \ell([\![\Gamma]\!]_{\mathcal{M}})$. Therefore, $\ell(\mathcal{E}(x)) = \ell([\![\Gamma]\!]_{\mathcal{M}})$

[5] Of course, it is also a property that can be trivially satisfied by taking Γ to be large enough—assuming that each $X \in \mathcal{E}(x)$ can be represented by a formula. Of course, if Γ has other properties, e.g., finiteness, that makes a better case for a non-trivial unification.

[6] X^c is the relative complement of X with respect to W.

In the modal language introduced so far, we can express covering, but not unification. The above result thus gives us reason to introduce an operator which allows us to express in the object language that a set of sentences unifies one's evidence. This operator, having variable arity, will be somewhat unorthodox. However, a similar operator has been introduced by [3] in the development of Instantial Neighbourhood Logic (INL).[7] The operator is constructed as follows:

If $\varphi_1, \dots, \varphi_n, \psi$ are formulas, so is $U(\varphi_1, \dots, \varphi_n; \psi)$.[8]

The inclusion of the formula at the end '$\dots; \psi)$' is an effort to build a logic that is parallel with INL. In future work we intend to investigate classes of operators—which we call 'pointed operators', where ψ is the point—that all have the same syntactic form and whose truth conditions have a similar shape. Having said that, the pointedness of the formula provides some very useful, and perhaps required, expressive power. The semantics of this operator is as one might expect given the discussion above:

$$\mathcal{M}, x \vDash U(\varphi_1, \dots, \varphi_n; \psi) \iff \text{ for all } X \in \mathcal{E}(x), \text{ if } X \subseteq [\![\psi]\!] \text{ then there is } i \leq n \text{ s.t. } [\![\varphi_i]\!] \subseteq X.$$

We have yet to bring syntactic and semantic conceptions of level together and a major stumbling block is that the syntactic consistency of a set of formulas requires looking at *all* the models whereas semantic level is determined merely by the model at hand. In some cases this gap can be bridged. Let $\mathbf{At}(\Gamma) = \{ p \in \mathbf{At} : p \text{ is mentioned in } \Gamma \}$ where $\mathbf{At}$ is the set of atomic sentences. Note that in the following observation we will just be using the conceptions of syntactic level derived from classical consequence. Let's call a model $\mathcal{M}$ **consistency comprehensive** for Γ when for all $X \subseteq \mathbf{At}(\Gamma)$, there is $x \in W$ such that for all $p \in \mathbf{At}(\Gamma)$, $\mathcal{M}, x \vDash p$ iff $p \in X$.

Observation 2. *Suppose Γ is a set of pure Boolean formulas. If $\mathcal{M} = \langle \mathfrak{F}, V \rangle$ is consistency for Γ, then $\ell(\Gamma) = \ell([\![\Gamma]\!]_{\mathcal{M}})$.*

Proof. First notice that if Γ contains some formula equivalent to $\bot$, then $\varnothing \in [\![\Gamma]\!]_{\mathcal{M}}$, and $\ell(\Gamma) = \infty = \ell([\![\Gamma]\!]_{\mathcal{M}})$.

Next, notice that if $\Gamma' \subseteq \Gamma$ and is propositionally consistent, there is a truth value assignment v to the atoms that are mentioned in Γ' such that $\vDash_v \Gamma'$. Let $X = \{ p : v(p) = T \} \cap \mathbf{At}(\Gamma)$. Then, by hypothesis there is a $x \in W$ such that $\mathcal{M}, x \vDash p$ iff $p \in X$ for all $p \in \mathbf{At}(\Gamma)$, so $\mathcal{M}, x \vDash \Gamma'$.

---

[7] The operator in [3] is $\Box(\varphi_1, \dots, \varphi_n; \psi)$ which is true at $\mathcal{M}, x$ iff there is $X \in \mathcal{E}(x)$ such that $X \subseteq [\![\psi]\!]$ and $X \cap [\![\varphi_i]\!] \neq \varnothing$ for each $i \leq n$. Its dual would be true, then, iff for all $X \in \mathcal{E}(x)$ if $X \subseteq [\![\psi]\!]$, then $X \subseteq [\![\varphi_i]\!]$ for some $i \leq n$. Whereas that operator says that all of the evidence is sufficient for at least one of φ_is—when it is sufficient for ψ, our U operator says that any piece of evidence is necessary for at least one of the φ_is, when it is sufficient for ψ.

[8] As abbreviations, we will write $\overline{\varphi}$ to mean $\varphi_1, \dots, \varphi_n$, and $(\overline{\varphi}/\psi)_i$ to mean

$$\varphi_1, \dots, \varphi_{i-1}, \psi, \varphi_{i+1}, \dots, \varphi_n$$

for $i \leq n$.

Let Π be a (syntatic) cover of Γ of width $\ell(\Gamma)$. Without loss of generality, we can assume that all logically equivalent formulas are in the same cells of the partition. Now form the following partition of $[\![\Gamma]\!]_{\mathcal{M}}$ by

$$\Pi' = \{ \{ [\![\gamma]\!] : \gamma \in \pi \} : \pi \in \Pi \} .$$

Claim: If $\pi' \in \Pi'$, then $\cap\pi' \neq \varnothing$. Since $\pi \subseteq \Gamma$ is a consistent subset of Γ (it is a cell in a cover of Γ), by the observation above there is $x \in W$ such that $\mathcal{M}, x \vDash \pi$. Thus, $x \in \cap\pi'$. Hence $\Pi'' = \{ \cap\pi' : \pi' \in \Pi' \}$ is a cover of $[\![\Gamma]\!]_{\mathcal{M}}$ and its width is $\ell(\Gamma)$ by construction. Thus $\ell([\![\Gamma]\!]_{\mathcal{M}}) \leq \ell(\Gamma)$. Since $\ell([\![\Gamma]\!]_{\mathcal{M}})$ cannot be less than $\ell(\Gamma)$, $\ell([\![\Gamma]\!]_{\mathcal{M}}) = \ell(\Gamma)$.

For a finite and purely Boolean Γ, consistency comprehensiveness can be expressible if we include a standard modal operator: $\Diamond\varphi$ meaning that φ is true at some "related" world. Although a relation could be added to interpret $\Diamond$, we will simply interpret $\Diamond$ as a global modality:

$$\mathcal{M}, x \vDash \Diamond\varphi \iff \text{ there is } w \in W \text{ s.t. } \mathcal{M}, w \vDash \varphi.$$

Now we can express consistency comprehensiveness. When Γ is finite and purely Boolean, let $\Diamond\mathbf{At}(\Gamma)$ be the formula:

$$\bigwedge_{\Gamma' \subseteq \mathbf{At}(\Gamma)} \Diamond((\bigwedge_{p \in \Gamma'} p) \wedge \bigwedge_{q \in \mathbf{At}(\Gamma) \setminus \Gamma'} (\neg q)).$$

So, for example, if $\mathbf{At}(\Gamma) = \{ p, q, r \}$, then $\Diamond\mathbf{At}(\Gamma)$ is

$$\begin{aligned} &\Diamond(p \wedge q \wedge r) \wedge \Diamond(p \wedge q \wedge \neg r) \wedge \Diamond(p \wedge \neg q \wedge r) \wedge \\ &\Diamond(\neg p \wedge q \wedge r) \wedge \Diamond(p \wedge \neg q \wedge \neg r) \wedge \Diamond(\neg p \wedge q \wedge \neg r) \wedge \\ &\Diamond(\neg p \wedge \neg q \wedge r) \wedge \Diamond(\neg p \wedge \neg q \wedge \neg r) \end{aligned}$$

As is easily verified, $\mathcal{M}$ satisfies $\Diamond\mathbf{At}(\Gamma)$ iff $\mathcal{M}$ is consistency comprehensive for Γ.

While the results above assumed a particular proof theoretic relation to define the syntactic covers, none of its specifics beyond being an extension of classical propositional logic were used. It can be replaced by any extension of CPL even one that is merely determined by a semantics. In the latter case we replace consistency with satisfiability and consequence with entailment, both relative to whatever semantics is being used. As long as the resulting (semantic) consequence relation is reflexive, transitive and monotonic, the syntactic covers and thus the level function will have all the necessary properties. This is fortunate since the subsequent extensions we have made to the language have not been given any sort of axiomatization so far. Nonetheless, the results above still hold for our new language which includes the U operator and $\Diamond/\Box$. With this observation in mind we can then show the following:

Lemma 1. *Suppose that* $\Gamma = \{ \gamma_1, \ldots, \gamma_n \}$. *If* $\mathcal{M}, x \vDash (E\gamma_1 \wedge \ldots \wedge E\gamma_n) \wedge U(\gamma_1, \ldots, \gamma_n; \top)$, *then* $\ell(\mathcal{E}(x)) = \ell([\![\Gamma]\!]_{\mathcal{M}})$. *If in addition* Γ *is purely boolean and* $\mathcal{M}$ *is consistency comprehensive for* Γ, $\ell(\Gamma) = \ell(\mathcal{E}(x))$.

Proof. From the previous observations and the definition of U.

Just as a reminder of the goal, we are trying to find a representation for forcing in terms of evidence logic in the same way that the K_n logics represent 'fixed-level forcing'. We have so far been able to find a way to express the level of a set of formulas, at least in the Boolean case (which is all we need).

To express that a formula is a forcing consequence we also need a way to canvas all the relevant covers of a set of sentences. While we will have to add an operator to the language to express the relevant relationship, it is expressible by a relation definable on the *frames* rather than on the models given. To define this relation we first need the idea of the *core* of $\mathcal{E}(x)$, denoted '$cor(\mathcal{E}(x))$' which is the set of any $\subseteq$-minimal elements of $\mathcal{E}(x)$. More precisely, $cor(\mathcal{E}(x)) = \{ X \in \mathcal{E}(x) : \nexists Y \in \mathcal{E}(x), Y \subsetneq X \}$, i.e., the set of elements of $\mathcal{E}(x)$ for which there is no proper subset also in $\mathcal{E}(x)$. A frame will be said to be *core complete* iff the core represents all the sets in $\mathcal{E}(x)$ in the sense that if $Y \in \mathcal{E}(x)$ there is some set $X \in cor(\mathcal{E}(x))$ such that $X \subseteq Y$.

It is fairly easy to see that $\ell(cor(\mathcal{X})) = \ell(\mathcal{X})$. What is also fairly easy to see is that if all the elements in the core are mutually exclusive, then the size of the core is the level of the set, i.e., $|cor(\mathcal{X})| = \ell(\mathcal{X})$, if for all distinct $X, Y \in cor(\mathcal{X})$, $X \cap Y = \varnothing$.

Now we define a relation $\text{cov}_{\mathfrak{F}} \subseteq W \times W$ as follows:

Definition 3. Let $\mathfrak{F} = \langle W, \mathcal{E} \rangle$ be a hyperframe. For all $x, y \in W$, $\text{cov}_{\mathfrak{F}}(x, y)$ holds iff

1. for all $X \in \mathcal{E}(x)$ there is $Y \in \mathcal{E}(y)$ such that $Y \subseteq X$,
2. for all $Y \in cor(\mathcal{E}(y))$ there is $X \in \mathcal{E}(x)$ such that $Y \subseteq X$, and
3. $|cor(\mathcal{E}(y))| = \ell(\mathcal{E}(x))$.

The idea is to have $\text{cov}_{\mathfrak{F}}(x, y)$ iff the "evidence set" at y forms a cover of minimal width of the evidence at x relative to $\mathfrak{F}$. So, if we were to look at all models on all frames we would be able to find all possible covers of Γ of width $\ell(\Gamma)$. We can now extend the language to include a new operator F to interpret the $\text{cov}_{\mathfrak{F}}$ relation on the frames, but the relation which interprets F needn't be all of $\text{cov}_{\mathfrak{F}}$. In fact, it needn't be a subset of $\text{cov}_{\mathfrak{F}}$ for the application that we have in mind. All that matters is that R_F and $\text{cov}_{\mathfrak{F}}$ agree when $\mathcal{E}(x)$ is finitely unifiable, but we will discuss this in more detail in Sect. 5. So, we can simply use a relation R_F on W which we will assume agrees with $\text{cov}_{\mathfrak{F}}$ on the relevant pairs (x, y).

$$\mathcal{M}, x \vDash F\varphi \Longleftrightarrow \forall w \in W, R_F(x, w), \mathcal{M}, w \vDash \varphi.$$

Define $\Gamma \vDash_{\mathbf{F}} \varphi$ iff for all hypermodels over the language defined so far with the semantics developed so far, if $\mathcal{M}, x \vDash \Gamma$, then $\mathcal{M}, x \vDash \varphi$. $\vDash_{\mathbf{F}} \varphi$ when φ is true at all worlds in all hypermodels. As discussed above, we can use $\vDash_{\mathbf{F}}$ to define syntactic covers and observations 1 and 2 will carry over to the current context. Just to be explicit about how that is done: Π is a syntactic cover of Γ relative to $\mathbf{F}$ iff for each $\pi \in \Pi$, π is satisfiable and for each $\gamma \in \Gamma$ there is $\pi \in \Pi$ such that $\pi \vDash_{\mathbf{F}} \gamma$.

Lemma 2. *Let $\mathcal{M}$ be a hypergraph model. If $\mathcal{M}$ covers and is unified by Γ at x and $\mathrm{cov}_{\mathfrak{F}}(x, y)$, then $\Pi_{\mathcal{E}(y)} = \{ \{ \varphi : Y \subseteq [\![\varphi]\!] \} : Y \in cor(\mathcal{E}(y)) \}$ is a syntactic cover of Γ (not necessarily a partition cover). If, in addition, Γ is pure boolean and $\mathcal{M}$ is consistency comprehensive for Γ, then the width of $\Pi_{\mathcal{E}(y)}$ is $\ell(\Gamma)$.*

Proof. Consider $\Pi_{\mathcal{E}(y)} = \{ \{ \varphi : Y \subseteq [\![\varphi]\!] \} : Y \in cor(\mathcal{E}(y)) \}$. Since $\mathcal{M}$ covers Γ at x, for each $\gamma \in \Gamma$, there is $X \in \mathcal{E}(x)$ such that $X \subseteq [\![\gamma]\!]$. By condition 1 in the definition of cov, there is $Y \in \mathcal{E}(y)$ such that $Y \subseteq X$ and, by definition of the core, there is $Y' \in cor(\mathcal{E}(y))$ such that $Y' \subseteq Y$. Thus, for any $\gamma \in \Gamma$, there is a $Y' \in cor(\mathcal{E}(y))$ such that $Y' \subseteq [\![\gamma]\!]$. That means, for each $\gamma \in \Gamma$, there is $\pi \in \Pi_{\mathcal{E}(y)}$ such that $\gamma \in \pi$, hence $\pi \vDash_F \gamma$. Furthermore, by definition of $\Pi_{\mathcal{E}(y)}$, for each $\pi \in \Pi_{\mathcal{E}(y)}$ there is a $Y \in cor(\mathcal{E}(y))$ such that $Y \subseteq \cap[\![\pi]\!]$ and of course $Y \neq \varnothing$ since $\mathcal{M}$ is a hypermodel and so $\varnothing \notin \mathcal{E}(y)$. Hence each π is satisfiable. Thus $\Pi_{\mathcal{E}(y)}$ is a syntactic cover of Γ. Also notice that the width of $\Pi_{\mathcal{E}(y)}$ is $|cor(\mathcal{E}(y))|$.

From observation 1 we know that $\ell(\mathcal{E}(x)) = \ell([\![\Gamma]\!])$ since $\mathcal{M}$ covers and is unified by Γ at x. So, $|cor(\mathcal{E}(y))| = \ell([\![\Gamma]\!])$ by condition 3 in the definition of cov. If we also assume that Γ is pure boolean and that $\mathcal{M}$ is consistency comprehensive for Γ, then by observation 2, $\ell([\![\Gamma]\!]) = \ell(\Gamma)$. Hence the width of $\Pi_{\mathcal{E}(y)}$ is $\ell(\Gamma)$.

Now we can ask the relevant question: is this logic one that allows us to capture classical forcing in at least the finite cases? The answer, fortunately, is 'yes'.

Theorem 1. *Suppose $\Gamma = \{ \gamma_1, \ldots, \gamma_m \}$ and φ are purely Boolean.*

$$\Gamma \Vdash \varphi \Longleftrightarrow \vDash_{\mathbf{F}} [(E\gamma_1 \wedge \ldots \wedge E\gamma_m) \wedge U(\gamma_1, \ldots, \gamma_m; \top) \wedge \Diamond \mathbf{At}(\Gamma)] \rightarrow FE\varphi$$

Proof. The only if direction follows by proving the contrapositive using Lemma 2. If $\nvDash_{\mathbf{F}} [(E\gamma_1 \wedge \ldots \wedge E\gamma_m) \wedge U(\gamma_1, \ldots, \gamma_m; \top) \wedge \Diamond \mathbf{At}(\Gamma)] \rightarrow FE\varphi$, then there is a model and world $\mathcal{M}, x$ such that $\mathcal{M}, x \vDash (E\gamma_1 \wedge \ldots \wedge E\gamma_m) \wedge U(\gamma_1, \ldots, \gamma_m; \top) \wedge \Diamond \mathbf{At}(\Gamma)$, but $\mathcal{M}, x \nvDash FE\varphi$. Since $\Diamond \mathbf{At}(\Gamma)$ is true at x, $\mathcal{M}$ is consistency comprehensive for Γ. And since $\mathcal{M}, x \nvDash FE\varphi$, there is a $y \in W$ such that $\mathrm{cov}(x, y)$ such that $\mathcal{M}, y \nvDash E\varphi$. Now we can apply Lemma 2 to $\mathcal{E}(y)$ and given that $\mathrm{cov}(x, y)$, $|cor(\mathcal{E}(y))| = \ell(\Gamma)$. Given that $|\Pi_{\mathcal{E}(y)}| = |cor(\mathcal{E}(y))|$, we get a cover of Γ such that each cell does not entail φ.

For the if direction again we will argue contrapositively. Assume that $\Gamma \nVdash \varphi$. Note that in this case, $\Vdash$ is forcing based on classical propositional logic.

Notice that since Γ is finite $\ell(\Gamma) = n \leq |\Gamma|$. Since Γ does not force φ there is a syntactic partition cover $\Pi = \{ \pi_i : 1 \leq i \leq n \}$ of Γ (of width n) such that for all $\pi \in \Pi$, $\pi \nvdash \varphi$. Hence there are n truth value assignments $v_1, \ldots, v_n$ such that for each $\pi \in \Pi$, there is $i \leq n$ such that $\vDash_{v_i} \pi$ and $\nvDash_{v_i} \varphi$ by the completeness of CPL with respect to two-valued truth value assignments.

Define $W_\Pi = \{ v : \exists X \subseteq \mathbf{At}(\Gamma), \forall p \in \mathbf{At}, p \in X \text{ only if } v(p) = T \}$; and $V_\Pi : \mathbf{At} \to \mathcal{P}(W)$ such that $V_\Pi(p) = \{ v \in W_\Pi : v(p) = T \}$. Finally, define $\mathcal{E}_\Pi(v)$ as

$$\mathcal{E}_\Pi(v) = \begin{cases} \{ [\![\gamma]\!] : \gamma \in \Gamma \} & v = v_1 \\ \{ \cap [\![\pi]\!] : \pi \in \Pi \} & v \neq v_1 \end{cases}$$

Let $\mathcal{M}_\Pi = \langle W_\Pi, \mathcal{E}_\Pi, V_\Pi \rangle$. In the case where Γ is consistent $\mathcal{E}(v) = \{ \cap [\![\Gamma]\!] \}$ for all $v \neq v_1$. Since all the v_i above are in W_Π, $W_\Pi \neq \varnothing$. Similarly, for all $v \in W$, and $X \in \mathcal{E}_\Pi(v)$, there is v_i from above such that $v_i \in X$. So, $\varnothing \notin \mathcal{E}_\Pi(v)$. As we have defined $\mathcal{M}_\Pi$, it is consistency comprehensive for Γ, and since $\mathcal{E}_\Pi(v_1) = \{ [\![\gamma]\!] : \gamma \in \Gamma \}$, $\mathcal{M}_\Pi, v_1 \vDash [(E\gamma_1 \wedge \ldots \wedge E\gamma_m) \wedge U(\gamma_1, \ldots, \gamma_m; \top) \wedge \Diamond \mathbf{At}(\Gamma)]$.

Let $\mathfrak{F}_\Pi = \langle W_\Pi, \mathcal{E}_\Pi \rangle$. Now we must exhibit at least one world that v_1 relates to by $\text{cov}_{\mathfrak{F}_\Pi}$ at which φ is false. Again since $\mathcal{M}_\Pi$ is consistency comprehensive for Γ, by Lemma 2,

$$\ell(\mathcal{E}_\Pi(v_1)) = \ell(cor(\mathcal{E}_\Pi(v_1))) = \ell([\![\Gamma]\!]) = \ell(\Gamma) = |\mathcal{E}_\Pi(v)|$$

for all $v \neq v_1$. Since for each $\gamma \in \Gamma$, there is $\pi \in \Pi$ such that $\pi \vDash \gamma$, $\cap [\![\pi]\!] \subseteq [\![\gamma]\!]$. But also, since each π is a consistent non-empty subset of Γ, there is $\gamma \in \Gamma$ such that $\cap [\![\pi]\!] \subseteq [\![\gamma]\!]$, since $\gamma \in \pi$. Thus, $\text{cov}_{\mathfrak{F}_\Pi}(v_1, v)$ for all $v \neq v_1$. So, in particular $\text{cov}_{\mathfrak{F}_\Pi}(v_1, v_2)$ and we can set $R_F = \text{cov}_{\mathfrak{F}_\Pi}$.

Finally, each of the v_i from above are such that $v_i \in \cap [\![\pi_i]\!]$ but $v_i \notin [\![\varphi]\!]$ for $i \leq n$; hence $\cap [\![\pi_i]\!] \nsubseteq [\![\varphi]\!]$. So, by definition, $\mathcal{M}_\Pi, v_2 \nvDash E\varphi$ and so $\mathcal{M}_\Pi, v_1 \nvDash FE\varphi$. Therefore, $\nvDash_F [(E\gamma_1 \wedge \ldots \wedge E\gamma_m) \wedge U(\gamma_1, \ldots, \gamma_m; \top) \wedge \Diamond \mathbf{At}(\Gamma)] \to FE\varphi$.

Thus, this logic allows one to represent classical forcing via a modal evidence logic. The next step, is to axiomatize the system. We will first give an axiomatization for a logic with the operators $E, \Box, F$, and U relative to all hypergraph models for which $\varnothing \notin \mathcal{E}(x) \neq \varnothing$. The logic **F** required by Theorem 1 is obtained by adding axioms to the logic **U** and is discussed in Sect. 5.

4 Semantics and Axiomatization for U

We start with the language $\mathcal{L}_{\mathbf{U}}$. It is defined by the following BNF:

$$\varphi := \bot \mid p \mid \neg\varphi \mid F\varphi \mid E\varphi \mid \Box\varphi \mid \varphi \to \varphi \mid U(\underbrace{\varphi, \ldots, \varphi}_{n-\text{times}}; \varphi) \; n \in Z^+$$

Where $p \in \mathbf{At}$ the set of atoms. The operators $\Diamond$, $\langle F \rangle$, and $\langle E \rangle$ are defined via their duals $\neg \blacksquare \neg \varphi$ for $\blacksquare \in \{ \Box, F, E \}$, and the other Boolean connectives are defined in the usual way. In the interest of limiting the number of operators to keep it in line with the literature on evidence logics we won't introduce additional neighbourhood operators like: for all $X \in \mathcal{E}(x), X \subseteq [\![\varphi]\!]$ which have been discussed elsewhere [10]. Next we have a frame and then a model:

Definition 4. A structure $\mathfrak{F} = \langle W, \mathcal{E} \rangle$ is a hypergraph frame iff:

1. $W \neq \varnothing$, and
2. $\mathcal{E} : W \to \mathcal{P}(\mathcal{P}(W))$ such that for all $x \in W$
   (a) $\varnothing \notin \mathcal{E}(x)$, and
   (b) $\mathcal{E}(x) \neq \varnothing$
3. R_F is a relation on W
4. The frame is **augmented** when there is an equivalence relation $R_{\Box} \subseteq W \times W$ added to the frame.

A hypergraph[9] **model** is a structure $\mathcal{M} = \langle \mathfrak{F}, V \rangle$ where $\mathfrak{F}$ is a hypergraph frame and $V : \mathbf{At} \to \mathcal{P}(W)$.

Let $\mathcal{M} = \langle \mathfrak{F}, V \rangle$ be a hypermodel. The semantics for the logic **U** for hypermodels is:

- $\mathcal{M}, x \vDash p$ iff $x \in V(p)$ for all $p \in \mathbf{At}$
- Boolean cases as usual,
- $\mathcal{M}, x \vDash E\varphi$ iff there is $X \in \mathcal{E}(x)$ such that $X \subseteq [\![\varphi]\!]$,
- $\mathcal{M}, x \vDash \langle E \rangle \varphi$ iff for all $X \in \mathcal{E}(x)$, $X \cap [\![\varphi]\!] \neq \varnothing$,
- $\mathcal{M}, x \vDash \Box\varphi$ iff $[\![\varphi]\!] = W$,
- $\mathcal{M}, x \vDash \Diamond\varphi$ iff $[\![\varphi]\!] \neq \varnothing$,
- $\mathcal{M}, x \vDash F\varphi$ iff $R_F(x) \subseteq [\![\varphi]\!]$,
- $\mathcal{M}, x \vDash U(\varphi_1, \ldots, \varphi_n; \psi)$ iff for all $X \in \mathcal{E}(x)$, $X \subseteq [\![\psi]\!]$ only if for some $i \leq n$, $[\![\varphi_i]\!] \subseteq X$

This semantics gives rise to a semantic consequence relation $\vDash_{\mathbf{U}}$, defined in the usual way. This system is complete with respect to the following axioms, which will give rise to the syntactic system $\vdash_{\mathbf{U}}$. In the following axioms $\overline{\varphi}$ refers to a tuple of formulas $\varphi_1, \ldots, \varphi_n$ as before, but in cases where it is not the only argument on the left of the ';' in a U operator it can be empty. $n!$ refers to all permutations of $\{1, 2, \ldots, n\}$ and σ will be a specific permutation in $n!$ where $\sigma(k)$ is the number that k is permuted to by the permutation σ. Let p, q, r, s, p_i be in **At**.

CL	All theorems of classical propositional logic.
S5	The axioms of S5 for $\Box$.
KF	$(Fp \wedge Fq) \longleftrightarrow F(p \wedge q)$
$\Box$F	$\Box p \to Fp$
D	$\neg E\bot$
N	$E\top$
E$\Box$	$\Box(p \to q) \to (Ep \to Eq)$
MergeE	$(Ep \wedge \Box q) \to E(p \wedge q)$
U$\bot$	$U(\bot; q)$

[9] We are using a 'hypergraph model' in the sense found in [9] rather than in [6]. Our hypergraph models are what they call neighbourhood models and what [5] calls 'Minimal Models'. Topologically speaking, it would make more sense to call neighbourhood models those minimal models $\langle W, \mathcal{E} \rangle$ in which for each $x \in W$, $x \in \bigcap \mathcal{E}(x)$ since a neighbourhood of x would usually contain x.

U! $U(p_1,\dots,p_n;\psi) \to (\bigwedge_{\sigma\in n!} U(p_{\sigma(1)},\dots,p_{\sigma(n)};q))$
UE $\neg U(\overline{p};q) \to Eq$
U+ $U(\overline{p};q) \to U(\overline{p},r;q)$
U- $U(\overline{p},r,r;q) \to U(\overline{p},r;q)$
UV $(U(\overline{p};q) \wedge Eq) \to \bigvee_{i=1}^{n} \Box(p_i \to q)$
U□R $\Box(q \to r) \to (U(\overline{p};r) \to U(\overline{p};q))$
U□L $\Box(q \to r) \to (U((\overline{p}/r)_i;s) \to U((\overline{p}/q)_i;s))$
With rules
US Uniform Substitution,
MP Modus Ponens,
Nec $\vdash \varphi$ only if $\vdash \Box\varphi$
UInf

$$\frac{\vdash \theta \to (\Box(p \to \psi) \to (\bigwedge_{j=1}^{n} \Diamond(\varphi_j \wedge \neg p) \to \neg Ep))}{\vdash \theta \to U(\varphi_1,\dots,\varphi_n;\psi)} \; p \text{ foreign to } \varphi_1,\dots,\varphi_n,\psi,\theta$$

The usual definitions for Hilbert-style proof theory are used: $\Gamma \vdash_{\mathbf{U}} \varphi$ iff there are $\gamma_1,\dots,\gamma_k \in \Gamma$ such that $\vdash_{\mathbf{U}} (\gamma_1 \wedge \dots \wedge \gamma_n) \to \varphi$. As will be shown in Sect. 6:

Theorem 2. *The system* $\vdash_{\mathbf{U}}$ *is sound and complete with respect to* $\vDash_{\mathbf{U}}$.

A few comments about the system are in order. The axiomatization is obviously not finite, but it is recursive. We can also treat the tuple of formulas before the semicolon in the U operators as a set given axioms U! and U-. The contrapositive of UV is equivalent to $\bigwedge_{i=1}^{n} \Diamond(\varphi_i \wedge \neg\psi) \to (E\psi \to \neg U(\overline{\varphi};\psi))$, and given UE, $\bigwedge_{i=1}^{n} \Diamond(\varphi_i \wedge \neg\psi) \to (E\psi \longleftrightarrow \neg U(\overline{\varphi};\psi))$ is derivable for any formulas $\overline{\varphi},\psi$. That also indicates how to interpret the UInf rule. UInf formalizes the idea that if no proposition that both implies ψ and is not implied by any of the φ_is in $U(\varphi_1,\dots,\varphi_n;\psi)$, can be in an evidence set at a world when θ is also true, then $U(\varphi_1,\dots,\varphi_n;\psi)$ must be true.

5 Definability and the Logic F

The first thing we will point out is that we know the logic **U** is distinct from Instantial Neighbourhood Logic (INL) of [3]. The reason for this is that using the U and E operators we can define $\Box$ in the context of the $E\Box$ axiom[10]:

$$(U(\neg\varphi;\varphi) \wedge E\varphi) \longleftrightarrow \Box\varphi$$

[10] A syntactic derivation of this equivalence proceeds as follows: Suppose $U(\neg\varphi;\varphi) \wedge E\varphi$. An instance of UV is $(U(\neg\varphi;\varphi) \wedge E\varphi) \to \Box(\neg\varphi \to \varphi)$, so we can infer $\Box(\neg\varphi \to \varphi)$ which is equivalent to $\Box\varphi$ in any normal modal logic. Conversely, suppose $\Box\varphi$. Thus, in any normal modal logic $\Box\varphi \to \Box(\neg\varphi \to \bot)$ is a theorem. By U⊥, $U(\bot;\varphi)$ is a theorem of U, and by U□L, $\Box(\neg\varphi \to \bot) \to (U(\bot;\varphi) \to U(\neg\varphi;\varphi))$ is a theorem and thus, $U(\neg\varphi;\varphi)$ follows. Since $\Box\varphi \to \Box(\top \to \varphi)$ is a theorem of any normal modal logic, using N, and E□, we can derive $E\varphi$.

While the E operator can be defined in INL—it is a special case of it—the authors show that $\Box$ is not definable in INL. Although this means that $\Box$ isn't needed in **U**, it is convenient to treat it as separate.

The system U is complete with respect to the class of all hypermodels. But the system needed to meet the requirements for the proof of Theorem 1 asks more of the relation R_F which interprets the F operator. The condition that is sufficient for Theorem 1 is the following: If $\mathcal{M}$ is a hypergraph model based on the frame $\mathfrak{F} = \langle W, \mathcal{E} \rangle$, then for all $x \in W$, $\mathcal{E}(x)$ is finitely unifiable only if for all $y \in W$ such that $R_F(x, y)$, $\text{cov}_{\mathfrak{F}}(x, y)$. I.e., when $\mathcal{E}(x)$ is finitely unifiable, all the R_F-realted worlds are minimal covers of $\mathcal{E}(x)$.

The task is to find axioms which guarantee that the conditions in the definition of $\text{cov}_{\mathfrak{F}}(x, y)$ are met. Thus, we need to show that if, $R_F(x, y)$ and $\mathcal{E}(x)$ is finitely unifiable, then 1) for all $X \in \mathcal{E}(x)$ there is $Y \in \mathcal{E}(y)$ such that $Y \subseteq X$, 2) for all $Y \in cor(\mathcal{E}(y))$ there is $X \in \mathcal{E}(x)$ such that $Y \subseteq X$, and 3) $|cor(\mathcal{E}(y))| = \ell(\mathcal{E}(x))$.

These requirements can be achieved by imposing axioms which define certain properties of the frames, since, after all, the properties that are required depend on the frames rather than the models. As per usual, a formula α is valid on a frame $\mathfrak{F} = \langle W, \mathcal{E}, R_F \rangle$ iff for all models $\mathcal{M}$ based on $\mathfrak{F}$, and all $x \in W$, $\mathcal{M}, x \vDash \alpha$, and we will denote that α is valid on $\mathfrak{F}$ by $\mathfrak{F} \vDash \alpha$.

Ensuring that condition 1 is met requires a fairly simple axiom which we refer to as EF: $Ep \rightarrow FEp$.

Proposition 1. *Let $\mathfrak{F}$ be a hyperframe. $\mathfrak{F} \vDash Ep \rightarrow FEp$ iff for all $x, y \in W$, if $R_F(x, y)$, then for all $X \in \mathcal{E}(x)$ there is $Y \in \mathcal{E}(y)$ such that $Y \subseteq X$. The proof is standard and uncomplicated, so we will omit it.*

To capture the other conditions we will first define some operators as abbreviations to simplify the expression of the axioms. One of the first things that we can notice is the one can express that a (finite) set of formulas forms a cover of $\mathcal{E}(x)$. We define $\text{cov}(\overline{p})$:

$$\text{cov}(p_1, \ldots, p_n) := \bigwedge_{i=1}^{n} \Diamond p_i \wedge U(p_1, \ldots, p_n; \top)$$

When $\text{cov}(\overline{\varphi})$ is true at $x \in W$, then $\overline{\varphi}$ unifies $\mathcal{E}(x)$ so $\{ [\![\varphi]\!] : \varphi \in \overline{\varphi} \}$ could serve as a semantic cover for $\mathcal{E}(x)$ since none of the $[\![\varphi]\!]$ is empty, but not necessarily a partition cover. But $\overline{\varphi}$ may not strongly unify $\mathcal{M}$ at x when $\text{cov}(\overline{\varphi})$ is true.

The next operator indicates that the extensions of the formulas to which it applies are found in the core of $\mathcal{E}(x)$:

$$\text{core}(p_1, \ldots, p_n) := \bigwedge_{i=1}^{n} (Ep_i \wedge U(p_i; p_i))$$

The ability to express that the extension of a formula is in the core of an evidence set is a great side-effect of making the U operator parallel with those found in the

Instantial Neighbourhood Logic of [3]. Without the operator's "point"—the formula after the semicolon—we could not guarantee that, when Ep is also true, $[\![p]\!]$ is the only element of $\mathcal{E}(x)$ which is contained in $[\![p]\!]$. If we add $U(p_1,\ldots,p_n;\top)$ to $\text{core}(p_1,\ldots,p_n)$, we get a formula that expresses that $\mathcal{E}(x)$ contains a cover of itself as its core, i.e., $cor(\mathcal{E}(x)) = \{\, [\![p_i]\!] : i \leq n\,\}$. This operator expresses that the sequence of formulas constitute the entire core of $\mathcal{E}(x)$:

$$\text{totalcore}(p_1,\ldots,p_n) := \bigwedge_{i=1}^{n}(Ep_i \wedge U(p_i;p_i)) \wedge U(p_1,\ldots,p_n;\top).$$

To capture conditions 2 and 3 in the definition of $\text{cov}_{\mathfrak{F}}(x,y)$ we use recursive sets of formulas. While EF provided condition 1 without the assumption that $\mathcal{E}(x)$ is finitely unifiable, our next "axioms" make that assumption explicit.

While we usually work with individual axioms or collections of various axioms to define frame conditions, the following "axioms" are actually recursive sets of formulas. Define the set of formulas Cor by

$$\text{Cor} := \left\{ \text{totalcore}(p_1,\ldots,p_n) \to (\langle F\rangle\, \text{core}(q) \to \bigvee_{i=1}^{n} \Box(q \to p_i)) : n > 0 \;\&\; p_i, q \in \mathbf{At} \right\}$$

Proposition 2. *Let $\mathfrak{F}$ be a hyperframe. $\mathfrak{F} \vDash Cor$ iff for all $x, y \in W$ if $R_F(x,y)$ and $\mathcal{E}(x)$ has a finite and non-empty core, then for all $Y \in cor(\mathcal{E}(y))$ there is $X \in cor(\mathcal{E}(x))$ such that $Y \subseteq X$.*

The condition on frames above is stronger than what condition 2 requires since it says that for each set in the core of any world that x R_F-relates to will imply all the elements of the core of $\mathcal{E}(x)$, provided there is a core. That could pose a problem since condition 2 doesn't require that all evidence sets have cores. However, as we discuss at the end of Sect. 6.2, **U** is complete with respect to the class of core-complete hyperframes, so we can restrict attention to only core-complete hyperframes in this context as well. In addition, the proof of Theorem 1 only use a model which is core-complete, so the assumption of core-completeness leaves all results intact.

Now we can home in on finding conditions for $|cor(\mathcal{E}(y))| = \ell(\mathcal{E}(x))$. The first thing to notice is that in section three the results were limited to finite cases, so while there can be evidence sets which have infinite levels, we are setting those to the side for the moment. We shall define another set of formulas UpLev as follows:

$$\begin{aligned}\text{UpLev} = \{&\text{cov}(q_1,\ldots,q_k) \to (\langle F\rangle\, \text{totalcore}(p_1,\ldots,p_n) \to U(p_1,\ldots,p_n;\top)) :\\ &n,k \in \mathbb{N} \;\&\; p_i, q_j \in \mathbf{At}\}\end{aligned}$$

Proposition 3. *Let $\mathfrak{F}$ be a hyperframe. $\mathfrak{F} \vDash$ UpLev iff for all $x, y \in W$, if $cor(\mathcal{E}(y))$ and $\ell(\mathcal{E}(x))$ are finite, then $R_F(x,y)$ only if $cor(\mathcal{E}(y))$ is a cover of $\mathcal{E}(x)$.*

The effect of this result is to enforce an upper bound on $\ell(\mathcal{E}(x))$ when it is finite; hence the name. Notice that if $|cor(\mathcal{E}(y))|$ is finite and $R_F(x, y)$ in an UpLev-frame, i.e., a frame $\mathfrak{F}$ where all formulas in UpLev are valid on $\mathfrak{F}$, then $\ell(\mathcal{E}(x)) \leq |cor(\mathcal{E}(y))|$. That follows since if $cor(\mathcal{E}(y))$ is a cover of $\mathcal{E}(x)$, then the level of $\ell(\mathcal{E}(x))$ can't be any larger than the size of that cover. What is needed, then, is a lower bound. For that we define:

$$\mathrm{LowLev} = \left\{ \mathrm{cov}(r_1, \ldots, r_n) \to (\langle F\rangle\, \mathrm{core}(p_1, \ldots, p_k) \to (U(q_1, \ldots, q_m; \top) \to \bigvee_{i=1}^{m} \neg\Diamond q_i)) : n, k, m \in \mathbb{N} \;\&\; m < k \right\}$$

Proposition 4. *Let $\mathfrak{F}$ be an hyperframe. $\mathfrak{F} \vDash$ LowLev iff for all $x, y \in W$, if $R_F(x, y)$, then $\ell(\mathcal{E}(x)) \geq |cor(\mathcal{E}(y))|$ when $\ell(\mathcal{E}(x))$ is finite.*

Proof. Suppose that $\mathfrak{F}$ is a hyperframe such that for all $x, y \in W$, if $R_F(x, y)$, then $\ell(\mathcal{E}(x)) \in \mathbb{N}$ only if $\ell(\mathcal{E}(x)) \geq |cor(\mathcal{E}(y))|$. Now suppose that $\mathcal{M}$ is a model based on $\mathfrak{F}$ and that $x \in W$ such that $\mathcal{M}, x \vDash \mathrm{cov}(r_1, \ldots, r_n) \wedge \langle F\rangle\, \mathrm{core}(p_1, \ldots, p_k) \wedge U(q_1, \ldots, q_m; \top)$ where $m < k$. From $\mathcal{M}, x \vDash \mathrm{cov}(r_1, \ldots, r_n)$, we can infer that $\ell(\mathcal{E}(x))$ is finite and from $\mathcal{M}, x \vDash \langle F\rangle\, \mathrm{core}(p_1, \ldots, p_k)$ we can infer that there is $y \in W$ such that $R_F(x, y)$ (and that $\mathcal{M}, y \vDash \mathrm{core}(p_1, \ldots, p_k)$). Thus, by our assumption about $\mathfrak{F}$, $\ell(\mathcal{E}(x)) \geq |cor(\mathcal{E}(y))|$. Since $\mathcal{M}, y \vDash \mathrm{core}(p_1, \ldots, p_k)$, $\{ [\![p_i]\!] : i \leq k \} \subseteq cor(\mathcal{E}(y))$, hence, $|cor(\mathcal{E}(y))| \geq k$. Now suppose for reductio that for each $i \leq m$, $[\![q_i]\!] \neq \varnothing$. Since $\mathcal{M}, x \vDash U(q_1, \ldots, q_m; \top)$, $\{ [\![q_i]\!] : i \leq m \}$ is a cover of $\mathcal{E}(x)$. In general, if $\mathcal{X}$ is a cover of $\mathcal{Y}$, then $\ell(\mathcal{Y}) \leq \ell(\mathcal{X})$, and $\ell(\mathcal{X}) \leq |\mathcal{X}|$. Thus, $\ell(\mathcal{E}(x)) \leq \ell(\{ [\![q_i]\!] : i \leq m \}) \leq |\{ [\![q_i]\!] : i \leq m \}| \leq m$. Thus,

$$\ell(\mathcal{E}(x)) \geq |cor(\mathcal{E}(y))| \geq k > m \geq \ell(\{ [\![q_i]\!] : i \leq m \}) \geq \ell(\mathcal{E}(x)),$$

a contradiction. So, some $[\![q_i]\!] = \varnothing$. Therefore, $\mathcal{M}, x \vDash \bigvee_{i=1}^{m} \neg\Diamond q_i$. Since, n, m, and k were arbitrary as was the model $\mathcal{M}$ based on $\mathfrak{F}$, $\mathfrak{F} \vDash$ LowLev.

Conversely, suppose that $\mathfrak{F}$ is such that there are $x, y \in W$ such that $R_F(x, y)$ and $\ell(\mathcal{E}(x))$ is finite, but that $\ell(\mathcal{E}(x)) < |cor(\mathcal{E}(y))|$. Since $\ell(\mathcal{E}(x))$ is finite suppose it is n and then suppose that $\{ X_1, \ldots, X_n \}$ is a cover of minimal width of $\mathcal{E}(x)$. Suppose that $\{ Y_1, \ldots, Y_{n+1} \} \subseteq cor(\mathcal{E}(y))$ which must exist since $|cor(\mathcal{E}(y))| > n$. Define $\mathcal{M}$ in which $V(r_i) = X_i = V(q_i)$ for $i \leq n$ and $V(p_j) = Y_j$ for $j \leq n + 1$. Since $n < n + 1$, the formula:

$$\mathrm{cov}(r_1, \ldots, r_n) \to (\langle F\rangle\, \mathrm{core}(p_1, \ldots, p_{n+1}) \to (U(q_1, \ldots, q_m; \top) \to \bigvee_{i=1}^{n} \neg\Diamond q_i))$$

is in LowLev. Furthermore, since the X_i's form a cover of $\mathcal{E}(x)$ none of them is empty nor are any of the Y_js since they are from the core of $\mathcal{E}(y)$. As we have assumed that $R_F(x, y)$, $\mathcal{M}, y \vDash \mathrm{core}(p_1, \ldots, p_{n+1})$ and so $\mathcal{M}, x \vDash \langle F\rangle\, \mathrm{core}(p_1, \ldots, p_{n+1})$. And as we have assumed the X_is are a cover of $\mathcal{E}(x)$, $\mathcal{M}, x \vDash \mathrm{cov}(r_1, \ldots, r_n)$, but also as part of that $\mathcal{M}, x \vDash U(q_1, \ldots, q_n)$. However, since none of the X_is is empty $\mathcal{M}, x \nvDash \bigvee_{i=1}^{n} \neg\Diamond q_i$. Thus

$$\mathcal{M}, x \nvDash \mathrm{cov}(r_1, \ldots, r_n) \to (\langle F\rangle\, \mathrm{core}(p_1, \ldots, p_{n+1}) \to (U(q_1, \ldots, q_m; \top) \to \bigvee_{i=1}^{n} \neg\Diamond q_i))$$

and so $\mathfrak{F} \nvDash$ LowLev.

Suppose now that $\mathfrak{F}$ is a (core-complete) frame on which EF, UpLev, LowLev, and Cor are all valid. If $x \in W$ and $\ell(\mathcal{E}(x))$ is finite, then for any $y \in W$ such that $R_F(x, y)$, $|cor(\mathcal{E}(y))|$ must also be finite by LowLev. Hence, by UpLev, if $R_F(x, y)$, $cor(\mathcal{E}(y))$ must be a cover of $\mathcal{E}(x)$. Thus, $|cor(\mathcal{E}(y))| = \ell(\mathcal{E}(x))$.

We will refer to a hyperframe $\mathfrak{F}$ which is an EF, LowLev, UpLev, and Cor frame as a **forcing**-frame. If $\mathfrak{F}$ is a forcing-frame, then if $x \in W$ and $\ell(\mathcal{E}(x))$ is finite, then $R_F(x, y)$ only if $\mathrm{cov}_{\mathfrak{F}}(x, y)$. Thus, the relation $\vDash_{\mathbf{F}}$ and its underlying semantics needed to prove Theorem 1 is the class of forcing frames. As an example of a forcing frame, one can consider the frame constructed in the proof of Theorem 1.

We can then get the proof theory of the logic **F** by adding to the logic **U** the additional axioms:

EF $Ep \to FEp$
Cor $\mathrm{totalcore}(p_1, \ldots, p_n) \to (\langle F\rangle\, \mathrm{core}(q) \to \bigvee_{i=1}^{n} \Box(q \to p_i))$ where $n > 0$
UpLev $\mathrm{cov}(q_1, \ldots, q_k) \to (\langle F\rangle\, \mathrm{totalcore}(p_1, \ldots, p_n) \to U(p_1, \ldots, p_n; \top))$ where $n > 0$
LowLev $\mathrm{cov}(r_1, \ldots, r_n) \to (\langle F\rangle\, \mathrm{core}(p_1, \ldots, p_k) \to (U(q_1, \ldots, q_m; \top) \to \bigvee_{i=1}^{m} \neg\Diamond q_i))$ where $m < k$ and $n > 0$

6 Soundness and Completeness of U

6.1 Soundness

The validity of most of the axioms is straightforward. The $\Box$ operator is supposed to be a global necessity, and F is, at this point, just a normal modal operator. D ensures that $\varnothing \notin \mathcal{E}(x)$ and N ensures that $\mathcal{E}(x) \neq \varnothing$. The E operator is a classical modal operator in Segerberg's sense, hence E$\Box$. The other things to notice is that since $\Box$ is global necessity, the truth of $\Box(\varphi \to \psi)$ anywhere in a model translates to $[\![\varphi]\!] \subseteq [\![\psi]\!]$.

U$\bot$ is valid becuase $\varnothing$ is a subset of any set and U! is valid because the disjunction used to give the truth condition of U is communative. Similarly, U+ and U- are valid because of properties of disjunction. The U$\Box$ axioms show that the operator is anti-monotonic on both the left and right side of ';' and follows because of the transitivity of the subset relation. The validity of UE can be seen by inspecting the truth condition for U and noticing that it is a conditional with $X \subseteq [\![\psi]\!]$ as its antecedent. Finally, UV is valid again because of the transitivity of the subset relation.

The only really interesting inference rule/axiom is UInf, and to prove that it is sound we need the following standard fact. Say that $\mathcal{M} = \langle W, R_F, \mathcal{E}, V\rangle$ and $\mathcal{M}' = \langle W', R'_F, \mathcal{E}', V'\rangle$ differ at most on $p \in \mathbf{At}$ iff $W = W'$, $\mathcal{E} = \mathcal{E}'$, $R_F = R'_F$ and $V(q) = V'(q)$ for all $q \neq p$ from **At**. Then we have that:

Lemma 3. *If $\mathcal{M}$ and $\mathcal{M}'$ differ at most on p, then $[\![\varphi]\!]_{\mathcal{M}} = [\![\varphi]\!]_{\mathcal{M}'}$ for all φ which do not mention p.*

Proof. The usual induction on the complexity of φ.

Proposition 5. UInf *is sound.*

Proof. Suppose that p is foreign to all of $\varphi_1, \ldots, \varphi_n, \psi, \theta$ and $\nvDash \theta \rightarrow U(\varphi_1, \ldots, \varphi_n; \psi)$. So there is a model $\mathcal{M} = \langle W, R_F, \mathcal{E}, V \rangle$ and $x \in W$ such that $\mathcal{M}, x \vDash \theta$, but $\mathcal{M}, x \nvDash U(\overline{\varphi}; \psi)$. By definition there is $X \in \mathcal{E}(x)$ such that $X \subseteq [\![\psi]\!]_{\mathcal{M}}$ and $[\![\varphi_j]\!] \nsubseteq X$ for all $j \leq n$. The last fact means that $[\![\varphi_j]\!]_{\mathcal{M}} \cap X^c \neq \varnothing$ for all $j \leq n$. Define $\mathcal{M}'$ to be just like $\mathcal{M}$ other than $V'(p) = X$. Then, since $\mathcal{M}$ and $\mathcal{M}'$ differ at most on p, by the lemma above, $[\![\theta]\!]_{\mathcal{M}} = [\![\theta]\!]_{\mathcal{M}'}$, $[\![\psi]\!]_{\mathcal{M}} = [\![\psi]\!]_{\mathcal{M}'}$, and $[\![\varphi_j]\!]_{\mathcal{M}} = [\![\varphi_j]\!]_{\mathcal{M}'}$ for all $j \leq n$. Immediately we have $\mathcal{M}', x \vDash \theta$. Furthermore, $[\![\varphi_j]\!]_{\mathcal{M}'} \cap [\![p]\!]^c_{\mathcal{M}'} \neq \varnothing$ for all $j \leq n$, so $[\![\varphi_j]\!]_{\mathcal{M}'} \cap [\![\neg p]\!]_{\mathcal{M}'} \neq \varnothing$ for all $j \leq n$. Hence, $\mathcal{M}', x \vDash \bigwedge_{j=1}^{n} \Diamond(\varphi_j \wedge \neg p)$. Since

$$[\![p]\!]_{\mathcal{M}'} = X \subseteq [\![\psi]\!]_{\mathcal{M}} = [\![\psi]\!]_{\mathcal{M}'},$$

$\mathcal{M}', x \vDash \Box(p \rightarrow \psi)$. Since $[\![p]\!]_{\mathcal{M}'} = X \in \mathcal{E}(x) = \mathcal{E}'(x)$, there is an $X \in \mathcal{E}'(x)$ such that $X \subseteq [\![p]\!]_{\mathcal{M}'}$, thus $\mathcal{M}', x \vDash Ep$, i.e. $\mathcal{M}', x \nvDash \neg Ep$. Therefore, $\nvDash \theta \rightarrow (\Box(p \rightarrow \psi) \rightarrow (\bigwedge_{j=1}^{n} \Diamond(\varphi_j \wedge \neg p) \rightarrow \neg Ep))$.

6.2 Completeness

The completeness proof resembles the Henkin-style completeness proofs for first-order and hybrid logics in that the domain of the canonical model isn't simply the collection of all maximally consistent sets of formulas. The sets need to have an additional property since $\neg U(\overline{\varphi}; \psi)$ can be true, while there is no formula which witnesses that fact, i.e., no formula θ such that $E\theta$, $\Box(\theta \rightarrow \psi)$ and $\bigvee_{j=1}^{n} \Diamond(\varphi_j \wedge \neg\theta)$ are all in the set. Naturally, the fix is to choose maximally consistent subsets Γ which are "filled-up" with enough formulas to witness each case where $\neg U(\overline{\varphi}, \psi) \in \Gamma$. We will call sets with this property U-**saturated**. Let $\Phi(U(\overline{\varphi}; \psi), p) = \left\{ Ep, \Box(p \rightarrow \psi), \bigwedge_{j=1}^{n} \Diamond(\varphi_j \wedge \neg p) \right\}$.

Proposition 6. *Each U-consistent set of sentences Γ can be extended to a maximally consistent and U-saturated set of sentences Γ^+.*

Proof. Suppose Γ is an U-consistent set of sentences. Then let $\{ p_i : i \in \mathbb{N} \}$ be a set of atoms not mentioned in Γ. Define a new language which includes the language of Γ and the new atoms and let $\{ \psi_i : i \in \mathbb{Z}^+ \}$ be an enumeration of that language. Then we define the following sequence of sets. Let $\Gamma_0 = \Gamma$ and

$$\Gamma_n = \begin{cases} \Gamma_{n-1} \cup \{ \neg\psi_n \} & \Gamma_{n-1} \cup \{ \psi_n \} \vdash \bot \\ \Gamma_{n-1} \cup \{ \psi_n \} & \Gamma_{n-1} \cup \{ \psi_n \} \nvdash \bot \ \& \ \psi_n \neq \neg U(\overline{\varphi}, \psi), \text{ or} \\ \Gamma_{n-1} \cup \{ \psi_n \} \cup \Phi(\psi_n, p_i) & \text{where } i \text{ is the least } i \text{ such that } p_i \text{ is not mentioned in } \Gamma_{n-1} \cup \{ \psi_n \} \end{cases}$$

We can see that each Γ_n is consistent by induction on n. The only case that is non-standard to see this in the inductive step is the third clause in the definition of Γ_n.

Suppose for reductio that $\Gamma_n = \Gamma_{n-1} \cup \{ \psi_n \} \cup \Phi(\psi_n, p) \vdash \bot$ where p is the first p_i not mentioned in $\Gamma_{n-1} \cup \{ \psi_n \}$. That can happen only when $\Gamma_{n-1} \cup \{ \psi_n \} \nvdash \bot$. By definition of ψ_n and $\Phi(\psi_n, p)$, then,

$$\Gamma_{n-1}, \neg U(\overline{\varphi}; \psi), Ep, \Box(p \to \psi), \bigwedge_{j=1}^{n} \Diamond(\varphi_j \wedge \neg p) \vdash \bot.$$

By the definition of provability, there is a finite subset of Γ_{n-1}, Γ' such that $\Gamma', \neg U(\overline{\varphi}; \psi), Ep, \Box(p \to \psi), \bigwedge_{j=1}^{n} \Diamond(\varphi_j \wedge \neg p) \vdash \bot$. It then follows by classical logic that
$\Gamma', Ep, \Box(p \to \psi), \bigwedge_{j=1}^{n} \Diamond(\varphi_j \wedge \neg p) \vdash U(\overline{\varphi}; \psi)$ and by U□R, that $\Gamma', Ep, \Box(p \to \psi), \bigwedge_{j=1}^{n} \Diamond(\varphi_j \wedge \neg p) \vdash U(\overline{\varphi}; p)$. By the contrapositive of UV, $\vdash \bigwedge_{j=1}^{n} \Diamond(\varphi_j \wedge \neg p) \to (Ep \to \neg U(\overline{\varphi}; p))$ so by MP, and the transitivity and monotonicity of $\vdash$, $\Gamma', Ep, \Box(p \to \psi), \bigwedge_{j=1}^{n} \Diamond(\varphi_j \wedge \neg p) \vdash \neg U(\overline{\varphi}; p)$ hence $\Gamma', Ep, \Box(p \to \psi), \bigwedge_{j=1}^{n} \Diamond(\varphi_j \wedge \neg p)$ is inconsistent. But that means, by classical logic, that

$$\Gamma', \Box(p \to \psi) \vdash \bigwedge_{j=1}^{n} \Diamond(\varphi_j \wedge \neg p) \to \neg Ep$$

and so by UInf $\Gamma' \vdash U(\overline{\varphi}; \psi)$. But that implies that $\Gamma_{n-1}, \neg U(\overline{\varphi}; \psi) \vdash \bot$ contrary to assumption. Naturally, let $\Gamma^+ = \bigcup_{i \in \mathbb{N}} \Gamma_n$.

To define the canonical model, we will start with the set of all maximally U-consistent and U-saturated sets of sentences, but we will always select only all the $R_\Box$-related worlds in order for the $\Box$ operator to represent global necessity. The canonical model $\mathcal{M}^* = \left\langle W^*, R^*_\Box, R^*_F, \mathcal{E}^*, V^* \right\rangle$ is augmented and defined in the following way:

- W^* is the set of maximally U-consistent and saturated sets of formulas,
- $R^*_\Box(x) = \{ y \in W^* : \forall \psi, \Box\psi \in x \Rightarrow \psi \in y \}$,
- $R^*_F(x) = \{ y \in W^* : \forall \psi, F\psi \in x \Rightarrow \psi \in y \}$,
- $V^*(p) = \{ x \in W^* : p \in x \}$, and
- $\mathcal{E}^*$ is defined by: $X \in E^*(x)$ iff there is $\{ \theta_i : i \in I \} \subseteq \{ \theta : E\theta \in x \}$ such that
  (a) $\bigcap_{i \in I} |\theta_i| = X$, and
  (b) For all δ, if $\bigcap_{i \in I} |\theta_i| \subseteq |\delta|$, then $E\delta \in x$

Observation 3. *If $E\theta \in x$ then $|\theta| \in \mathcal{E}^*(x)$.*

Proof. This follows since if $|\theta| \subseteq |\delta|$, then $\vdash \theta \to \delta$, so $\vdash \Box(\theta \to \delta)$, hence if $E\theta \in x$, $E\delta \in x$. Thus $\{ |\theta| \}$ satisfies conditions (a) and (b).

From the canonical model we can define the model which will be used for counterexamples. For each $y \in W^*$ define $\mathcal{M}^{*,y}$ as follows:

- $W^{*,y} = R^*_\Box(y)$,
- $R^{*,y}_F(x) = R^*_F(x) \cap W^{*,y}$,
- $\mathcal{E}^{*,y}(x) = \{ X \cap W^{*,y} : X \in \mathcal{E}^*(x) \}$, and

– $V^{*,y}(p) = V^*(p) \cap W^{*,y}$.

It is possible to give alternative representations of $W^{*,y}$. For example, $W^{*,y} = \{ z \in W^* : \Box(y) \subseteq z \}$, where $\Box(y) = \{ \varphi : \Box\varphi \in y \}$. Since $\Box$ is an S5 operator it follows that if $z \in W^{*,y}$, then $\Box(z) = \Box(y)$. So we can also represent $W^{*,y}$ as $\{ z \in W^* : \Box(z) = \Box(y) \}$. In fact, since $\Box$ is an S5 operator, for all $z \in W^{*,y}$, $\Box\varphi \in z$ iff $\Box\varphi \in y$, i.e., all elements of $W^{*,y}$ agree on $\Box$ed formulas. We can also show the following:

Lemma 4. *If $x \in W^{*,y}$ for some $y \in W^*$, then all $X \in \mathcal{E}^{*,y}(x)$ are non-empty.*

Proof. Let $X \in \mathcal{E}^{*,y}(x)$. Suppose that $X = \varnothing$. By definition, there is $\{ \theta_i \}_{i \in I} \subseteq \{ \theta : E\theta \in x \}$ such that $\bigcap_{i \in I} |\theta_i| \in \mathcal{E}^*(x)$ and $\bigcap_{i \in I} |\theta_i| \cap W^{*,y} = X = \varnothing$. Thus, $\bigcap_{i \in I} |\theta_i| \cap W^{*,y} = \varnothing$ iff $\{ \theta_i : i \in I \} \cup \Box(y) \vdash \bot$. By the compactness of $\vdash$, then there are $\{ \varphi_1, \dots, \varphi_k \} \subseteq \Box(y)$ and $\{ \theta_1, \dots, \theta_n \} \subseteq \{ \theta_i : i \in I \}$ such that $\bigwedge_{i=1}^k \varphi_i \wedge \bigwedge_{j=1}^n \theta_j \vdash \bot$. Since each θ_j is one of the θ_is for some $i \in I$, $\bigcap_{i \in I} |\theta_i| \subseteq |\bigwedge_{j=1}^n \theta_j|$. So by condition b in the definition of $\mathcal{E}^*$, $E(\bigwedge_{j=1}^n \theta_j) \in x$. Due to the fact that $x \in W^{*,y}$, $\Box(x) = \Box(y)$, thus $\Box(\bigwedge_{i=1}^k \varphi_i) \in x$. Then, by MergeE, $E(\bigwedge_{i=1}^k \varphi_i \wedge \bigwedge_{j=1}^n \theta_j)$. But then, by E$\Box$, $E\bot \in x$ which is impossible since $\neg E\bot \in x$ and x is consistent.

Now we can show that the truth lemma for $\mathcal{M}^{*,y}$ is true for any $y \in W^*$.

Lemma 5 (Truth Lemma). *For all φ, and $x \in W^{*,y}$, $\mathcal{M}^{*,y}, x \vDash \varphi$ iff $\varphi \in x$, i.e., $[\![\varphi]\!]_{\mathcal{M}^{*,y}} = |\varphi| \cap W^{*,y}$.*

Proof. Let $x \in W^{*,y}$. The proof is by induction on the complexity of φ. The atomic case follows by the definition of $V^{*,y}$. The induction hypothesis (IH) is that for all δ of less complexity than φ, $[\![\delta]\!]_{\mathcal{M}^{*,y}} = |\delta| \cap W^{*,y}$. We will omit the subscript '$\mathcal{M}^{*,y}$' and the $\cap W^{*,y}$ from here on, unless it is important. The Boolean cases are standard and the case for $\Box$ follows since all members of $W^{*,y}$ agree on $\Box$ed formulas. The F and E cases are also fairly straightforward, so we will just do the U case.

Suppose $\varphi = U(\varphi_1, \dots, \varphi_n; \delta)$. Assume that $\mathcal{M}^{*,y}, x \vDash U(\varphi_1, \dots, \varphi_n; \delta)$. By the truth condition for U, $\forall X \in E^{*,y}(x),\ X \subseteq [\![\delta]\!]$ only if $[\![\varphi_j]\!] \subseteq X$ for some $j \leq n$. Then, by the IH, $\forall X \in E^{*,y}(x),\ X \subseteq |\delta| \cap W^{*,y}$ only if $|\varphi_j| \cap W^{*,y} \subseteq X$ for some $j \leq n$.

Now suppose for reductio that $U(\varphi_1, \dots, \varphi_n; \delta) \notin x$, by x's maximality, $\neg U(\varphi_1, \dots, \varphi_n; \delta) \in x$. Since x is U-saturated there is θ such that $\varPhi(U(\overline{\varphi}; \delta), \theta) \subseteq x$. From observation 4 above, then, $|\theta| \cap W^{*,y} \in \mathcal{E}^{*,y}(x)$ since $E\theta \in x$. It also follows that, since $\Box(\theta \to \delta) \in x$, $|\theta| \cap W^{*,y} \subseteq |\delta| \cap W^{*,y}$ because $\Box(\theta \to \delta) \in z$ for all $z \in W^{*,y}$. By IH, $|\delta| \cap W^{*,y} = [\![\delta]\!]$, so there is $X \in \mathcal{E}^{*,y}(x)$ such that $X \subseteq [\![\delta]\!]$. But we also have that $\bigwedge_{j=1}^n \Diamond(\varphi_j \wedge \neg\theta) \in x$, thus for each $j \leq n$, $\Diamond(\varphi_j \wedge \neg\theta) \in x$ which implies $|\varphi_j| \cap W^{*,y} \not\subseteq |\theta| \cap W^{*,y}$. Since $|\theta| \cap W^{*,y} = X \in \mathcal{E}^{*,y}(x)$, that should be impossible according to our first assumption. Thus, $U(\varphi_1, \dots, \varphi_n; \delta) \in x$.

Conversely, suppose $U(\varphi_1, \dots, \varphi_n; \delta) \in x$. Further, suppose that $X \in \mathcal{E}^{*,y}(x)$ and that $X \subseteq [\![\delta]\!]$; if not the conclusion follows vacuously. We need to show that

$[\![\varphi_j]\!] \subseteq X$ for some $j \leq n$. By the IH, we get that $X \subseteq |\delta| \cap W^{*,y}$ and by the definition of $\mathcal{E}^{*,y}$ we get $X = \bigcap_{i \in I} |\theta_i| \cap W^{*,y}$ for some $\bigcap_{i \in I} |\theta_i| \in \mathcal{E}^*(x)$.

Suppose for reductio that $[\![\varphi_j]\!] \not\subseteq \bigcap_{i \in I} |\theta_i| \cap W^{*,y}$ for all $j \leq n$. By IH $|\varphi_j| \cap W^{*,y} \not\subseteq \bigcap_{i \in I} |\theta_i| \cap W^{*,y}$ for all $j \leq n$. For each j there is at least one $x_j \in |\varphi_j| \cap W^{*,y}$ and $x_j \notin \bigcap_{i \in I} |\theta_i| \cap W^{*,y}$. Thus, there is θ_{i_j} such that $x_j \notin |\theta_{i_j}| \cap W^{*,y}$ which means that for each j, $|\varphi_j| \cap W^{*,y} \not\subseteq \bigcap_{k=1}^{n} |\theta_{i_k}| \cap W^{*,y}$. Hence, for each j, $|\varphi_j| \cap |\neg(\bigwedge_{k=1}^{n} \theta_{i_k})| \cap W^{*,y} \neq \varnothing$, which implies that $\varphi_j \wedge \neg(\bigwedge_{k=1}^{n} \theta_{i_k}) \in z$ for some $z \in W^{*,y}$ for each $j \leq n$. But that means $\Diamond(\varphi_j \wedge \neg(\bigwedge_{k=1}^{n} \theta_{i_k})) \in x$ for each j and thus, due to x's maximal consistency, $\bigwedge_{j=1}^{n} \Diamond(\varphi_j \wedge \neg(\bigwedge_{k=1}^{n} \theta_{i_k})) \in x$.

Since $\bigcap_{i \in I} |\theta_i| \cap W^{*,y} \subseteq |\delta| \cap W^{*,y} \subseteq |\delta|$, $\bigcap_{i \in I} |\theta_i| \cap \bigcap \{\, |\psi| : \Box\psi \in y \,\} \subseteq |\delta|$. By standard facts about proof sets, then, $\{\, \theta_i : i \in I \,\} \cup \{\, \psi : \Box\psi \in y \,\} \vdash \delta$. So there are finite sets $\Theta \subseteq \{\, \theta_i : i \in I \,\}$ and $\Psi \subseteq \{\, \psi : \Box\psi \in y \,\}$ such that $\Theta \cup \Psi \vdash \delta$. Hence, by monotonicity and classical logic $\bigwedge \Theta \wedge \bigwedge \Psi \wedge \bigwedge_{k=1}^{n} \theta_{i_k} \vdash \delta$. Thus,

$$\Box[(\bigwedge \Theta \wedge \bigwedge \Psi \wedge \bigwedge_{k=1}^{n} \theta_{i_k}) \to \delta] \in x.$$

Also, $\vdash \varphi_j \wedge \neg(\bigwedge_{k=1}^{n} \theta_{i_k}) \to (\varphi_j \wedge \neg(\bigwedge \Theta \wedge \bigwedge \Psi \wedge \bigwedge_{k=1}^{n} \theta_{i_k}))$, which implies $\vdash \Diamond(\varphi_j \wedge \neg(\bigwedge_{k=1}^{n} \theta_{i_k})) \to \Diamond(\varphi_j \wedge \neg(\bigwedge \Theta \wedge \bigwedge \Psi \wedge \bigwedge_{k=1}^{n} \theta_{i_k}))$ since $\Box$ is normal, so $\Diamond(\varphi_j \wedge \neg(\bigwedge \Theta \wedge \bigwedge \Psi \wedge \bigwedge_{k=1}^{n} \theta_{i_k})) \in x$ for each $j \leq n$ and thus,

$$\bigwedge_{j=1}^{n} \Diamond(\varphi_j \wedge \neg(\bigwedge \Theta \wedge \bigwedge \Psi \wedge \bigwedge_{k=1}^{n} \theta_{i_k})) \in x.$$

By the contrapositive of UV, then,

$$\vdash [\bigwedge_{j=1}^{n} \Diamond(\varphi_j \wedge \neg\, (\bigwedge \Theta \wedge \bigwedge \Psi \wedge \bigwedge_{k=1}^{n} \theta_{i_k}))] \to [E(\bigwedge \Theta \wedge \bigwedge \Psi \wedge \bigwedge_{k=1}^{n} \theta_{i_k}) \\ \to \neg U(\varphi_1, \ldots, \varphi_n; (\bigwedge \Theta \wedge \bigwedge \Psi \wedge \bigwedge_{k=1}^{n} \theta_{i_k}))]$$

is a theorem of U and so in x. But that means, since x is closed under modus ponens, that

$$E(\bigwedge \Theta \wedge \bigwedge \Psi \wedge \bigwedge_{k=1}^{n} \theta_{i_k}) \to \neg U(\varphi_1, \ldots, \varphi_n; (\bigwedge \Theta \wedge \bigwedge \Psi \wedge \bigwedge_{k=1}^{n} \theta_{i_k})) \in x.$$

Notice, however, that since $\bigcap_{i \in I} |\theta_i| \subseteq \bigcap \{\, |\theta| : \theta \in \Theta \,\} \cap \bigcap_{k=1}^{n} |\theta_{i_k}|$, by standard facts about proof sets we have $\bigcap \{\, |\theta| : \theta \in \Theta \,\} \cap \bigcap_{k=1}^{n} |\theta_{i_k}| = |\bigwedge \Theta \wedge \bigwedge_{k=1}^{n} \theta_{i_k}|$, so by condition (b) on $\mathcal{E}^*(x)$, $E(\bigwedge \Theta \wedge \bigwedge_{k=1}^{n} \theta_{i_k}) \in x$. But since $\Psi \subseteq \Box(x) = \Box(y)$, and $\Box$ is a normal operator, $\Box(\bigwedge \Psi) \in x$. But then, by MergeE, $E(\bigwedge \Theta \wedge \bigwedge \Psi \wedge \bigwedge_{k=1}^{n} \theta_{i_k}) \in x$. Thus,

$$\neg U(\varphi_1, \ldots, \varphi_n; (\bigwedge \Theta \wedge \bigwedge \Psi \wedge \bigwedge_{k=1}^{n} \theta_{i_k})) \in x.$$

Since we have already established that $\Box[(\bigwedge \Theta \wedge \bigwedge \Psi \wedge \bigwedge_{k=1}^{n} \theta_{i_k}) \to \delta] \in x$, it follows from U$\Box$R, that $U(\varphi_1, \ldots, \varphi_n; (\bigwedge \Theta \wedge \bigwedge \Psi \wedge \bigwedge_{k=1}^{n} \theta_{i_k})) \in x$. That means x is inconsistent. But x is maximally consistent, so $[\![\varphi_j]\!] \subseteq X$ for some $j \leq n$. Therefore, $\mathcal{M}^{*,y}, x \vDash U(\varphi_1, \ldots, \varphi_n; \delta)$.

If we then notice that assuming $E\top \in x$ for each $x \in W^{*,y}$, we get that $[\![\top]\!] \in \mathcal{E}^{*,y}(x)$ by the truth lemma and the definition of $\mathcal{E}^{*,y}$, thus, $\mathcal{E}^{*,y}(x) \neq \varnothing$. Also, if we were to assume $\varnothing \in \mathcal{E}^{*,y}(x)$, then $[\![\bot]\!] \in \mathcal{E}^{*,y}(x)$. However, given the definition of $\mathcal{E}^{*,y}(x)$ and the truth lemma, we would have $E\bot \in x$ which is impossible since x must be consistent. Therefore $\mathcal{M}^{*,y}$ is a hypergraph model as given in definition 4. The standard argument shows that the logic **U** is complete with respect to the arguments validated in all hypergraph models. In fact, it shows something stronger.

The proof of completeness from the previous section shows that the logic **U** is actually complete with respect to the class of all core-complete models. Thus, we may assume that all frames considered are core-complete: i.e., $cor(\mathcal{E}(x)) \neq \varnothing$ and for all $X \in \mathcal{E}(x)$ there is $X' \in cor(\mathcal{E}(x))$ such that $X' \subseteq X$. This observation was made in [6]. Their construction of a canonical model resulted in a core-reduced model rather than just one that is core-complete because they only kept the cores of each $\mathcal{E}^*(x)$ they defined. The core of $\mathcal{E}^*(x)$ consists of all sets $\bigcap \{ |\theta_i| : i \in I \}$ that are maximal subsets of $\{ |\theta| : E\theta \in x \}$ which also satisfy the second condition in the definition of $\mathcal{E}^*(x)$. Since $\mathcal{E}^{*,y}$ is defined from $\mathcal{E}^*(x)$, $\mathcal{E}^{*,y}(x)$ is also core-complete.

Thus, we have given a modal evidence logic for general Schotch-Jennings forcing, not simply the fixed-level versions. Now that we have this logic, in future work we can extend this semantics to a doxastic logic in the style of the evidence logics of van Benthem et al. and Baltag et al. We will also explore generalizations of the U operator which we have called ‘pointed operators’.

References

1. Apostoli, P., Brown, B.: A solution to the completeness problem for weakly aggregative modal logic. J. Symbolic Logic **60**(3), 832–842 (1995). https://doi.org/10.2307/2275759
2. Baltag, A., Bezhanishvili, N., Özgün, A., Smets, S.: Justified belief and the topology of evidence. In: Väänänen, J., Hirvonen, Å., de Queiroz, R. (eds.) WoLLIC 2016. Lecture Notes in Computer Science, vol. 9803, pp. 83–103. Springer, Berlin (2016). https://doi.org/10.1007/978-3-662-52921-8_6
3. van Benthem, J., Bezhanishvili, N., Enqvist, S., Yu, J.: Instantial neighbourhood logic. Rev. Symbolic Logic **10**(1), 116–144 (2017). https://doi.org/10.1017/s1755020316000447
4. van Benthem, J., Pacuit, E., Fernández-Duque, D.: Evidence and plausibility in neighborhood structures. Ann. Pure Appl. Logic **165**, 106–133 (2014)
5. Chellas, B.F.: Modal Logic: An Introduction. Cambridge University Press, Cambridge (1980)
6. Ding, Y., Liu, J., Wang, Y.: Someone knows that local reasoning on hypergraphs is a weakly aggregative modal logic. Synthese **201**(2), 1–27 (2023). https://doi.org/10.1007/s11229-022-04032-y

7. Jennings, R.E., Schotch, P.K.: The preservation of coherence. Stud. Logica. **43**, 89–106 (1984). https://doi.org/10.1007/bf00935743
8. Jennings, R.E., Brown, B., Schotch, P.: On Preserving: Essays on Preservationism and Paraconsistency. Toronto Studies in Philosophy, University of Toronto Press, Toronto (2009)
9. Nicholson, D.: A dualization of neighbourhood structures. In: Jennings, R., Brown, B., Schotch, P. (eds.) On Preserving: Essays on Preservationism and Paraconsistent Logic, pp. 49–60. University of Toronto Press (2009)
10. Pacuit, E.: Neighborhood Semantics for Modal Logic. Springer, Cham, Switzerland (2017). https://doi.org/10.1007/978-3-319-67149-9
11. Payette, G.: Getting the most out of inconsistency. J. Philos. Logic **44**(5), 573–592 (2015). https://doi.org/10.1007/s10992-014-9343-5
12. Rescher, N., Manor, R.: On inferences from inconsistent premises. Theor. Decis. **1**(2), 179–217 (1970). https://doi.org/10.1007/bf00154005
13. Schotch, P.K., Jennings, R.E.: Inference and necessity. J. Philos. Logic **9**(3), 327–340 (1980). https://doi.org/10.1007/bf00248398
14. Schotch, P.K., Jennings, R.E.: On detonating. In: Priest, G., Routley, R., Norman, J. (eds.) Paraconsistent Logic: Essays On The Inconsistent, pp. 306–327 (1989)

A Separation Logic with Histories of Epistemic Actions as Resources

Hans van Ditmarsch[1(✉)], Didier Galmiche[2], and Marta Gawek[2]

[1] Université de Toulouse, CNRS, IRIT, Toulouse, France
hans.van-ditmarsch@irit.fr
[2] Université de Lorraine, CNRS, LORIA, Nancy, France

Abstract. We propose a separation logic where resources are histories (sequences) of epistemic actions so that resource update means concatenation of histories and resource decomposition means splitting of histories. This separation logic, called AMHSL, allows us to reason about the past: does what is true now depend on what was true in the past, before certain actions were executed? We show that the multiplicative connectives can be eliminated from a logical language with also epistemic and action model modalities, if the horizon of epistemic actions is bounded.

1 Introduction

In an action that is an informative update, what the agents know about facts and about each other may change (I learn that it rains in Spain), and these facts themselves may also change (it stopped raining). We present a logic wherein the amount of change, as measured by sequences of actions that are informative updates, is considered as a resource. In an epistemic context such updates often depend on each other (after it stopped raining, I cannot learn that it rains in Spain), so it is relevant when, as resources, they can be separated and combined with the multiplicative connectives of the Bunched Implications logic (BI) [15]. Let us survey the relevant areas dynamic epistemic logic and bunched separation logic, and describe prior proposals to combine both.

Knowledge and change of knowledge, and in particular for multiple agents, are the abode of epistemic logic [19], a modal logic interpreted on relational models consisting of possible worlds. The analysis of multiple agents publicly informing each other of their ignorance and knowledge culminated in Public Announcement Logic [14], and a further generalization non-public information change such as private or secret announcements resulted in Action Model Logic [3], further extended with factual change in [17]. Another source of our ideas is the logic of Bunched Implications (BI) and its variants, like Boolean BI (BBI) [15], that mainly focus on resource sharing and separation. These logics combine additive ($\wedge$, $\rightarrow$, $\vee$) and multiplicative ($*$, $-\!\!*$) connectives. The multiplicative conjunction $*$ expresses separation of resources and the multiplicative implication $-\!\!*$ expresses resource update [15]. Here the term "separation logics" denotes the

H. H. Hansen et al. (Eds.): WoLLIC 2023, LNCS 13923, pp. 161–177, 2023.
https://doi.org/10.1007/978-3-031-39784-4_10

class of logics based on BI or BBI and their modal extensions, even if so-called Separation Logic is such a logic with resources being memory areas [10].

How can we combine knowledge and resources? It is a two-way traffic. One can go in the direction of modelling uncertainty about resources [7,8]. But one can also go in the direction of modelling information as a resource. We very clearly go in that, novel, direction. We notice that this is a dangerous road: incoming information is highly dependent on context and may have side effects, so it is difficult to separate/decompose, which goes against the grain of separation logics. But it is therefore a challenge we propose to meet. Both directions, inasfar as discussed here, have in common that we add modalities to separation logics (either epistemic or dynamic) [8]. Epistemic extensions of separation logic include Public Announcement Separation Logic [7], and the further generalization called Action Model Separation Logic [18]. In these logics the states or worlds of an epistemic model represent resources, resource decomposition and update relate different states in the domain of the model, and the members of the domain of a Kripke model should therefore represent a resource monoid. In [18] the valuation of a state is a resource, instead of the state, so that different states with the same valuation can represent the same resource.

In this work we consider *histories of epistemic actions* as resources. It is both according to the philosophy of separation, as in many epistemic contexts one can run of out resources, such as exceeding the permitted number of calls in a gossip protocol or the number of manipulations in epistemic planning [5]; but also somewhat against the philosophy of separation, as the knowledge consequences of epistemic actions highly depend on their order and may also lack certain monotonicity of knowledge consequences. However, in the special case where factual change is absent, ignorance can only be lost, whereas positive knowledge (the universal fragment) continues to grow.

We propose a new separation logic with sequences of actions (informative updates) as resources, called Action Model History Separation Logic (AMHSL). Instead of states we consider sequences of actions (histories) to be resources, and consequently we define resource composition as the concatenation of histories. This requires another interpretation of the multiplicative connectives. As the order of actions is non-trivial, the multiplicative conjunction interpreting resource composition is non-commutative, and there are two ways of resource update: appending a history to the end of a given history, or before its beginning. We therefore need two multiplicative implications in the logical language. After defining this logical semantics of separation and composition of actions histories we illustrate the interest of AMHSL with an example about gossip protocols. Finally we show that, given a maximum length of action histories, any AMHSL formula with multiplicative connectives is equivalent to a formula without them: a so-called reduction. As the latter is a formula in action model logic, we have thus also axiomatized AMSHL.

2 Semantics with Informative Actions as Resources

We first present the syntax of the logical language and the semantical structures.

Let a finite set of *agents* A and a (disjoint) countable set of *atoms* (or *propositional variables*) P be given.

Definition 1 (Language). *The logical language* $\mathcal{L}^{K*\otimes}(A,P)$ *is defined by a BNF, where* $p \in P$*,* $a \in A$*, and* $\mathcal{E}_e$ *is a pointed action model, defined below.*

$$\psi ::= \quad p \mid I \mid \bot \mid \neg\psi \mid (\psi \wedge \psi) \mid (\psi * \psi) \mid (\psi \mathrel{-\!\!*} \psi) \mid (\psi \mathrel{*\!\!-} \psi) \mid K_a\psi \mid [\mathcal{E}_e]\psi$$

The parameters A and P are often omitted from $\mathcal{L}^{K*\otimes}(A,P)$. We also consider the sublanguage $\mathcal{L}^{K\otimes}$ without the constructs containing $*$, $\mathrel{-\!\!*}$ and $\mathrel{*\!\!-}$ and without the constant I (the language of action model logic), the sublanguage $\mathcal{L}^{K*}$ without the construct $[\mathcal{E}_e]\psi$ (the language of epistemic separation logic), and the sublanguage $\mathcal{L}^{K}$ without either (the language of epistemic logic). It is implicit in the definition that the pre- and postcondition formulas of $\mathcal{E}$ are in $\mathcal{L}^{K\otimes}$ (Definition 4). For the sublanguage of $\mathcal{L}^{K*\otimes}$ only allowing the unique action model $\mathcal{E}$ we write $\mathcal{L}^{K*\mathcal{E}}$ and similarly for other fragments. Other propositional connectives are defined by notational abbreviation and also the dual modality $\langle\mathcal{E}_c\rangle\varphi := \neg[\mathcal{E}_e]\neg\varphi$.

Definition 2 (Resource monoid). *A* partial resource monoid *(or* resource monoid*) is a structure* $\mathcal{R} = (R, \circ, n)$ *where* R *is a set of* resources *(denoted* $r, r', r_1, r_2, \ldots$*) containing a* neutral element n*, and where* $\circ : R \times R \to R$ *is a* resource composition operator *that is associative, that may be partial and such for all* $r \in R$*,* $r \circ n = n \circ r = r$*. If* $r \circ r'$ *is defined we write* $r \circ r'{\downarrow}$ *and if* $r \circ r'$ *is undefined we write* $r \circ r'{\uparrow}$*. When writing* $r \circ r' = r''$ *we assume that* $r \circ r'{\downarrow}$*.*

Definition 3 (Epistemic model). *An* epistemic model *is a structure* $\mathcal{M} = (S, \sim, V)$ *such that* S *is a non-empty* domain *of* states *(or* worlds*),* $\sim : A \to \mathcal{P}(S \times S)$ *is a function that maps each agent* a *to an equivalence relation* $\sim_a$*, and* $V : P \to \mathcal{P}(S)$ *is a* valuation function*, where* $V(p)$ *denotes where variable* p *is true. Given* $s \in S$*, the pair* $(\mathcal{M}, s)$ *is a* pointed epistemic model*, denoted* $\mathcal{M}_s$*.*

Definition 4 (Action model). *An* action model *is a structure* $\mathcal{E} = (E, \approx, pre, post)$*, where* E *is a non-empty finite domain of* actions *(denoted* $e, f, g, \ldots$*),* $\approx_a$ *an equivalence relation on* E *for all* $a \in A$*,* $pre : E \to \mathcal{L}^{K\otimes}$ *is a* precondition *function, and* $post : E \to P \to \mathcal{L}^{K\otimes}$ *is a* postcondition *function such that every* $post(e)$ *is only finitely different from the identity: we can see its domain as a finite set of variables* $Q \subseteq P$*. Given* $e \in E$*, a* pointed action model *(or* epistemic action*) is a pair* $(\mathcal{E}, e)$*, denoted* $\mathcal{E}_e$*.*

2.1 Knowledge and Informative Actions

We distinguish the semantics of knowledge and action model execution on epistemic models, from the more involved semantics of the full language on epistemic history models. The distinction is made to keep the exposition transparant, because we wish to focus on information change as separation and composition, and because it allows us to use a simpler, abbreviated, notation for the latter.

For the satisfaction relation of the former we write $\models_0$ and for that of the latter we write $\models$. The $\models_0$ update semantics is standard fare (although less so with the variation involving factual change) and can be found in, for example [13,17]. Definitions 5 and 6 are assumed to be given by simultaneous recursion.

Definition 5 (Satisfaction relation for the restricted language). *The* satisfaction relation $\models_0$ *between pointed epistemic models $\mathcal{M}_s$ and formulas in $\mathcal{L}^{K\otimes}(A,P)$, where $\mathcal{M} = (S, \sim, V)$ and $s \in S$, is defined by induction on formula structure.*

$$\begin{array}{lll}
\mathcal{M}_s \models_0 p & \textit{iff} & s \in V(p) \\
\mathcal{M}_s \models_0 \bot & \textit{iff} & \textit{false} \\
\mathcal{M}_s \models_0 \neg\varphi & \textit{iff} & \mathcal{M}_s \not\models_0 \varphi \\
\mathcal{M}_s \models_0 \varphi \wedge \psi & \textit{iff} & \mathcal{M}_s \models_0 \varphi \textit{ and } \mathcal{M}_s \models_0 \psi \\
\mathcal{M}_s \models_0 K_a\varphi & \textit{iff} & \mathcal{M}_{s'} \models_0 \varphi \textit{ for all } s' \in S \textit{ such that } s \sim_a s' \\
\mathcal{M}_s \models_0 [\mathcal{E}_e]\varphi & \textit{iff} & \mathcal{M}_s \models_0 \textit{pre}(e) \textit{ implies } (\mathcal{M} \otimes \mathcal{E})_{(s,e)} \models_0 \varphi
\end{array}$$

Definition 6 (Action model execution). *Given are epistemic model $\mathcal{M} = (S, \sim, V)$ and action model $\mathcal{E} = (E, \approx, \textit{pre}, \textit{post})$. The updated epistemic model $\mathcal{M} \otimes \mathcal{E} = (S', \sim', V')$ is such that—where $s, t \in S$, $a \in A$, $e, f \in E$, $p \in P$:*

$$\begin{array}{lll}
S' & = & \{(s,e) \mid \mathcal{M}_s \models_0 \textit{pre}(e)\} \\
(s,e) \sim_a (t,f) & \textit{iff} & s \sim_a t \textit{ and } e \approx_a f \\
(s,e) \in V'(p) & \textit{iff} & \mathcal{M}_s \models_0 \textit{post}(e)(p)
\end{array}$$

2.2 Semantics for Separation and Composition of Action Histories

We now present the $\models$ semantics, that is defined on the full language. The semantics interprets formulas with respect to states in an initial model and sequences of informative actions (or events). This is known as a history-based semantics, where the sequence of actions is the history of past actions [16]. The corresponding semantic objects are often known as 'history-based models' and called here *history models*. Updates of models with action models construct such history models. However, as constructing history models requires evaluating formulas and as formulas are interpreted in history models, the semantics are given by simultaneous induction involving both.

Definition 7 (Epistemic history model). *Given are epistemic model $\mathcal{M} = (S, \sim, V)$ and action model $\mathcal{E} = (E, \approx, \textit{pre}, \textit{post})$. First, we define $\mathcal{M} \otimes \mathcal{E}^n$ by induction on $n \in \mathbb{N}$ as: $\mathcal{M} \otimes \mathcal{E}^0 := \mathcal{M}$, and $\mathcal{M} \otimes \mathcal{E}^{n+1} := (\mathcal{M} \otimes \mathcal{E}^n) \otimes \mathcal{E}$. The* epistemic history model *$\mathcal{ME}^\omega$ is now defined as $\oplus_{n \in \mathbb{N}}(\mathcal{M} \otimes \mathcal{E}^n)$, where $\oplus$ is the direct sum. We also distinguish the* bounded *epistemic history model $\mathcal{ME}^{\mathbf{max}}$ defined as $\oplus_{n \leq \mathbf{max}}(\mathcal{M} \otimes \mathcal{E}^n)$, where we assume that* $\mathbf{max} \geq 1$.

Histories of Actions. The elements of the domain of $\mathcal{ME}^\omega$ have the shape $(s, e_1, \ldots, e_n)$ where $e_1, \ldots, e_n \in E$ for $n \in \mathbb{N}$, and where for $n = 0$ the domain element is s. The tuple of actions $(e_1, \ldots, e_n)$ is called a *history*, denoted h, where ϵ is the empty history. Given $(s, e_1, \ldots, e_n)$, we also say that the history $(e_1, \ldots, e_n)$ can be *executed* in the state s. For $(s, e_1, \ldots, e_n)$ we write $se_1 \ldots e_n$ or sh, where $h = e_1 \ldots e_n$. In other words, we consider a history h to be a member of E^*. Given history h, $|h|$ denotes its length, and for concatenation of histories h, h' we write hh'. We let $\sqsubseteq$ be the prefix relation on histories, ($\epsilon \sqsubseteq h$, and if $h \sqsubseteq h'$, then $h \sqsubseteq h'e$), and if $h' \sqsubseteq h$, then $h \backslash h'$ is the 'postfix' following h', that is, $h = h'(h \backslash h')$. Indistinguishability of histories is defined as: $\epsilon \sim_a \epsilon$, and if $h \sim_a h'$ for histories h, h' and also $e \sim_a e'$, then $he \sim_a h'e'$. Finally, given $sh, s'h' \in \mathcal{D}(\mathcal{ME}^\omega)$, $sh \sim_a s'h'$ means that $s \sim_a s'$ and $h \sim_a h'$. Note that indistinguishable histories are of the same length (in this synchronous semantics).

Alternative History Models. Another way to define history-based models seems more common in the literature [16,20]. We then enrich the model $\mathcal{ME}^\omega$ with relations $\rightarrow_e$ for all $e \in E$ defined as: $sh \rightarrow_e she$ for all $sh, she \in \mathcal{D}(\mathcal{ME}^\omega)$. Note that this assumes $\mathcal{ME}^\omega_{sh} \models pre(e)$. In other words, the model transforming updates induced by action models $\mathcal{E}$ are internalized as transitions between the (state,history) pairs of the domain of the epistemic history model. This modelling facilitates the comparison with temporal epistemic logics.

Histories as Resources. Inspired by the action monoids of [6], we now take histories as resources, such that the set of histories of actions is a resource monoid with concatenation of histories as resource composition and the empty history ϵ as neutral element. For $h \circ h'$ we write hh', as above. Evidently this 'resource composition' (concatenation) is associative, and also $\epsilon \circ h = h \circ \epsilon = h$. As histories can always be concatenated, resource composition is always defined. However, for some applications there is a maximum length **max** of histories, such that $hh' \uparrow$ then means that $|hh'| > \mathbf{max}$. Seeing histories as resources, it seems to make sense that you run out of actions if you execute too many. As the order of actions, and histories, matters, the multiplicative conjunction ($*$) is not commutative, and to maintain duality we need two different multiplicative implications: one for what is true after appending an arbitrary history to a given history ($-\!\!*$), and another one for what is true after appending a given history to an arbitrary history ($*\!\!-$).

We now define the semantics. Instead of interpreting a formula in a state of an epistemic model, we interpret it in a (state,history) pair of an epistemic history model.

Below, 'there is sh' means 'there is h such that $sh \in \mathcal{D}(\mathcal{ME}^\omega)$', in other words, there is a history h such that h can be executed in state s; and similarly for 'for all sh'. Note that both imply that $h\downarrow$, that is, $|h| \leq \mathbf{max}$. For example, "for all sh, shh'" in the clause for $-\!\!*$ means "for all $h' \in E^*$ such that $|hh'| \leq \mathbf{max}$ and $shh' \in \mathcal{D}(\mathcal{M})$". We recall that $h = h'h''$ means that $h'h''\downarrow$ and $h = h'h''$.

Definition 8 (Satisfaction relation). *The* satisfaction relation $\models$ *between a pointed epistemic history model* $\mathcal{ME}^\omega_{sh}$ *and formulas in* $\mathcal{L}^{K*\mathcal{E}}(A, P)$*, where* $\mathcal{M} = (S, \sim, V)$*,* $\mathcal{E} = (E, \approx, pre, post)$*,* $s \in S$*, and* $h \in E^*$*, is defined by induction on formula structure. Model* $\mathcal{ME}^\omega$ *is left implicit in the notation, and* $\mathcal{E}$ *is left* *implicit in* $[\mathcal{E}_e]\varphi$*.*

$$\begin{array}{lll} sh \models p & \textit{iff} & s \models post(h)(p) \\ sh \models I & \textit{iff} & h = \epsilon \\ sh \models \bot & \textit{iff} & \textit{false} \\ sh \models \neg\varphi & \textit{iff} & sh \not\models \varphi \\ sh \models \varphi \wedge \psi & \textit{iff} & sh \models \varphi \textit{ and } sh \models \psi \\ sh \models \varphi * \psi & \textit{iff} & \textit{there are } sh', sh'' \textit{ with } h = h'h'' \textit{ such that } sh' \models \varphi \textit{ and } sh'' \models \psi \\ sh \models \varphi \mathrel{-\!\!*} \psi & \textit{iff} & \textit{for all } sh', shh' : sh' \models \varphi \textit{ implies } shh' \models \psi \\ sh \models \varphi \mathrel{*\!\!-} \psi & \textit{iff} & \textit{for all } sh', sh'h : sh' \models \varphi \textit{ implies } sh'h \models \psi \\ sh \models K_a\varphi & \textit{iff} & s'h' \models \varphi \textit{ for all } s'h' \textit{ such that } sh \sim_a s'h' \\ sh \models [e]\varphi & \textit{iff} & sh \models pre(e) \textit{ implies } she \models \varphi \end{array}$$

On $\mathcal{ME}^{\mathbf{max}}$ *all clauses are the same except the last one, that then becomes:*

$$sh \models [e]\varphi \textit{ iff } |h| < \mathbf{max} \textit{ and } sh \models pre(e) \textit{ imply } she \models \varphi$$

For $s\epsilon \models \varphi$ *we write* $s \models \varphi$*. The simplified notation is justified because all formulas are interpreted in the one and only model* $\mathcal{ME}^\omega$*, unlike in the* $\models_0$ *semantics. We emphasize that the language of interpretation is* $\mathcal{L}^{K*\mathcal{E}}$ *(with action modalities only for* $\mathcal{E}$*) and not* $\mathcal{L}^{K*\otimes}$ *(for arbitrary action model modalities).*

There are two notions of validity. A formula φ *is* valid*, notation* $\models \varphi$*, iff for all* $\mathcal{M} = (S, \sim, V)$ *and* $s \in S$*,* $s \models \varphi$*. A formula* φ *is* $*$-valid*, or* always-valid*, notation* $\models^* \varphi$*,*[1] *iff for all* $\mathcal{M} = (S, \sim, V)$ *and* $\mathcal{E} = (E, \approx, pre, post)$ *and for all* $sh \in \mathcal{D}(\mathcal{ME}^\omega)$*,* $sh \models \varphi$*. Validity is similarly defined on* $\mathcal{ME}^{\mathbf{max}}$*.*

In fact we defined two semantics, one without a bound on action histories and one with the bound **max**, but we write $\models$ for both satisfaction relations (and $\models^*$). The validities in Sect. 4 are restricted to the semantics with bound **max**.

Lemma 1.

1. *For all* $\varphi \in \mathcal{L}^{K*\mathcal{E}}$*:* $\models \varphi$ *iff* $\models^* I \to \varphi$*.*
2. *For all* $\varphi \in \mathcal{L}^{K*\mathcal{E}}$*:* $\models^* \varphi$ *implies* $\models \varphi$*.*
3. *For all* $\varphi \in \mathcal{L}^{K\mathcal{E}}$*:* $\models_0 \varphi$ *iff* $\models \varphi$*.*

Proof.

1. Observe that I is only true for the empty history.
2. If a formula is true for arbitrary histories, then also for the empty history.
3. Let $\mathcal{M} = (S, \sim, V)$, and $s \in S$ be given. Then $\mathcal{M}_s \models_0 \varphi$, iff $s \models \varphi$, where the latter is in model $\mathcal{ME}^\omega$. The proof by induction on φ is obvious except for the case $[e]\varphi$ that directly follows from the semantics.

[1] The $*$ of multiplicative conjunction $\varphi * \psi$ is as the $*$ in $*$-valid, but the latter is motivated by the Kleene-$*$ of arbitrary iteration.

Histories in the Language. A fair number of properties of our history semantics are more elegantly presented if we allow histories in the language. For example it is convenient to think of the precondition or the postcondition of a history, not only of an action. We recursively define by notational abbreviation: (i) $[\epsilon]\varphi := \varphi$ and $[he]\varphi := [h][e]\varphi$; (ii) $pre(\epsilon) := \top$ and $pre(he) := \langle h\rangle pre(e)$; (iii) $post(\epsilon)(p) := p$ and $post(he)(p) := \langle h\rangle post(e)(p)$.

Given modalities for histories, the usual reduction axioms for action model logic can generalized in an obvious way. That is, all except the reduction axiom $[e][f]\varphi \leftrightarrow [e \circ f]\varphi$, where $\circ$ is action model composition, as $\mathcal{E} \circ \mathcal{E}$ is typically another action model than $\mathcal{E}$, that is not in the language $\mathcal{L}^{K\mathcal{E}}$ for the unique action model $\mathcal{E}$. As we reduce history modalities instead of action modalities we do not need that axiom.

Proposition 1. *All valid in the $\models_0$ semantics are*

$[e]p \leftrightarrow pre(e) \rightarrow post(e)(p)$	$[h]p \leftrightarrow pre(h) \rightarrow post(h)(p)$
$[e]\neg\varphi \leftrightarrow pre(e) \rightarrow \neg[e]\varphi$	$[h]\neg\varphi \leftrightarrow pre(h) \rightarrow \neg[h]\varphi$
$[e](\varphi \wedge \psi) \leftrightarrow [e]\varphi \wedge [e]\psi$	$[h](\varphi \wedge \psi) \leftrightarrow [h]\varphi \wedge [h]\psi$
$[e]K_a\varphi \leftrightarrow pre(e) \rightarrow \bigwedge_{e\sim_a f} K_a[f]\varphi$	$[h]K_a\varphi \leftrightarrow pre(h) \rightarrow \bigwedge_{h\sim_a h'} K_a[h']\varphi$

Proof. All the left are standard [21]. All the right follow from the left. The proof is by induction on the length of history h. The inductive clauses are all elementary (omitted, however for inductive case $[he]K_a\varphi$ observe that $he \sim_a h'e'$ if $h \sim_a h'$ and $e \sim_a e'$), and only the basic clause $h = \epsilon$ may need some attention.

- $[\epsilon]p = p$ which is equivalent to $pre(\epsilon) \rightarrow post(\epsilon)(p) = \top \rightarrow p$.
- $[\epsilon]\neg\varphi = \neg\varphi$, which is equivalent to $pre(\epsilon) \rightarrow \neg[\epsilon]\varphi = \top \rightarrow \neg\varphi$.
- $[\epsilon](\varphi \wedge \psi) = \varphi \wedge \psi$, which is equivalent to $[\epsilon]\varphi \wedge [\epsilon]\psi = \varphi \wedge \psi$.
- $[\epsilon]K_a\varphi = K_a\varphi$, which is equivalent to $pre(\epsilon) \rightarrow \bigwedge_{\epsilon\sim_a h'} K_a[h']\varphi = \top \rightarrow K_a[\epsilon]\varphi = \top \rightarrow K_a\varphi$, which is equivalent to $K_a\varphi$.

A corollary of Lemma 1 and Proposition 1 is that these history reduction axioms are also valid for the $\models$ semantics, where the formulas φ, ψ ocurring in them are from $\mathcal{L}^{K\mathcal{E}}$, and it is also straightforward to observe that they remain $\models$ valid if $\varphi, \psi \in \mathcal{L}^{K*\mathcal{E}}$. This is what we need in Sect. 4.[2]

3 Gossip Protocols with AMHSL

In gossip protocols we investigate dissemination of information through a network by way of peer-to-peer calls. Each agent holds a 'secret', that is, some piece of information private to that agent only. The goal of the information exchanges is that all agents know all secrets. In a call the callers exchange all the secrets they know. In an epistemic gossip protocol [22] only calls are permitted that

[2] They are all even $*$-valid in the $\models$ semantics, on models $\mathcal{ME}^{\omega}$, but not on models $\mathcal{ME}^{\mathbf{max}}$ as that would need relativization of each axiom to $\neg[h]\bot \rightarrow$. However we will not use (nor claim) that.

satisfy a certain logical condition. In the protocol LNS [1] you may only call another agent if you do not know that agent's secret. In the protocol CMO [22] you may only call another agent if you have not been involved in a call with that agent. Note that a LNS-permitted call is also CMO-permitted.

In our setting, a permitted call sequence is a resource, and a call is represented as an action model [1]. We provide (novel) action models for synchronous CMO- and LNS-calls.

Given a set A of n agents, and $a, b \in A$, propositional variables a_b represent that the secret of agent a is known by agent b, a *call* is a pair (a, b) denoted ab, a *call sequence* σ is a finite sequence $ab.cd.\ldots$ of calls, and variables ab^+ represent that call ab took place. A *secret distribution* is an n-tuple of subsets of A. We execute gossip protocols in the model $\mathcal{I}$ with the *initial secret distribution* wherein all agents only know their own secret (a_b is only true when $a = b$, and all ab^+ are false). An agent who knows all secrets is an expert. We let *Exp* represent that all agents know all secrets, that is, $\bigwedge_{a,b\in A} a_b$. In protocol LNS the condition for making a call ab is $\neg b_a$ and in CMO the condition is $\neg ab^+ \wedge \neg ba^+$.

The action model representing a synchronous call in CMO is defined as $\mathcal{G} = (E, \approx, \mathit{pre}, \mathit{post})$ where $E = \{ab \mid a, b \in A, a \neq b\}$, $ab \approx_c de$ iff ($c \neq a, b, c, d$, or $c = a = d$ and $b = e$, or $c = b = e$ and $a = d$), $\mathit{pre}(ab) = \neg ab^+ \wedge \neg ba^+$, and $\mathit{post}(ab)(c_a) = \mathit{post}(ab)(c_b) = c_a \vee c_b$ (a secret c is known by a after the call ab if before the call it was known by a or by b, and similarly for b), $\mathit{post}(ab)(ab^+) = \top$, and otherwise facts do not change value (i.e., $\mathit{post}(ab)(p) = p$). The action model for a synchronous LNS call is the same except that $\mathit{pre}(ab) = \neg b_a$.

Given n agents, we now investigate $\mathcal{I}\mathcal{G}^{\mathbf{max}}$ for synchronous CMO so that $\mathbf{max} = \binom{n}{2}$. Given three agents, a call sequence after which all agents are experts is $ab.ac.bc$. We now represent some scenarios involving K, $*$, $-\!\!*$, and $*\!\!-$.

– $ab.ac \models c_b -\!\!* \mathit{Exp}$:
Given three agents a, b, c and call sequence $ab.ac$, after which a and c but not b are experts (in the second call ac, a informs c of a, b and c informs a of c, so that both are now experts), any subsequent call resulting in b knowing the secret of c makes all agents experts. For example, $bc \models c_b$ and indeed $ab.ac.\boldsymbol{bc} \models \mathit{Exp}$. But also $ac.ac.ab \models c_b$ and $ab.ac.\boldsymbol{ac.ac.ab} \models \mathit{Exp}$.

– $\models \varphi_{ab} *\!\!- K_a K_b (b_c \to a_c)$: (where $\varphi_{ab} := a_b \wedge b_a \wedge \bigwedge_{c\neq a} \neg c_b \wedge \bigwedge_{c\neq b} \neg c_a$)
Formula φ_{ab} holds after any call sequence σ wherein the only call(s) involving a and b was (were) to each other. Any extension $\sigma\tau$ of a σ satisfying φ_{ab} will pass along the secrets of a and b jointly. Therefore, $\models ab^+ *\!\!- (b_c \to a_c)$ and also $\models \varphi_{ab} *\!\!- K_a K_b (b_c \to a_c)$. On the other hand, $\not\models \varphi_{ab} -\!\!* K_a K_b (b_c \to a_c)$: when appending σ to a τ containing a call between b and another agent c, $b_c \to a_c$ is false, and a subsequent call ab also fails to guarantee that it holds. For example, $cd.ab \models \varphi_{ab}$, and therefore $\boldsymbol{cd.ab}.bc \models b_c \to a_c$, whereas $bc.\boldsymbol{cd.ab} \not\models b_c \to a_c$. So this example showed that there are φ and ψ for which $\models \varphi *\!\!- \psi$ but $\not\models \varphi -\!\!* \psi$.

– $\not\models a_c \wedge b_c \to a_c * b_c$:
Agent c may know the secrets of a and b now but not necessarily after fewer calls, although agent c may still know the secret of a or the secret of b. For example, $ab.ac \models a_c \wedge b_c$ but $ab.ac \not\models a_c * b_c$.

4 Reduction from $\mathcal{L}^{K*\mathcal{E}}$ to $\mathcal{L}^{K\mathcal{E}}$ Given a Bound max

In this section we show that every formula in $\mathcal{L}^{K*\mathcal{E}}$ (we recall that $\mathcal{L}^{K*\mathcal{E}}$ is the language $\mathcal{L}^{K*\otimes}$ where only action model $\mathcal{E}$ is allowed) is equivalent to a formula in $\mathcal{L}^{K\mathcal{E}}$, without $*$, $-\!\!*$, and $-\!\!\circledast$ modalities, and without I. We show this by the time-honoured technique of a reduction system: a number of validities that are equivalences [11]. As every formula in $\mathcal{L}^{K\mathcal{E}}$ is equivalent to a formula in $\mathcal{L}^{K}$ [3,17], we then have shown that AMHSL is as expressive as the base multi-agent epistemic logic S5.

Our result is restricted in two ways. First, it is with respect to truth in the empty history models. Without that restriction already the language $\mathcal{L}^{K*}$ is more expressive than the language $\mathcal{L}^{K}$, as it is easy to see: a model wherein a knows that p and p is announced, is different from a model wherein a is uncertain about p and p is announced. However, after the announcement they satisfy the same epistemic formulas. However, to restrict validities to those for models with empty histories is usual in history-based semantics. The first restriction therefore keeps our result still relevant. Second, we can only show this if there is a bound **max** $\in \mathbb{N}$ on the number of actions that can be executed. Without that we do not have a reduction, and we conjecture that one may not exist, given the well-known theoretical issues with arbitrary iteration of updates (undecidable logics, etc.) [12], and given that the semantics of $-\!\!\circledast$ and $\circledast\!\!-$ involve arbitrarily large histories of actions. The second restriction makes our result less relevant.

A dual question is whether every formula in $\mathcal{L}^{K*\mathcal{E}}$ is equivalent to a formula in $\mathcal{L}^{K*}$: can we also get rid of the action model modalities and stick with the epistemic separation language only? We are uncertain about the answer to this question. However, the language $\mathcal{L}^{K*}$ wherein we can only indirectly refer to actions by way of $*$ and $-\!\!*$, already permits some $*$-validities of interest. It is succinctly discussed in Sect. 5.

4.1 Validities for Empty Histories and a Bound Max

We assume bound **max** throughout Sect. 4, and also that $\mathcal{E} = (E, \approx, \mathit{pre}, \mathit{post})$. The crucial validities in the reduction are as follows. They will be successively shown in subsequent lemmas and propositions. Recall that $\models$ is validity with respect to empty history models. The obvious proof of Lemma 2 is omitted.

$$\begin{array}{lll}
\models I & \leftrightarrow \top & \\
\models [h]\varphi & \leftrightarrow \top & \text{where } |h| > \mathbf{max} \\
\models [h](\varphi * \psi) & \leftrightarrow \mathit{pre}(h) \to \bigvee_{h' \sqsubseteq h}(\langle h' \rangle \varphi \wedge \langle h \backslash h' \rangle \psi) & \text{where } |h| \leq \mathbf{max} \\
\models [h](\varphi -\!\!\circledast \psi) & \leftrightarrow \mathit{pre}(h) \to \bigwedge_{|h'| \leq \mathbf{max} - |h|}(\langle h' \rangle \varphi \to [hh']\psi) & \text{where } |h| \leq \mathbf{max} \\
\models [h](\varphi \circledast\!\!- \psi) & \leftrightarrow \mathit{pre}(h) \to \bigwedge_{|h'| \leq \mathbf{max} - |h|}(\langle h' \rangle \varphi \to [h'h]\psi) & \text{where } |h| \leq \mathbf{max}
\end{array}$$

Lemma 2. $\models I \leftrightarrow \top$

Lemma 3. $\models [h]\varphi \leftrightarrow \top$, *where* $|h| > \mathbf{max}$.

Proof. We show that $\models [h]\varphi$, which is equivalent to $\models [h]\varphi \leftrightarrow \top$. Given $\mathcal{ME}^{\mathbf{max}}$ with $s \in \mathcal{D}(\mathcal{ME}^{\mathbf{max}})$. Let $h' \sqsubset h$ be the prefix of h with $|h'| = \mathbf{max}$, and assume $s \models pre(h')$. We need to show that $sh' \models [h \backslash h']\varphi$. Let $h \backslash h' = eh''$. According to the semantics of dynamic modalities, $sh' \models [e][h'']\varphi$ is equivalent to ($|h'| < \mathbf{max}$ and $sh' \models pre(e)$ imply $sh'e \models [h'']\varphi$). As $|h'| < \mathbf{max}$ is false, the whole implication is true.

Proposition 2. *$\models [h](\varphi * \psi) \leftrightarrow (pre(h) \rightarrow \bigvee_{h' \sqsubseteq h}(\langle h'\rangle\varphi \wedge \langle h \backslash h'\rangle\psi))$, where $|h| \leq \mathbf{max}$.*

Proof. Given $\mathcal{ME}^{\mathbf{max}}$ and $s \in \mathcal{D}(\mathcal{ME}^{\mathbf{max}})$, assume $s \models [h](\varphi * \psi)$. In order to prove that $s \models pre(h) \rightarrow \bigvee_{h' \sqsubseteq h}(\langle h'\rangle\varphi \wedge \langle h \backslash h'\rangle\psi)$, let us further assume that $s \models pre(h)$. From that and the initial assumption we obtain that $sh \models \varphi * \psi$. Then, there are h', h'' such that $h = h'h''$, $sh' \models \varphi$, and $sh'' \models \psi$ (note that $h'' = h \backslash h'$). From that we obtain $s \models \langle h'\rangle\varphi$ respectively $s \models \langle h''\rangle\psi$, and therefore $sh' \models \langle h'\rangle\varphi \wedge \langle h''\rangle\psi$, and therefore (using that $h'' = h \backslash h'$) $s \models \bigvee_{h' \sqsubseteq h}(\langle h'\rangle\varphi \wedge \langle h \backslash h'\rangle\psi)$, as required. For the other direction, now assume $s \models pre(h) \rightarrow \bigvee_{h' \sqsubseteq h}(\langle h'\rangle\varphi \wedge \langle h \backslash h'\rangle\psi)$, and towards showing that $s \models [h](\varphi * \psi)$, let us again further assume that $s \models pre(h)$. Thus $s \models \bigvee_{h' \sqsubseteq h}(\langle h'\rangle\varphi \wedge \langle h \backslash h'\rangle\psi)$. Let h' be such that $s \models \langle h'\rangle\varphi \wedge \langle h \backslash h'\rangle\psi$. Then, as before, $sh' \models \varphi$ and $s(h \backslash h') \models \psi$ so that $sh \models \varphi * \psi$.

Proposition 3. *$\models [h](\varphi \twoheadrightarrow \psi) \leftrightarrow (pre(h) \rightarrow \bigwedge_{|h'| \leq \mathbf{max} - |h|}(\langle h'\rangle\varphi \rightarrow [hh']\psi))$, where $|h| \leq \mathbf{max}$.*

Proof. Given $\mathcal{ME}^{\mathbf{max}}$ and $s \in \mathcal{D}(\mathcal{ME}^{\mathbf{max}})$, assume $s \models [h](\varphi \twoheadrightarrow \psi)$. Towards showing that $s \models pre(h) \rightarrow \bigwedge_{|h'| \leq \mathbf{max} - |h|}(\langle h'\rangle\varphi \rightarrow [hh']\psi)$, further assume $s \models pre(h)$, let h' be such that $|h'| \leq \mathbf{max} - |h|$ and let $s \models \langle h'\rangle\varphi$. It then remains to show that $s \models [hh']\psi$. In order to obtain that we make one final assumption namely $s \models pre(hh')$, so that $shh' \in \mathcal{D}(\mathcal{ME}^{\mathbf{max}})$. It then remains to show that $shh' \models \psi$. From $s \models \langle h'\rangle\varphi$ we obtain that $s \models pre(h')$ and $sh' \models \varphi$. From $s \models [h](\varphi \twoheadrightarrow \psi)$ and $s \models pre(h)$ we deduce $sh \models \varphi \twoheadrightarrow \psi$. From that, $sh' \models \varphi$, and $shh' \in \mathcal{D}(\mathcal{ME}^{\mathbf{max}})$ we then get $shh' \models \psi$, as required.

For the other direction, we now assume $s \models (pre(h) \rightarrow \bigwedge_{|h'| \leq \max - |h|}(\langle h'\rangle\varphi \rightarrow [hh']\psi))$, and towards showing that $s \models [h](\varphi \twoheadrightarrow \psi)$ we further assume that $s \models pre(h)$, so that it remains to show that $sh \models \varphi \twoheadrightarrow \psi$. Let now h' be such that $|h'| \leq \mathbf{max} - |h|$, $s \models pre(h')$, $s \models pre(hh')$, and $sh' \models \varphi$. We need to show that $shh' \models \psi$. From $sh' \models \varphi$ we get $s \models \langle h'\rangle\varphi$. Now using the initial assumption, $s \models pre(h)$, $s \models \langle h'\rangle\varphi$, and $s \models pre(hh')$, we obtain that $shh' \models \psi$, as required.

Proposition 4. *$\models [h](\varphi \twoheadleftarrow \psi) \leftrightarrow (pre(h) \rightarrow \bigwedge_{|h'| \leq \mathbf{max} - |h|}(\langle h'\rangle\varphi \rightarrow [h'h]\psi))$, where $|h| \leq \mathbf{max}$.*

Proof. The proof is obtained from the proof of Proposition 3 by replacing hh' by $h'h$ everywhere in that proof. The order of h and h' does not play a role in the proof.

From Propositions 2 and 3 it follows in particular, as the empty history can only be decomposed into empty and empty, and as $pre(\epsilon) = \top$, that:

Corollary 1.

$$\begin{array}{l} \models \varphi * \psi \ \ \leftrightarrow \varphi \wedge \psi \\ \models \varphi \mathrel{-\!\!*} \psi \leftrightarrow \bigwedge_{|h| \leq \mathbf{max}} (\langle h \rangle \varphi \rightarrow [h]\psi) \\ \models \varphi \mathrel{*\!\!-} \psi \leftrightarrow \bigwedge_{|h| \leq \mathbf{max}} (\langle h \rangle \varphi \rightarrow [h]\psi) \end{array}$$

4.2 Termination of Reduction from $\mathcal{L}^{K*\mathcal{E}}$ to $\mathcal{L}^{K\mathcal{E}}$

We now show termination of the reduction. We define a translation t from $\mathcal{L}^{K*\mathcal{E}}$ to $\mathcal{L}^{K\mathcal{E}}$, and a complexity/weight measure c from $\mathcal{L}^{K*\mathcal{E}}$ to $\mathbb{N}$ and we then show that the translation is correct (is truth –value– preserving) and terminates.

For the translation it is of tantamount importance that we use an outside-in reduction strategy. This is because the reductions are $\models$ validities, they are **not** $\models^*$ validities: they are validities with respect to models with empty histories. In other words, the translation t to be defined is only correct when all modalities $[h]$ occurring in formulas are interpreted in models with empty histories only. For example, given $[h](K_a p \rightarrow [h']q)$, we can only rewrite $[h]$ and we cannot (at this stage) rewrite $[h']$. This can only happen at a later stage in the rewriting procedure after the formula has been massaged into a shape wherein $[h']$ (or some modality derived from it in the process of rewriting) can be interpreted in an empty history model. It is for this reason that the translation below does not contain a clause for $[h][h']\varphi$: in such a case we are compelled to reduce $[hh']\varphi$, or more precisely (as the formulas are identical by notational abbreviation), to find a clause in the translation function for the main logical connective of φ.

If an inside-out reduction had been possible, a proof by natural induction on the number of $*$, $\mathrel{-\!\!*}$, and $\mathrel{*\!\!-}$ occurrences would have been possible (in a slightly refined lexicographic way comparing triples of natural numbers). As the reduction is outside-in, applying an equivalence such as $[h](\varphi * \psi) \leftrightarrow (pre(h) \rightarrow \bigvee_{h' \sqsubseteq h} (\langle h' \rangle \varphi \wedge \langle h \backslash h' \rangle \psi)$ does not necessarily reduce the number of separation connectives on the righthand side of the equation. Any further $*$ occurring in φ on the left, will now occur as many times on the right as there as prefixes h' of h. Therefore we have to resort to the standard method of defining a weight/complexity measure on formulas.

Definition 9 (Complexity).

$$\begin{array}{ll} c(p) = c(\bot) = c(I) & = 1 \\ c(\neg\varphi) & = 1 + c(\varphi) \\ c(\varphi \wedge \psi) & = 1 + \max\{c(\varphi), c(\psi)\} \\ c(\varphi * \psi) & = \mathbf{max} + 1 + \max\{c(\varphi), c(\psi)\} \\ c(\varphi \mathrel{-\!\!*} \psi) = c(\varphi \mathrel{*\!\!-} \varphi) & = 3 + \Sigma_{i=0}^{\mathbf{max}} |E|^i + c(\mathcal{E})^{\mathbf{max}} \cdot \max\{c(\varphi), c(\psi)\} \\ c(K_a\varphi) & = 1 + c(\varphi) \\ c([e]\varphi) & = c(\mathcal{E}) \cdot c(\varphi) \\ c(\mathcal{E}) & = 3 + |E| + \max\{pre(e), post(e)(p) \mid e \in E, p \in P\} \end{array}$$

From $c([e]\varphi) = c(\mathcal{E}) \cdot c(\varphi)$ we obtain that $c([h]\varphi) = c(\mathcal{E})^{|h|} \cdot c(\varphi)$ for arbitrary histories h. We may abuse the language and write $c(h)$ for $c(\mathcal{E})^{|h|}$. In $c(\varphi \twoheadrightarrow \psi)$ and $c(\varphi \twoheadrightarrow \psi)$, the conjunction $\bigwedge_{h \leq \max}$ is over all histories of length at most **max**, where each action e in that history can be one of $|E|$. The total number of histories therefore involves a geometric series $\Sigma_{i=0}^{\mathbf{max}} |E|^i$.

Definition 10 (Translation). *Where* $1 \leq |h| \leq \mathbf{max}$ *except in clause* $t([h]\varphi)$.

$$
\begin{array}{lll}
t(p) & = p & \\
t(\bot) & = \bot & \\
t(I) & = \top & \\
t(\neg\varphi) & = \neg t(\varphi) & \\
t(\varphi \wedge \psi) & = t(\varphi) \wedge t(\psi) & \\
t(K_a\varphi) & = K_a t(\varphi) & \\
t(\varphi * \psi) & = t(\varphi \wedge \psi) & \\
t(\varphi \twoheadrightarrow \psi) & = t(\bigwedge_{|h| \leq \mathbf{max}}(\langle h \rangle \varphi \rightarrow [h]\psi)) & \\
t(\varphi \twoheadleftarrow \psi) & = t(\bigwedge_{|h| \leq \mathbf{max}}(\langle h \rangle \varphi \rightarrow [h]\psi)) & \\
t([h]\varphi) & = \top & \textit{where } |h| > \mathbf{max} \\
t([h]p) & = pre(h) \rightarrow post(h)(p) & \\
t([h]\bot) & = \neg pre(h) & \\
t([h]I) & = \neg pre(h) & \\
t([h]\neg\varphi) & = pre(h) \rightarrow t(\neg[h]\varphi) & \\
t([h](\varphi \wedge \psi)) & = t([h]\varphi \wedge [h]\psi) & \\
t([h]K_a\varphi) & = pre(h) \rightarrow t(\bigwedge_{h \sim_a h'} K_a[h']\varphi) & \\
t([h](\varphi * \psi)) & = pre(h) \rightarrow t(\bigvee_{h' \sqsubseteq h}(\langle h' \rangle \varphi \wedge \langle h \backslash h' \rangle \psi)) & \\
t([h](\varphi \twoheadrightarrow \psi)) & = pre(h) \rightarrow t(\bigwedge_{|h'| \leq \mathbf{max} - |h|}(\langle h' \rangle \varphi \rightarrow [hh']\psi)) & \\
t([h](\varphi \twoheadleftarrow \psi)) & = pre(h) \rightarrow t(\bigwedge_{|h'| \leq \mathbf{max} - |h|}(\langle h' \rangle \varphi \rightarrow [h'h]\psi)) &
\end{array}
$$

As action model pre- and postconditions are in $\mathcal{L}^{K\mathcal{E}}$ (contain no I, $*$, $\twoheadrightarrow$, and $\twoheadleftarrow$), we need not to translate (i.e., eliminate those operators from) those parts.

Lemma 4. *All the following hold:*

1. $c(\mathcal{E}) \geq 5$
2. $c(\varphi \vee \psi) \leq 3 + \max\{c(\varphi), c(\psi)\}$
3. $c(\varphi \rightarrow \psi) \leq 3 + \max\{c(\varphi), c(\psi)\}$
4. $c(pre(h)) \leq c(h)$

Proof. We prove the successive items.

1. $c(\mathcal{E}) = 3 + |E| + \mathbf{max}\{pre(e), post(e)(p) \mid e \in E, p \in P\} \geq 3 + 1 + 1 = 5$.
2. $c(\varphi \vee \psi) = c(\neg(\neg\varphi \wedge \neg\psi)) = 1 + c(\neg\varphi \wedge \neg\psi)) \leq 3 + \max\{c(\varphi), c(\psi)\}$
3. $c(\varphi \rightarrow \psi) = c(\neg(\varphi \wedge \neg\psi) = 1 + c(\varphi \wedge \neg\psi) = 2 + \max\{c(\varphi), c(\psi) + 1\} \leq 3 + \max\{c(\varphi), c(\psi)\}$
4. This follows from: $c(h) = c(\mathcal{E})^{|h|}$, $c(pre(e)) \leq \max\{c(pre(e)), c(post(e)(p)) \mid e \in E, p \in P\}$, and (as $|h| > 1$ so that $h = h'e'$) $pre(h) = \langle h' \rangle pre(e')$.

Lemma 5. *The following inequalities hold for arbitrary formulas, where* $1 \leq |h| \leq \mathbf{max}$ *except in the clause for* $c([h]\varphi)$.

$$\begin{array}{lll}
c(\varphi * \psi) & > c(\varphi \wedge \psi) & \\
c(\varphi -\!\!* \psi) & > c(\bigwedge_{|h| \leq \mathbf{max}}(\langle h \rangle \varphi \rightarrow [h]\psi)) & \\
c(\varphi *\!\!- \psi) & > c(\bigwedge_{|h| \leq \mathbf{max}}(\langle h \rangle \varphi \rightarrow [h]\psi)) & \\
c([h]\varphi) & > c(\top) & \text{where } |h| > \mathbf{max} \\
c([h]p) & > c(pre(h) \rightarrow post(h)(p)) & \\
c([h]\bot) & > c(\neg pre(h)) & \\
c([h]I) & > c(\neg pre(h)) & \\
c([h]\neg\varphi) & > c(pre(h) \rightarrow \neg[h]\varphi) & \\
c([h](\varphi \wedge \psi)) & > c([h]\varphi \wedge [h]\psi) & \\
c([h]K_a\varphi) & > c(pre(h) \rightarrow \bigwedge_{h' \sim_a h} K_a[h']\varphi) & \\
c([h](\varphi * \psi)) & > c(pre(h) \rightarrow \bigvee_{h'.h''=h}(\langle h' \rangle \varphi \wedge \langle h'' \rangle \psi)) & \\
c([h](\varphi -\!\!* \psi)) & > c(pre(h) \rightarrow \bigwedge_{|h'| \leq \mathbf{max}-|h|}(\langle h' \rangle \varphi \rightarrow [hh']\psi)) & \\
c([h](\varphi *\!\!- \psi)) & > c(pre(h) \rightarrow \bigwedge_{|h'| \leq \mathbf{max}-|h|}(\langle h' \rangle \varphi \rightarrow [h'h]\psi)) &
\end{array}$$

Proof. We prove the separate items one by one.

$$\begin{aligned}
c(\varphi * \psi) &= \mathbf{max} + 1 + \max\{c(\varphi), c(\psi)\} \\
&> 1 + \max\{c(\varphi), c(\psi)\} \qquad \text{this bound is sharp} \\
&= c(\varphi \wedge \psi)
\end{aligned}$$

$$\begin{aligned}
c(\varphi -\!\!* \psi) &= 3 + \Sigma_{i=0}^{\mathbf{max}}|E|^i + c(\mathcal{E})^{\mathbf{max}} \cdot \max\{c(\varphi), c(\psi)\} \\
&> \Sigma_{i=0}^{\mathbf{max}}|E|^i - 1 + 3 + c(\mathcal{E})^{\mathbf{max}} \cdot \max\{c(\varphi), c(\psi)\} \qquad (@) \\
&\geq c(\bigwedge_{|h| \leq \mathbf{max}}(\langle h \rangle \varphi \rightarrow [h]\psi))
\end{aligned}$$

(@): The number of h with $|h| \leq \mathbf{max}$ is bounded by $\Sigma_{i=0}^{\mathbf{max}}|E|^i$, so one less for the number of $\wedge$-symbols. Then, $c(\mathcal{E})^{\mathbf{max}}$ is the weight of the largest such h.

The case $c(\varphi *\!\!- \psi)$ is treated just as the case $c(\varphi -\!\!* \psi)$.

$$c([h]\varphi) = c(h) \cdot c(\varphi) \geq 5c(\varphi) \geq 5 > 2 = c(\neg\bot) = c(\top) \qquad \text{when } |h| \geq \mathbf{max}$$

Note that $c(h) = c(\mathcal{E})^{|h|} \geq c(\mathcal{E}) \geq 5$. We do not use $|h| \geq \mathbf{max}$ but only $|h| \geq 1$.

$$\begin{aligned}
c([h]p) &= c(\mathcal{E})^{|h|} \cdot c(p) \\
&= c(\mathcal{E})^{|h|} \cdot 1 \\
&= c(\mathcal{E})^{|h|} \\
&\geq c(\mathcal{E}) \\
&= 3 + |E| + \max\{c(pre(e)), c(post(e)(p)) \mid e \in E, p \in P\} \\
&> 3 + \max\{c(pre(e)), c(post(e)(p)) \mid e \in E, p \in P\} \\
&\geq c(pre(e) \rightarrow post(e)(p))
\end{aligned}$$

$$\begin{aligned}
c([h]\bot) &= c(\mathcal{E})^{|h|} \cdot c(\bot) \\
&= c(\mathcal{E})^{|h|} \\
&\geq c(\mathcal{E}) \\
&= 3 + |E| + \max\{c(pre(e)), c(post(e)(p)) \mid e \in E, p \in P\} \\
&> 1 + c(pre(h)) \\
&= c(\neg pre(h))
\end{aligned}$$

$$c([h]I) > c(\neg pre(h))$$

The case $c([h]I)$ is treated as the case $c([h]\bot)$, as $c(I) = c(\bot) = 1$.

$$\begin{aligned}
c([h]\neg\varphi) &= c(\mathcal{E})^{|h|} \cdot c(\neg\varphi) \\
&= c(\mathcal{E})^{|h|} \cdot (1 + c(\varphi)) \\
&= c(\mathcal{E})^{|h|} + c(\mathcal{E})^{|h|} \cdot c(\varphi) && c(\mathcal{E}) \geq 5, |h| \geq 1 \\
&\geq 5 + c(\mathcal{E})^{|h|} \cdot c(\varphi) && \text{note that this bound is sharp} \\
&> 4 + c(\mathcal{E})^{|h|} \cdot c(\varphi) && \text{use that } c(pre(h)) \leq c(h) \\
&= 3 + \max\{c(pre(h)), 1 + c(\mathcal{E})^{|h|} \cdot c(\varphi)\} \\
&= 3 + \max\{c(pre(h)), 1 + c([h]\varphi)\} \\
&= 3 + \max\{c(pre(h)), c(\neg[h]\varphi)\} \\
&\geq c(pre(h) \rightarrow \neg[h]\varphi)
\end{aligned}$$

$$\begin{aligned}
c([h](\varphi \wedge \psi)) &= c(\mathcal{E})^{|h|} \cdot c(\varphi \wedge \psi) \\
&= c(\mathcal{E})^{|h|} \cdot (1 + \max\{c(\varphi), c(\psi)\}) \\
&= c(\mathcal{E})^{|h|} + c(\mathcal{E})^{|h|} \cdot \max\{c(\varphi), c(\psi)\} \\
&> 1 + c(\mathcal{E})^{|h|} \cdot \max\{c(\varphi), c(\psi)\} \\
&= 1 + \max\{c(\mathcal{E})^{|h|} \cdot c(\varphi), c(\mathcal{E})^{|h|} \cdot c(\psi)\} \\
&= 1 + \max\{c([h]\varphi), c([h]\psi)\} \\
&= c([h]\varphi \wedge [h]\psi)
\end{aligned}$$

$$\begin{aligned}
c([h]K_a\varphi) &= c(\mathcal{E})^{|h|} \cdot c(K_a\varphi) \\
&= c(\mathcal{E})^{|h|} \cdot (1 + c(\varphi)) \\
&= c(\mathcal{E})^{|h|} + c(\mathcal{E})^{|h|} \cdot c(\varphi) \\
&\geq c(\mathcal{E})^{|h|} + c(\mathcal{E})^{|h|} \cdot c(\varphi) \\
&\geq 3 + |E|^{|h|} + \max\{c(pre(e)), c(post(e)(p)) \mid \dots\} + c(\mathcal{E})^{|h|} \cdot c(\varphi) \\
&\geq 4 + |E|^{|h|} + c(\mathcal{E})^{|h|} \cdot c(\varphi) && \text{this bound is sharp when } h = 1 \\
&> 3 + |E|^{|h|} + c(\mathcal{E})^{|h|} \cdot c(\varphi) \\
&= 3 + |E|^{|h|} - 1 + c(K_a[h]\varphi) && (*) \\
&\geq 3 + c(\textstyle\bigwedge_{h' \sim_a h} K_a[h']\varphi) && \text{as } c(pre(h)) \leq c(h) \\
&\geq 3 + \max\{c(pre(h)), c(\textstyle\bigwedge_{h' \sim_a h} K_a[h']\varphi)\} \\
&\geq c(pre(h) \rightarrow \textstyle\bigwedge_{h' \sim_a h} K_a[h']\varphi)
\end{aligned}$$

$(*)$: There are at most $|E|$ indistinguishable f from a given e, therefore there are at most $|E|^{|h|}$ indistinguishable h' from a given h. Minus 1 when counting the number of $\wedge$-symbols in a conjunction of that length.

$$\begin{aligned}
c([h](\varphi * \psi)) &= c(\mathcal{E})^{|h|} \cdot c(\varphi * \psi) \\
&= c(\mathcal{E})^{|h|} \cdot (\mathbf{max} + 1 + \max\{c(\varphi), c(\psi)\}) \\
&= \mathbf{max} \cdot c(\mathcal{E})^{|h|} + c(\mathcal{E})^{|h|} + c(\mathcal{E})^{|h|} \cdot \max\{c(\varphi), c(\psi)\} \\
&\geq 5 + 3\mathbf{max} + c(\mathcal{E})^{|h|} \cdot \max\{c(\varphi), c(\psi)\} && \mathbf{max} \geq |h| \\
&> 3 + 3\,h| + 1 + c(\mathcal{E})^{|h|} \cdot \max\{c(\varphi), c(\psi)\} && (**) \\
&\geq 3 + c(\textstyle\bigvee_{h' \sqsubseteq h}(\langle h'\rangle\varphi \wedge \langle h \backslash h'\rangle\psi)) \\
&= 3 + \max\{c(pre(e)), c(\textstyle\bigvee_{h' \sqsubseteq h}(\langle h'\rangle\varphi \wedge \langle h \backslash h'\rangle\psi))\} \\
&\geq c(pre(h) \rightarrow \textstyle\bigvee_{h' \sqsubseteq h}(\langle h'\rangle\varphi \wedge \langle h \backslash h'\rangle\psi))
\end{aligned}$$

($**$): Each of $|h|$ disjunctions adds 3, plus 1 for the conjunction.

$$\begin{aligned}
c([h](\varphi \twoheadrightarrow \psi)) &= c(\mathcal{E})^{|h|} \cdot c(\varphi \twoheadrightarrow \psi) \\
&= c(\mathcal{E})^{|h|} \cdot (3 + \Sigma_{i=0}^{\mathbf{max}} |E|^i + c(\mathcal{E})^{\mathbf{max}} \cdot \max\{c(\varphi), c(\psi)\}) \\
&> 2 \cdot (3 + \Sigma_{i=0}^{\mathbf{max}} |E|^i + c(\mathcal{E})^{\mathbf{max}} \cdot \max\{c(\varphi), c(\psi)\}) \\
&> 3 + 3 + \Sigma_{i=0}^{\mathbf{max}-1} |E|^i - 1 + c(\mathcal{E})^{\mathbf{max}} \cdot \max\{c(\varphi), c(\psi)\} \\
&\geq 3 + c(\textstyle\bigwedge_{|h'| \leq \mathbf{max}-|h|} (\langle h' \rangle \varphi \rightarrow [hh']\psi)) \\
&= 3 + \max\{c(pre(h), c(\textstyle\bigwedge_{|h'| \leq \mathbf{max}-|h|} (\langle h' \rangle \varphi \rightarrow [hh']\psi))\} \\
&\geq c(pre(h) \rightarrow \textstyle\bigwedge_{|h'| \leq \mathbf{max}-|h|} (\langle h' \rangle \varphi \rightarrow [hh']\psi))
\end{aligned}$$

The case $c([h](\varphi \twoheadrightarrow \psi))$ is serious overkill, as in $c(\varphi \twoheadrightarrow \psi)$ weight $c(h)$ is already factored in. But in case $c(\varphi \twoheadrightarrow \varphi)$ this was indispensable. We also use that the h' we quantify over must have length at most $\mathbf{max} - 1$ (as $|h| \geq 1$), one less therefore than in the case $c(\varphi \twoheadrightarrow \psi)$.

The case of $[h](\varphi \ast\!\!- \psi)$ is similar to the case $[h](\varphi \ast\!\!- \psi)$.

Theorem 1. *Every formula in $\mathcal{L}^{K*\mathcal{E}}$ is equivalent to a formula in $\mathcal{L}^{K\mathcal{E}}$*

Proof. We recall that the reduction is outside-in. Consider a formula $\varphi \in \mathcal{L}^{K*\mathcal{E}}$, and apply a clause of translation t (Definition 10) on φ. Consider $c(\varphi)$. If φ is one of p, $\bot$, or $\top$, termination is trivial (such as $t(p) = p$). If the main logical connective of φ commutes with t (such as in $t(\psi \wedge \chi) = t(\psi) \wedge t(\chi)$), then it is obvious that the complexities of subformulas of φ are strictly lower than the complexities of φ (such as $c(\psi) < c(\psi \wedge \chi)$ above, since the complexity of a conjunction is that of its conjuncts **plus one**). If the main logical connective of φ does not commute with t, we have one of the cases spelled out in Lemma 5 and use that for all those cases $c(\varphi) > c(t(\varphi))$. Therefore, in every step of translation t that we apply, the weight c is strictly less. As $c(\varphi)$ is a natural number, this is bounded by 0. Therefore the reduction terminates.

Corollary 2. *Every formula in $\mathcal{L}^{K*\mathcal{E}}$ is equivalent to a formula in $\mathcal{L}^{K}$.*

As a consequence, the logic AMHSL (for empty history models, and given bound **max**) is therefore completely axiomatized by the reduction axioms of Sect. 4.1 and the axiomatization of action model logic with factual change [17] (where we recall Proposition 1) that extends S5.

5 Remarks and Perspectives

Considering the history-based logical semantics with the bound **max** of the epistemic history model, it appears that model checking is decidable. Satisfiability may be a different matter as the $\twoheadrightarrow$ and $\ast\!\!-$ connectives quantify over histories of arbitrary finite length, even if we know that quantifying over action models results in a decidable logic [9]. If histories are unbounded, we are uncertain if $*$, $\ast\!\!-$ and $\twoheadrightarrow$ can be eliminated by reduction from the language $\mathcal{L}^{K*\mathcal{E}}$. It is also highly uncertain if action model modalities can be eliminated by reduction from the language $\mathcal{L}^{K*\mathcal{E}}$, so that we get a $\mathcal{L}^{K*}$ formula.

The logical semantics for the language $\mathcal{L}^{K*}$ is interesting in its own right. Dynamic epistemic logics allowing reasoning about the past are very rare [2,4]. In $\mathcal{L}^{K*}$ we can refer to the past in novel and unexpected ways. For example, $sh \models \psi * \top$ formalizes that ψ was true in the past (there must be h', h'' with $h'h'' = h$ such that $sh' \models \psi$ and $sh'' \models \top$, where the latter is trivially true). A formula like $\neg I * \neg I * \neg I$ is only true after at least three actions have been executed (etcetera). Would such a logic be axiomatizable? We can see that $(K_a\varphi * K_a\psi) \rightarrow K_a(\varphi * \psi)$ is valid. However, $K_a(\varphi * \psi) \rightarrow (K_a\varphi * K_a\psi)$ is invalid.

Finally, instead of decomposing action histories into prefixes and postfixes, such that resource update required distinct $-\!\!*$ (append postfix) and $*\!\!-$ (append prefix) connectives, and where $\varphi * \psi$ may not be equivalent to $\psi * \varphi$, we could also contemplate decomposing an action history into a subsequence and its complement (such as when decomposing $a.b.c.d$ into $a.c$ and $b.d$). Now, one $-\!\!*$ connective suffices that can be interpreting as 'enriching' a given history with bits and pieces of action sequences where it pleases us, and $*$ has become commutative. This comes closer to the philosophy of separation.

Acknowledgements. We very much wish to thank the reviewers for their comments. A reviewer pointed out an error in the proof of the case $c([h](\varphi * \psi))$ of Lemma 5, that needed repair by strengthening the weight of case $c(\varphi * \psi)$ in Definition 9. Another reviewer mentioned that AMHSL allows to reason about the length of histories. We added an example.

References

1. Attamah, M., van Ditmarsch, H., Grossi, D., van der Hoek, W.: Knowledge and gossip. In: Proceedings of 21st ECAI, pp. 21–26. IOS Press (2014)
2. Balbiani, P., van Ditmarsch, H., Herzig, A.: Before announcement. In: Beklemishev, L.D., Demri, S., Maté, A. (eds.) Advances in Modal Logic, vol. 11, pp. 58–77. College Publications (2016)
3. Baltag, A., Moss, L.S., Solecki, S.: The logic of public announcements, common knowledge, and private suspicions. In: Proceedings of 7th TARK, pp. 43–56 (1998)
4. Baltag, A., Özgün, A., Vargas Sandoval, A.L.: Arbitrary public announcement logic with memory. J. Philos. Log. **52**, 53–110 (2022). https://doi.org/10.1007/s10992-022-09664-6
5. Belle, V., Bolander, T., Herzig, A., Nebel, B.: Epistemic planning: Perspectives on the special issue. Artif. Intell. (2022). https://doi.org/10.1016/j.artint.2022.103842
6. Courtault, J.-R., Galmiche, D.: A modal separation logic for resource dynamics. J. Log. Comput. **28**(4), 733–778 (2018)
7. Courtault, J.-R., van Ditmarsch, H., Galmiche, D.: A public announcement separation logic. Math. Struct. Comput. Sci. **29**(6), 828–871 (2019)
8. Galmiche, D., Kimmel, P., Pym, D.: A substructural epistemic resource logic: theory and modelling applications. J. Log. Comput. **29**(8), 1251–1287 (2019)
9. Hales, J.: Arbitrary action model logic and action model synthesis. In: Proceedings of 28th LICS, pp. 253–262. IEEE (2013)
10. Ishtiaq, S., O'Hearn, P.: BI as an assertion language for mutable data structures. In: Proceedings of 28th POPL, pp. 14–26 (2001)

11. Kooi, B.: Expressivity and completeness for public update logics via reduction axioms. J. Appl. Non-Classical Log. **17**(2), 231–254 (2007)
12. Miller, J.S., Moss, L.S.: The undecidability of iterated modal relativization. Stud. Log. **79**(3), 373–407 (2005)
13. Moss, L.S.: Dynamic epistemic logic. In: van Ditmarsch, H., Halpern, J.Y., van der Hoek, W., Kooi, B. (eds.) Handbook of Epistemic Logic, pp. 261–312. College Publications (2015)
14. Plaza, J.A.: Logics of public communications. In: Proceedings of the 4th ISMIS, pp. 201–216. Oak Ridge National Laboratory (1989)
15. Pym, D.: The Semantics and Proof Theory of the Logic of Bunched Implications. Applied Logic Series, vol. 26. Springer, Heidelberg (2002). https://doi.org/10.1007/978-94-017-0091-7
16. van Benthem, J., Gerbrandy, J.D., Hoshi, T., Pacuit, E.: Merging frameworks for interaction. J. Philos. Log. **38**, 491–526 (2009)
17. van Benthem, J., van Eijck, J., Kooi, B.: Logics of communication and change. Inf. Comput. **204**(11), 1620–1662 (2006)
18. van Ditmarsch, H., Galmiche, D., Gawek, M.: An epistemic separation logic with action models. J. Logic Lang. Inform. **32**(1), 89–116 (2023). https://doi.org/10.1007/s10849-022-09372-z
19. van Ditmarsch, H., Halpern, J.Y., van der Hoek, W., Kooi, B.: An introduction to logics of knowledge and belief. In: van Ditmarsch, H., Halpern, J.Y., van der Hoek, W., Kooi, B. (eds.) Handbook of Epistemic Logic, pp. 1–51 (2015)
20. van Ditmarsch, H., Ruan, J., van der Hoek, W.: Connecting dynamic epistemic and temporal epistemic logics. Log. J. IGPL **21**(3), 380–403 (2013)
21. van Ditmarsch, H., van der Hoek, W., Kooi, B.: Dynamic Epistemic Logic. Synthese Library, vol. 337. Springer, Heidelberg (2008). https://doi.org/10.1007/978-1-4020-5839-4
22. van Ditmarsch, H., van Eijck, J., Pardo, P., Ramezanian, R., Schwarzentruber, F.: Epistemic protocols for dynamic gossip. J. Appl. Log. **20**, 1–31 (2017)

Conditional Obligations in Justification Logic

Federico L. G. Faroldi, Atefeh Rohani(✉), and Thomas Studer

Institute of Computer Science, University of Bern, Bern, Switzerland
{federico.faroldi,atefeh.rohani,thomas.studer}@unibe.ch

Abstract. This paper presents a justification counterpart for dyadic deontic logic, which is often argued to be better than Standard Deontic Logic at representing conditional and contrary-to-duty obligations, such as those exemplified by the notorious Chisholm's puzzle. We consider the alethic-deontic system (E) and present the explicit version of this system (JE) by replacing the alethic Box-modality with proof terms and the dyadic deontic Circ-modality with justification terms. The explicit representation of strong factual detachment (SFD) is given and finally soundness and completeness of the system (JE) with respect to basic models and preference models is established.

Keywords: dyadic deontic logic · justification logic · preference models

1 Introduction

Dyadic Deontic Logic (**DDL**) is an extension of Monadic Deontic Logic (**MDL**) that employs a dyadic conditional represented by $\bigcirc(B/A)$, which is weaker than the expression $A \rightarrow \bigcirc B$ from **MDL**. The conditional $\bigcirc(B/A)$ is read as "B is obligatory, given A" so that A is the antecedent and B is the consequent [7]. In contrast to Monadic Deontic Logic, which relies on Kripke-style possible world models, Dyadic Deontic Logic works with preference-based semantics, in which the possible worlds are related according to their betterness or relative goodness. Under this semantics, $\bigcirc(B/A)$ is true when all best A-worlds are B-worlds [17]. One of the puzzles that is solved by preference models is the so-called *Chisholm's set*.

1.1 Chisholm's Set

Chisholm [6] was the initiator of the so-called "contrary-to-duty" problem, which deals with the question of what to do when primary obligations are violated. The main goal of **DDL** was to deal with these obligations, which works with setting an order on the set of worlds [15,23,24]. Here is an example of Chisholm's set. Consider the following sentences:

F. Faroldi was supported by the Ambizione grant PZ00P1_201906, A. Rohani and T. Studer were supported by the Swiss National Science Foundation grant 200020_184625.

H. H. Hansen et al. (Eds.): WoLLIC 2023, LNCS 13923, pp. 178–193, 2023.
https://doi.org/10.1007/978-3-031-39784-4_11

1. Thomas should take the math exam.
2. If he takes the math exam, he should register for it.
3. If he does not take the math exam, he should not register for it.
4. He does not take the math exam.

(1) is a primary obligation. (2) is an according-to-duty (ATD) obligation, which says what is obligatory when the primary obligation is satisfied. (3) is a contrary-to-duty obligation (CTD), which says what is obligatory when the primary obligation is violated. (4) is a descriptive premise, saying that the primary obligation is violated. Now we consider how these sentences are formalized in **MDL** and in **DDL** [25].

The paradox raises from formulating the set of formulas:

$$\Gamma = \{(1), (2), (3), (4)\}$$

in monadic deontic logic, where this set is either inconsistent or one sentence is derivable from another sentence in this set. However, Chisholm's set seems intuitively consistent and they also seem to be logically independent sentences. There are four ways to formalize this set in **MDL** as follows:

(1.1) $\bigcirc E$	(2.1) $\bigcirc E$	(3.1) $\bigcirc E$	(4.1) $\bigcirc E$
(1.2)$E \to \bigcirc R$	(2.2) $\bigcirc (E \to R)$	(3.2) $\bigcirc (E \to R)$	(4.2)$E \to \bigcirc R$
(1.3) $\bigcirc (\neg E \to \neg R)$	(2.3)$\neg E \to \bigcirc \neg R$	(3.3) $\bigcirc (\neg E \to \neg R)$	(4.3)$\neg E \to \bigcirc \neg R$
(1.4)$\neg E$	(2.4)$\neg E$	(3.4)$\neg E$	(4.4)$\neg E$

We use Γ_i to denote the set $\{(i.1), (i.2), (i.3), (i.4)\}$. Observe that

$$P \to (\neg P \to Q) \tag{1}$$

is a propositional tautology. Using (1) we find that (1.4) implies (1.2). The set Γ_2 is inconsistent: from (2.1) and (2.2) we get $\bigcirc R$ whereas from (2.3) and (2.4) we get $\bigcirc \neg R$; but in **MDL** obligations must not contradict each other. For Γ_3, note that applying necessitation to (1) and then using distributivity of $\bigcirc$ over $\to$ yields

$$\bigcirc P \to \bigcirc(\neg P \to Q).$$

Therefore, (3.1) implies (3.3). For Γ_4 we again obtain by (1) that (4.4) implies (4.2).

In **DDL**, where there is a ranking on the set of worlds according to their betterness, Chisholm's set does not yield an inconsistency because of the layers of betterness. This ranking can be defined based on the number of obligations violated in each state. Where more obligations are violated, the distance to the ideal state is bigger. The set Γ that models Chisholm's set is given by

$$\Gamma := \{\bigcirc E, \bigcirc(R/E), \bigcirc(\neg R/\neg E), \neg E\}.$$

The following diagram shows a model for Γ. Both R and E are true in w_1, so w_1 is the best world since no obligation of Γ is violated there. E is true in w_2 and neither E nor R is true in w_3. So w_2, w_3 are second best because one obligation is violated there. R is true in w_4 and w_4 is the worst world where two obligations of Γ are violated there.

best		$\bullet w_1,\ R, E$	
2nd best	$\bullet w_2,\ E$		$\bullet w_3$
worst		$\bullet w_4,\ R$	

1.2 Factual Detachment (FD) and Strong Factual Detachment (SFD)

In **DDL**, we do not have the validity of *Factual Detachment* (FD), which is sometimes called "deontic modus-ponens" [16]:

$$(\bigcirc(A/B) \wedge B) \rightarrow \bigcirc A$$

However, a restricted form of factual detachment, namely *strong factual detachment* (SFD),

$$(\bigcirc(A/B) \wedge \Box B) \rightarrow \bigcirc A$$

is valid in DDL. One can interpret SFD as *if A is obligatory given B, and B is settled or proved, then A is obligatory.* An example is as follows:

1. It is obligatory to pay a fine in case someone doesn't pay taxes. ($\bigcirc(F/\neg T)$)
2. The deadline for paying taxes is over and it is proved that someone didn't pay the tax. ($\Box\neg T$)
3. from (1) and (2) and SFD we conclude that it's obligatory for this person to pay the fine. ($\bigcirc F$)

One can consider $\Box A$ as A is proved, which guarantees that from now on we can believe that the person has not paid the taxes. Another principle, which is not valid in **DDL**, is the law of Strengtheninng of the Antecedent (SA):

$$\bigcirc(A/B) \rightarrow \bigcirc(A/B \wedge C)$$

However, the restricted form of strengthening the antecedent is valid in some systems of **DDL**, which is called "Rational Monotony", where $P(A/B)$ is read as A is permissible, given B:

$$P(A/B) \wedge \bigcirc(C/B) \rightarrow \bigcirc(C/B \wedge A)$$

Replacing a modal operator with explicit justifications first appeared in the Logic of Proofs [1], the first justification logic, which was developed by Artemov in order to introduce an explicit counterpart of the modal logic S4 by using classical provability semantics. Various interpretations of justification logic combing justifications with traditional possible worlds models were presented after Fitting [14]. The combination of justification logic and traditional possible world models leads to various interpretations of justification logics [3,21,22]. They make it possible to apply justification logic in many different epistemic and deontic contexts [2,5,18,29,32,33].

Using justification logic for resolving deontic puzzles is already discussed by Faroldi in [9,10,12,13] where the advantages of using explicit reasons are thoroughly explained. In particular, the fact that deontic modalities are hyperintensional, i.e., they can distinguish between logically equivalent formulas, is a good motivation to use justification logic. By replacing the modal operator with a justification term, hyperintensionality is guaranteed by design in justification logic, because two logically equivalent formulas can be justified by different terms. Moreover, the problem of conflicting obligations can be handled well in justification logic [8,11].

This article aims to present an explicit version of **DDL**, where the $\Box$-operator is replaced with *proof terms* satisfying an $\mathsf{S5}$-type axioms and the $\bigcirc$-operator is replaced with suitable *justification terms*. The idea of using two types of terms is already used in [20] and also in our work on explicit non-normal modal logic [30, 31]. We are going to extend the latter framework so that justification terms represent conditional obligations. One of the main motivations for developing justification counterpart of **DDL** is to find explicit reasons for countrary-to-duty and according-to-duty conditional obligations.

The problem with explicit non-normal logics is that the logic is too weak and hardly derives a formula. In the present paper, we remedy this by introducing an explicit version of dyadic deontic logic. This is much stronger than non-normal modal logic and we have appropriate formulations of according-to-duty and contrary-to-duty obligations.

In this article an axiomatization of the justification counterpart of the minimal **DDL** system JE is presented and based on this axiomatization, we provide examples that show the explicit derivation of some well-known formulas such as strong factual detachment (SFD) in our new system. For semantics, basic models are defined, and based on this, preference models are adopted for this system. Soundness and completeness of system $\mathsf{JE}_{\mathsf{CS}}$ with respect to basic models and then preference models are established.

2 Proof Systems for Alethic-Deontic Logic

We consider the proof system for alethic-deontic logic as a basis for our work. In this system, which is denoted by E, two types of modal operators are used: the alethic $\Box$-operator and dyadic deontic $\bigcirc$-operator.

2.1 Modal System

Let Prop be a countable set of atomic propositions. The set of formulas of the language of Dyadic Deontic Logic is constructed inductively as follows: [26]

$$F := P_i \mid \neg F \mid F \rightarrow F \mid \Box F \mid \bigcirc (F/F)$$

such that $P_i \in \mathsf{Prop}$, $\Box F$ is read as "F is settled true" and $\bigcirc(F/G)$ as "F is obligatory, given G". $P(F/G)$ is a short form for $\neg \bigcirc (\neg F/G)$, $\Diamond F$ is a short form of $\neg\Box\neg F$, and $\bigcirc F$ is an abbreviation for $\bigcirc(F/\top)$ which is read as "F is unconditionally obligatory". Formulas with iterated modalities, such as $\bigcirc(p/(\bigcirc(p/q) \wedge q))$, are well-formed formulas. System E with the two operators $\Box$ and $\bigcirc$ is axiomatized as follows:

Axioms of classical propositional logic	CL
$\mathsf{S5}$-scheme axioms for $\Box$	$\mathsf{S5}$
$\bigcirc(B/A) \rightarrow \Box \bigcirc (B/A)$	(Abs)
$\Box A \rightarrow \bigcirc(A/B)$	(Nec)
$\Box(A \leftrightarrow B) \rightarrow (\bigcirc(C/A) \leftrightarrow \bigcirc(C/B))$	(Ext)
$\bigcirc(A/A)$	(Id)
$\bigcirc(C/A \wedge B) \rightarrow \bigcirc(B \rightarrow C/A)$	(Sh)
$\bigcirc(B \rightarrow C/A) \rightarrow (\bigcirc(B/A) \rightarrow \bigcirc(C/A))$	(COK)

$$\frac{A \quad A \rightarrow B}{B} \text{ (MP)} \qquad\qquad \frac{A}{\Box A} \text{ (Neccesitation)}$$

As we see, these axioms can be categorized as follows:

– The axioms containing one operator $\Box$. These are axiom schemas of $\mathsf{S5}$, namely K, T, and 5.

$\Box(A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B)$	(K)
$\Box A \rightarrow A$	(T)
$\Diamond A \rightarrow \Box\Diamond A$	(5)

– The axioms containing one operator $\bigcirc$. (COK) is a deontic version of the K-axiom, (Id) is the principle of identity, and (Sh), named after Shoham, is a deontic analogue of the deduction theorem.
– Finally, the axioms containing two operators $\Box$ and $\bigcirc$. (Abs), which is Lewis' principle of absoluteness, shows that the betterness relation is not world-relative. (Nec) is a deontic version of necessitation. (Ext), extensionality, makes it possible to replace necessarily equivalent sentences in the antecedent of deontic conditionals.

The following principles are derived in system E:

if $A \leftrightarrow B$ then $\bigcirc(C/A) \leftrightarrow \bigcirc(C/B)$	(LLE)
if $A \rightarrow B$ then $\bigcirc(A/C) \rightarrow \bigcirc(B/C)$	(RW)
$\bigcirc(B/A) \wedge \bigcirc(C/A) \rightarrow \bigcirc(B \wedge C/A)$	(AND)
$\bigcirc(C/A) \wedge \bigcirc(C/B) \rightarrow \bigcirc(C/A \vee B)$	(OR)
$\bigcirc(C/A) \wedge \bigcirc(D/B) \rightarrow \bigcirc(C \vee D/A \vee B)$	(OR')

2.2 Preference Models

Now we review the preference model semantics for system E as follows:

Definition 1 (Preference model). *A preference model is a tuple*

$$\mathcal{M} = (W, \preceq, V),$$

where:

- *W is a non-empty set of worlds;*
- *$\preceq$ is a binary relation on W, called betterness relation, which orders the set of worlds according to their relative goodness. So for $w, v \in W$ we read $w \preceq v$ as "state v is at least as good as state w";*
- *V is a valuation function assigning a set $V(p) \subseteq W$ to each atomic formula p.*

Definition 2 (Truth under preference model). *Given a preference model $\mathcal{M} = (W, \preceq, V)$, for $w, v \in W$ and $A, B \in \mathsf{Fm}$, the truth for formulas under $\mathcal{M}$ is defined as follows:*

- *for propositional formulas is in standard way;*
- *$\mathcal{M}, w \Vdash \Box A$ iff, for all $v \in W$, $\mathcal{M}, v \Vdash A$;*
- *$\mathcal{M}, w \Vdash \bigcirc(A/B)$ iff $\mathrm{best}\|B\| \subseteq \|A\|$;*

where $\|A\|$ is truth set of A, i.e., the set of all worlds in which A is true. $\mathrm{best}\|B\|$ *is the subset of $\|B\|$ which is best according to $\preceq$.*

2.3 Justification Version of System E

Now we present the explicit version of E denoted by JE. We first define the set of terms and formulas as follows.

Definition 3. *The set of* proof terms*, shown by* PTm*, and* justification terms*, shown by* JTm*, are defined as follows:*

$$\lambda ::= \alpha_i \mid \xi_i \mid \Delta t \mid (\lambda + \lambda) \mid (\lambda \cdot \lambda) \mid\ !\lambda \mid\ ?\lambda$$

$$t ::= \mathsf{i} \mid x_i \mid t \cdot t \mid \nabla t \mid \mathsf{e}(t, \lambda) \mid \mathsf{n}(\lambda)$$

where α_i are proof constants, ξ_i are proof variables, i *is a justification constant and x_i are justification variables.*

Formulas are inductively defined as follows:

$$F := P_i \mid \neg F \mid (F \rightarrow F) \mid \lambda : F \mid [t](F/F),$$

where $P_i \in \mathsf{Prop}$, $\lambda \in \mathsf{PTm}$, and $t \in \mathsf{JTm}$. $[t]F$ is an abbreviation for $[t](F/\top)$. We use Fm for the set of formulas.

Definition 4 (Axiom Schemas of JE).

Axioms of Classical Propositional Logic	CL
$\lambda : (F \to G) \to (\kappa : F \to \lambda \cdot \kappa : G)$	j
$(\lambda : F \vee \kappa : F) \to (\lambda + \kappa) : F$	j+
$\lambda : F \to F$	jt
$\lambda : F \to !\lambda : \lambda : F$	j4
$\neg\lambda : A \to ?\lambda : (\neg\lambda : A)$	j5
$[t](B/A) \to \Delta t : [t](B/A)$	(Abs)
$\lambda : B \to [\mathsf{n}(\lambda)](B/A)$	(Nec)
$\lambda : (A \leftrightarrow B) \to ([t](C/A) \to [\mathsf{e}(t,\lambda)](C/B))$	(Ext)
$[\mathsf{i}](A/A)$	(Id)
$[t](C/A \wedge B) \to [\nabla t](B \to C/A)$	(Sh)
$[t](B \to C/A) \to ([s](B/A) \to [t \cdot s](C/A))$	(COK)

Definition 5 (Constant Specification). *A* constant specification CS *is any subset:*

$$\mathsf{CS} \subseteq \{(\alpha, A) \mid \alpha \textit{ is a proof constant and } A \textit{ is an axiom of } \mathsf{JE}\}.$$

A constant specification CS *is called* axiomatically appropriate *if for each axiom A of* JE*, there is a constant α with $(\alpha, A) \in \mathsf{CS}$.*

Definition 6 (System $\mathsf{JE}_{\mathsf{CS}}$). *For a constant specification* CS*, the system* $\mathsf{JE}_{\mathsf{CS}}$ *is defined by a Hilbert-style system with the axioms of* JE *and the following inference rules:*

$$\frac{A \quad A \to B}{B}\ (MP) \qquad \frac{}{\alpha : A}\ AN_{\mathsf{CS}} \textit{ where } (\alpha : A) \in \mathsf{CS}$$

As usual in justification logic [1,4,19], $\mathsf{JE}_{\mathsf{CS}}$ internalizes its own notion of proof.

Lemma 1 (Internalization). *Let* CS *be an axiomatically appropriate constant specification. For any formula A with $\mathsf{JE}_{\mathsf{CS}} \vdash A$, there exists a proof term λ such that $\mathsf{JE}_{\mathsf{CS}} \vdash \lambda : A$.*

To have a better understanding of the axiomatic system of JE, we provide Hilbert-style proofs of some typical formulas in the following examples. It is notable how terms are constructed as a justification for obligations.

Example 1. The explicit version of

$$\text{if } A \to B \text{ then } \bigcirc(A/C) \to \bigcirc(B/C) \qquad \text{(RW)}$$

is derivable in $\mathsf{JE}_{\mathsf{CS}}$ as follows for an axiomatically appropriate CS and a suitable term λ:

$A \to B$	
$\lambda : (A \to B)$	(Internalization)
$[\mathsf{n}(\lambda)](A \to B/C)$	(Nec)
$[s](A/C) \to [\mathsf{n}(\lambda) \cdot s](B/C)$	(COK)

Example 2. The explicit version of

$$\bigcirc(B/A) \wedge \bigcirc(C/A) \rightarrow \bigcirc(B \wedge C/A) \qquad \text{(AND)}$$

is derivable in $\mathsf{JE_{CS}}$ as follows for an axiomatically appropriate CS and a suitable term λ:

$[t](B/A) \wedge [s](C/A)$	
$B \rightarrow (C \rightarrow B \wedge C)$	(Tautology)
$[t](B/A) \rightarrow [\mathsf{n}(\lambda) \cdot t](C \rightarrow B \wedge C/A)$	(RW)
$[\mathsf{n}(\lambda) \cdot t](C \rightarrow B \wedge C/A)$	(MP)
$[s](C/A) \rightarrow [\mathsf{n}(\lambda) \cdot t \cdot s](B \wedge C/A)$	(COK)
$[\mathsf{n}(\lambda) \cdot t \cdot s](B \wedge C/A)$	(MP)

Example 3. The explicit version of

$$(\bigcirc(A/B) \wedge \Box B) \rightarrow \bigcirc A \qquad \text{(SFD)}$$

strong factual detachment is derivable in $\mathsf{JE_{CS}}$ as follows for an axiomatically appropriate CS and a suitable term γ:

$[t](A/B) \wedge \lambda : B$	
$\gamma : ((B \wedge \top) \leftrightarrow B)$	Tautology and internalization
$[t](A/B) \rightarrow [\mathsf{e}(t, \gamma)](A/B \wedge \top)$	(Ext)
$[\mathsf{e}(t, \gamma)](A/B \wedge \top)$	(MP)
$[\nabla \mathsf{e}(t, \gamma)](B \rightarrow A/\top)$	(Sh)
$[\mathsf{n}(\lambda)](B/\top)$	(Nec)
$[\nabla \mathsf{e}(t, \gamma) \cdot \mathsf{n}(\lambda)](A/\top)$	(COK)

3 Semantics

We first consider the following operations on the sets of formulas and sets of pairs of formulas in order to define basic evaluations.

Definition 7. *Let X, Y be sets of formulas, U, V be sets of pairs of formulas, and λ be a proof term. We define the following operations:*

$\lambda : X := \{\lambda : F \mid F \in X\}$;
$X \cdot Y := \{F \mid G \rightarrow F \in X \text{ for some } G \in Y\}$;
$U \ominus V := \{(F, G) \mid (H \rightarrow F, G) \in U \text{ for some } (H, G) \in V\}$;
$X \odot V := \{(F, G) \mid (G \leftrightarrow H) \in X \text{ for some } (F, H) \in V\}$;
$\mathsf{n}(X) := \{(F, G) \mid F \in X, G \in \mathsf{Fm}\}$;
$\nabla U := \{(F \rightarrow G, H) \mid (G, (H \wedge F)) \in U\}$.

Definition 8 (Basic Evaluation). *A* basic evaluation *for* $\mathsf{JE_{CS}}$ *is a function* ε *that*

– *maps atomic propositions to 0 and 1:*

$$\varepsilon(P_i) \in \{0,1\}, \quad \textit{for } P_i \in \mathsf{Prop}$$

– *maps proof terms to sets of formulas:*

$$\varepsilon(\lambda) \in \mathcal{P}(\mathsf{Fm}) \textit{ for } \lambda \in \mathsf{PTm}$$

such that for arbitrary $\lambda, \kappa \in \mathsf{PTm}$*:*
(i) $\varepsilon(\lambda) \cdot \varepsilon(\kappa) \subseteq \varepsilon(\lambda \cdot \kappa)$
(ii) $\varepsilon(\lambda) \cup \varepsilon(\kappa) \subseteq \varepsilon(\lambda + \kappa)$
(iii) $F \in \varepsilon(\alpha)$ *if* $(\alpha, F) \in \mathsf{CS}$
(iv) $\lambda : \varepsilon(\lambda) \subseteq \varepsilon(!\lambda)$
(v) $F \notin \varepsilon(\lambda)$ *implies* $\neg\lambda : F \in \varepsilon(?\lambda)$

– *maps justification terms to sets of pairs of formulas:*

$$\varepsilon(t) := \{(A, B) \mid A, B \in \mathsf{Fm}\}, \textit{ for } t \in \mathsf{JTm}$$

such that for any proof term λ *and justification terms* t, s*:*
1. $\varepsilon(t) \ominus \varepsilon(s) \subseteq \varepsilon(t \cdot s)$
2. $\varepsilon(\lambda) \odot \varepsilon(t) \subseteq \varepsilon(\mathsf{e}(t, \lambda))$
3. $\mathsf{n}(\varepsilon(\lambda)) \subseteq \varepsilon(\mathsf{n}(\lambda))$
4. $\nabla\varepsilon(t) \subseteq \varepsilon(\nabla t)$
5. $\varepsilon(\Delta t) = \{[t](A/B) \mid (A, B) \in \varepsilon(t)\}$
6. $\varepsilon(\mathsf{i}) = \{(A, A) \mid A \in \mathsf{Fm}\}$.

Definition 9 (Truth Under a Basic Evaluation). *We define truth of a formula* F *under a basic evaluation* ε *inductively as follows:*

1. $\varepsilon \Vdash P$ *iff* $\varepsilon(P) = 1$ *for* $P \in \mathsf{Prop}$*;*
2. $\varepsilon \Vdash F \to G$ *iff* $\varepsilon \nVdash F$ *or* $\varepsilon \Vdash G$*;*
3. $\varepsilon \Vdash \neg F$ *iff* $\varepsilon \nVdash F$*;*
4. $\varepsilon \Vdash \lambda : F$ *iff* $F \in \varepsilon(\lambda)$*;*
5. $\varepsilon \Vdash [t](F/G)$ *iff* $(F, G) \in \varepsilon(t)$*.*

Definition 10 (Factive Basic Evaluation). *A basic evaluation* ε *is called* factive *if for any formula* $\lambda : F$ *we have* $\varepsilon \Vdash \lambda : F$ *implies* $\varepsilon \Vdash F$*.*

Definition 11 (Basic Model). *Given an arbitrary* CS*, a* basic model *for* $\mathsf{JE}_{\mathsf{CS}}$ *is a basic evaluation that is factive.*

The following theorem gives us the expected soundness and completeness with respect to basic models which is proved in Appendix A.

Theorem 1 (Soundness and Completeness w.r.t. Basic Models). *Let* CS *be an arbitrary constant specification. System* $\mathsf{JE}_{\mathsf{CS}}$ *is sound and complete with respect to the class of all basic models. For any formula* F*,*

$$\mathsf{JE}_{\mathsf{CS}} \vdash F \quad \textit{iff} \quad \varepsilon \Vdash F \textit{ for all basic models } \varepsilon \textit{ for } \mathsf{JE}_{\mathsf{CS}}.$$

4 Preference Models

In this section, we introduce preference models for $\mathsf{JE_{CS}}$, which feature a set of possible worlds together with a *betterness* or *comparative goodness* relation on them.

Definition 12 (Quasi-model). *A quasi-model for* $\mathsf{JE_{CS}}$ *is a triple*

$$\mathcal{M} = \langle W, \preceq, \varepsilon \rangle$$

where:

- W *is a non-empty set of worlds;*
- $\preceq \subseteq W \times W$ *is a binary relation on the set of worlds where* $w_1 \preceq w_2$ *is read as* world w_2 is at least as good as world w_1.
- ε *is an evaluation function that asigns a basic evaluation* ε_w *to each world* w.

Definition 13 (Truth in Quasi-model). *Let* $\mathcal{M} = \langle W, \preceq, \varepsilon \rangle$ *be a quasi-model.* Truth of a formula at a world w *in a quasi-model is defined inductively as follows:*

1. $\mathcal{M}, w \Vdash P$ *iff* $\varepsilon_w(P) = 1$, *for* $P \in \mathsf{Prop}$
2. $\mathcal{M}, w \Vdash F \rightarrow G$ *iff* $\mathcal{M}, w \nVdash F$ *or* $\mathcal{M}, w \Vdash G$
3. $\mathcal{M}, w \Vdash \neg F$ *iff* $\mathcal{M}, w \nVdash F$
4. $\mathcal{M}, w \Vdash \lambda : F$ *iff* $F \in \varepsilon_w(\lambda)$
5. $\mathcal{M}, w \Vdash [t](F/G)$ *iff* $(F, G) \in \varepsilon_w(t)$.

We will write $\mathcal{M} \Vdash F$ *if* $\mathcal{M}, w \Vdash F$ *for all* $w \in W$.

Remark 1. As usual for quasi-models for justification logic [3,19,21], truth is local, i.e., for a quasi-model $\mathcal{M} = \langle W, \preceq, \varepsilon \rangle$ and $w \in W$, we have for any $F \in \mathsf{Fm}$:

$$\mathcal{M}, w \Vdash F \text{ iff } \varepsilon_w \Vdash F.$$

Remark 2. Let $\mathcal{M} = \langle W, \preceq, \varepsilon \rangle$ be a quasi-model. The *truth set* of $F \in \mathsf{Fm}$ is the set of all worlds in which F is true (denoted by $\|F\|^{\mathcal{M}}$),

$$\|F\|^{\mathcal{M}} := \{w \in W \mid \mathcal{M}, w \Vdash F\}.$$

Moreover, the best worlds in which F is true, according to $\preceq$, are called *best* F*-worlds* and are denoted by $\text{best}_{\preceq}\|F\|^{\mathcal{M}}$. For simplicity we often write $\|F\|$ for $\|F\|^{\mathcal{M}}$ and $\text{best}\|F\|$ for $\text{best}_{\preceq}\|F\|^{\mathcal{M}}$ when the model is clear from the context.

Remark 3 (Two Notions of "Best"). There are two ways to formalize the notion of "best world" respecting optimality and maximality [27]:

- $\text{best}\|A\|$ under "opt rule":

$$opt_{\preceq}(\|A\|) = \{w \in \|A\|^{\mathcal{M}} \mid \forall v(\mathcal{M}, v \Vdash A \rightarrow v \preceq w)\}$$

– best$\|A\|$ under "max rule":

$$max_{\preceq}(\|A\|) = \{w \in \|A\|^{\mathcal{M}} \mid \forall v((\mathcal{M}, v \Vdash A \wedge w \preceq v) \rightarrow v \preceq w)\}$$

Definition 14 (Preference Model). *A* preference model *is a quasi-model where ε_w is factive and satisfies the following condition:*

$$\textit{for any } t \in \mathsf{JTm} \textit{ and } w \in W,$$

$$(A, B) \in \varepsilon_w(t) \textit{ implies } \text{best}\|B\| \subseteq \|A\| \qquad \text{(JYB)}$$

in other words, all best B-worlds are A-worlds. This condition is called justification yields belief.

Definition 15 (Properties of $\preceq$). *We can require additional properties for the relation $\preceq$ such as:*

- *reflexivity: for all $w \in W, w \preceq w$*
- *totalness: for all $w, v \in W, w \preceq v$ or $v \preceq w$*
- *limitedness: if $\|A\| \neq \emptyset$ then* best$\|A\| \neq \emptyset$.

Limitedness avoids the case of not having a best state, i.e., of having infinitely many strictly better states. Morover, totalness yields reflexivity.

Lemma 2. *$max_{\preceq}(\|A\|) = opt_{\preceq}(\|A\|)$ if $\preceq$ is total.*

Proof. If $\preceq$ is total, then clearly from the definition $opt_{\preceq}(\|A\|) \subseteq max_{\preceq}(\|A\|)$. For the converse inclusion, suppose $w \in max_{\preceq}(\|A\|)$. By totalness, for any $v \in W$ with $\mathcal{M}, v \Vdash A$, either $v \preceq w$ or $w \preceq v$. In first case $w \in opt_{\preceq}(\|A\|)$ and in latter case, by definition of $max_{\preceq}$, $v \preceq w$ and $w \in opt_{\preceq}(\|A\|)$.

4.1 Soundness and Completeness w.r.t. Preference Models

Theorem 2. *System $\mathsf{JE}_{\mathsf{CS}}$ is sound and complete with respect to the class of all preference models under opt rule.*

Proof. To prove soundness, suppose $\mathcal{M} = \langle W, \preceq, \varepsilon \rangle$ is a preference model and $\mathsf{JE} \vdash A$. We show that A is true in every world $w \in W$. By soundness of JE with respect to basic models, we get $\varepsilon_w \Vdash A$ for all ε_w and by locality of truth in quasi-models, we conclude $\mathcal{M}, w \Vdash A$.

To prove completeness, suppose that $\mathsf{JE} \nvdash A$. By completeness of JE with respect to basic models, there is a basic model ε such that $\varepsilon \nVdash A$. Now construct a preference model $\mathcal{M} := \langle \{w_1\}, \preceq, \varepsilon' \rangle$ with $\varepsilon'_{w_1} := \varepsilon$ and $\preceq := \emptyset$. Then by locality of truth, we have $\mathcal{M}, w_1 \nVdash A$. It is easy to see that $\mathcal{M}$ is a preference model, i.e., to show (JYB). For any $t \in \mathsf{Tm}$ if $(B, C) \in \varepsilon(t)$, we have best$\|C\| \subseteq \|B\|$ since best$\|C\| = \emptyset$.

Remark 4. The above proof does not give us completeness under the max rule. The problem is that for the max rule, we cannot define the relation $\preceq$ such that best$\|C\| = \emptyset$.

However, by proving the following theorem we get desired results analogous to result in [28].

Theorem 3. *For every preference model $\mathcal{M} = \langle W, \preceq, \varepsilon\rangle$ under opt rule, there is an equivalent preference model $\mathcal{M}' = \langle W', \preceq', \varepsilon'\rangle$, such that $\preceq'$ is total (and hence reflexive).*

Proof. Let $\mathcal{M} = \langle W, \preceq, \varepsilon\rangle$. We define $\mathcal{M}' = \langle W', \preceq', \varepsilon'\rangle$ as follows:

- $W' = \{\langle w, n\rangle \mid w \in W, n \in \omega\}$;
- $\langle w, n\rangle \preceq' \langle v, m\rangle$ iff $w \preceq v$ or $n \leq m$;
- $\varepsilon'(p) = \{\langle w, n\rangle \mid w \in \varepsilon(p)\}$, for $p \in \mathsf{Prop}$;
- $\varepsilon'_{\langle w,n\rangle}(\lambda) = \varepsilon_w(\lambda)$;
- $\varepsilon'_{\langle w,n\rangle}(t) = \varepsilon_w(t)$;

where ω is the set of natural numbers. One can easily see that $\preceq'$ is total, since for any $\langle w, n\rangle$ and $\langle v, m\rangle$ in W', we have either $\langle w, n\rangle \preceq' \langle v, m\rangle$ or $\langle v, m\rangle \preceq' \langle w, n\rangle$, by totality of $\leq$ on the set of natural numbers. By locality of truth, for any formula $F \in \mathsf{Fm}$, we have $\mathcal{M}, w \Vdash F$ iff $\mathcal{M}', \langle w, n\rangle \Vdash F$ for all $n \in \omega$.

In order to show (JYB) in $\mathcal{M}'$, suppose $\mathcal{M}', \langle w, n\rangle \Vdash [t](A/B)$. By definition of $\mathcal{M}'$ we get $(A, B) \in \varepsilon'_{\langle w,n\rangle}(t)$ and so $(A, B) \in \varepsilon_w(t)$.

By applying (JYB) in $\mathcal{M}$, we get $\text{best}\|B\|^{\mathcal{M}} \subseteq \|A\|^{\mathcal{M}}$. We need to show that $\text{best}\|B\|^{\mathcal{M}'} \subseteq \|A\|^{\mathcal{M}'}$. Suppose $\langle v, k\rangle \in \text{best}\|B\|^{\mathcal{M}'}$, which means $\mathcal{M}', \langle v, k\rangle \Vdash B$. Then by definition of $\mathcal{M}'$ we have $\mathcal{M}, v \Vdash B$. We will show that $v \in \text{best}\|B\|^{\mathcal{M}}$. Suppose towards contradiction that $v \notin \text{best}\|B\|^{\mathcal{M}}$. Based on this, there is a world $u \in W$ such that $u \not\preceq v$ and $\mathcal{M}, u \Vdash B$. From this we get $\langle u, k\rangle \in W'$ and $\langle u, k+1\rangle \in W'$ as well. By definition of $\mathcal{M}'$ we have $\mathcal{M}', \langle u, k+1\rangle \Vdash B$, where $\langle v, k\rangle \preceq' \langle u, k+1\rangle$. This is a contradiction with the assumption that $\langle v, k\rangle \in \text{best}\|B\|^{\mathcal{M}'}$. As a result $v \in \text{best}\|B\|^{\mathcal{M}}$ and by (JYB) in $\mathcal{M}$ we get $v \in \|A\|^{\mathcal{M}}$, which means $\mathcal{M}, v \Vdash A$. As a result $\mathcal{M}', \langle v, k\rangle \Vdash A$, which means $\langle v, k\rangle \in \|A\|^{\mathcal{M}'}$.

We conclude that the following strengthening of Theorem 2 holds.

Corollary 1. *System* $\mathsf{JE}_{\mathsf{CS}}$ *is sound and complete with respect to preference models with a total betterness relation.*

By Lemma 2 this implies completeness of $\mathsf{JE}_{\mathsf{CS}}$ with respect to preference models under max rule.

Corollary 2. *System* $\mathsf{JE}_{\mathsf{CS}}$ *is sound and complete with respect to preference models under max rule.*

5 Conclusion and Future Work

Having explicit counterparts of modalities is valuable not only in epistemic but also in deontic contexts, where justification terms can be interpreted as reasons for obligations. Explicit non-normal modal logic [30] avoids the usual deontic paradoxes at the cost of being very (too) weak with respect to deductive power [31]. In the present paper, we introduced an explicit version $\mathsf{JE_{CS}}$ of the alethic-deontic system E, which features dyadic modalities to capture deontic conditionals. Semantics for E is given in terms of preference models, where the set of worlds is ordered according to a betterness relation. The language of $\mathsf{JE_{CS}}$ includes proof terms for the alethic modality and justification terms for the deontic modality.

We established soundness and completeness of $\mathsf{JE_{CS}}$ with respect to basic models and preference models. In preference models, the property "justification yields belief" (JYB) holds, which means justified formulas act like obligatory formulas.

The converse direction, however, only holds in fully explanatory models. A preference model is *fully explanatory* if the converse of (JYB) holds, that is for any world w and any formulas A, B:

$$\text{best}\|B\| \subseteq \|A\| \text{ implies } (A, B) \in \varepsilon_w(t) \text{ for some } t \in \mathsf{JTm}.$$

To prove the completeness for $\mathsf{JE_{CS}}$ with respect to fully explanatory preference models, one would have to follow the strategy of the completeness proof for the modal system E [28]. That is define so-called selection function models for $\mathsf{JE_{CS}}$, establish completeness with respect to the selection function models, and show that for each selection function model, there is an equivalent preference model.

Another line of future work is to study justification logic for preference models where the betterness relation satisfies the limitedness condition. The modal axiom that corresponds to this is $\Diamond A \rightarrow (\bigcirc(B/A) \rightarrow \mathsf{P}(B/A))$, where $\Diamond A$ and $\mathsf{P}(B/A))$ stand for $\neg\Box\neg A$ and $\neg \bigcirc (\neg B/A)$, respectively. The problem of finding a justification logic version for this axiom is that terms in justification logic usually stand for $\Box$-type modalities. A notable exception is the work on justified constructive modal logic [20].

A Soundness and Completeness with Respect to Basic Models

Theorem 4. *System* $\mathsf{JE_{CS}}$ *is sound with respect to the class of all basic models.*

Proof. The proof is by induction on the length of derivations in $\mathsf{JE_{CS}}$. For an arbitrary basic model ε, soundness of the propositional axioms is trivial and soundness of $\mathsf{S5}$ axioms $\mathsf{j}, \mathsf{jt}, \mathsf{j4}, \mathsf{j5}, \mathsf{j+}$ immediately follows from the definition of basic evaluation and factivity. We just check the cases for the axioms containing justification terms. Suppose $\mathsf{JE_{CS}} \vdash F$ and F is an instance of:

- (COK): Suppose $\varepsilon \Vdash [t](B \rightarrow C/A)$ and $\varepsilon \Vdash [s](B/A)$. Thus we have

$$(B \rightarrow C, A) \in \varepsilon(t) \quad \text{and} \quad (B, A) \in \varepsilon(s).$$

By the definition of basic model, we have $\varepsilon(t) \ominus \varepsilon(s) \subseteq \varepsilon(t \cdot s)$ and as a result $(C, A) \in \varepsilon(t \cdot s)$, which means $\varepsilon \Vdash [t \cdot s](C/A)$.

- (Nec): Suppose $\varepsilon \Vdash (\lambda : A)$. Thus $A \in \varepsilon(\lambda)$. By the definition of $\mathsf{n}(\varepsilon(\lambda))$ we have $(A, B) \in \mathsf{n}(\varepsilon(\lambda))$ for any $B \in \mathsf{Fm}$ and by the definition of basic evaluation $\mathsf{n}(\varepsilon(\lambda)) \subseteq \varepsilon(\mathsf{n}(\lambda))$, so $(A, B) \in \varepsilon(\mathsf{n}(\lambda))$, which means $\varepsilon \Vdash [\mathsf{n}(\lambda)](A/B)$.
- (Ext): Suppose $\varepsilon \Vdash \lambda : (A \leftrightarrow B)$, so $(A \leftrightarrow B) \in \varepsilon(\lambda)$. Since $\varepsilon(\lambda) \odot \varepsilon(t) \subseteq \varepsilon(\mathsf{e}(t, \lambda))$, we have $(C, B) \in \varepsilon(\mathsf{e}(t, \lambda))$ if $(C, A) \in \varepsilon(t)$. Hence $\varepsilon \Vdash ([t](C/A) \rightarrow [\mathsf{e}(t, \lambda)](C/B))$.
- (Sh): Suppose $\varepsilon \Vdash [t](C/A \wedge B)$, then $(C, (A \wedge B)) \in \varepsilon(t)$. By definition of $\nabla(\varepsilon(t))$ we have $(B \rightarrow C, A) \in \nabla(\varepsilon(t))$ and by definition of basic models, $\nabla\varepsilon(t) \subseteq \varepsilon(\nabla t)$. As a result, $((B \rightarrow C), A) \in \varepsilon(\nabla t)$ which means $\varepsilon \Vdash [\nabla t](B \rightarrow C/A)$.

For the axioms (Abs) and (Id) soundness is immediate from the definition of basic evaluation. □

Theorem 5. *System* $\mathsf{JE}_{\mathsf{CS}}$ *is complete with respect to the class of all basic models.*

Proof. Given a maximal consistent Γ, we define the canonical model ε^c induced by Γ as follows:

- $\varepsilon^c_\Gamma(P) := 1$, if $P \in \Gamma$ and $\varepsilon^c := 0$, if $P \notin \Gamma$;
- $\varepsilon^c_\Gamma(\lambda) := \{F \mid \lambda : F \in \Gamma\}$;
- $\varepsilon^c_\Gamma(t) := \{(F, G) \mid [t](F/G) \in \Gamma\}$.

We first show that ε^c is a basic evaluation. Conditions (i)–(v) follow immediately from the maximal consistency of Γ and axioms of $\mathsf{j} - \mathsf{j5}$. Conditions (1)–(6) are obtained from the axioms (Abs), (COK), (Nec), (Id), (Ext), and (Sh). Let us only show (1) and (3).

To check condition (1), suppose $(C, B) \in \varepsilon^c(t) \ominus \varepsilon^c(s)$. Then there is an $A \in \mathsf{Fm}$ such that $(A \rightarrow C, B) \in \varepsilon^c(t)$ and $(A, B) \in \varepsilon^c(s)$. By the definition of canonical model $[t](A \rightarrow C/B) \in \Gamma$ and $[s](A/B) \in \Gamma$, by maximal consistency of Γ and axiom (COK) we have $[t \cdot s](C/B) \in \Gamma$, which gives $(C/B) \in \varepsilon^c(t \cdot s)$.

To check condition (3), suppose $(A, B) \in \mathsf{n}(\varepsilon^c(\lambda))$. Then $A \in \varepsilon^c(\lambda)$, which means $\lambda : A \in \Gamma$. By maximal consistency of Γ and axiom (Nec) we get $[\mathsf{n}(\lambda)](A/B) \in \Gamma$. By the definition of canonical model we conclude $(A, B) \in \varepsilon^c(\mathsf{n}(\lambda))$. Thus ε^c is a basic evaluation.

The truth lemma states:

$$F \in \Gamma \text{ iff } \varepsilon^c \Vdash F,$$

which is established as usual by induction on the structure of F. In case $F = [t](A/B)$, we have $[t](A/B) \in \Gamma$ iff $(A, B) \in \varepsilon^c(t)$ iff $\varepsilon^c \Vdash [t](A/B)$.

Due to axiom jt, ε^c is factive by the following reasoning: if $\varepsilon^c \Vdash \lambda : F$, we get by the truth lemma that $\lambda : F \in \Gamma$. By the maximal consistency of Γ we have $F \in \Gamma$ which means $\varepsilon^c \Vdash F$ by the truth lemma. □

References

1. Artemov, S.: Explicit provability and constructive semantics. BSL **7**(1), 1–36 (2001)
2. Artemov, S.: Justified common knowledge. TCS **357**(1–3), 4–22 (2006). https://doi.org/10.1016/j.tcs.2006.03.009
3. Artemov, S.: The ontology of justifications in the logical setting. Stud. Log. **100**(1–2), 17–30 (2012). https://doi.org/10.1007/s11225-012-9387-x
4. Artemov, S., Fitting, M.: Justification Logic: Reasoning with Reasons. Cambridge Tracts in Mathematics, Cambridge University Press, Cambridge (2019). https://doi.org/10.1017/9781108348034
5. Bucheli, S., Kuznets, R., Studer, T.: Justifications for common knowledge. Appl. Non-Classical Log. **21**(1), 35–60 (2011). https://doi.org/10.3166/JANCL.21.35-60
6. Chisholm, R.M.: Contrary-to-duty imperatives and deontic logic. Analysis **24**(2), 33–36 (1963). https://doi.org/10.1093/analys/24.2.33
7. Danielsson, S.: Preference and Obligation. Studies in the Logic of Ethics. Filosofiska föreningen (1968)
8. Faroldi, F., Ghari, M., Lehmann, E., Studer, T.: Impossible and conflicting obligations in justification logic. In: Marra, A., Liu, F., Portner, P., Van De Putte, F. (eds.) Proceedings of DEON 2020 (2020)
9. Faroldi, F.L.G.: Deontic modals and hyperintensionality. Log. J. IGPL **27**, 387–410 (2019). https://doi.org/10.1093/jigpal/jzz011
10. Faroldi, F.L.G.: Hyperintensionality and Normativity. Springer, Heidelberg (2019). https://doi.org/10.1007/978-3-030-03487-0
11. Faroldi, F.L.G., Ghari, M., Lehmann, E., Studer, T.: Consistency and permission in deontic justification logic. J. Log. Comput. (2022). https://doi.org/10.1093/logcom/exac045
12. Faroldi, F.L.G., Protopopescu, T.: All-things-considered ought via reasons in justification logic (2018). Preprint
13. Faroldi, F.L.G., Protopopescu, T.: A hyperintensional logical framework for deontic reasons. Log. J. IGPL **27**, 411–433 (2019). https://doi.org/10.1093/jigpal/jzz013
14. Fitting, M.: The logic of proofs, semantically. APAL **132**(1), 1–25 (2005). https://doi.org/10.1016/j.apal.2004.04.009
15. van Fraassen, B.C.: The logic of conditional obligation. J. Philos. Log. **1**(3/4), 417–438 (1972)
16. Gabbay, D., Horty, J., Parent, X., van der Meyden, R., van der Torre, L.: Handbook of Deontic Logic and Normative System, vol. 2. College Publications (2021)
17. Hansson, B.: An analysis of some deontic logics. Noûs **3**(4), 373–398 (1969)
18. Kokkinis, I., Maksimović, P., Ognjanović, Z., Studer, T.: First steps towards probabilistic justification logic. Log. J. IGPL **23**(4), 662–687 (2015). https://doi.org/10.1093/jigpal/jzv025
19. Kuznets, R., Studer, T.: Logics of Proofs and Justifications. Studies in Logic. College Publications (2019)
20. Kuznets, R., Marin, S., Straßburger, L.: Justification logic for constructive modal logic. J. Appl. Log. **8**, 2313–2332 (2021)
21. Kuznets, R., Studer, T.: Justifications, ontology, and conservativity. In: Bolander, T., Braüner, T., Ghilardi, S., Moss, L. (eds.) Advances in Modal Logic, vol. 9, pp. 437–458. College Publications (2012)
22. Lehmann, E., Studer, T.: Subset models for justification logic. In: Iemhoff, R., Moortgat, M., de Queiroz, R. (eds.) WoLLIC 2019. LNCS, vol. 11541, pp. 433–449. Springer, Heidelberg (2019). https://doi.org/10.1007/978-3-662-59533-6_26

23. Lewis, D.K.: Counterfactuals. Blackwell, Cambridge (1973)
24. Lewis, D.K.: Semantic analyses for dyadic deontic logic. In: Stenlund, S., Henschen-Dahlquist, A.M., Lindahl, L., Nordenfelt, L., Odelstad, J. (eds.) Logical Theory and Semantic Analysis. Synthese Library, vol. 63, pp. 1–14. Springer, Dordrecht (1974)
25. Parent, X., van der Torre, L.: Introduction to Deontic Logic and Normative Systems. Texts in Logic and Reasoning. College Publications (2018)
26. Parent, X.: A complete axiom set for Hansson's deontic logic DSDL2. Log. J. IGPL **18**(3), 422–429 (2010). https://doi.org/10.1093/jigpal/jzp050
27. Parent, X.: Maximality vs. optimality in dyadic deontic logic. J. Philos. Log. **43**(6), 1101–1128 (2013). https://doi.org/10.1007/s10992-013-9308-0
28. Parent, X.: Completeness of Åqvist's systems E and F. Rev. Symb. Log. **8**(1), 164–177 (2015). https://doi.org/10.1017/S1755020314000367
29. Renne, B.: Dynamic epistemic logic with justification. Ph.D. thesis, City University of New York (2008)
30. Rohani, A., Studer, T.: Explicit non-normal modal logic. In: Silva, A., Wassermann, R., de Queiroz, R. (eds.) WoLLIC 2021. LNCS, vol. 13038, pp. 64–81. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-88853-4_5
31. Rohani, A., Studer, T.: Explicit non-normal modal logic. J. Log. Comput. (in print)
32. Studer, T.: Decidability for some justification logics with negative introspection. JSL **78**(2), 388–402 (2013). https://doi.org/10.2178/jsl.7802030
33. Xu, C., Wang, Y., Studer, T.: A logic of knowing why. Synthese **198**, 1259–1285 (2021)

Structural Completeness and Superintuitionistic Inquisitive Logics

Thomas Ferguson and Vít Punčochář(✉)

Institute of Philosophy, Czech Academy of Sciences, Jilská 1, 11000 Prague, Czechia
tferguson@gradcenter.cuny.edu, puncochar@flu.cas.cz

Abstract. In this paper, the notion of structural completeness is explored in the context of a generalized class of superintuitionistic logics involving also systems that are not closed under uniform substitution. We just require that each logic must be closed under D-substitutions assigning to atomic formulas only $\vee$-free formulas. For these systems we introduce four different notions of structural completeness and study how they are related. We focus on superintuitionistic inquisitive logics that validate a schema called Split and have the disjunction property. In these logics disjunction can be interpreted in the sense of inquisitive semantics as a question forming operator. It is shown that a logic is structurally complete with respect to D-substitutions if and only if it includes the weakest superintuitionistic inquisitive logic. Various consequences of this result are explored. For example, it is shown that every superintuitionistic inquisitive logic can be characterized by a Kripke model built up from D-substitutions. Additionally, we resolve a conjecture concerning superintuitionistic inquisitive logics due to Miglioli *et al.*.

Keywords: Inquisitive logic · superintuitionistic logics · Structural completeness · Substitution · Kripke semantics

1 Introduction

The property of *structural completeness* of a logic Λ is satisfied when every rule admissible in Λ is also derivable in Λ. Since its introduction by Pogorzelski [25], the property has been studied in a number of contexts, *e.g.* in substructural logics in [24] or fuzzy logics in [8], but most exhaustively in the context of superintuitionistic (also known as intermediate) logics (e.g. in [9,10]).

Research on structural completeness has largely assumed the requirement of closure under uniform substitutions. A notable exception to this requirement is found in *inquisitive logic* [6]. Inquisitive logic makes up a framework in which declarative propositions (asserting information) are distinguished from inquisitive propositions (in which a question is posed). In order to preserve the distinction, uniform substitution must be relaxed.

As [28] documents, there is an interesting class of superintuitionistic inquisitive logics that inhabit the space of theories extending intuitionistic logic but

H. H. Hansen et al. (Eds.): WoLLIC 2023, LNCS 13923, pp. 194–210, 2023.
https://doi.org/10.1007/978-3-031-39784-4_12

are not closed under all substitutions. Both [22] and [20] touch on properties concerning structural completeness relevant to inquisitive logics. [22] considers variants of structural completeness over a wide range of theories in which substitutivity might fail, emphasizing two particular inquisitive logics, while [20] considers the inquisitive logic InqB among related systems of logics of dependency. While both studies have lessons for the structure of inquisitive logics, neither focuses on the whole class of these logics.

In what follows, we investigate structural completeness from the perspective of intuitionistic inquisitive logic and its extensions. Several interesting facts become clear from this standpoint, including the fact that all superintuitionistic inquisitive logics enjoy a property of hereditary structural completeness and that hereditary structural completeness coincides with structural completeness *simpliciter* in the space of extensions of intuitionistic inquisitive logic. We apply these results to expose a relationship with extensions of Gödel-Dummett logic LC and to explore the features of Kripke models for inquisitive logics built from substitutions. Along the way, we resolve a two-part conjecture concerning inquisitive logics due to Miglioli *et al.* made in [22].

2 Intuitionistic Inquisitive Logic and Its Extensions

The standard inquisitive logic [2–4,6] is based on an "information-based" semantics for classical logic that allows us to add to the language and characterize semantically some question forming operators, like inquisitive disjunction on the propositional level and an inquisitive existential quantifier on the first-order level. Since we will focus in this paper on the propositional level, we will be concerned only with inquisitive disjunction. This operator, when applied to statements S_1, S_2, forms the question *whether* S_1 *or* S_2, which can be contrasted with the statement *that* S_1 *or* S_2. Interestingly, this construction can be embedded under other operators, thus allowing one to form, for example, conjunctive questions and conditional questions. It is also possible to form disjunctive questions with more than two alternatives (*whether* S_1*,* S_2*, or* S_3, and so on), and polar (yes/no) questions as a special kind of disjunctive question (*whether* S_1 *or not* S_1).

Inquisitive disjunction has some constructive features and resembles intuitionistic disjunction, though, as we will see, it is not identical with it. Standard inquisitive logic can thus be viewed as classical logic extended with this constructive operator. It is possible to vary almost arbitrarily the background logic of declarative sentences, while keeping fixed the most characteristic features of inquisitive semantics. In this way we obtain non-classical inquisitive logics. A general semantic and syntactic theory of these logics is developed in [30]. An important example of such a logic is intuitionistic inquisitive logic [5,28,31]. It can be presented in two different ways. One can take the standard language of intuitionistic logic and add inquisitive disjunction to this language (as in [5,29]). Alternatively, one can work just with the standard language involving only one disjunction, which is however interpreted in the inquisitive way (as in [2,6,31]). In this paper we will take the latter approach, which has the advantage that the axiomatization of the corresponding logic is much simpler and more elegant, and it can be compared directly with other logical systems in the same language that have been already studied in the literature on superintuitionistic logics. But the main reason for the decision to work with only one disjunction is that our main results concerning structural completeness hold only for this restricted language and they cease to be valid if the second disjunction is added. So, in this paper we will work with this basic language:

$$\varphi ::= p \mid \bot \mid \varphi \to \varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi$$

where $\vee$ will be viewed as inquisitive disjunction. Negation, equivalence and a constant for validity are defined in the following usual way: $\neg\varphi =_{def} \varphi \to \bot$, $\varphi \leftrightarrow \psi =_{def} (\varphi \to \psi) \wedge (\psi \to \varphi)$, $\top =_{def} \bot \to \bot$.

Fix any Hilbert style axiomatization of intuitionistic logic in this language, which has modus ponens as the only rule of inference. Let IL denote the set of its derivable formulas. We write $\vdash_{\mathsf{IL}} \varphi$ instead of $\varphi \in \mathsf{IL}$, $\varphi \vdash_{\mathsf{IL}} \psi$ instead of $\varphi \to \psi \in \mathsf{IL}$, and $\varphi \equiv_{\mathsf{IL}} \psi$ instead of $\varphi \leftrightarrow \psi \in \mathsf{IL}$. (The same notation will be used also for other logics).

An axiomatization of intuitionistic inquisitive logic is obtained by extending the system for intuitionistic logic with the following schema called *Split*:

$$(\alpha \to (\psi \vee \chi)) \to ((\alpha \to \psi) \vee (\alpha \to \chi)),$$

where α ranges over $\vee$-free formulas.[1] The set of derivable formulas will be denoted as InqIL. The Split schema can be viewed as a piece that is missing in intuitionistic logic in order to prove inductively the following disjunctive form theorem, which is a cornerstone of propositional inquisitive logic that has been proved and used for many of its incarnations.

Theorem 1. *For any formula φ there are $\vee$-free formulas $\alpha_1, \ldots, \alpha_n$ such that*

$$\varphi \equiv_{\mathsf{InqIL}} \alpha_1 \vee \ldots \vee \alpha_n.$$

Proof. One can proceed by induction. We show just the inductive step for implication. Assume that $\psi \equiv_{\mathsf{InqIL}} \beta_1 \vee \ldots \vee \beta_m$ and $\chi \equiv_{\mathsf{InqIL}} \gamma_1 \vee \ldots \vee \gamma_n$. Let $I = \{1, \ldots, m\}$ and $J = \{1, \ldots, n\}$. Then (using Split in the equivalence between the second and the third line):

$$\begin{aligned} \psi \to \chi &\equiv_{\mathsf{InqIL}} \bigvee_{i \in I} \beta_i \to \bigvee_{j \in J} \gamma_j \\ &\equiv_{\mathsf{InqIL}} \bigwedge_{i \in I} (\beta_i \to \bigvee_{j \in J} \gamma_j) \\ &\equiv_{\mathsf{InqIL}} \bigwedge_{i \in I} \bigvee_{j \in J} (\beta_i \to \gamma_j) \\ &\equiv_{\mathsf{InqIL}} \bigvee_{f \in I \to J} \bigwedge_{i \in I} (\beta_i \to \gamma_{f(i)}). \end{aligned}$$

□

[1] For a formulation of intuitionistic inquisitive logic as a system of natural deduction, see [29].

The Split schema can be formulated in a stronger form, using the notion of a Harrop formula. A formula is called a *Harrop formula* if disjunction occurs in it only within antecedents of implications. So, the only allowed occurrences of disjunction are in the following context: $(\ldots((\ldots \vee \ldots) \to \ldots)\ldots)$. It can be observed that for every Harrop formula φ there is a $\vee$-free formula α such that $\varphi \equiv_{\mathsf{InqIL}} \alpha$. To see this, assume that φ contains a subformula $\psi \to \chi$. The antecedent ψ may possibly involve a disjunction but all such disjunctions can be eliminated, since, according to Theorem 1, there are $\vee$-free formulas $\beta_1, \ldots, \beta_n$ such that $\psi \equiv_{\mathsf{InqIL}} \beta_1 \vee \ldots \vee \beta_m$, and so $\psi \to \chi \equiv_{\mathsf{InqIL}} \bigvee_i \beta_i \to \chi \equiv_{\mathsf{InqIL}} \bigwedge_i(\beta_i \to \chi)$. As a consequence of this observation, InqIL can be equivalently axiomatized by extending intuitionistic logic with the schema that we can call *H-Split* and which is like Split except that α ranges over arbitrary Harrop formulas.

Nevertheless, the restriction on the antecedents in Split (or H-Split) is important and cannot be completely avoided. Not all substitutional instances of Split hold in InqIL. For example, $((p \vee q) \to (p \vee q)) \to (((p \vee q) \to p) \vee ((p \vee q) \to q)) \notin \mathsf{InqIL}$. In this respect, InqIL is an unusual logic because it is not closed under uniform substitution. This fact is however well-motivated, given that the logic deals with two different categories of propositions, statements and questions. Questions are generated only by the inquisitive operator, and so every $\vee$-free formula represents a statement. It is not surprising that questions behave differently, and in some contexts cannot be substituted for statements.

In this paper, we want to explore InqIL within the context of superintuitionistic logics. However, since superintuitionistic logics are required to be closed under uniform substitution, we need to introduce a more general notion, that will encompass also InqIL (and other inquisitive logics). For this purpose, we use the following definitions.

Definition 1. *A* substitution *is a function that assigns a formula to each atomic formula. An H-*substitution *is a substitution that assigns to each atomic formula a Harrop formula. A D-*substitution *is a substitution that assigns to each atomic formula a $\vee$-free formula.*[2] *If s is a substitution and φ is a formula then $s(\varphi)$ denotes the formula that is obtained from φ by replacing simultaneously every occurrence of each atomic formula p in φ with the formula $s(p)$.*

Definition 2. *A* gsi-logic *(generalized superintuitionistic logic) is any set of formulas which (a) contains all intuitionistically valid formulas; (b) does not contain $\bot$; (c) is closed under every D-substitution; (d) is closed under modus ponens.*[3] *A gsi-logic is* standard *if it is closed under every substitution.*

Note that the notion of a standard gsi-logic coincides with the standard notion of a superintuitionistic logic. Next we define, in accordance with [31], the notion of an inquisitive gsi-logic.

[2] Intuitively, given the inquisitive interpretation of disjunction, D-substitutions assign only declarative sentences to atomic formulas.

[3] In [31] these sets of formulas were called superintuitionistic logics*, using the star to indicate that the notion of a logic is used in a non-standard way, since closure under uniform substitution is not generally required.

Definition 3. *We say that a gsi-logic Λ is* inquisitive *if it (a) contains all instances of Split, and (b) has the disjunction property, i.e.* $\alpha \vee \beta \in \Lambda$ *only if* $\alpha \in \Lambda$ *or* $\beta \in \Lambda$. *Instead of saying that a gsi-logic Λ contains all instances of Split we will sometimes say that Split is valid in Λ.*

It is clear that InqIL is a gsi-logic, though not standard. Another example of a non-standard gsi-logic is the classical inquisitive logic, often called basic inquisitive logic and denoted as InqB (see [3]), which can be obtained by adding to InqIL the restricted double negation law DN: $\neg\neg\alpha \to \alpha$, where α ranges over $\vee$-free (or, equivalently, Harrop) formulas. Note that the $\vee$-free fragment of InqB is identical with classical logic, while it can be shown that the $\vee$-free fragment of InqIL is identical with (the $\vee$-free fragment of) intuitionistic logic.

It can also be shown that both InqIL and InqB have the disjunction property and thus are inquisitive also according to our general definition. In fact, InqIL is the weakest and InqB the strongest inquisitive gsi-logic. No inquisitive gsi-logic is standard. However, for any standard gsi-logic Λ there is exactly one inquisitive gsi-logic that conservatively extends the $\vee$-free fragment of Λ. As a consequence, there are uncountably many inquisitive gsi-logics. For a justification of these claims, see [28], where inquisitive gsi-logics are called G-logics.

To the best of our knowledge, InqIL and InqB were both studied for the first time in [22], under the names $\mathsf{F}_{\mathsf{int}}$ and F_{cl}, where it was proved, for instance, that the schematic fragment of InqB (called the standardization of F_{cl} in [22]), i.e. the set $S(\mathsf{InqB}) = \{\varphi \mid s(\varphi) \in \mathsf{InqB}, \text{ for every substitution } s\}$, is identical with Medvedev's logic of finite problems ML. Interestingly, the same was stated without proof also for InqIL. That this quite non-trivial claim is true follows from the main result of [17].

The system of InqB was later rediscovered in [2,6] and proved to be sound and complete with respect to the modern version of inquisitive semantics. This logic was applied to linguistic phenomena (see [4] for an overview) but also studied from algebraic and topological perspectives [1,13,34]. There is also an extensive literature on various modal extensions of InqB [7,14,15,27,32].

A generalization of inquisitive semantics that corresponds to InqIL was introduced in [28]. Intuitionistic inquisitive logic was further studied in [5,29,35], and from an algebraic perspective in [31,33]. A slightly different approach to intuitionistic inquisitive logic was proposed in [18]. This approach is based on a more general framework that does not validate Split.

A recent interesting result also shows that the system of InqIL provides a sound and complete axiomatization of proof-theoretic semantics [36]. This connection to proof-theoretic semantics nicely stresses the significance of this logic.

3 Structural Completeness

In this section, we show that the Split schema is quite intimately connected to the notion of structural completeness. Structural completeness is usually studied in relation to logics that are closed under arbitrary substitutions. For a more general notion of a logic we need a more flexible notion of structural completeness.

In particular, we can consider various kinds of structural completeness defined in terms of some restricted classes of substitutions. Such notions were studied in depth in [22] and they were employed also in [20] where it was shown that InqB and some related propositional dependence logics are structurally complete with respect to a suitably adapted sense of the term. In [22], the authors considered, besides other options, structural completeness defined in terms of H-substitutions and they called the corresponding notion *H-smoothness*. We will call it *SH-completeness*. Besides that we introduce three other notions of structural completeness.

Definition 4. *Let Λ be a gsi-logic. By $sub(\Lambda)$ we denote the set of all substitutions under which Λ is closed, i.e., $s \in sub(\Lambda)$ iff $s(\varphi) \in \Lambda$, for every $\varphi \in \Lambda$. We say that Λ is SF-*complete *(structurally fully complete) if it holds:*

$\varphi \vdash_\Lambda \psi$ *iff for any substitution s, if $\vdash_\Lambda s(\varphi)$ then $\vdash_\Lambda s(\psi)$.*

*We say that Λ is SG-*complete *(structurally generally complete) if it holds:*

$\varphi \vdash_\Lambda \psi$ *iff for any $s \in sub(\Lambda)$, if $\vdash_\Lambda s(\varphi)$ then $\vdash_\Lambda s(\psi)$.*

*We say that Λ is SH-*complete *(structurally Harrop complete) if it holds:*

$\varphi \vdash_\Lambda \psi$ *iff for any H-substitution s, if $\vdash_\Lambda s(\varphi)$ then $\vdash_\Lambda s(\psi)$.*

*We say that Λ is SD-*complete *(structurally declaratively complete) if it holds:*

$\varphi \vdash_\Lambda \psi$ *iff for any D-substitution s, if $\vdash_\Lambda s(\varphi)$ then $\vdash_\Lambda s(\psi)$.*

We say that Λ is hereditarily SF-complete *(SG-complete, SH-complete, SD-complete) if every gsi-logic $\Lambda' \supseteq \Lambda$ is SF-complete (SG-complete, SH-complete, SD-complete).*

Note that the notions of SF-completeness and SG-completeness both generalize, in a clear sense, the standard notion of structural completeness. For standard gsi-logics both these notions are equivalent and coincide with the usual notion that is restricted to this field. Even though the notion of SF-completeness preserves the literal formulation of the usual definition and just applies it to a broader context, we think that SG-completeness provides a more natural generalization of structural completeness. This should be clear from the following observation that indicates that the notion of SF-completeness is reasonably applicable only to standard gsi-logics. We can also immediately observe that SD-completeness is also stronger than SG-completeness.

Proposition 1. *Let Λ be a gsi-logic. Then (a) if Λ is SD-complete then it is SG-complete; (b) Λ is SF-complete if and only if it is standard and SG-complete.*

Proof. (a) Assume that Λ is SD-complete. Obviously, if $\varphi \vdash_\Lambda \psi$ then for any $s \in sub(\Lambda)$, if $\vdash_\Lambda s(\varphi)$ then $\vdash_\Lambda s(\psi)$. For the opposite direction, assume that for any $s \in sub(\Lambda)$, if $\vdash_\Lambda s(\varphi)$ then $\vdash_\Lambda s(\psi)$. Then also for any D-substitution s, if $\vdash_\Lambda s(\varphi)$ then $\vdash_\Lambda s(\psi)$, and thus $\varphi \vdash_\Lambda \psi$.

(b) Assume that Λ is SF-complete and $\vdash_\Lambda \varphi$. Then also $\vdash_\Lambda \top \to \varphi$. Take any substitution s. Since $\vdash_\Lambda s(\top)$, by SF-completeness we obtain $\vdash_\Lambda s(\varphi)$. We have shown that SF-complete gsi-logics are standard. The rest follows from the observation that for standard gsi-logics the notions of SF-completeness and SG-completeness coincide. □

Our aim in this section is to show that Split is closely related to SH-completeness and SD-completeness. Through this connection we will also be able to see that these two notions of structural completeness are in fact equivalent. First we can observe the following.

Proposition 2. *Every gsi-logic in which Split is valid is closed under all H-substitutions.*

Proof. Assume that Λ is a gsi-logic in which Split is valid and take any H-substitution s. We have already observed that Split guarantees that every Harrop formula is equivalent to a $\vee$-free formula. In particular, for any p there is a $\vee$-free formula α_p such that $s(p) \equiv_\Lambda \alpha_p$. Now we can define a D-substitution s^* fixing $s^*(p) = \alpha_p$, for each atomic p. If $\vdash_\Lambda \varphi$ then also $\vdash_\Lambda s^*(\varphi)$, since Λ is closed under D-substitutions. It follows that also $\vdash_\Lambda s(\varphi)$. □

Let Λ be a gsi-logic that validates Split. Proposition 2 says that if s is an H-substitution then $s \in sub(\Lambda)$. One can ask whether also the converse holds. If we formulate the converse directly, it is false for trivial reasons. If a substitution s assigns formulas that are not Harrop but are all equivalent to Harrop formulas then Λ must be closed also under s. A more interesting question is whether Λ may be closed also under substitutions that are not equivalent to any H-substitution. Let us formulate it more precisely. We say that two substitutions, s and t, are Λ-equivalent if $s(p) \equiv_\Lambda t(p)$, for every atomic formula p. Then the converse of Proposition 2 would state that if $s \in sub(\Lambda)$ then s is Λ-equivalent to an H-substitution. Interestingly, this is a property that distinguishes InqIL and InqB.

Proposition 3. *Every $s \in sub(\mathsf{InqB})$ is InqB-equivalent to an H-substitution (and even D-substitution). In contrast, there is an $s \in sub(\mathsf{InqIL})$ that is not InqIL-equivalent to any H-substitution.*

Proof. For the first part, let s be a substitution under which InqB is closed and let p be an arbitrary atomic formula. Then as InqB proves $\neg\neg p \leftrightarrow p$, we obtain $\neg\neg s(p) \equiv_{\mathsf{InqB}} s(p)$. Also, by Theorem 1, $s(p) \equiv_{\mathsf{InqB}} \alpha_1 \vee \ldots \vee \alpha_n$, where each α_i is $\vee$-free, whence $s(p) \equiv_{\mathsf{InqB}} \neg\neg(\alpha_1 \vee \ldots \vee \alpha_n)$. But as $(\neg\varphi \wedge \neg\psi) \leftrightarrow \neg(\varphi \vee \psi)$ holds intuitionistically, $s(p) \equiv_{\mathsf{InqB}} \neg(\neg\alpha_1 \wedge \ldots \wedge \neg\alpha_n)$, where $\neg(\neg\alpha_1 \wedge \ldots \wedge \neg\alpha_n)$ is $\vee$-free and thus Harrop.

For the second part of the proposition, fix an atomic formula p and take the substitution s that assigns to every atomic formula the formula $p \vee \neg p$. Assume, for the sake of contradiction, that $p \vee \neg p \equiv_{\mathsf{InqIL}} \alpha$, for some Harrop formula α. Then, by Split and disjunction property, we would obtain $\vdash_{\mathsf{InqIL}} \alpha \to p$ or $\vdash_{\mathsf{InqIL}} \alpha \to \neg p$, and thus $\vdash_{\mathsf{InqIL}} (p \vee \neg p) \to p$ or $\vdash_{\mathsf{InqIL}} (p \vee \neg p) \to \neg p$. This leads to a contradiction because none of these formulas is valid in classical logic which extends InqIL. So, s is not InqIL-equivalent to any H-substitution.

It remains to be shown that $s \in sub(\mathsf{InqIL})$. In order to show that a logic generated by modus ponens from a set of axioms is closed under a substitution s, it is sufficient to show that $s(\chi)$ is provable for every axiom χ. So, we have to show $\vdash_{\mathsf{InqIL}} s(\chi)$, for our specific s and every instance χ of Split. First, observe that for every formula χ, $s(\chi)$ is intuitionistically equivalent to one of these formulas: $\bot, p \vee \neg p, \top$. To see this, note that $\bot$ and $\top$ are generated from $p \vee \neg p$ by negation and the set $\{\bot, p \vee \neg p, \top\}$ is, up to intuitionistic equivalence, closed under the operations $\to, \wedge, \vee$. But then, if $s(\alpha), s(\varphi), s(\psi)$ are intuitionistically equivalent to any of the formulas $\bot, p \vee \neg p, \top$, the formula

$$(s(\alpha) \to (s(\varphi) \vee s(\psi))) \to ((s(\alpha) \to s(\varphi)) \vee (s(\alpha) \to s(\varphi)))$$

is intuitionistically valid. Hence, if we apply s to any instance of Split we always obtain a formula that is valid in InqIL. □

Proposition 3 shows that H-substitutions do not necessarily cover the space of all substitutions under which a logic validating Split is closed. Nevertheless, it will be clear from our main result, Theorem 2 below, that these substitutions play a special role in these logics.

We will rely heavily on a standard technique of proving structural completeness developed by Prucnal [26], or more precisely, on its refinement introduced later in [23]. Let α be a Harrop formula such that $\nvdash_{\mathsf{IL}} \neg\alpha$. Then there is a classical valuation v such that $v(\alpha) = 1$. Following [23], we define, relative to α and v, the following H-substitution:

$$s^v_\alpha(p) = \begin{cases} \alpha \to p & \text{if } v(p) = 1 \\ \neg\neg\alpha \wedge (\alpha \to p) & \text{otherwise} \end{cases}$$

Note that if α is a $\vee$-free formula, then s^v_α is a D-substitution.

Lemma 1. *Let φ be any formula, α, β Harrop formulas, and v a classical valuation such that $v(\beta) = 1$. Then (a) $s^v_\alpha(\varphi) \vdash_{\mathsf{IL}} \alpha \to \varphi$; (b) $s^v_\alpha(\beta) \equiv_{\mathsf{IL}} \alpha \to \beta$.*

For a proof of this crucial result, see [23]. Note that, as a direct consequence of Lemma 1-b, it holds that if α is a Harrop formula, and $v(\alpha) = 1$ then $\vdash_{\mathsf{IL}} s^v_\alpha(\alpha)$. This lemma implies the following one that generalizes the main result of [23], which was originally formulated for standard gsi-logics.

Lemma 2. *Let Λ be a gsi-logic, ψ, χ any formulas, and α a Harrop formula. Then if $\alpha \to (\psi \vee \chi) \in \Lambda$ then $(\alpha \to \psi) \vee (\alpha \to \chi) \in \Lambda$.*

Proof. Assume $\alpha \to (\psi \vee \chi) \in \Lambda$. We want to prove $(\alpha \to \psi) \vee (\alpha \to \chi) \in \Lambda$. If $\neg\alpha \in \Lambda$ then the required conclusion is immediate. Assume $\neg\alpha \notin \Lambda$. Then, due to Glivenko's theorem, $\neg\alpha$ is not classically valid and so there is a valuation v such that $v(\alpha) = 1$. Then, by Lemma 1-b, $s^v_\alpha(\alpha) \in \Lambda$. Since Λ is closed under H-substitutions, $s^v_\alpha(\alpha) \to (s^v_\alpha(\psi) \vee s^v_\alpha(\chi)) \in \Lambda$, and hence also $s^v_\alpha(\psi) \vee s^v_\alpha(\chi) \in \Lambda$. Thus, by Lemma 1-a, $(\alpha \to \psi) \vee (\alpha \to \chi) \in \Lambda$. □

The previous lemma plays a crucial role in the proof of one part of the next theorem which is our main result relating Split with the notions of SH-completeness and SD-completeness.

Theorem 2. *For every gsi-logic Λ the following claims are equivalent: (a) Λ is SH-complete; (b) Λ is SD-complete; (c) Split is valid in Λ.*

Proof. First, assume that Λ is SH-complete. Let α be a $\vee$-free formula and φ, ψ arbitrary formulas. By Lemma 2, for any H-substitution s, if $s(\alpha \to (\psi \vee \chi)) \in \Lambda$ then $s((\alpha \to \psi) \vee (\alpha \to \chi)) \in \Lambda$. It follows from SH-completeness that Split is valid in Λ. In the same way, one can prove that SD-completeness implies the validity of Split.

Second, assume that Split is valid in Λ. We show that $\varphi \vdash_\Lambda \psi$ iff for any H-substitution s, if $\vdash_\Lambda s(\varphi)$ then $\vdash_\Lambda s(\psi)$. The left-to-right direction follows immediately from Proposition 2. We prove the right-to-left direction. Assume that

(i) for any H-substitution s, if $\vdash_\Lambda s(\varphi)$ then $\vdash_\Lambda s(\psi)$.

We have to show that $\varphi \vdash_\Lambda \psi$. If $\vdash_\Lambda \neg\varphi$, we are done, so we can assume that $\nvdash_\Lambda \neg\varphi$. As Split is valid in Λ, we can take, due to the disjunctive form theorem, Harrop formulas $\alpha_1, \ldots, \alpha_n$ such that

(ii) $\varphi \equiv_\Lambda \alpha_1 \vee \ldots \vee \alpha_n$.

Let us assume that the disjunction is minimal, i.e. φ is not equivalent with the disjunction of any proper subset of $\{\alpha_1, \ldots, \alpha_n\}$. It follows that for every α_i $(1 \leq i \leq n)$, $\nvdash_{\mathsf{IL}} \neg\alpha_i$ (otherwise the disjunction in (ii) would not be minimal). Thus, due to Glivenko's theorem, for each $1 \leq i \leq n$, $\nvdash_{\mathsf{CL}} \neg\alpha_i$, and so there is a classical valuation v_i such that $v_i(\alpha_i) = 1$. Then the following holds:

1. $\vdash_\Lambda s^{v_i}_{\alpha_i}(\alpha_i)$ (by Lemma 1-b),
2. $\vdash_\Lambda s^{v_i}_{\alpha_i}(\alpha_i) \to s^{v_i}_{\alpha_i}(\varphi)$ (since Λ is closed under H-substitutions),
3. $\vdash_\Lambda s^{v_i}_{\alpha_i}(\varphi)$ (from 1. and 2.),
4. $\vdash_\Lambda s^{v_i}_{\alpha_i}(\psi)$ (from 3. and the assumption (i)),
5. $s^{v_i}_{\alpha_i}(\psi) \vdash_\Lambda \alpha_i \to \psi$ (by Lemma 1-a),
6. $\alpha_i \vdash_\Lambda \psi$ (from 4. and 5.).

Since the last point holds for any $1 \leq i \leq n$ we obtain $\varphi \vdash_\Lambda \psi$ as required. In the same way, one can prove that the validity of Split implies that the logic is SD-complete. □

In the rest of this paper, we will explore some consequences of this result. The first one is immediate and it shows an interesting difference between SF-completeness/SG-completeness (on the one hand) and SH-completeness/SD-completeness (on the other).

Corollary 1. *Every SH-complete (SD-complete) gsi-logic is hereditarily SH-complete (SD-complete).*

In contrast to this corollary, there are examples of gsi-logics that are SF- and SG-complete but neither hereditarily SF-complete nor hereditarily SG-complete. For instance, the logic of finite problems ML has this property (see [9,12]). The next corollary is based on the observation that SD-completeness is a stronger property than SG-completeness.

Corollary 2. InqIL *is hereditarily SG-complete.*

In other words, every gsi-logic that validates Split is SG-complete. The converse does not hold. ML does not validate Split but it is SG-complete.

The relations between different kinds of structural completeness are summarized in the following picture:

$$\begin{array}{ccc} SF & \overset{\Longrightarrow}{\not\Longleftarrow} & SG \\ \not\Downarrow\Uparrow & & \Uparrow\not\Downarrow \\ SD & \Longleftrightarrow & SH \end{array}$$

ML is a counterexample to $SF \Longrightarrow SD(H)$ and $SG \Longrightarrow SD(H)$, and any inquisitive gsi-logic is a counterexample to $SD(H) \Longrightarrow SF$ and $SG \Longrightarrow SF$.

4 Schematic Closures of Inquisitive Gsi-Logics

In this section we discuss some issues concerning SF-completeness and for this purpose we employ the following notation. If Λ is a gsi-logic and Δ a set of formulas then $\Lambda \oplus \Delta$ will denote the set of formulas derivable from $\Lambda \cup \Delta$ by modus ponens. More precisely,

$$\Lambda \oplus \Delta = \{\varphi \mid \psi \wedge \chi_1 \wedge \ldots \wedge \chi_n \vdash_{\mathsf{IL}} \varphi \text{ for some } \psi \in \Lambda \text{ and } \chi_1, \ldots, \chi_n \in \Delta\}.$$

Note that if Δ is closed under D-substitutions then so is $\Lambda \oplus \Delta$, and if, moreover, $\Lambda \cup \Delta$ is consistent, i.e. $\bot \notin \Lambda \oplus \Delta$, then $\Lambda \oplus \Delta$ is the smallest gsi-logic including $\Lambda \cup \Delta$.

Let us recall that LC is the Gödel-Dummett fuzzy logic [11,16]. It is often presented as an extension of intuitionistic logic by the prelinearity schema:

$$(\varphi \to \psi) \vee (\psi \to \varphi).$$

Let us denote the set of all instances of this schema as $PreLin$ and let $FullSplit$ denote the set of instances of the schema (where χ is not restricted)

$$(\chi \to (\varphi \vee \psi)) \to ((\chi \to \varphi) \vee (\chi \to \psi)).$$

An obvious connection of LC to the logics we are focused on in this paper is given by the following observation made in [11].

Proposition 4. $\mathsf{LC} = \mathsf{IL} \oplus PreLin = \mathsf{IL} \oplus FullSplit.$

So, LC is a standard extension of InqIL ($= \mathsf{IL} \oplus Split$). It is well-known that the logic LC is hereditarily structurally complete in the class of standard gsi-logics (shown in [12]). This result can also be obtained as a direct consequence of our Corollary 2, if we recall that SG-completeness generalizes the usual structural completeness. A natural question arises whether LC is also hereditarily SF-complete over the space of gsi-logics in general.

When one considers results over the case of standard gsi-logics, there is reason to be cautious before importing them to our general setting. That a logic is hereditarily structurally complete over the standard gsi-logics does not *a priori* entail hereditary SF-completeness over the broader space of gsi-logics. To see that LC is indeed hereditarily SF-complete, we need to use the following observation that is due to Dummett [11].

Lemma 3. $\varphi \vee \psi \equiv_{\mathsf{LC}} ((\varphi \to \psi) \to \psi) \wedge ((\psi \to \varphi) \to \varphi).$

This result shows that disjunction is definable in LC. In fact, as pointed out in [23], every (standard) gsi-logic in which disjunction is definable includes LC. The previous lemma implies the following one.

Lemma 4. *Every gsi-logic that includes* LC *is standard.*

Proof. Take any gsi-logic Λ that includes LC, any $\varphi \in \Lambda$ and any substitution s. It follows from Lemma 3 that for every formula ψ there is a $\vee$-free formula α_ψ such that $\psi \equiv_\Lambda \alpha_\psi$. We define a D-substitution s^* fixing $s^*(p) = \alpha_{s(p)}$, for every atom p. Note that s^* is Λ-equivalent to s. Since $s^*(\varphi) \in \Lambda$, we also have $s(\varphi) \in \Lambda$. Hence, Λ is standard. $\square$

Using this Lemma we can prove the following result.

Theorem 3. LC *is hereditarily* SF*-complete over all gsi-logics.*

We can even strengthen this result in the following way.

Theorem 4. *Let* Λ *be a gsi-logic including* InqIL*. Then the following claims are equivalent: (a)* Λ *is hereditarily* SF*-complete; (b)* Λ *is* SF*-complete; (c)* Λ *is standard; (d)* Λ *includes all instances of* $FullSplit$*; (e) if* $\vdash_\Lambda \varphi \to (\psi \vee \chi)$ *then* $\vdash_\Lambda (\varphi \to \psi) \vee (\varphi \to \chi)$*; (f)* $\mathsf{LC} \subseteq \Lambda$*.*

Proof. $(a) \Rightarrow (b)$ is immediate by definition. $(b) \Rightarrow (c)$ is from Proposition 1. $(c) \Rightarrow (d)$: As a standard extension of InqIL, Λ must contain all instances of $FullSplit$. $(d) \Rightarrow (e)$ is immediate. $(e) \Rightarrow (f)$: As an extension of IL, Λ includes all instances of $(\varphi \vee \psi) \to (\varphi \vee \psi)$. By the assumption (e), we obtain $((\varphi \vee \psi) \to \varphi) \vee ((\varphi \vee \psi) \to \psi) \in \Lambda$. Since $\psi \to (\varphi \vee \psi), \varphi \to (\varphi \vee \psi)$ are theorems, we obtain $(\varphi \to \psi) \vee (\psi \to \varphi) \in \Lambda$. By Proposition 4, $\mathsf{LC} \subseteq \Lambda$. $(f) \Rightarrow (a)$: By Theorem 3, LC is hereditarily SF-complete, whence in virtue of the inclusion of LC in Λ, Λ is hereditarily SF-complete. $\square$

It is clear that the logic LC is what we obtain if we close InqIL under all substitutions. Let us consider this operation of *schematic closure* from a more general perspective. We have already indicated that any gsi-logic Λ determines its *schematic fragment* $S(\Lambda) = \{\varphi \mid s(\varphi) \in \Lambda, \text{for every substitution } s\}$, which is a standard gsi-logic. All inquisitive gsi-logics have the same schematic fragment, namely ML. The operation of schematic closure that we denote as C can be generally defined as follows: $\varphi \in C(\Lambda)$ iff $s_1(\psi_1) \wedge \ldots \wedge s_n(\psi_n) \vdash_{\mathsf{IL}} \varphi$, for some substitutions $s_1, \ldots, s_n$ and some $\psi_1, \ldots, \psi_n \in \Lambda$. Note that $C(\Lambda)$ is indeed a standard gsi-logic. In general, it holds that $S(\Lambda) \subseteq \Lambda \subseteq C(\Lambda)$. $S(\Lambda)$ is the greatest standard gsi-logic below Λ, and $C(\Lambda)$ is the smallest standard gsi-logic above Λ. So, if Λ is itself standard, we obtain $S(\Lambda) = \Lambda = C(\Lambda)$. In the rest of this section we explore the schematic closures of inquisitive gsi-logics.

It is clear that $C(\mathsf{InqIL}) = \mathsf{LC}$ and $C(\mathsf{InqB}) = \mathsf{CL}$. What do the schematic closures of the other inquisitive gsi-logics look like? The standard gsi-logics that include LC form a chain consisting of the multivalued logics G_n, $n \geq 3$, plus the classical logic CL on the top. If we denote classical logic as G_2 and the logic LC as G_ω then we have (see [16]):

$$\mathsf{G}_\omega \subseteq \ldots \subseteq \mathsf{G}_5 \subseteq \mathsf{G}_4 \subseteq \mathsf{G}_3 \subseteq \mathsf{G}_2.$$

The schematic closure of every gsi-logic extending InqIL will be one of the G-logics. For those gsi-logics which include InqIL and have the disjunction property, i.e. for the inquisitive gsi-logics, we can characterize the schematic closures in an elegant and systematic way.

For any set of formulas Δ, let Δ^{df} denote the $\vee$-free fragment of Δ, i.e. $\Delta^{df} = \{\varphi \in \Delta \mid \varphi \text{ is } \vee\text{-free}\}$. The disjunction property of inquisitive logics is crucial for the proof of the following lemma.

Lemma 5. *Let Λ be any inquisitive gsi-logic and G_n, where $n \in \{2, 3, 4, \ldots, \omega\}$, any gsi-logic including LC. Then (a) $\Lambda = \mathsf{InqIL} \oplus \Lambda^{df}$; (b) $\mathsf{G}_n = \mathsf{LC} \oplus \mathsf{G}_n^{df}$.*

Proof. (a) Let Λ be any inquisitive gsi-logic. Clearly, it holds that $\mathsf{InqIL} \oplus \Lambda^{df} \subseteq \Lambda$. Assume $\varphi \in \Lambda$. Let $\varphi \equiv_{\mathsf{InqIL}} \alpha_1 \vee \ldots \vee \alpha_n$, where each α_i is $\vee$-free. By the disjunction property, for some i, $\alpha_i \in \Lambda$ and thus $\alpha_i \in \Lambda^{df}$. Hence, $\varphi \in \mathsf{InqIL} \oplus \Lambda^{df}$. (b) Clearly, $\mathsf{LC} \oplus \mathsf{G}_n^{df} \subseteq \mathsf{G}_n$. Assume $\varphi \in \mathsf{G}_n$. Then, by Lemma 3, there is $\vee$-free α such that $\varphi \equiv_{\mathsf{LC}} \alpha$. Then $\alpha \in \mathsf{G}_n^{df}$ and thus $\varphi \in \mathsf{LC} \oplus \mathsf{G}_n^{df}$.

With the help of this lemma we can characterize the schematic closures of inquisitive gsi-logics in the following way.

Theorem 5. *Let Λ be an inquisitive gsi-logic. Then*

$$C(\Lambda) = \mathsf{LC} \oplus \Lambda^{df} = \mathsf{G}_n, \ \text{for } n = \max\{m \mid \Lambda^{df} \subseteq \mathsf{G}_m^{df}\}.$$

Proof. Assume that Λ is an inquisitive gsi-logic. Clearly, it holds that $\mathsf{LC} \oplus \Lambda^{df} \subseteq C(\Lambda)$. By Lemma 5-a, $\Lambda = \mathsf{InqIL} \oplus \Lambda^{df}$ and thus $\Lambda \subseteq \mathsf{LC} \oplus \Lambda^{df}$. By Lemma 4, $\mathsf{LC} \oplus \Lambda^{df}$ is standard and since $C(\Lambda)$ is the smallest standard gsi-logic extending Λ, we obtain $C(\Lambda) \subseteq \mathsf{LC} \oplus \Lambda^{df}$. So, we have proved that $C(\Lambda) = \mathsf{LC} \oplus \Lambda^{df}$.

Clearly, $C(\Lambda) = \mathsf{G}_n$, for some n. Then it must hold $\Lambda^{df} \subseteq \mathsf{G}_n^{df}$. Assume, for the sake of contradiction, that $n \neq \omega$ and $\Lambda^{df} \subseteq \mathsf{G}_{n+1}^{df}$. Then, using Lemma 5-b, we obtain $C(\Lambda) = \mathsf{LC} \oplus \Lambda^{df} \subseteq \mathsf{LC} \oplus \mathsf{G}_{n+1}^{df} = \mathsf{G}_{n+1}$. But $C(\Lambda) \subseteq \mathsf{G}_{n+1}$ is in contradiction with the assumption $C(\Lambda) = \mathsf{G}_n$. Thus $n = \max\{m \mid \Lambda^{df} \subseteq \mathsf{G}_m^{df}\}$. □

So, while $\Lambda = \mathsf{InqIL} \oplus \Lambda^{df}$, $C(\Lambda) = \mathsf{LC} \oplus \Lambda^{df}$. This result shows that the schematic closure operation collapses the uncountable space of inquisitive gsi-logics to a countable linear order of standard gsi-logics. Moreover, this mapping to G_n-logics is surjective. In particular, each $\mathsf{InqIL} \oplus \mathsf{G}_n^{df}$ is inquisitive and $C(\mathsf{InqIL} \oplus \mathsf{G}_n^{df}) = \mathsf{G}_n$.

5 Kripke Models

Structural completeness is an important property of classical logic that intuitionistic logic lacks. On the other hand, the disjunction property is an important property of intuitionistic logic that classical logic violates. An interesting question is whether there are logics that have both these properties.

Definition 5. *A gsi-logic is* optimal *if it is SG-complete and has the disjunction property.*

The only standard gsi-logic that is known to be optimal is ML (see [37]). As another direct consequence of Theorem 2 we obtain the following result showing that among non-standard gsi-logics there are uncountably many optimal logics.

Theorem 6. *Every inquisitive gsi-logic is optimal.*

Let us point out that at the end of [22] a conjecture is formulated which, when translated in our terminology, says that (a) InqIL is optimal, and (b) InqIL and InqB are the only optimal gsi-logics. Theorem 6 thus serves to prove the first half of the conjecture and refutes the latter half.

In this section, we show that Theorem 6 is related to the possibility of an interesting canonical model construction for inquisitive gsi-logics. For any gsi-logic one can construct, in the usual way, a canonical Kripke model built out of prime theories. A peculiar feature of inquisitive gsi-logics is that they can also be characterized by a Kripke model built directly out of consistent $\vee$-free formulas.[4] We will briefly formulate this construction and compare it with another unusual canonical model construction that we obtain as an application of our main results. For the rest of this section, let us fix an inquisitive gsi-logic Λ. Moreover, let us say that α is *consistent* if $\nvdash_\Lambda \neg\alpha$.

Now we introduce the Kripke semantics for intuitionistic logic. A *Kripke frame* is a pair $\langle S, \leq \rangle$ where $\leq$ is a preorder, i.e. a reflexive and transitive relation on S. A Kripke model is a Kripke frame equipped with a valuation V, i.e. a

[4] This is related to a fact that was already observed in [28], namely that any inquisitive gsi-logic Λ can be characterized by a canonical Kripke model built up from the Lindenbaum-Tarski algebra of the $\vee$-free fragment of Λ.

function that assigns to each atomic formula an upward closed subset of S (that is, if $w \in V(p)$ and $w \leq v$ then $v \in V(p)$). Given any Kripke model the relation $\Vdash$ between states of the model and formulas is defined in the usual recursive way. For the atomic formulas, we set $w \Vdash p$ iff $w \in V(p)$. For the constant $\bot$ and complex formulas, the relation is determined as follows:

(a) $w \nVdash \bot$,
(b) $w \Vdash \varphi \rightarrow \psi$ iff for any $v \geq w$, if $v \Vdash \varphi$ then $v \Vdash \psi$,
(c) $w \Vdash \varphi \wedge \psi$ iff $w \Vdash \varphi$ and $w \Vdash \psi$,
(d) $w \Vdash \varphi \vee \psi$ iff $w \Vdash \varphi$ or $w \Vdash \psi$.

The relation is persistent: if $w \Vdash \varphi$ and $w \leq v$ then $v \Vdash \varphi$. We say that φ is valid in a model $\langle S, \leq, V\rangle$ if $w \Vdash \varphi$ holds in that model for every $w \in S$. It is well-known that a formula is intuitionistically valid iff it is valid in all Kripke models [21]. The recursive clauses for $\Vdash$ actually mirror some characteristic properties of $\vdash$ in inquisitive gsi-logics, as is shown in the following proposition.

Proposition 5. *Let α be a consistent $\vee$-free formula and φ, ψ arbitrary formulas. Then*

(a) $\alpha \nvdash_\Lambda \bot$,
(b) $\alpha \vdash_\Lambda \varphi \rightarrow \psi$ *iff for any consistent $\vee$-free β s.t. $\beta \vdash_\Lambda \alpha$ if $\beta \vdash_\Lambda \varphi$ then $\beta \vdash_\Lambda \psi$,*[5]
(c) $\alpha \vdash_\Lambda \varphi \wedge \psi$ *iff* $\alpha \vdash_\Lambda \varphi$ *and* $\alpha \vdash_\Lambda \psi$,
(d) $\alpha \vdash_\Lambda \varphi \vee \psi$ *iff* $\alpha \vdash_\Lambda \varphi$ *or* $\alpha \vdash_\Lambda \psi$.

Proof. (a) and (c) are immediate. Due to Lemma 2, (d) holds for every gsi-logic which has the disjunction property. Let us prove (b). The left-to-right direction is immediate. For the right-to-left direction assume that for any consistent $\vee$-free $\beta \vdash_\Lambda \alpha$, if $\beta \vdash_\Lambda \varphi$ then $\beta \vdash_\Lambda \psi$. Obviously, $\beta \vdash_\Lambda \psi$ holds also in the case that β is not consistent. Assume that $\varphi \equiv_\Lambda \gamma_1 \vee \ldots \vee \gamma_n$, where $\gamma_1, \ldots, \gamma_n$ are $\vee$-free. Then $\gamma_i \wedge \alpha \vdash_\Lambda \alpha$ and $\gamma_i \wedge \alpha \vdash_\Lambda \varphi$. Our assumption implies that $\gamma_i \wedge \alpha \vdash_\Lambda \psi$. So, for all i, $\gamma_i \vdash_\Lambda \alpha \rightarrow \psi$, and thus $\varphi \vdash_\Lambda \alpha \rightarrow \psi$. It follows that $\alpha \vdash_\Lambda \varphi \rightarrow \psi$. □

This observation motivates the construction of a canonical Kripke model $\mathcal{M}_\Lambda = \langle S_\Lambda, \leq_\Lambda, V_\Lambda\rangle$, where $S_\Lambda = \{\alpha \mid \alpha \text{ is } \vee\text{-free and consistent}\}$, $\alpha \leq_\Lambda \beta$ iff $\beta \vdash_\Lambda \alpha$, and $\alpha \in V_\Lambda(p)$ iff $\alpha \vdash_\Lambda p$. Then Proposition 5 directly implies that $\Vdash$ in $\mathcal{M}_\Lambda$ coincides with $\vdash_\Lambda$.

Theorem 7. *For each φ and each consistent $\vee$-free α, $\alpha \Vdash \varphi$ in $\mathcal{M}_\Lambda$ if and only if $\alpha \vdash_\Lambda \varphi$. As a consequence, $\varphi \in \Lambda$ if and only if φ is valid in $\mathcal{M}_\Lambda$.*

There is also a remarkable correspondence between the semantic clauses of Kripke semantics and the behaviour of D-substitutions in inquisitive gsi-logics. This correspondence is based on Theorem 6. To make it more visible, let us write $s \succ \varphi$ instead of $\vdash_\Lambda s(\varphi)$ and let us define for any D-substitutions s, t that $s \leq t$ iff there is a D-substitution u such that $t = u \circ s$ (where $\circ$ is the composition of substitutions).

[5] The word "consistent" could be omitted here and the equivalence would hold too but in a moment we will need this particular form of the statement that quantifies over consistent formulas.

Proposition 6. *Let s be a D-substitution and φ, ψ arbitrary formulas. Then*

(a) $s \not\succ \bot$,
(b) $s \succ \varphi \rightarrow \psi$ *iff for any D-substitution $t \geq s$, if $t \succ \varphi$ then $t \succ \psi$,*
(c) $s \succ \varphi \wedge \psi$ *iff* $s \succ \varphi$ *and* $s \succ \psi$,
(d) $s \succ \varphi \vee \psi$ *iff* $s \succ \varphi$ *or* $s \succ \psi$.

Proof. All cases are straightforward. We will just comment on the case (b). This case can be reformulated in this way: $s(\varphi) \vdash s(\psi)$ iff for any D-substitution t, if $\vdash t(s(\varphi))$ then $\vdash t(s(\psi))$. But this is exactly SD-completeness applied to the implication $s(\varphi) \rightarrow s(\psi)$. □

Using this observation we can build from D-substitutions a particular canonical Kripke model $\mathcal{M}^{\Lambda} = \langle S^{\Lambda}, \leq^{\Lambda}, V^{\Lambda} \rangle$, where S^{Λ} is the set of all D-substitutions, $s \leq^{\Lambda} t$ iff there is a D-substitution u such that $t = u \circ s$, $s \in V^{\Lambda}(p)$ iff $s \succ p$.[6]

Let id be the identity function on atomic formulas. Then id is a D-substitution and $id \leq^{\Lambda} s$, for every D-substitution s. So, φ is valid in $\mathcal{M}^{\Lambda}$ iff $id \Vdash \varphi$ in $\mathcal{M}^{\Lambda}$. Proposition 6 implies the following result.

Theorem 8. *For each φ and each D-substitution s, $s \Vdash \varphi$ in $\mathcal{M}^{\Lambda}$ if and only if $s \succ \varphi$. As a consequence, $\varphi \in \Lambda$ if and only if φ is valid in $\mathcal{M}^{\Lambda}$.*

6 Conclusion

Let us summarize the main results of this paper. We have studied four notions of structural completeness (SF-, SG-, SH- and SD-completeness) in a class of generalized superintuitionistic logics that are not required to be closed under all substitutions but only under substitutions assigning disjunction free formulas. We have shown a connection between these notions and the schema Split that axiomatizes intuitionistic inquisitive logic InqIL. A characteristic feature of Split is that it allows one to transform every formula to a disjunctive normal form (Theorem 1).

Our main result (Theorem 2) shows that SH-completeness is equivalent to SD-completeness and these properties hold exactly for those logics that validate Split. As a consequence, $SH(D)$-completeness is hereditary (Corollary 1). We have also shown that InqIL is hereditarily SG-complete (Corollary 2) and its closure under substitutions, i.e. the Gödel-Dummett logic LC, remains hereditarily SF-complete (Theorem 3 and its extension Theorem 4).

We have further studied inquisitive logics, i.e. those logics that include InqIL and have the disjunction property. We have proved that the operation that closes every such logic under substitutions maps the uncountably large class of inquisitive logics onto the countably infinite chain of those logics that include LC (Theorem 5). It follows directly from our main result that inquisitive logics are optimal, i.e. they are structurally complete and have disjunction property

[6] An analogous construction was described in [37] for the standard optimal gsi-logic ML.

(Theorem 6). They can be characterized by a canonical Kripke model built from consistent disjunction free formulas (Theorem 7). Interestingly, their optimality means that they can be alternatively characterized by a canonical Kripke model built from substitutions assigning disjunction free formulas (Theorem 8).

In future work, we would like to study structural completeness in the more general context of substructural inquisitive logics [30]. We also plan to explore the notion of structural completeness for these logics in the setting of multi-conclusion consequence relation [19].

Acknowledgements. This paper is an outcome of the project Logical Structure of Information Channels, no. 21-23610M, supported by the Czech Science Foundation and carried out at the Institute of Philosophy of the Czech Academy of Sciences.

References

1. Bezhanishvili, N., Grilletti, G., Holliday, W.H.: Algebraic and topological semantics for inquisitive logic via choice-free duality. In: Iemhoff, R., Moortgat, M., de Queiroz, R. (eds.) WoLLIC 2019. LNCS, vol. 11541, pp. 35–52. Springer, Heidelberg (2019). https://doi.org/10.1007/978-3-662-59533-6_3
2. Ciardelli, I.: Inquisitive semantics and intermediate logics. Master's thesis, University of Amsterdam, Amsterdam (2009)
3. Ciardelli, I.: Questions in logic. Ph.D. thesis, University of Amsterdam, Amsterdam (2016)
4. Ciardelli, I., Groenendijk, J., Roelofsen, F.: Inquisitive Semantics. Oxford University Press, Oxford (2019)
5. Ciardelli, I., Iemhoff, R., Yang, F.: Questions and dependency in intuitionistic logic. Notre Dame J. Formal Log. **61**, 75–115 (2020)
6. Ciardelli, I., Roelofsen, F.: Inquisitive logic. J. Philos. Log. **40**, 55–94 (2011)
7. Ciardelli, I., Roelofsen, F.: Inquisitive dynamic epistemic logic. Synthese **192**, 1643–1687 (2015)
8. Cintula, P., Metcalfe, G.: Structural completeness in fuzzy logics. Notre Dame J. Formal Log. **50**, 153–182 (2009)
9. Citkin, A.I.: On structurally complete superintuitionistic logics. Sov. Math. Dokl. **19**, 816–819 (1978)
10. Citkin, A.I.: Hereditarily structurally complete superintuitionistic deductive systems. Stud. Log. **106**, 827–856 (2018)
11. Dummett, M.: A propositional calculus with denumerable matrix. J. Symb. Log. **24**, 97–106 (1959)
12. Dzik, D., Wroński, A.: Structural completeness of Gödel's and Dummett's propositional calculi. Stud. Log. **32**, 69–73 (1973)
13. Frittella, S., Greco, G., Palmigiano, A., Yang, F.: A multi-type calculus for inquisitive logic. In: Väänänen, J., Hirvonen, Å., de Queiroz, R. (eds.) WoLLIC 2016. LNCS, vol. 9803, pp. 215–233. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-52921-8_14
14. Gessel, T.: Action models in inquisitive logic. Synthese **197**, 3905–3945 (2020)
15. Gessel, T.: Questions in two-dimensional logic. Rev. Symb. Log. **15**, 859–879 (2022)
16. Gödel, K.: Zum intuitionistischen Aussagenkalkül. Anzeiger Akad. Wissenschaften Wien **69**, 65–66 (1932)

17. Grilletti, G.: Medvedev logic is the logic of finite distributive lattices without top element. In: Fernández-Duque, D., Palmigiano, A., Pinchinat, S. (eds.) AiML 2022. Advances in Modal Logic, vol. 14. College Publications (2022)
18. Holliday, W.H.: Inquisitive intuitionistic logic. In: Olivetti, N., Verbrugge, R., Negri, S., Sandu, G. (eds.) AiML 2020. Advances in Modal Logic, vol. 13, pp. 329–348. College Publications (2020)
19. Iemhoff, R.: Consequence relations and admissible rules. J. Philos. Log. **45**, 327–348 (2016)
20. Iemhoff, R., Yang, F.: Structural completeness in propositional logics of dependence. Arch. Math. Log. **55**, 955–975 (2016)
21. Kripke, S.A.: Semantical analysis of intuitionistic logic I. In: Dummett, M.A.E., Crossley, J.N. (eds.) Studies in Logic and the Foundations of Mathematics, vol. 40, pp. 92–130. Elsevier (1965)
22. Miglioli, P., Moscato, U., Ornaghi, M., Quazza, S., Usberti, G.: Some results on intermediate constructive logics. Notre Dame J. Formal Log. **30**, 543–562 (1989)
23. Minari, P., Wroński, A.: The property (HD) in intermediate logics: a partial solution of a problem of H. Ono. Rep. Math. Log. **22**, 21–25 (1988)
24. Olson, J., Raftery, J.G., van Alten, C.J.: Structural completeness in substructural logics. Log. J. IGPL **16**, 455–495 (2008)
25. Pogorzelski, W.A.: Structural completeness of the propositional calculus. Bull. l'Acad. Polonaise Sci. Sér. Sci. Math. **19**, 349–351 (1971)
26. Prucnal, T.: On the structural completeness of some pure implicational propositional calculi. Stud. Log. **32**, 45–50 (1973)
27. Punčochář, V.: Weak negation in inquisitive semantics. J. Logic Lang. Inform. **23**, 47–59 (2015)
28. Punčochář, V.: A generalization of inquisitive semantics. J. Philos. Log. **45**, 399–428 (2016)
29. Punčochář, V.: Algebras of information states. J. Log. Comput. **27**, 1643–1675 (2017)
30. Punčochář, V.: Substructural inquisitive logics. Rev. Symb. Log. **12**, 296–330 (2019)
31. Punčochář, V.: Inquisitive Heyting algebras. Stud. Log. **109**, 995–1017 (2021)
32. Punčochář, V., Sedlár, I.: Inquisitive propositional dynamic logic. J. Log. Lang. Inform. **30**, 91–116 (2021)
33. Quadrellaro, D.E.: On intermediate inquisitive and dependence logics: an algebraic study. Ann. Pure Appl. Log. **173** (2022). Article 103143
34. Roelofsen, F.: Algebraic foundations for the semantic treatment of inquisitive content. Synthese **190**(1), 79–102 (2013). https://doi.org/10.1007/s11229-013-0282-4
35. Sano, K.: Goldblatt-Thomason-style characterization for intuitionistic inquisitive logic. In: Olivetti, N., Verbrugge, R., Negri, S., Sandu, G. (eds.) AiML 2020. Advances in Modal Logic, vol. 13, pp. 541–560. College Publications (2020)
36. Stafford, W.: Proof-theoretic semantics and inquisitive logic. J. Philos. Log. **50**, 1199–1229 (2021)
37. Wojtylak, P.: On a problem of H. Friedman and its solution by T. Prucnal. Rep. Math. Log. **38**, 69–86 (2004)

Validity in Choice Logics

A Game-Theoretic Investigation

Robert Freiman(✉) and Michael Bernreiter

Institute of Logic and Computation, TU Wien, Vienna, Austria
robert@logic.at, michael.bernreiter@tuwien.ac.at

Abstract. Qualitative Choice Logic (QCL) is a framework for jointly dealing with truth and preferences. We develop the concept of degree-based validity by lifting a Hintikka-style semantic game [10] to a provability game. Strategies in the provability game are translated into proofs in a novel labeled sequent calculus where proofs come in degrees. Furthermore, we show that preferred models can be extracted from proofs.

Keywords: choice logics · game semantics · sequent calculus

1 Introduction

Preferences are important in many research areas, including computer science and artificial intelligence [14]. A formalism for preference representation that has gained considerable attention is Qualitative Choice Logic (QCL) [6], which extends classical propositional logic with a connective $\vec{\times}$ called ordered disjunction. $F \vec{\times} G$ expresses that F or G should be satisfied, but satisfying F is preferable to satisfying only G. QCL and its variations [2,4,5] have been studied with regards to applications [1,7,12,15,16], computational properties [4], and proof systems [3].

Recently, QCL has been reexamined through the lens of game theoretic semantics (GTS). Specifically, Game-induced Choice Logic (GCL) [10] was introduced as an extension of Hintikka's semantic game for classical logic [11]. In this semantic game, two players – *Me* and *You*– play over a fixed formula F and an interpretation $\mathcal{I}$. Hintikka's modeling of truth as a win for *Me* and falsity as a loss is refined by more fine-grained outcomes. The more preferences I am able to satisfy during the game, the higher the payoff for *Me*. Besides providing a new understanding of ordered disjunction, GCL addresses some contentious behavior of negation in QCL, where a formula F is not necessarily semantically equivalent to the double negation $\neg\neg F$. GCL redefines negation using game-theoretic methods and thus provides semantics where F is equivalent to $\neg\neg F$ and negation behaves more similarly to classical negation in general.

A natural question not yet addressed in existing work on GCL is whether there is an algorithm that finds strategies for *Me* which guarantee a fixed payoff for the game over a fixed formula F and *all* interpretations. Reduced to winning strategies, this corresponds to the question of the validity of F. We answer this

H. H. Hansen et al. (Eds.): WoLLIC 2023, LNCS 13923, pp. 211–226, 2023.
https://doi.org/10.1007/978-3-031-39784-4_13

question by lifting the GTS to a dialogue game (we prefer the term *provability game*). Intuitively, in this game, the players play the semantic game over all interpretations simultaneously but *I* am allowed to create backup copies of game states. This technique has been demonstrated to lead to adequate proof systems for a variety of logics [8,9,13]. Our approach is the first to interpret non-classical truth values with non-binary outcomes in both the semantic and the disjunctive game. Our main result states that from a strategy σ for *Me* in the disjunctive game one can extract strategies for the semantic game over every interpretation yielding a payoff at least as good as σ's. Furthermore, from *Your* strategy one can extract an interpretation $\mathcal{I}$ and a strategy for the semantic game over $\mathcal{I}$ yielding at least the same payoff for *You*. In logical terminology, this corresponds to counter-model extraction; in the realm of preference handling, this corresponds to the construction of a preferred model.

While the exposition in this paper is mostly game-theoretic, we demonstrate that strategies for *Me* in the disjunctive game can be formulated as proofs in a labeled sequent calculus. Unlike the system for QCL [3], in our proof system $\mathbf{GS}^*$, proofs have degrees where positive degrees represent proofs of validity, while negative degrees represent refutations of a formula.

This paper is structured as follows: In Sect. 2, we recall game-theoretic notions and GCL. In Sect. 3, we lift the semantic game for GCL to a provability game. In Sect. 4, we reformulate *My* strategies as a proof system.

2 Preliminaries

In this section, we recall the game-induced choice logic GCL and its two semantics – game-theoretic and degree-based. The language of GCL is the same as QCL's, i.e., it extends the usual propositional language by the choice connective $\vec{\times}$. We assume an infinite countable set of propositional variables $a, b, \ldots$. Compound formulas are built according to the following grammar:

$$F ::= a \mid \neg F \mid F \wedge F \mid F \vee F \mid F \vec{\times} F.$$

An *interpretation* $\mathcal{I}$ is a set of propositional variables, with $\mathcal{I} \models a$ iff $a \in \mathcal{I}$.

2.1 Game-Theoretic Semantics

We start by recalling Hintikka's game [11] over a formula F in the language restricted to the connectives $\vee, \wedge, \neg$ and over a classical interpretation $\mathcal{I}$. The game is played between two players, *Me* and *You*, both of which can act either in the role of Proponent (**P**) or Opponent (**O**). At formulas of the form $G_1 \vee G_2$, **P** chooses a formula G_i that the game continues with. At formulas of the form $G_1 \wedge G_2$ it is **O**'s choice. At negations $\neg G$, the game continues with G and a role switch. Every outcome (final state of the game) is an occurrence of a propositional variable a. The player currently in the role of **P** wins the game (and $\mathbf{O}$ loses) iff $a \in \mathcal{I}$. The central result is that I have a winning strategy for the game starting in $\mathbf{P} : F$ iff $\mathcal{I} \models F$.

To deal with ordered disjunction ($\vec{\times}$), Hintikka's game is extended as follows [10]: at $G_1 \vec{\times} G_2$ it is $\mathbf{P}$'s choice whether to continue with G_1 or with G_2, *but this player prefers* G_1. The preferences of $\mathbf{O}$ are the exact opposite of $\mathbf{P}$. For both players, the aim in the game is now not only to win the game but to do so with as little compromise to their preferences as possible. Thus, it is natural to express $\mathbf{P}$'s preference of G_1-outcomes O_1 over G_2-outcomes O_2 via the relation $O_1 \gg O_2$. We leave the formal treatment of this game for the next section and proceed with some standard game-theoretic definitions.

Definition 1. *A* game *is a pair* $\mathbf{G} = (T, d)$*, where*

1. $T = (V, E, l)$ *is a tree with set of nodes* V *(called* (game) states*) and edges* E*. The leaves of* T *are called* outcomes *and are denoted* $\mathcal{O}(\mathbf{G})$*. The* labeling function l *maps nodes of* T *to the set* $\{I, Y\}$*.*
2. d *is a payoff-function mapping outcomes to elements of a linear order* $(\Lambda, \preceq)$*.*

We write $x \approx y$ *if* $x \preceq y$ *and* $y \preceq x$*.* Λ *is partitioned into two sets,* W *and* L*, where* W *is upward-closed and* $L = \Lambda \setminus W$*. Outcomes* O *are called* winning *if* $d(O) \in W$ *and* losing *if* $d(O) \in L$*. A* run *of the game is a maximal path in* T *starting at the root.*

Hintikka's game can be seen as a game in the sense of this definition: the game tree is the formula tree of F where each occurrence of a subformula G of F is decorated with either $\mathbf{P}$ or $\mathbf{O}$, we write $\mathbf{P} : G$ and $\mathbf{O} : G$, respectively. Let F be decorated with $\mathbf{Q}_0 \in \{\mathbf{P}, \mathbf{O}\}$. If $G = G_1 \vee G_2$, or $G = G_1 \wedge G_2$ then the children of $\mathbf{Q} : G$ are decorated the same. If $G = \neg G'$, then the child of $\mathbf{Q} : G$ is $\bar{\mathbf{Q}} : G'$, where $\bar{\mathbf{Q}}$ is $\mathbf{O}$ if $\mathbf{Q} = \mathbf{P}$, and $\mathbf{P}$ otherwise. As for the labeling function, game states of the form $\mathbf{P} : G_1 \vee G_2$, $\mathbf{O} : G_1 \wedge G_2$ are I-states and all other states are Y-states.

As for payoffs, we write $\mathcal{I} \models \mathbf{P} : a$ iff $\mathcal{I} \models a$ and $\mathcal{I} \models \mathbf{O} : a$ iff $\mathcal{I} \not\models a$. The payoff functions maps outcomes to $P = \{0, 1\}$, where $d(o) = 1$ iff $\mathcal{I} \models o$. P carries the usual ordering $0 < 1$ and $W = \{1\}$.

A strategy σ for *Me* in a game can be understood as *My* complete game plan. For every node of the underlying game tree labeled "I", σ tells *Me* to which node I have to move. Here is a formal definition:

Definition 2. *A* strategy σ *for* Me *for the game* $\mathbf{G}$ *is a subtree of the underlying tree such that (1) the root of* T *is in* σ *and for all* v *in* σ*, (2) if* $l(v) = I$*, then at least one successor of* v *is in* σ *and (3) if* $l(v) = Y$*, then all successors of* v *are in* σ*. A strategy for* You *is defined symmetrically. We denote by* Σ_I *and* Σ_Y *the set of all strategies for* Me *and* You*, respectively.*

Conditions (1) and (3) make sure that all possible moves by the other player are taken care of by the game plan. Each pair of strategies $\sigma_I \in \Sigma_I$, $\sigma_Y \in \Sigma_Y$ defines a *unique* outcome of $\mathbf{G}$, denoted by $O(\sigma_I, \sigma_Y)$. We abbreviate $d(O(\sigma_I, \sigma_Y))$ by $d(\sigma_I, \sigma_Y)$. A strategy σ_I^* for *Me* is called *winning* if, playing according to this strategy, I win the game, no matter how *You* move, i.e. for all

$\sigma_Y \in \Sigma_Y$, $d(\sigma_I^*, \sigma_Y) \in W$. Let $k \in \Lambda$. A strategy σ_I^k for *Me* guaranteeing a payoff of at least k, i.e. $\min^{\preceq}_{\sigma_Y}(\sigma_I^k, \sigma_Y) \succeq k$ is called a *k-strategy for Me.* A strategy for *You* guaranteeing a payoff of at most k is called a *k-strategy for You.* An outcome O that maximizes *My* pay-off in light of *Your* best strategy is called *maxmin-outcome.* Formally, O is a maxmin-outcome iff $d(O) = \max^{\preceq}_{\sigma_I} \min^{\preceq}_{\sigma_Y} d(\sigma_I, \sigma_Y)$ and $d(O)$ is called the *maxmin-value* of the game. A strategy σ_I^* for *Me* is a *maxmin-strategy* for $\mathbf{G}$ if $\sigma_I^* \in \arg\max^{\preceq}_{\sigma_I} \min^{\preceq}_{\sigma_Y} d(\sigma_I, \sigma_Y)$, i.e., the maximum is reached at σ_I^*. *Minmax* values and strategies for *You* are defined symmetrically.

The class of games that we have defined falls into the category of *zero-sum games of perfect information* in game theory. They are characterized by the fact that the players have strictly opposing interests. In these games, the minimax and maximin values always coincide and are referred to as the *value of the game.*

2.2 Game Choice Logics GCL

We now define the game semantics for GCL [10]. Let $\mathbf{Q} \in \{\mathbf{P}, \mathbf{O}\}$. The game over the interpretation $\mathcal{I}$ starting with *Me* in the role $\mathbf{Q}$ of formula F is denoted by $\mathbf{NG}(\mathbf{Q} : F, \mathcal{I})$. The game tree for the *semantic game* $\mathbf{NG}(\mathbf{Q} : F, \mathcal{I})$ is the same as in Hintikka's game, where $\vec{\times}$ is treated like $\vee$.

The main difference is that we now wish to deal with preferences induced by $\vec{\times}$. *My* preferences are expressed via the strict partial order $\ll$ on outcomes of the game tree: If $\mathbf{P} : G_1 \vec{\times} G_2$ appears in the tree, then outcomes reachable from $\mathbf{P} : G_1$ are in $\gg$-relation with outcomes reachable from $\mathbf{P} : G_2$. Similarly, for $\mathbf{O} : G_1 \vec{\times} G_2$, outcomes reachable from $\mathbf{O} : G_1$ are in $\ll$-relation with outcomes reachable from $\mathbf{O} : G_2$.

A sensible payoff function must respect both truth (winning conditions) and preferences (the relation $\ll$). Our payoff function $\delta_{\mathcal{I}}$ takes values in the domain $Z := (\mathbb{Z} \setminus \{0\}, \trianglelefteq)$. The ordering $\trianglelefteq$ is the inverse ordering on $\mathbb{Z}^+$ and on $\mathbb{Z}^-$, for $a \in \mathbb{Z}^+, b \in \mathbb{Z}^-$ we set $b \vartriangleleft a$, i.e. $-1 \vartriangleleft -2 \vartriangleleft \dots 2 \vartriangleleft 1$. For each outcome o, let $\pi_{\ll}(o)$ be the longest $\ll$-chain starting in o, i.e. pairwise different outcomes $o_1, \dots, o_n$ such that $o = o_1 \ll \dots \ll o_n$. Let $|\pi_{\ll}(o)| = n$ denote its length. For an interpretation $\mathcal{I}$, and an outcome $\mathbf{Q} : a$, we define[1]

$$\delta_{\mathcal{I}}(\mathbf{Q} : a) = \begin{cases} |\pi_{\ll}(\mathbf{Q} : a)|, & \text{if } \mathcal{I} \models \mathbf{Q} : a, \\ -|\pi_{\gg}(\mathbf{Q} : a)|, & \text{if } \mathcal{I} \not\models \mathbf{Q} : a. \end{cases}$$

By design, $\delta_{\mathcal{I}}$ maps true outcomes to $\mathbb{Z}^+$ and false outcomes to $\mathbb{Z}^-$. We, therefore, declare all outcomes with a payoff in $\mathbb{Z}^+$ as winning, and all other outcomes as losing for *Me*. The game can be thus seen as a refined extension of Hintikka's game. Indeed, let F^* be F with all $\vec{\times}$s replaced by $\vee$s. Then I have a winning strategy for $\mathbf{NG}(\mathbf{P} : F, \mathcal{I})$ iff I have a winning strategy for F^* in Hintikka's game over $\mathcal{I}$. Furthermore, $\delta_{\mathcal{I}}$ respects the relation $\ll$: if $o_1 \ll o_2$ and both are winning (or both are losing) for *Me*, then $\delta_{\mathcal{I}}(o_1) \vartriangleleft \delta_{\mathcal{I}}(o_2)$.

[1] Notice the flipped $\ll$-sign in the second case.

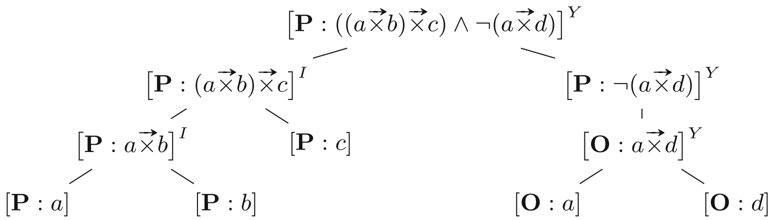


Fig. 1. The game tree for $\mathbf{NG}(\mathbf{P} : ((a \overrightarrow{\times} b) \overrightarrow{\times} c) \wedge \neg(a \overrightarrow{\times} d))$.

Example 1. Consider the formula $((a \overrightarrow{\times} b) \overrightarrow{\times} c) \wedge \neg(a \overrightarrow{\times} d)$. The game tree, where *I* am initially the Proponent can be found in Fig. 1. The order on outcomes is $\mathbf{P} : c \ll \mathbf{P} : b \ll \mathbf{P} : a$ and $\mathbf{O} : a \ll \mathbf{O} : d$.

Let $\mathcal{I} = \{b\}$. If *You* go to the left at the root node, *I* will move to reach the outcome $\mathbf{P} : b$, winning the game with payoff 2. Therefore, *You* might choose to go right at the root to reach $\mathbf{O} : a$ or $\mathbf{O} : d$ with payoff 2 and 1 respectively. It is better for *You* to reach $\mathbf{O} : a$ with payoff 2. Thus, the value of the game is 2.

Now consider the game starting in $\mathbf{O} : ((a \overrightarrow{\times} b) \overrightarrow{\times} c) \wedge \neg(a \overrightarrow{\times} d)$, again with $\mathcal{I} = \{b\}$. The game tree is the same, except that **P** and **O** are flipped everywhere, as are the labels I, Y and the order over outcomes. *You* can now win the game: if *I* go left at the root, *You* will move to $\mathbf{O} : b$ with payoff -2. The alternative is not better for *Me*: if *I* go right, *I* can choose between $\mathbf{P} : a$ and $\mathbf{P} : d$ with payoffs -1 and -2 respectively. Thus, the value of this game is -2.

2.3 Degree-Based Semantics for GCL

Although the motivation for GCL is game-theoretic, it also admits a degree semantics that is more common in choice logics. We first need the following notion of optionality:

Definition 3. *The optionality of GCL-formulas is defined inductively as follows: (i)* $\mathrm{opt}(a) = 1$ *for variables a, (ii)* $\mathrm{opt}(\neg F) = \mathrm{opt}(F)$*, (iii)* $\mathrm{opt}(F \circ G) = \max(\mathrm{opt}(F), \mathrm{opt}(G))$ *for* $\circ \in \{\vee, \wedge\}$*, and (iv)* $\mathrm{opt}(F \overrightarrow{\times} G) = \mathrm{opt}(F) + \mathrm{opt}(G)$.

In [10], we show that $\mathrm{opt}(F)$ computes the length of the longest $\ll$-chain in the outcomes reachable from $\mathbf{P} : F$ in the semantic game. The degree function of GCL is denoted by $\deg^{\mathcal{G}}_{\mathcal{I}}$.[2] It assigns to each formula a degree relative to an interpretation $\mathcal{I}$ and is defined inductively as follows:

$$\deg^{\mathcal{G}}_{\mathcal{I}}(a) = 1 \text{ if } a \in \mathcal{I}, -1 \text{ otherwise}$$
$$\deg^{\mathcal{G}}_{\mathcal{I}}(\neg F) = -\deg^{\mathcal{G}}_{\mathcal{I}}(F)$$
$$\deg^{\mathcal{G}}_{\mathcal{I}}(F \wedge G) = \min(\deg^{\mathcal{G}}_{\mathcal{I}}(F), \deg^{\mathcal{G}}_{\mathcal{I}}(G))$$

[2] The superscript $\mathcal{G}$ is used to differentiate from the standard degree function $\deg_{\mathcal{I}}$ of QCL used in the literature.

$$\deg^{\mathcal{G}}_{\mathcal{I}}(F \vee G) = \max(\deg^{\mathcal{G}}_{\mathcal{I}}(F), \deg^{\mathcal{G}}_{\mathcal{I}}(G))$$

$$\deg^{\mathcal{G}}_{\mathcal{I}}(F \vec{\times} G) = \begin{cases} \deg^{\mathcal{G}}_{\mathcal{I}}(F) & \text{if } \deg^{\mathcal{G}}_{\mathcal{I}}(F) \in \mathbb{Z}^+ \\ \text{opt}(F) + \deg^{\mathcal{G}}_{\mathcal{I}}(G) & \text{if } \deg^{\mathcal{G}}_{\mathcal{I}}(F) \in \mathbb{Z}^-, \\ & \deg^{\mathcal{G}}_{\mathcal{I}}(G) \in \mathbb{Z}^+ \\ \deg^{\mathcal{G}}_{\mathcal{I}}(F) - \text{opt}(G) & \text{otherwise} \end{cases}$$

Here min and max are relative to $\trianglelefteq$. If $\deg^{\mathcal{G}}_{\mathcal{I}}(F) \in \mathbb{Z}^+$ then we say that $\mathcal{I}$ classically satisfies F, or that $\mathcal{I}$ is a model of F. A model $\mathcal{I}$ of F is *preferred*, if for every other model $\mathcal{I}'$ of F we have $\deg^{\mathcal{G}}_{\mathcal{I}}(F) \trianglerighteq \deg^{\mathcal{G}}_{\mathcal{I}'}(F)$.

Theorem 4 (Theorem 4.7 in [10]). *The value of* $\mathbf{NG}(\mathbf{P} : F, \mathcal{I})$ *is* $\deg^{\mathcal{G}}_{\mathcal{I}}(F)$. *The value of* $\mathbf{NG}(\mathbf{O} : F, \mathcal{I})$ *is* $-\deg^{\mathcal{G}}_{\mathcal{I}}(F)$.

3 A Provability Game

Usually, in a semantic view of a logic, validity of a formula F is defined as truth of F in all interpretations. In our context of graded truth, however, we can refine this notion to *graded validity*. Thus, we define the *degree (of validity)* of F to be the least possible degree of F in an interpretation:

$$\deg^{\mathcal{G}}(F) := \min_{\mathcal{I}} \deg^{\mathcal{G}}_{\mathcal{I}}(F)$$

In this section, we give a game-theoretic characterization of this degree. To this end, we lift the semantic game **NG** to a provability game that adequately characterizes validity in GCL. Our framework will be able to deal with the following central notion in preference handling:

Definition 5. *An interpretation* $\mathcal{I}$ *is a* preferred model of F *iff* $\deg^{\mathcal{G}}_{\mathcal{I}}(F) \in \mathbb{Z}^+$ *and for all other interpretations* $\mathcal{J}$, $\deg^{\mathcal{G}}_{\mathcal{J}}(F) \trianglelefteq \deg^{\mathcal{G}}_{\mathcal{I}}(F)$.

We now describe the lifting of **NG** to the provability game **DG** (we call our game *disjunctive game*). We want a winning strategy for *Me* for the provability game starting at g to imply the existence of winning strategies in *all* semantic games starting at g. Note that the game trees of g over different interpretations $\mathcal{I}$ are identical, except for the payoff at outcomes. Therefore, a simultaneous play can be modeled by changing the pay-off at outcomes o to be the worst possible pay-off of o under all interpretations: $\delta(o) = \min_{\mathcal{I}}(o)$.

However, this variant does not capture validity yet, as *I* do not have winning strategies for this game even for simple cases, like $\mathbf{P} : a \vee \neg a$. This variant of the game is too restrictive, as it would require the existence of a *uniform strategy* – a single strategy that works in all semantic games. To remedy this shortcoming, we allow *Me* to create "backup copies" of game states during the provability game. If the game is unfavorable for *Me* in one copy, *I* can always come back to have another shot at the other copy. *My* goal is to win at least one of these copies. The game states of this game can be thus read as disjunctions, and are therefore called *disjunctive game states*[3], (hence the name of the game).

[3] To avoid confusion, we always refer to game states of the disjunctive game **DG** as "*disjunctive (game) states*". "*(Game) states*" is reserved for the semantic game **NG**.

Formally, game states of the disjunctive game are finite multisets of game states of the game **NG**. We prefer to write $g_1 \bigvee ... \bigvee g_n$ for the disjunctive game state $D = \{g_1, ..., g_n\}$, but keep the convenient notation $g \in D$ if g belongs to the multiset D. A disjunctive state is called *elementary* if all its game states are leaf-states of **NG**. Following the intuition of backup states, the payoff at an elementary disjunctive state D is the maximum of the payoffs of its game states:

$$\delta(D) = \min_{\mathcal{I}} \max_{1 \le i \le n} \delta_{\mathcal{I}}(g_i).$$

Additionally, *I* take the rule of a scheduler who decides which of the copies is played upon next.

At the disjunctive state $D \bigvee g$, *I* can point at a non-leaf state g, codified by underlining: $D \bigvee \underline{g}$. After the corresponding player takes their turn in $\mathbf{NG}(g_i, \mathcal{I})$ and moves to a state, say g', the game continues with $D \bigvee g'$.

As mentioned, instead of pointing to a game state of the disjunctive state, *I* can *duplicate* any of its states, i.e. create a backup copy. If *I* decide to duplicate g, the game continues with $D \bigvee g \bigvee g$. Due to this rule, it is now possible to have infinite runs of the game. In these runs, *I* repeatedly create backup copies. To prevent such behavior, we punish the "delaying" of the game by declaring infinite runs losing for *Me* with the worst possible pay-off -1.

Formally, we define the game tree of the disjunctive game $\mathbf{DG}(D, \mathcal{I})$ recursively as follows. We say that $D' \bigvee \underline{g}$ is obtained from $D = D' \bigvee g$ by *underlining* a game state and $D \bigvee g \bigvee g$ is obtained by *duplicating* a game state. If no states in D are underlined, it is an "I"-disjunctive state and its successor nodes are all disjunctive states obtainable by underlining, or duplicating a game state. If a game state is underlined, say we are in $D = D' \bigvee \underline{g}$, then this disjunctive state is labeled the same as g in the semantic game. The children of D are all $D' \bigvee g'$, where g' ranges over the children of g in the semantic game. For example, if $D = D' \bigvee \underline{\mathbf{P} : G_1 \vee G_2}$, then it is an "I"-disjunctive state and its children are $D' \bigvee \mathbf{P} : G_1$ and $D' \bigvee \mathbf{P} : G_2$.

Example 2. Let F be $((a \vec{\times} b) \vec{\times} c) \wedge \neg(a \vec{\times} b)$. Figure 2 shows a compact representation of a strategy for *Me* for the game $\mathbf{DG}(\mathbf{O} : F)$. Underlining moves are clear from context and are therefore hidden. First, *I* duplicate $\mathbf{O} : F$ and move to $\mathbf{P} : ((a \vec{\times} b) \vec{\times} c)$ in one copy and to $\mathbf{O} : \neg(a \vec{\times} d)$ in the other. The latter is immediately converted to $\mathbf{P} : a \vec{\times} d$, for which *I* repeat the strategy of duplicating and moving into both options. Finally, *I* point to $\mathbf{O} : (a \vec{\times} b) \vec{\times} c$, where it is *Your* turn. All *Your* possible choices are shown in the strategy. The payoffs are

$$\begin{aligned}
\delta(\mathbf{O} : a \bigvee \mathbf{P} : a \bigvee \mathbf{P} : d) &= \min_{\mathcal{I}} \max\{\delta_{\mathcal{I}}(\mathbf{O} : a), \delta_{\mathcal{I}}(\mathbf{P} : a), \delta_{\mathcal{I}}(\mathbf{P} : d)\} \\
&= \max\{\delta_{\emptyset}(\mathbf{O} : a), \delta_{\emptyset}(\mathbf{P} : a), \delta_{\emptyset}(\mathbf{P} : d)\} = \max\{3, -2, -1\} = 3, \\
\delta(\mathbf{O} : b \bigvee \mathbf{P} : a \bigvee \mathbf{P} : d) &= \min_{\mathcal{I}} \max\{\delta_{\mathcal{I}}(\mathbf{O} : b), \delta_{\mathcal{I}}(\mathbf{P} : a), \delta_{\mathcal{I}}(\mathbf{P} : d)\} \\
&= \max\{\delta_{\{b\}}(\mathbf{O} : b), \delta_{\{b\}}(\mathbf{P} : a), \delta_{\{b\}}(\mathbf{P} : d)\} = \max\{-2, -2, -1\} = -2, \\
\delta(\mathbf{O} : c \bigvee \mathbf{P} : a \bigvee \mathbf{P} : d) &= \min_{\mathcal{I}} \max\{\delta_{\mathcal{I}}(\mathbf{O} : c), \delta_{\mathcal{I}}(\mathbf{P} : a), \delta_{\mathcal{I}}(\mathbf{P} : d)\}
\end{aligned}$$

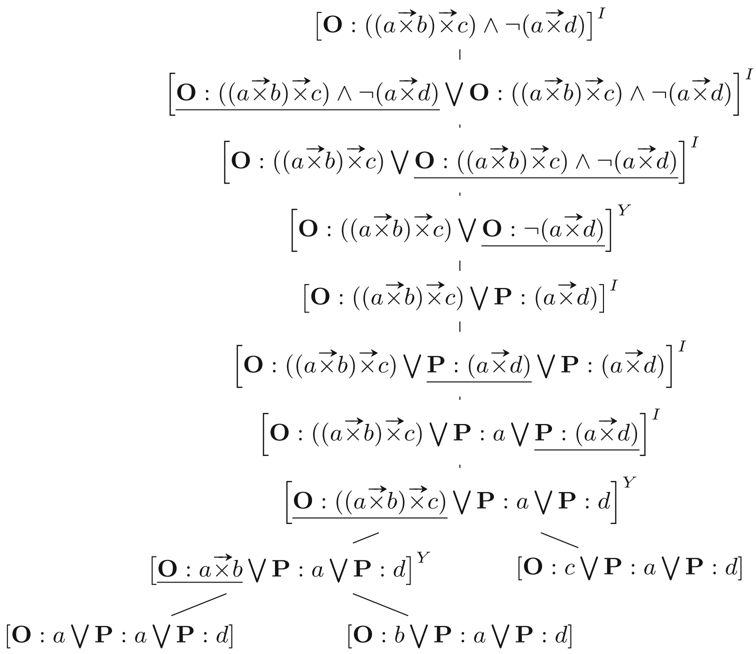


Fig. 2. A compact representation of the strategy for *Me* for an instance of **DG**

$$= \max\{\delta_{\{c\}}(\mathbf{O} : c), \delta_{\{c\}}(\mathbf{P} : a), \delta_{\{c\}}(\mathbf{P} : d)\} = \max\{-3, -2, -1\} = -3.$$

Given these payoffs, *You* prefer the second outcome, giving *Me* a payoff of -2. We note two things. First, *I* cannot do better by playing another strategy. If the outcomes do not contain game states resulting from $\mathbf{O} : (a \vec{\times} b) \vec{\times} c$, then their pay-offs are the same, or even less. The strategy of first duplicating, then exploiting all possible moves is therefore – in a way – optimal for *Me*. Hence, we can conclude that the value of the game is -2.

The remainder of this section is devoted to proving the adequacy of **DG**.

Theorem 6. I *have a k-strategy in* $\mathbf{DG}(D)$ *iff for every interpretation* $\mathcal{I}$, *there is some* $g \in D$ *such that* I *have a k-strategy in* $\mathbf{NG}(g, \mathcal{I})$. You *have a k-strategy in* $\mathbf{DG}(D)$ *iff there is an interpretation* $\mathcal{I}$ *such that* You *have k-strategies in* $\mathbf{NG}(g, \mathcal{I})$, *for all* $g \in D$.

We prove the above theorem with the help of two lemmas.

Lemma 7. *Let* π *be a finite run of the game* $\mathbf{DG}(D)$ *such that for every I-state* g *in* π, *all of its children appear in* π, *too. Let* k *be the payoff of* π. *Then there is a model* $\mathcal{I}_0$ *such that* You *have a k-strategy for* $\mathbf{NG}(g, \mathcal{I}_0)$, *for each* $g \in D$.

Proof (of Lemma 7). Let D_{fin} be the outcome of π and let $\mathcal{I}_0$ be such that $\delta_{\mathcal{I}_0}(D_{fin}) \trianglelefteq k$. For $g_0 \in D$, let σ_{g_0} be the set of successors of g appearing in

π. Note that σ_{g_0} carries the structure of a subtree of $\mathbf{NG}(g_0, \mathcal{I})$. Indeed, it is a strategy for *You*: by assumption, for every I-state in σ_{g_0}, all successors appear in π, and thus in σ_{g_0}. Every Y-state g in σ_{g_0} comes from a disjunctive state $D' \bigvee g$ appearing in π. At some point, $D' \bigvee \underline{g}$ is in π. The next disjunctive state in π is $D' \bigvee g'$, so g' is the unique successor of the Y-state g in σ_{g_0}.

To verify that σ_{g_0} is a k-strategy, it is enough to notice that all outcomes o in σ_{g_0} appear in D_{fin}. Thus, $\delta_{\mathcal{I}'}(o) \trianglelefteq \max_{h \in D} \delta_{\mathcal{I}_0}(h) \trianglelefteq \delta_{\mathcal{I}_0}(D_{fin}) \trianglelefteq k$. □

Lemma 8. *Let σ be a strategy for* Me *for $\mathbf{DG}(D_0)$ and let S be a set containing exactly one game state of each outcome of σ. Then for every interpretation $\mathcal{I}$, there is a strategy for* Me *for $\mathbf{NG}(g_0, \mathcal{I})$ with $g_0 \in D_0$ and outcomes in S.*

Proof (of Lemma 8). We define recursively for each $D \in \sigma$ a strategy σ^D for $\mathbf{NG}(g, \mathcal{I})$, where $g \in D$ and outcomes are in S. In the base case, D is an outcome of σ, so we set σ^D to be the singleton $S \cap D$.

If D is an I-state, and its unique child $H \in \sigma$ is obtained by duplicating or underlining a game state, we use the inductive hypothesis and set $\sigma^D = \sigma^H$. If $D = D' \bigvee \underline{g}$, where g is an I-state, then $H = D' \bigvee g'$, where g' is a child of g. If σ^H is a strategy for $\mathbf{NG}(h, \mathcal{I})$ with $h \in D'$, we can simply set $\sigma^D = \sigma^H$. Otherwise, σ^H is a strategy for $\mathbf{NG}(g', \mathcal{I})$. We can thus set $\sigma^D = \{g\} \cup \sigma^H$.

If D is a Y-state, then it is of the form $D = D' \bigvee \underline{g}$, where g is a Y-state. The children of D are of the form $D' \bigvee g'$, where g' ranges over the children of g. Since σ is a strategy for *Me*, all these children appear in σ. If for some g', $\sigma^{D' \bigvee g'}$ is a strategy for $\mathbf{NG}(h, \mathcal{I})$ and $h \in D'$, we can set $\sigma^D = \sigma^{D' \bigvee g'}$. Otherwise, all $\sigma^{D' \bigvee g'}$ are strategies for $\mathbf{DG}(g', \mathcal{I})$, and we can set $\sigma^D = \{g\} \cup \bigcup \sigma^{D' \bigvee g'}$.

In all the inductive steps it is clear that σ^D contains only outcomes from S. The claim follows for $D = D_0$. □

Proof (of Theorem 6). We prove the left-to-right directions (ltr) of both statements. The right-to-left directions (rtl) then follow easily: for example, suppose, for every $\mathcal{I}$, there is a $g \in D$, such that I have a k-strategy in $\mathbf{NG}(g, \mathcal{I})$. Let $l \triangleleft k$ be maximal. We infer that for every $\mathcal{I}$, there is some $g \in D$ such that *You* do not have a k-strategy in $\mathbf{NG}(g, \mathcal{I})$. By ltr of Statement 2, *You* do not have an l-strategy for $\mathbf{DG}(g, \mathcal{I})$. Since *You* cannot enforce the payoff to be below k, I have a k-strategy. The rtl of the other statement is similar.

Let us prove the ltr of Statement 1. Fix a k-strategy σ for *Me* in $\mathbf{DG}(D)$ and an interpretation $\mathcal{I}$. By assumption, for every outcome of the disjunctive game O in σ, there is a game state $o \in O$ such that $\delta_{\mathcal{I}}(o) \trianglerighteq k$. Collect for each outcome such an o into a set S. We apply Lemma 8 to obtain a strategy μ for *Me* for $\mathbf{NG}(g, \mathcal{I})$, for some $g \in D$ with outcomes in S. These outcomes have a payoff of at least k, i.e., μ is a k-strategy.

Ltr of Statement 2: suppose *You* have a k-strategy for $\mathbf{DG}(D)$. Let π be the run of the game where I play according to the following strategy: if the current disjunctive state is H, I underline an arbitrary $h \in H$. If h is an I-state and has only one child h', I go to that child in the corresponding copy. If h has two children h_1 and h_2, I first duplicate h, then go to h_1 in the first and to h_2 in the second copy. Let L be the outcome of π. By assumption, $\delta(L) \trianglelefteq k$. By Lemma 7, there is $\mathcal{I}$ such that *You* have k-strategies for $\mathbf{NG}(g, \mathcal{I})$, for each $g \in D$. □

Corollary 9. *The values of the games* $\mathbf{DG}(\mathbf{P} : F)$ *and* $\mathbf{DG}(\mathbf{O} : F)$ *are given by* $\deg^{\mathcal{G}}_{\mathcal{I}}(F) = \min_{\mathcal{I}} \deg^{\mathcal{G}}_{\mathcal{I}}(F)$ *and* $-\max_{\mathcal{I}} \deg^{\mathcal{G}}_{\mathcal{I}}(F)$*, respectively.*

Proof. For each interpretation $\mathcal{I}$, let $v_{\mathcal{I}}$ be the value of $\mathbf{DG}(D, \mathcal{I})$. It follows from the theorem that the value of $\mathbf{DG}(D)$ is $\min_{\mathcal{I}} v_{\mathcal{I}}$. Thus, by Theorem 4, the values of $\mathbf{DG}(\mathbf{P} : F)$ and $\mathbf{DG}(\mathbf{O} : F)$ are $\min_{\mathcal{I}} \deg^{\mathcal{G}}_{\mathcal{I}}(F)$ and $\min_{\mathcal{I}} - \deg^{\mathcal{G}}_{\mathcal{I}}(F)) = -\max_{\mathcal{I}} \deg^{\mathcal{G}}_{\mathcal{I}}(F)$, respectively. □

Corollary 10. *Let* $\mathcal{I}$ *be a preferred model of* F *and let* k *be the value of* $\mathbf{DG}(\mathbf{O} : F)$*. Then* $k = -\deg^{\mathcal{G}}_{\mathcal{I}}(F)$ *and a preferred model of* F *can be extracted from* Your *k-strategy for* $\mathbf{DG}(\mathbf{O} : F)$*.*

Proof. The first statement immediately follows from Corollary 9. Let σ be *Your* k-strategy for $\mathbf{DG}(\mathbf{O} : F)$. Since there is an interpretation making F true, k must be negative and thus winning for *You*. By the proof of Theorem 6, all the information for a preferred model is contained in the outcome of the run of the game, where *I* play according to the strategy sketched in that proof and *You* play according to σ. Let L be the outcome of that run. L must be winning for *You*. We, therefore, set $\mathcal{I}^{\pi} = \{a \mid \mathbf{O} : a \in L\}$ and obtain a k-strategy for *You* for $\mathbf{DG}(\mathbf{O} : F, \mathcal{I}^{\pi})$. Let v be the value of that game. We have that $v \trianglelefteq k$, by the existence of *Your* k-strategy and $v \trianglerighteq k$, since by Theorem 4 and Corollary 9, $v = -\deg^{\mathcal{G}}_{\mathcal{I}^{\pi}}(F) \trianglerighteq -\max_{\mathcal{I}} \deg^{\mathcal{G}}_{\mathcal{I}}(F) = k$.

This shows $\deg^{\mathcal{G}}_{\mathcal{I}^{\pi}}(F) = \max_{\mathcal{I}} \deg^{\mathcal{G}}_{\mathcal{I}}(F)$, i.e., $\mathcal{I}^{\pi}$ is a preferred model of F. □

4 Proof Systems

In this section, we study the proof-theoretic content of the provability game by reinterpreting strategies as proofs in three different labeled sequent calculi. Essentially, proofs in these systems are nothing but representations of *My* strategies for the disjunctive game. Sequents in these calculi consist of labeled formulas: each formula is decorated with two numbers $k, l \geq 1$ and we write ${}^{k}_{l}F$. The intuitive reading is that all winning outcomes of $\mathbf{Q} : F$ have a longest $\ll$-chain of at least l, and thus their payoff is at most l. Losing outcomes have a longest $\gg$-chain of at least k and thus their payoff is at least $-k$.

Sequents are of the form $\Gamma \Rightarrow \Delta$, where Γ and Δ are multisets of labeled formulas. There is a direct translation from disjunctive states of the game $\mathbf{DG}(D_0)$ into sequents: Each disjunctive state D is translated into the sequent

$$\{{}^{k}_{l}F \mid \mathbf{O} : F \in D\} \Rightarrow \{{}^{k}_{l}F \mid \mathbf{P} : F \in D\},$$

where $k = \min\{|\pi_{\gg}(o)| : o \in \mathcal{O}(\mathbf{Q} : F)\}$ and $l = \min\{|\pi_{\ll}(o)| : o \in \mathcal{O}(\mathbf{Q} : F)\}$. In particular, we have $k = l = 1$ if the game starts at $\mathbf{Q} : F$. We assign degrees to sequents $\Gamma \Rightarrow \Delta$ as follows: for each interpretation $\mathcal{I}$,

$$\text{if } {}^{k}_{l}F \in \Delta, \text{ we set } \deg^{\mathcal{G}}_{\mathcal{I}}({}^{k}_{l}F) = \begin{cases} l + \deg^{\mathcal{G}}_{\mathcal{I}}(F) - 1, & \text{if } \deg^{\mathcal{G}}_{\mathcal{I}}(F) \in \mathbb{Z}^{+}, \\ -k + \deg^{\mathcal{G}}_{\mathcal{I}}(F) + 1, & \text{if } \deg^{\mathcal{G}}_{\mathcal{I}}(F) \in \mathbb{Z}^{-}, \end{cases}$$

$$\text{if } {}^k_l F \in \Gamma \text{, we set } \deg^{\mathcal{G}}_{\mathcal{I}}({}^k_l F) = \begin{cases} l - \deg^{\mathcal{G}}_{\mathcal{I}}(F) - 1 & \text{if } \deg^{\mathcal{G}}_{\mathcal{I}}(F) \in \mathbb{Z}^-, \\ -k - \deg^{\mathcal{G}}_{\mathcal{I}}(F) + 1 & \text{if } \deg^{\mathcal{G}}_{\mathcal{I}}(F) \in \mathbb{Z}^+. \end{cases}$$

We then set

$$\deg^{\mathcal{G}}(\Gamma \Rightarrow \Delta) = \min_{\mathcal{I}} \max_{{}^k_l F \in \Gamma \cup \Delta} \deg^{\mathcal{G}}_{\mathcal{I}}({}^k_l F).$$

In the simplest case, $\deg^{\mathcal{G}}(\Rightarrow^1_1 F)$ coincides with $\deg^{\mathcal{G}}(F)$. We now have all ingredients to present our proof systems.

The first proof system, **GS** in Fig. 1, is closer to the game-theoretic view. Proofs are (bottom-up) representations of *My* strategies for the disjunctive game. What is unusual is that *all* sequents consisting of labeled propositional variables are allowed as initial sequents. A proof with all initial sequents of degree $\trianglerighteq k$, therefore, represents a k-strategy for *Me*. Hence, in this case, we speak of a k-*proof*. Note that in accordance with a k-strategy, k-proofs are not per se optimal: they merely witness that the degree of the proved sequent is at least k. In particular, every k proof is also an l-proof, if $k \trianglerighteq l$.

The second proof system is a proof-theoretically more orthodox system. In fact, it is actually a family of proof systems: for each $k \in Z$, the system $\mathbf{S}^k$ is defined in Fig. 1. These proof systems share all the rules with **GS**, but initial sequents are valid iff their degree is at least k. Such initial sequents are axioms in the usual sense.

The conceptual difference between the two approaches is as follows: in **GS**, the value k can be *computed* from the initial sequents. In the second approach, k is *guessed* (implicitly, by picking the proof system $\mathbf{S}^k$, for a concrete k).

Example 3. Figure 3 shows a derivation of $((a \vec{\times} b) \vec{\times} c) \wedge \neg(a \vec{\times} d) \Rightarrow$ in **GS**. Essentially, it is *My* strategy from Example 2 bottom-up. Degrees of initial sequents:

$$\begin{aligned} \deg^{\mathcal{G}}({}^3_1 a \Rightarrow^1_2 a, {}^2_1 d) &= \deg^{\mathcal{G}}_{\{a\}}({}^3_1 a \Rightarrow^1_2 a, {}^2_1 d) = 2, \\ \deg^{\mathcal{G}}({}^2_2 b \Rightarrow^1_2 a, {}^2_1 d) &= \deg^{\mathcal{G}}_{\{b\}}({}^2_2 b \Rightarrow^1_2 a, {}^2_1 d) = -2, \\ \deg^{\mathcal{G}}({}^1_3 c \Rightarrow^1_2 a, {}^2_1 d) &= \deg^{\mathcal{G}}_{\{c\}}({}^1_3 c \Rightarrow^1_2 a, {}^2_1 d) = -3. \end{aligned}$$

Therefore, the derivation is a -2-proof and thus a proof in $\mathbf{S}^{-2}$.

It follows directly from the translation of *My* strategies into proofs:

Theorem 11. *The following are equivalent:*

1. I *have a k-strategy for* $\mathbf{DG}(\mathbf{O} : F_1 \bigvee ... \bigvee \mathbf{O} : F_n \bigvee \mathbf{P} : G_1 \bigvee ... \bigvee \mathbf{P} : G_m)$.
2. $\deg^{\mathcal{G}}({}^1_1 F_1, ..., {}^1_1 F_n \Rightarrow^1_1 G_1, ..., {}^1_1 G_m) \trianglerighteq k$.
3. *There is a k-proof of* ${}^1_1 F_1, ..., {}^1_1 F_n \Rightarrow^1_1 G_1, ..., {}^1_1 G_m$ *in* **GS**.
4. *There is a proof of* ${}^1_1 F_1, ..., {}^1_1 F_n \Rightarrow^1_1 G_1, ..., {}^1_1 G_m$ *in* $\mathbf{S}^k$.

Corollary 12. *Let $k \in \mathbb{Z}^-$. Then there is a k-proof of ${}^1_1 F \Rightarrow$ in* **GS** *iff there is a proof of ${}^1_1 F \Rightarrow$ in $\mathbf{S}^k$ iff the degree of F in a preferred model is at most $-k$.*

Table 1. Proof systems **GS** and $\mathbf{S}^k$.

Initial Sequents for GS

$\Gamma \Rightarrow \Delta$, where Γ and Δ consist of labeled variables

Axioms for $\mathbf{S}^k$

$\Gamma \Rightarrow \Delta$, where $\deg^G(\Gamma \Rightarrow \Delta) \trianglerighteq k$, and Γ and Δ consist of labeled variables

Structural Rules

$$\frac{\Gamma,_l^k F,_l^k F \Rightarrow \Delta}{\Gamma,_l^k F \Rightarrow \Delta}(L_c) \qquad \frac{\Gamma \Rightarrow_l^k F,_l^k F, \Delta}{\Gamma \Rightarrow_l^k F, \Delta}(R_c)$$

Propositional rules

$$\frac{\Gamma,_l^k F \Rightarrow \Delta \quad \Gamma,_l^k G \Rightarrow \Delta}{\Gamma,_l^k (F \vee G) \Rightarrow \Delta}(L_\vee) \qquad \frac{\Gamma \Rightarrow_l^k F, \Delta}{\Gamma \Rightarrow_l^k (F \vee G), \Delta}(R_\vee^1)$$

$$\frac{\Gamma,_l^k F \Rightarrow \Delta}{\Gamma,_l^k (F \wedge G) \Rightarrow \Delta}(L_\wedge^1) \qquad \frac{\Gamma \Rightarrow_l^k G, \Delta}{\Gamma \Rightarrow_l^k (F \vee G), \Delta}(R_\vee^2)$$

$$\frac{\Gamma,_l^k G \Rightarrow \Delta}{\Gamma,_l^k (F \wedge G) \Rightarrow \Delta}(L_\wedge^2) \qquad \frac{\Gamma \Rightarrow_l^k F, \Delta \quad \Gamma \Rightarrow_l^k G, \Delta}{\Gamma \Rightarrow_l^k (F \wedge G), \Delta}(R_\wedge)$$

$$\frac{\Gamma \Rightarrow_l^k F, \Delta}{\Gamma,_l^k \neg F \Rightarrow \Delta}(L_\neg) \qquad \frac{\Gamma,_l^k F \Rightarrow \Delta}{\Gamma \Rightarrow_l^k \neg F, \Delta}(R_\neg)$$

Choice rules

$$\frac{\Gamma, {}_{l+\mathrm{opt}(G)}^{k} F \Rightarrow \Delta \quad \Gamma, {}_{l}^{k+\mathrm{opt}(F)} G \Rightarrow \Delta}{\Gamma, {}_{l}^{k} (F \vec{\times} G) \Rightarrow \Delta}(L_{\vec{\times}}) \qquad \frac{\Gamma \Rightarrow {}_{l}^{k+\mathrm{opt}(G)} F, \Delta}{\Gamma \Rightarrow {}_{l}^{k} (F \vec{\times} G), \Delta}(R_{\vec{\times}}^1)$$

$$\frac{\Gamma \Rightarrow {}_{l+\mathrm{opt}(G)}^{k} G, \Delta}{\Gamma \Rightarrow {}_{l}^{k} (F \vec{\times} G), \Delta}(R_{\vec{\times}}^2)$$

My strategy in Example 2, is not only a -2-strategy but also a minmax-strategy for *Me*. This implies that *I* cannot do better than -2, i.e. the value of the game is -2. How does this translate into the proof-theoretic interpretation of Example 3? There, the minmax-strategy takes the form of invertibility of rule applications: rule applications S'/S and $(S_1, S_2)/S$ are called *invertible* iff $\deg^{\mathcal{G}}(S') = \deg^{\mathcal{G}}(S)$ and $\min\{\deg^{\mathcal{G}}(S_1), \deg^{\mathcal{G}}(S_2)\} = \deg^{\mathcal{G}}(S)$. In Example 3 only invertible rule applications are used.

In Table 2 we give a calculus $\mathbf{GS}^*$ which is equivalent to **GS** but has only invertible rules, i.e. all rule applications are invertible. The contraction rules are admissible in this system. The motivation behind this calculus is the same as in *My* maxmin-strategy: in every I-state, *I* first duplicate and then exhaustively take all the available options. Every proof produced in this system corresponds

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{{}^{1}_{3}a \Rightarrow {}^{2}_{1}a, {}^{1}_{2}d \qquad {}^{2}_{2}b \Rightarrow {}^{2}_{1}a, {}^{1}_{2}d}{{}^{2}_{1}(a \overrightarrow{\times} b) \Rightarrow {}^{2}_{1}a, {}^{1}_{2}d}\,(L_{\overrightarrow{\times}}) \qquad {}^{3}_{1}c \Rightarrow {}^{2}_{1}a, {}^{1}_{2}d}{{}^{1}_{1}((a \overrightarrow{\times} b) \overrightarrow{\times} c) \Rightarrow {}^{2}_{1}a, {}^{1}_{2}d}\,(L_{\overrightarrow{\times}})}{{}^{1}_{1}((a \overrightarrow{\times} b) \overrightarrow{\times} c) \Rightarrow {}^{2}_{1}a, {}^{1}_{1}(a \overrightarrow{\times} d)}\,(R^{2}_{\overrightarrow{\times}})}{{}^{1}_{1}((a \overrightarrow{\times} b) \overrightarrow{\times} c) \Rightarrow {}^{1}_{1}(a \overrightarrow{\times} d), {}^{1}_{1}(a \overrightarrow{\times} d)}\,(R^{1}_{\overrightarrow{\times}})}{{}^{1}_{1}((a \overrightarrow{\times} b) \overrightarrow{\times} c) \Rightarrow {}^{1}_{1}(a \overrightarrow{\times} d)}\,(R_C)}{{}^{1}_{1}((a \overrightarrow{\times} b) \overrightarrow{\times} c), {}^{1}_{1}(\neg(a \overrightarrow{\times} d)) \Rightarrow}\,(L_\neg)}{{}^{1}_{1}((a \overrightarrow{\times} b) \overrightarrow{\times} c), {}^{1}_{1}(((a \overrightarrow{\times} b) \overrightarrow{\times} c) \wedge \neg(a \overrightarrow{\times} d)) \Rightarrow}\,(L_\wedge)}{{}^{1}_{1}(((a \overrightarrow{\times} b) \overrightarrow{\times} c) \wedge \neg(a \overrightarrow{\times} d)), {}^{1}_{1}(((a \overrightarrow{\times} b) \overrightarrow{\times} c) \wedge \neg(a \overrightarrow{\times} d)) \Rightarrow}\,(L_\wedge)}{{}^{1}_{1}(((a \overrightarrow{\times} b) \overrightarrow{\times} c) \wedge \neg(a \overrightarrow{\times} d)) \Rightarrow}\,(L_C)
$$

Fig. 3. A −2-proof in **GS**.

to an optimal strategy and has, therefore, an optimal degree. The below results follow directly from the invertibility of the rules:

Proposition 13. *Every* $\mathbf{GS}^*$*-proof of a sequent* S *has degree* $\deg^{\mathcal{G}}(S)$.

Corollary 14. *Let* $k = \deg^{\mathcal{G}}({}^{1}_{1}F \Rightarrow) \in \mathbb{Z}^-$. *Then the degree of* F *in a preferred model is equal to* $-k$. *Furthermore, a preferred model of* F *can be extracted from every* $\mathbf{GS}^*$*-proof of* ${}^{1}_{1}F \Rightarrow$.

Example 4. Figure 4 shows a $\mathbf{GS}^*$-proof of ${}^{1}_{1}((a \overrightarrow{\times} b) \overrightarrow{\times} c) \wedge \neg(a \overrightarrow{\times} d) \Rightarrow$. The proof is essentially a compact representation of the proof in Fig. 3, and has therefore degree −2. We conclude that in a preferred model, $((a \overrightarrow{\times} b) \overrightarrow{\times} c) \wedge \neg(a \overrightarrow{\times} d)$ has degree 2. Furthermore, we can extract the preferred model $\{b\}$ from the position where the $\deg^{\mathcal{G}}$-function is minimal on the initial sequents, as computed in Example 3.

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{{}^{1}_{3}a \Rightarrow {}^{2}_{1}a, {}^{1}_{2}d \qquad {}^{2}_{2}b \Rightarrow {}^{2}_{1}a, {}^{1}_{2}d}{{}^{2}_{1}(a \overrightarrow{\times} b) \Rightarrow {}^{2}_{1}a, {}^{1}_{2}d}\,(L_{\overrightarrow{\times}}) \qquad {}^{3}_{1}c \Rightarrow {}^{2}_{1}a, {}^{1}_{2}d}{{}^{1}_{1}((a \overrightarrow{\times} b) \overrightarrow{\times} c) \Rightarrow {}^{2}_{1}a, {}^{1}_{2}d}\,(L_{\overrightarrow{\times}})}{{}^{1}_{1}((a \overrightarrow{\times} b) \overrightarrow{\times} c) \Rightarrow {}^{1}_{1}(a \overrightarrow{\times} d)}\,(R_{\overrightarrow{\times}})}{{}^{1}_{1}((a \overrightarrow{\times} b) \overrightarrow{\times} c), {}^{1}_{1}\neg(a \overrightarrow{\times} d) \Rightarrow}\,(L_\neg)}{{}^{1}_{1}(((a \overrightarrow{\times} b) \overrightarrow{\times} c) \wedge \neg(a \overrightarrow{\times} d)) \Rightarrow}\,(L_\wedge)
$$

Fig. 4. A proof in $\mathbf{GS}^*$.

Table 2. The proof system $\mathbf{GS}^*$ for GCL with invertible rules.

Initial Sequents

$\Gamma \Rightarrow \Delta$, where Γ and Δ consist of labeled variables

Propositional rules

$$\frac{\Gamma,{}^k_l F \Rightarrow \Delta \quad \Gamma,{}^k_l G \Rightarrow \Delta}{\Gamma,{}^k_l (F \vee G) \Rightarrow \Delta}(L_\vee) \qquad \frac{\Gamma \Rightarrow^k_l F,{}^k_l G, \Delta}{\Gamma \Rightarrow^k_l (F \vee G), \Delta}(R_\vee)$$

$$\frac{\Gamma,{}^k_l F,{}^k_l G, \Rightarrow \Delta}{\Gamma,{}^k_l (F \wedge G) \Rightarrow \Delta}(L_\wedge) \qquad \frac{\Gamma \Rightarrow^k_l F, \Delta \quad \Gamma \Rightarrow^k_l G, \Delta}{\Gamma \Rightarrow^k_l (F \wedge G), \Delta}(R_\wedge)$$

$$\frac{\Gamma \Rightarrow^k_l F, \Delta}{\Gamma,{}^k_l \neg F \Rightarrow \Delta}(L_\neg) \qquad \frac{\Gamma,{}^k_l F \Rightarrow \Delta}{\Gamma \Rightarrow^k_l \neg F, \Delta}(R_\neg)$$

Choice rules

$$\frac{\Gamma,{}^{\;k}_{l+\mathrm{opt}(G)}F \Rightarrow \Delta \quad \Gamma,{}^{\;k+\mathrm{opt}(F)}_{l}G \Rightarrow \Delta}{\Gamma,{}^k_l(F \vec{\times} G) \Rightarrow \Delta}(L_{\vec{\times}})$$

$$\frac{\Gamma \Rightarrow {}^{k+\mathrm{opt}(G)}_{l}F,{}^{\;k}_{l+\mathrm{opt}(F)}G, \Delta}{\Gamma \Rightarrow {}^k_l(F \vec{\times} G), \Delta}(R_{\vec{\times}})$$

Note that the following degree-version of cut does not hold. The existence of k-strategies for $D \bigvee \mathbf{P} : F$ and $D \bigvee \mathbf{O} : F$ does not imply that a k-strategy for D exists. For example, note that the values of $\mathbf{O} : \top \bigvee \mathbf{O} : \bot \vec{\times} \top$ and $\mathbf{O} : \top \bigvee \mathbf{P} : \bot \vec{\times} \top$ are -2 and 2, respectively. But the value of the "conclusion" of the cut, $\mathbf{O} : \top$, has value -1.

What is more, there is no function computing the value of the conclusion of cut from the values of the premises. To see this, note that the values of $\mathbf{O} : \bot \vec{\times} \top \bigvee \mathbf{O} : \bot \vec{\times} \top$ and $\mathbf{O} : \bot \vec{\times} \top \bigvee \mathbf{P} : \bot \vec{\times} \top$ are -2 and 2 respectively, as in the above example. However, in contrast to the above example, the conclusion of this cut, $\mathbf{O} : \bot \vec{\times} \top$, has value -2.

Lastly, we demonstrate that $\mathbf{GS}$ and $\mathbf{S}^k$ are useful systems, i.e. that computing the degree of initial sequents is easier than the degree of general sequents.

Proposition 15. *Deciding whether* $\deg^{\mathcal{G}}(\Gamma \Rightarrow \Delta) \trianglerighteq k$ *is* coNP*-hard in general. If* $\Gamma \Rightarrow \Delta$ *is initial, then* $\deg^{\mathcal{G}}(\Gamma \Rightarrow \Delta)$ *can be computed in polynomial time.*

Proof. coNP-hardness of deciding $\deg^{\mathcal{G}}(\Gamma \Rightarrow \Delta) \trianglerighteq k$ follows by coNP-hardness of the validity problem in classical logic: if F is a classical formula, then it holds that $\deg^{\mathcal{G}}(\Rightarrow^1_1 F) \in \mathbb{Z}^+$ if and only if F is valid (true under all interpretations).

We now show that $\deg^{\mathcal{G}}(\Gamma \Rightarrow \Delta)$ can be computed in polynomial time if $\Gamma \Rightarrow \Delta$ is initial. We start with the empty interpretation $\mathcal{I} = \emptyset$. Now, go through every variable x occurring in $\Gamma \Rightarrow \Delta$. Consider $\Gamma_x \Rightarrow \Delta_x$ where ${}^l_k x \in \Gamma_x$ iff ${}^l_k x \in \Gamma$ and ${}^l_k x \in \Delta_x$ iff ${}^l_k x \in \Delta$. If we have $\deg^{\mathcal{G}}_{\{x\}}(\Gamma_x \Rightarrow \Delta_x) \lhd \deg^{\mathcal{G}}_{\emptyset}(\Gamma_x \Rightarrow \Delta_x)$

then let $\mathcal{I} = \mathcal{I} \cup \{x\}$, otherwise leave $\mathcal{I}$ unchanged. In other words, since $\Gamma \Rightarrow \Delta$ is initial, we can simply choose the 'better' option for any given variable x without side effects. Thus, this procedure gives us the minimal $\mathcal{I}$ for $\Gamma \Rightarrow \Delta$. □

5 Conclusion and Future Work

In this paper, we investigate the notion of validity in choice logics. Specifically, we lift a previously established [10] semantic game **NG** for the language of QCL to a provability game **DG**. This allows us to examine formulas with respect to *all* interpretations. Similar to truth, validity in choice logic comes in degrees. We show that the value of **DG** adequately models these validity degrees. Strategies for *Me* in **DG** correspond to proofs in an analytic labeled sequent calculi **GS**. The unique feature of this system is that its proofs have degrees that represent the degree of validity. We give two variants of **GS** – $\mathbf{GS}^*$ with invertible rules corresponding to *My* optimal strategy, and the more orthodox system $\mathbf{S}^k$ where proofs do not have degrees, but a "degree-profile" is guessed similar to [3].

For future work, it will be interesting to adapt **NG** to capture related logics such as Conjunctive Choice Logic [5] or Lexicographic Choice Logic [4], both of which introduce another choice connective in place of ordered disjunction. Using the methods established in this paper, provability games for these semantic games can then be derived. Indeed, our systems are quite modular in this sense, since most aspects of our provability game and our calculi require no adaptation if ordered disjunction were to be exchanged with another choice connective.

Acknowledgments. We thank all our anonymous reviewers for their valuable feedback. This work was funded by the Austrian Science Fund (FWF) under grants P32830 and P32684, the Vienna Science and Technology Fund (WWTF) under grant ICT19-065, and partially funded by the EU (Marie Skłodowska-Curie RISE) project MOSAIC, grant 101007624.

References

1. Benferhat, S., Sedki, K.: Alert correlation based on a logical handling of administrator preferences and knowledge. In: Fernández-Medina, E., Malek, M., Hernando, J. (eds.) SECRYPT 2008, Porto, Portugal, 26–29 July 2008, SECRYPT is part of ICETE, pp. 50–56. INSTICC Press (2008)
2. Benferhat, S., Sedki, K.: Two alternatives for handling preferences in qualitative choice logic. Fuzzy Sets Syst. **159**(15), 1889–1912 (2008). https://doi.org/10.1016/j.fss.2008.02.014
3. Bernreiter, M., Lolic, A., Maly, J., Woltran, S.: Sequent calculi for choice logics. In: Blanchette, J., Kovács, L., Pattinson, D. (eds.) IJCAR 2022. LNAI, vol. 13385, pp. 331–349. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-10769-6_20
4. Bernreiter, M., Maly, J., Woltran, S.: Choice logics and their computational properties. Artif. Intell. **311**, 103755 (2022). https://doi.org/10.1016/j.artint.2022.103755
5. Boudjelida, A., Benferhat, S.: Conjunctive choice logic. In: ISAIM 2016, Fort Lauderdale, Florida, USA, 4–6 January 2016 (2016)

6. Brewka, G., Benferhat, S., Berre, D.L.: Qualitative choice logic. Artif. Intell. **157**(1–2), 203–237 (2004). https://doi.org/10.1016/j.artint.2004.04.006
7. Brewka, G., Niemelä, I., Syrjänen, T.: Logic programs with ordered disjunction. Comput. Intell. **20**(2), 335–357 (2004). https://doi.org/10.1111/j.0824-7935.2004.00241.x
8. Fermüller, C.G., Metcalfe, G.: Giles's game and the proof theory of Łukasiewicz logic. Stud. Logica. **92**(1), 27–61 (2009). https://doi.org/10.1007/s11225-009-9185-2
9. Freiman, R.: Games for hybrid logic. In: Silva, A., Wassermann, R., de Queiroz, R. (eds.) WoLLIC 2021. LNCS, vol. 13038, pp. 133–149. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-88853-4_9
10. Freiman, R., Bernreiter, M.: Truth and preferences - a game approach for qualitative choice logic. In: M-PREF@IJCAI (2022). https://doi.org/10.48550/arXiv.2209.12777
11. Hintikka, J.: Logic, Language-Games and Information: Kantian Themes in the Philosophy of Logic. Clarendon Press, Oxford (1973)
12. Liétard, L., Hadjali, A., Rocacher, D.: Towards a gradual QCL model for database querying. In: Laurent, A., Strauss, O., Bouchon-Meunier, B., Yager, R.R. (eds.) IPMU 2014. CCIS, vol. 444, pp. 130–139. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08852-5_14
13. Pavlova, A., Freiman, R., Lang, T.: From semantic games to provability: the case of Gödel logic. Stud. Logica. **110**(2), 429–456 (2021). https://doi.org/10.1007/s11225-021-09966-x
14. Onyepunuka, U., Alassad, M., Nwana, L., Agarwal, N.: Multilingual Analysis of YouTube's Recommendation System: Examining Topic and Emotion Drift in the 'Cheng Ho' Narrative (2023)
15. Sedki, K., Lamy, J., Tsopra, R.: Learning preferences in prioritized qualitative choice logic. In: 32nd IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2020, Baltimore, MD, USA, 9–11 November 2020, pp. 368–375. IEEE (2020). https://doi.org/10.1109/ICTAI50040.2020.00065
16. Sedki, K., Lamy, J., Tsopra, R.: Qualitative choice logic for modeling experts recommendations of antibiotics. In: Barták, R., Keshtkar, F., Franklin, M. (eds.) Proceedings of the Thirty-Fifth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2022, Hutchinson Island, Jensen Beach, Florida, USA, 15–18 May 2022 (2022). https://doi.org/10.32473/flairs.v35i.130677

Aleatoric Propositions: Reasoning About Coins

Tim French(✉)

School of Physics, Mathematics and Computing, The University of Western Australia, Perth, WA, Australia
tim.french@uwa.edu.au

Abstract. Aleatoric propositions are a generalisation of Boolean propositions, that are intrinsically probabilistic, or determined by the toss of a (biased) coin. Rather than let propositions take a true/false valuation, we assume they act as a biased coin, that will sometimes land heads (*true*), and sometimes land tails (*false*). Complex propositions then correspond to a conditional series of tosses of these coins. We extend the syntax and semantics for Aleatoric Logic to include a novel fixed-point operator that is able to represent a weak form of iteration. We examine the expressivity of the of the language, showing a correspondence to classes of rational functions over $(0, 1)$

Keywords: Probabilistic Reasoning · Expressivity · Correspondence Theory

1 Introduction

Logic is the study of truth and deduction. In both philosophical and mathematical contexts, a logic represents a reasoning process, where true statements are composed via some rules to infer new truths. However, the commitment to the study of *true* statements sets a very high bar for reasoning. In everyday live we are beset with uncertainty and absolute truth can rarely be assumed. Nonetheless, reason persists and we are able to act rationally and perform deductions within the bounds of our uncertainty. Furthermore, we are able to do this without ever quantifying our uncertainty: some facts are simply recognised on being contingent on things outside of our experience. As we accrue experience, our confidence in our judgements increases as does our trust in our reasoning, even though the standard of absolute truth is never attained. This is the experiential logic described by Hume [8].

As automated reasoning and artificial intelligence become more capable, there is a need for a foundation for reasoning and logic that is tolerant of the uncertainty that we find in every day life. This paper presents such a formalism: an analogue of propositional logic where the true false statements are replaced by independent probabilistic events: what may thought of as tosses of a biased coin. A coin toss is intrinsically uncertain, so while the proposition has a correspondence to the coin, we suppose that an agent only has access to this proposition via

H. H. Hansen et al. (Eds.): WoLLIC 2023, LNCS 13923, pp. 227–243, 2023.
https://doi.org/10.1007/978-3-031-39784-4_14

the sampling of tosses of the coin. Therefore, aleatoric propositions are defined over a series of events, where atomic events correspond to the toss of some biased coin. The complex expressions therefore describe coin-flipping protocols where the next coin chosen to flip is contingent on the outcome of the prior coin flips. Particularly, we introduce a fixed-point operator that is able to represent looping coin flipping protocols.

1.1 Related Work

There has been a considerable number of works that have considered probabilistic semantics. We will briefly cover some here and attempt to categorise them in relation to this work.

Early work includes Kolmogorov's [10] axiomatization, Ramsey's [14] and de Finetti's [2] characterisation of subjective probability, These works provide a foundation for what constitutes a probability distribution, and Hailperin [5] gives a good overview of early work.

There have been a number of works applying probabilistic elements to automated reasoning and deduction, including probabilistic description logic [12], reasoning about uncertainty [6], verification of randomised programs [11]. These approaches include a modality for the probability of some event occurring. As the probabilities are explicit in the syntax, these approaches reason about the probabilities of events, in the sense that while a pair of dice landing as two ones (snake eyes) is an uncertain statement, the statement "snake eyes has probability $\frac{1}{36}$" is a Boolean (true/false) statement. Gardenfors [4] has approached the topic of probability logics in a similar way, axiomatising a logic of relative probability, which contains statements such as "A is at least as probable as B".

We are interested in reasoning probabilistically, without necessarily quantifying probabilities. In this vein also are the Fuzzy approaches to probabilistic reasoning [16]. Fuzzy logic is typically applied to describe the concept of vagueness, where the semantics allow for the increasing or decreasing of plausibility, without necessarily committing to the absolute certainty of a proposition. When the *product semantics* are used (so the plausibility of two propositions taken together are multiplied) there is a natural correspondence with independent random events like the flipping of coins. While this has many similarities to our approach, it does not have the fixed-point operator capable of expressing events of unbounded magnitude.

This work is an extension of the propositional *aleatoric calculus* presented in [3] which also considered the modal extension of the logic (not considered here) but also did not include the fixed point operator discussed here.

1.2 Overview

The following section will present the syntax and semantics for aleatoric propositions, and provide some discussion of the novel operators. The next section will consider the expressivity of the language, and give some illustrative examples, such as the expression of fractional probabilities, and conditional reasoning. The main result of the paper is to show that the aleatoric propositions include the set of all rational functions that map products of the closed interval $[0,1]$ to the open interval $(0,1)$. This result is based on work by Mossel [13]. Finally we will discuss some remaining open problems, and briefly discuss how we may extend the system to talk about dependent events.

2 Syntax and Semantics

Here we present a minimal syntax and semantics for *Aleatoric Propositions*, extending the aleatoric calculus presented in [3]. Aleatoric propositions are a generalisation of Boolean propositions, and are defined over a set of atomic propositions $\mathfrak{P}$. To avoid confusion, we will not refer to propositions being *true* or *false*, but rather consider them as descriptions of sets of events (e.g. a coin landing head side up) that occurs with some probability. For this reason we use symbols $\bullet$ (*heads*) and $\circ$ (*tails*) for atoms corresponding to probability 1 (always heads) and probability 0 (always tails) respectively.

A complex proposition describes events comprised of sub-events. For example, given propositions A and B, we may consider a proposition which corresponds to events where the A-coin lands heads, and then the B-coin lands heads, *and* then the A-coin (tossed for a second time) lands heads once more.

As we consider a proposition describing the occurrence of an event, we can also consider a proposition describing an event *failing* to occur. Note, in this context, we consider the failure of an event meaning the event is explicitly tested, and that test fails. So given an event "the penny lands heads", the negation would be "the penny is flipped and and does not land heads", but it would not be "the penny is not flipped" or "the quarter is flipped and it lands heads".

The propositions can also describe conditional events: we could consider a proposition describing an event where the A-coin has the same result two times in a row, so *if* the A-coin lands heads, then it is tossed again, and lands heads again, but *if* it landed tails the first time, it is tossed again and lands tails the second time.

Finally, we also consider iterative events, where propositions are repeatedly sampled until come condition is met. A famous example is the scheme devised by von Neumann [15] to simulate a fair coin (with bias precisely 1/2) using any coin with probability in $(0,1)$. Here the coin is flipped twice: if we see a head followed by a tail, we report a (synthetic) head; if we see a tail followed by a head we report a (synthetic) tail; and otherwise, if we see two heads or two tails, we repeat the process. Since each coin flip is independently sampled with the same probability regardless of the bias the likelihood of the synthetic head is the same as the likelihood of the synthetic tail,

These complex propositions are built using a ternary *if-then-else* operator, a negation operator, and a *fixed-point* operator.

Definition 1. *The* syntax of aleatoric propositions *is given by:*

$$\alpha ::= \bullet \mid A \mid \neg\alpha \mid (\alpha?\alpha:\alpha) \mid \mathbb{F}X\alpha$$

where $A, X \in \mathfrak{P}$*, and* X *is* linear in α*. The set of aleatoric propositions is denoted as* $\mathcal{L}$*.*

We let $\mathrm{var}(\alpha)$ be the set of atomic propositions appearing in α, and say $X \in \mathrm{var}(\alpha)$ is *free in* α ($X \in \mathrm{free}(\alpha)$) if X does not appear in the scope of a $\mathbb{F}X$ operator. If $X \in \mathrm{var}(\alpha)$ does appear in the scope of a $\mathbb{F}X$ operator, we say X is *bound* in α ($X \in \mathrm{bnd}(\alpha)$).

The fixed point operator required that X *is linear in* α, and this is to ensure that the fixed point is unique and non-ambiguous.

Definition 2. *An atomic proposition* $X \in \mathfrak{P}$ is linear in α *if and only if for every subformula* $(\beta?\gamma_1 : \gamma_2)$ *of* α*, there is no occurrence of* X *appearing in* β*.*

The meaning and significance of linearity will be discussed once the semantics have been presented.

A brief description of these operators is as follows:

- $\bullet$ (*heads*) describes an event that invariably occurs (i.e. a coin that always lands heads, or a double headed coin).
- A is an *atom* that describes an event that occurs with some probability $p \in (0, 1)$.
- $\neg\alpha$, (*not* α) describes the *failure* of an event to occur. That is, the event is explicitly tested for, and that test fails.
- $(\alpha?\beta : \gamma)$ (*if* α *then* β *else* γ) describes a *conditional* event where the event described by α is tested (or sampled) and if it occurs, then an event described by β occurs, but if the alpha event does not occur, then an event described by γ occurs.
- $\mathbb{F}X\alpha$ (X *where* $X = \alpha$) is the *fixed point* proposition, and it describes an event with probability x such that if the event corresponding to the atom X had likelihood x, so would α. An alternative way to consider this operation is a description of a recursive event, so that whenever the proposition, X, is to be tested instead a test of the proposition $\mathbb{F}X\alpha$ is (recursively) substituted. If this process continues forever, (e.g., in the evaluation of $\mathbb{F}XX$, its value is deemed to be $\frac{1}{2}$. See the proof of Lemma 1 for a discussion of this.

As an example of this syntax, we can represent the scheme of von Neumann, mentioned above, as:

$$\textit{fair-coin} = \mathbb{F}X(A?(A?X : \bullet) : (A?\circ : X)).$$

We note that the notion of uncertainty is intrinsic in these operators. That an atomic proposition A "happened" does not mean that A is *true* in the common sense: we could repeat the process and the subsequent event satisfies $\neg A$. The propositions, themselves are mercurial and transient, so it does not make sense to say a proposition *is* true. Rather, we are interested in the *probability* that an event described by the proposition *will occur*, and this is what is presented in the following semantics.

The semantics of these operators is given over an *interpretation*, $\mathcal{I}$, which assigns a probability between 0 and 1 to every atomic proposition.

Definition 3. *An* interpretation *for propositional aleatoric logic is a function* $\mathcal{I} : \mathfrak{P} \longrightarrow (0,1)$. *Given an interpretation,* $\mathcal{I}$*, an atomic proposition* $X \in \mathfrak{P}$ *and some* $p \in (0,1)$*, we let the interpretation* $\mathcal{I}[X : p]$ *be such that for all* $Y \in \mathfrak{P}\backslash\{X\}$*,* $\mathcal{I}[X : p](Y) = \mathcal{I}(Y)$ *and* $\mathcal{I}[X : p] = p$.

We will use the notation $\alpha[X\backslash\beta]$ to represent the proposition α with all free occurrences of X in α replaced by β, and we say β *is free of* X *in* α if for every free variable Y in β, X is not in the scope of an operator $\mathbb{F}Y$ in α. We can now define the semantics as below[1]:

Definition 4. *Given an interpretation,* $\mathcal{I}$*, and some aleatoric proposition* α*, the interpretation assigns the probability* $\mathcal{I}(\alpha)$ *inductively as follows:*

$$\begin{aligned}
\bullet^{\mathcal{I}} &= 1\\
A^{\mathcal{I}} &= \mathcal{I}(A)\\
(\neg\alpha)^{\mathcal{I}} &= 1 - \alpha^{\mathcal{I}}\\
(\alpha?\beta{:}\gamma)^{\mathcal{I}} &= \alpha^{\mathcal{I}} \cdot \beta^{\mathcal{I}} + (1 - \alpha^{\mathcal{I}}) \cdot \gamma^{\mathcal{I}}\\
(\mathbb{F}X\alpha)^{\mathcal{I}} &= \begin{cases} 1 \textit{ if } \alpha^{\mathcal{I}} = 1\\ 0 \textit{ if } \alpha^{\mathcal{I}} = 0\\ x \textit{ if } x \textit{ is the unique value such that } \alpha^{\mathcal{I}[X:x]} = x\\ 1/2 \textit{ if } \forall x \in (0,1),\ \alpha^{\mathcal{I}[X:x]} = x \end{cases}
\end{aligned}$$

We must show that the semantic interpretation of the fixed point operator is well defined; that is, the fixed point always exists and has uniquely defined value.

Lemma 1. *The semantic interpretation of the fixed point operator is well defined. Given any* α *where* X *is linear in* α*, given any interpretation* $\mathcal{I}$*, either* $\alpha^{\mathcal{I}} \in \{0,1\}$*, or there is a unique* $x \in (0,1)$ *such that* $\alpha^{\mathcal{I}[X:x]} = x$*, or for every* $x \in (0,1)$*,* $\alpha^{\mathcal{I}[X:x]} = x$*.*

Proof. This proof will be given by induction over the complexity of formulas, and will also provide an alternative semantic definition for the fixed point operator.

The induction hypothesis is, for every aleatoric proposition $\alpha \in \mathcal{L}$, for each atomic proposition X where X is linear in α, given any interpretation $\mathcal{I}$, there are unique values h_α, $i_\alpha^X \in [0,1]$ such that $\alpha^{\mathcal{I}[X:x]} = h_\alpha + i_\alpha^X \cdot x$. For simplicity we will assume that free(α) and bnd(α) are disjoint sets. The induction is given over the complexity of formulas as follows:

- for $\psi = \bullet$, $h_\psi = 1$ and $i_\psi^X = 0$.
- for $\psi = A \in$ var, where $A \notin$ bnd(α), let $h_\psi = \mathcal{I}(A)$ and $i_\psi^X = 0$. Since $A^{\mathcal{I}[X;x]} = \mathcal{I}(A)$, it is clear that $h_\psi = \mathcal{I}(A)$ and $i_\psi^X = 0$ are the only values that satisfy the induction hypothesis.
- for $\psi = X$, where $X \in$ bnd(α), let $h_\psi = 0$ and $i_\psi^X = 1$. Since $X^{\mathcal{I}[X:x]} = x$, it is clear that $h_\psi = 0$ and $i_\psi^X = 1$ are the only values that satisfy the induction hypothesis.

[1] We use the notation where given the probabilities x and y, $x \cdot y$ is interpreted as the *product* of x and y.

- for $\psi = (\beta?\gamma_1:\gamma_2)$, $h_\psi = h_\beta \cdot h_{\gamma_1} + (1-h_\beta)\cdot h_{\gamma_2}$, and $i^X_\psi = h_\beta \cdot i^X_{\gamma_1} + (1-h_\beta)\cdot i^X_{\gamma_2}$. Since $\mathcal{I}(\psi) = \mathcal{I}(\beta)\cdot\mathcal{I}(\gamma_1) + (1-\mathcal{I}(\beta))\cdot\mathcal{I}(\gamma_2)$, it follows that $h_\psi = h_\beta \cdot h_{\gamma_1} + (1-h_\beta)\cdot h_{\gamma_2}$ and $i^X_\psi = h_\beta \cdot i^X_{\gamma_1} + (1-h_\beta)\cdot i^X_{\gamma_2}$, noting that as X is linear in α, by the induction hypothesis$i^X_\beta = 0$. The values for h_ψ and i^X_ψ are unique since these calculations are deterministic.
- for $\psi = \neg\beta$, let $h_\psi = 1 - h_\beta$, $i^X_\psi = -i^X_\beta$. This derivation follows from the induction hypothesis; as $\beta^{\mathcal{I}[X:x]}$ is described with respect to x by the function $\beta^{\mathcal{I}[X:x]} = h_\beta + i^X_\beta \cdot x$, it follows that

$$1 - \beta^{\mathcal{I}[X:x]} = 1 - (h_\beta + i^X_\beta \cdot x) = (1-h_\beta) + (-i^X_\beta)\cdot x)$$

  as required.
- for $\psi = \mathbb{F}Y\beta$, $h_\psi = \frac{h_\beta}{1-i^Y_\beta}$ or $h_\psi = 1/2$ if $i^Y_\beta = 1$, and $i^X_\psi = \frac{i^X_\beta}{1-i^Y_\beta}$ or 0 if $i^Y_\beta = 1$. For any y, we have $\beta^{\mathcal{I}[X:x,Y:y]} = h^X_\beta + i^X_\beta \cdot x + i^Y_\beta \cdot y$, noting the linearity of both X and Y in β. As $(\mathbb{F}Y\beta)^{\mathcal{I}[X:x]} = y$ where $y = h^X_\beta + i^X_\beta \cdot x + i^Y_\beta \cdot y$, solving for y, provided $i^Y_\beta \neq 1$, the unique solution $(\mathbb{F}Y\beta)^{\mathcal{I}[X:x]} = \frac{h^X_\beta}{1-i^Y_\beta} + \frac{i^X_\beta \cdot x}{1-i^Y_\beta}$ gives the definition of h^X_ψ and i^X_ψ. If $i^Y_\beta = 1$, the it must be the case that $h_\beta = 0$ and $\beta^{\mathcal{I}[Y:y]} = y$. Therefore, for every $y \in (0,1)$ we have $\beta^{\mathcal{I}[Y:y]} = y$ so the fixed point semantics gives $(\mathbb{F}Y\beta)^{\mathcal{I}[X:x]} = 1/2$. It follows that $h_\psi = 1/2$ and $i^X_\psi = 0$.

These definitions are complete and deterministic, and from the induction hypothesis, it follows $(\mathbb{F}X\alpha)^{\mathcal{I}} = \frac{h_\alpha}{1-i^X_\alpha}$ or $\frac{1}{2}$ if $i^X_\alpha = 1$. In either case, the semantic interpretation of the fixed point operator is well defined.

The fixed point is a genuine fixed point and the fact that in the formula $\mathbb{F}X\alpha$, X is always linear in α means that the fixed point is always unique (see Fig. 1, which also demonstrates $\neg\mathbb{F}X\alpha(X) = \mathbb{F}X\neg\alpha(\neg X)$). This gives an alternative semantic formulation of the fixed point operator, and a convenient way to visualise the fixed point as the intersection point of a line with intercept h_α and gradient i^X_α with the line with intercept 0 and gradient 1. This also motivates, Definition 2, where X is linear in α if α is a linear function of the interpretation of X, when all other arguments of α are fixed.

Definition 5. *Given an interpretation* $\mathcal{I}$, *and some* $\alpha \in \mathcal{L}$ *the* functional semantics *for propositional aleatoric logic assigns a value* $h_\alpha \in [0,1]$ *and a value* i^X_α

for each $X \in \text{bnd}(\alpha)$ *as follows:*

$$
\begin{array}{lll}
\psi = \bullet: & h_\psi = 1 & i_\psi^X = 0 \\
\psi = A \in \text{free}(alpha): & h_\psi = \mathcal{I}(A) & i_\psi^X = 0 \\
\psi = X \in \text{bnd}(\alpha): & h_\psi = 0 & i_\psi^X = 1 \\
\psi = Y \in \text{bnd}(\alpha): & h_\psi = 0 & i_\psi^X = 0 \\
\psi = (\alpha?\beta:\gamma): & h_\psi = h_\alpha \cdot h_\beta + (1-h_\alpha)\cdot h_\gamma & i_\psi^X = h_\alpha \cdot i_\beta^X + (1-h_\alpha)\cdot i_\gamma^X \\
\psi = \neg\alpha: & h_\psi = 1 - h_\alpha & i_\psi^X = -i_\alpha^X \\
\psi = \mathbb{F}X\alpha,\ i_\alpha^X \neq 1: & h_\psi = \frac{h_\alpha}{1-i_\alpha^X} & i_\psi^X = 0 \\
\psi = \mathbb{F}X\alpha,\ i_\alpha^X = 1: & h_\psi = 1/2 & i_\psi^X = 0 \\
\psi = \mathbb{F}Y\alpha,\ i_\alpha^Y \neq 1: & h_\psi = \frac{h_\alpha}{1-i_\alpha^Y} & i_\psi^X = \frac{i_\alpha^X}{1-i_\alpha^Y} \\
\psi = \mathbb{F}Y\alpha,\ i_\alpha^Y = 1: & h_\psi = 1/2 & i_\psi^X = 0
\end{array}
$$

Corollary 1. *For any proposition* $\alpha \in \mathcal{L}$*, and any interpretation* $\mathcal{I}$*, we have* $\alpha^{\mathcal{I}} = h_\alpha$*, where* h_α *is given in Definition 5.*

This corollary follows directly from the proof of Lemma 1

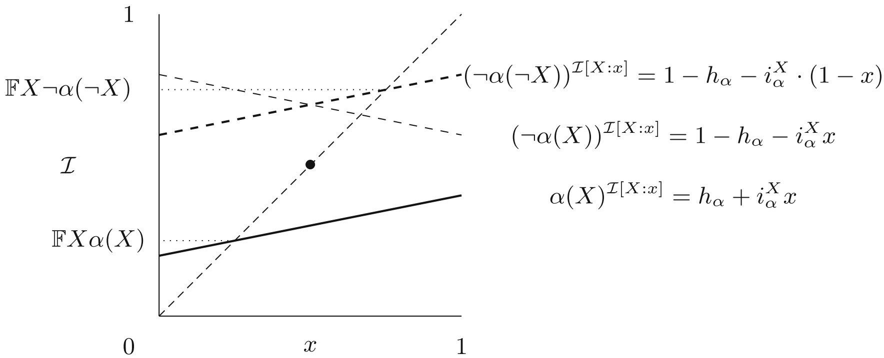


Fig. 1. The semantic interpretation of $\mathbb{F}X\alpha(X)$, showing how the value of $\mathbb{F}X\alpha$ corresponds to $\alpha^{\mathcal{I}[X:x]}$ with respect to x. The lower thick line is the function $\alpha(X)^{\mathcal{I}[X:x]}$, with intercept h_α and gradient i_α^X. The thick dashed line corresponds to the function $\neg\alpha(\neg X)^{\mathcal{I}[X:x]}$. The fixed point of each function is the point where the line crosses the diagonal.

2.1 Abbreviations

Within these semantics we may define conventional logic operators where the semantics loosely align with fuzzy logic using the product t-norm [16].

Table 1 contains some useful abbreviations. Note the notion of frequency in this set of abbreviations. The formula $\alpha^{\frac{n}{m}}$ or α at least n out of m times refers to the event that α when sampled m times, α occurred at least n times. This does

Table 1. Some useful abbreviations for aleatoric propositions.

Abbreviation	Expression	Description
○	¬●	*tails*
◐	$\mathbb{F}XX$	*fair coin flip*
$\alpha \wedge \beta$	$(\alpha?\beta:$○$)$	α *and* β
$\alpha \vee \beta$	$(\alpha?$●$:\beta)$	α *or* β
$\alpha \rightarrow \beta$	$(\alpha?\beta:$●$)$	α *implies* β
$\alpha \leftrightarrow \beta$	$(\alpha?\beta:\neg\beta)$	α *if and only if* β
$\alpha^{\frac{0}{m}}$	●	α 0 *out of* m *times.*
$\alpha^{\frac{n}{0}}$	○	α n *out of* 0 *times* $(n > 0)$.
$\alpha^{\frac{n}{m}}$	$(\alpha?\alpha^{\frac{n-1}{m-1}}:\alpha^{\frac{n}{m-1}})$	α *at least* n *out of* m *times*

not suggest that probability of α is at least $\frac{n}{m}$. It simply describes an event: if $\Pr(\alpha) = 0.1$ then $\alpha^{\frac{10}{10}}$ is simply a very unlikely event. In experiential logic, this gives a proxy for truth: $\alpha^{\frac{100}{100}}$ is the case only when we are very confident in α.

2.2 Motivation and Discussion

With the semantics established and shown to be well-founded it is worth taking some time to motivate the semantics choices made. We will consider the following motivating example:

Example 1. Suppose that Venus and Serena are playing a game of tennis, and are involved in a tie break. The tie break works by Venus serving first, then Serena serving twice, and then Venus serving twice, and so on, until one of them is two points ahead of the other. In tennis, it is often supposed the server has the advantage, so we let V be the probability Venus wins on her serve, and S be the probability that Serena wins on her serve. Then Venus winning the tie break can be represented as the following aleatoric proposition:

$$\mathbb{F}X(V?(S?(S?(V?X:\bigcirc):(V?\bullet:X)):\bullet):(S?\bigcirc:(S?(V?X:\bigcirc):(V?\bullet:X))))$$

Applying Corollary 1 (and several algebraic reductions) we can describe the probability of Venus winning the tie-break as a function of V and S:

$$\mathit{VenusWins}(V, S) = \frac{V - S \cdot V}{S + V - 2 \cdot S \cdot V} \quad (1)$$

with the contour plot given in Fig. 2.

This first example is effectively a representation of a Markov decision process. Where the probabilities correspond to discrete events. However, aleatoric propositions can also be used to represent situations with unknown variables, such as “Venus has an injury”, which are either true or false, but unknown to the reasoner. The following example demonstrates how aleatoric propositions could be applied to these epistemic variables

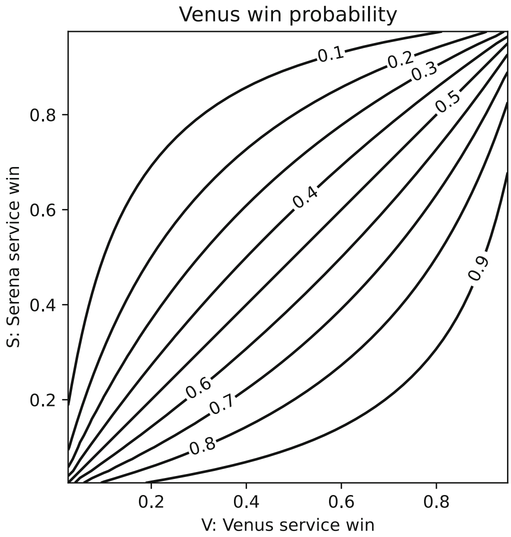


Fig. 2. A contour diagram of the probability of Venus winning a tie break (1, given Venus has probability V of winning on her own serve, and a probability of $(1 - S)$ of winning on Serena's serve.

Example 2. It is Alex's turn to get dinner and Blake is speculating what Alex may do. If Alex is not too tired, a home cooked meal is likely, but if Alex can afford it, Alex may (50%) order dinner from a food delivering service.

Blake considers Alex's current state base on discussion they have had during the week, and imagines *alex_ tired* and *alex_ rich* as two coins with biases reflecting Blake's assessment of Alex's current state. Blake is then able to synthesise a new coin representing whether there will be a home cooked meal:

$$home\_cooked = (alex\_rich?(◐?\neg alex\_tired : \bot) : \neg alex\_tired)$$

In this example, if Alex is rich, there is a 50% chance that they will get a food delivery service. Otherwise, if Alex is not too tired Alex will cook a home cooked meal. Note that it is possible to similarly devise a coin for whether they will order dinner from a service, but the outcomes of the coins will not necessarily be mutually exclusive, nor necessarily sum to 1 (if Alex is poor and tired breakfast cereal may be an option).

The use of aleatoric propositions can be thought of as mental simulations. Given an agent's experiences, they may imagine how the world might be: this is a test. Blake can ponder "Is Alex rich", and he may imagine it to be so or not. These mental simulations reflect Blake's belief. The structure of an aleatoric proposition describes this mental simulation process: what beliefs are considered and in what order.

3 A Correspondence for Aleatoric Propositions

In this section we will investigate the expressivity of aleatoric propositions, and give a correspondence result. The correspondence is based on earlier work by Keane and O'Brien [9] and Elchanan Mossel and Yuval Peres [13]. Keane and O'Brien originally showed that "Bernoulli factories", which are essentially coin flipping protocols that transform one probability to another[2], can simulate any continuous polynomial bounded function over $(0,1)$, and Mossel and Peres showed that with some restrictions on the coin flipping protocols, the resulting set of functions correspond to the set of rational functions over $(0,1)$. We will follow Mossel's and Peres's presentation here.

To be precise, we can consider a proposition $\alpha \in \mathcal{L}$ to be a function that given an interpretation, $\mathcal{I}$, returns a probability $\alpha^{\mathcal{I}}$. In turn, the interpretation, $\mathcal{I}$, with respect to α is simply an assignment from $\text{var}(\alpha)$ to $(0,1)$, so any aleatoric proposition, α, may be considered as a function from $(0,1)^{var(\alpha)}$ to $(0,1)$. To characterise the expressivity of aleatoric propositions, we will first describe the set of *rational functions* from $(0,1)^{\text{var}(\alpha)}$ to $(0,1)$. The next subsection will express the semantics of aleatoric propositions as functions, and establish a normal form for aleatoric propositions, showing every aleatoric proposition corresponds to a rational function. The final subsection completes the correspondence by showing that every rational function from $(0,1)^{\mathcal{X}}$ to $(0,1)$ agrees with the semantics of some aleatoric proposition defined over the atomic propositions $\mathcal{X}$.

In this section we suppose that $\alpha \in \mathcal{L}$ is an aleatoric proposition where $\text{var}(\alpha) = \{X_1, \ldots, X_n\} = \mathcal{X}$, and $f_\alpha : (0,1)^{\mathcal{X}} \longrightarrow (0,1)$ is a function such that $f_\alpha(X_1 \mapsto X_1^{\mathcal{I}}, \ldots, X_n \mapsto X_n^{\mathcal{I}}) = \alpha^{\mathcal{I}}$.

In general, rational functions over $\mathcal{X}$ maybe thought of as fractions of polynomials.

Definition 6. *A* rational function of degree k *from* $(0,1)^{\mathcal{X}}$ *to* $[0,1]$ *is a function of the form:*

$$f(\overline{x}) = \frac{\sum_{\overline{a}\in\sigma^k_{\mathcal{X}})} \ell_{\overline{a}} \prod_{x\in\mathcal{X}} x^{a_x}}{\sum_{\overline{a}\in\sigma^k_{\mathcal{X}}} m_{\overline{a}} \prod_{x\in\mathcal{X}} x^{a_x}}$$

where $\sigma^k_{\mathcal{X}} = \{\overline{a} \in \{0,\ldots,k\}^{\mathcal{X}} \mid \sum_{x\in\mathcal{X}} a_x = k\}$ *for all* $\overline{a} \in \sigma^k_{\mathcal{X}}$*,* $\ell_{\overline{a}}, m_{\overline{a}} \in \mathbb{Z}$*, and for all* $\overline{x} \in (0,1)^{\mathcal{X}}$*,* $f(\overline{x}) \in (0,1)$*.*

3.1 Aleatoric Functions

In this subsection we will define a special form for aleatoric propositions, and through a series of semantically invariant transformations, show that every aleatoric proposition can be represented in this form.

In this section we will suppose that we are dealing with formulas consisting of the free variables $A_1, \ldots, A_n$, and the fixed point variables $X_1, \ldots, X_m$, which are disjoint with the free variables.

The definition of *block normal form* for aleatoric propositions is as follows.

[2] For example by twice flipping a coin with bias p, we construct an event with probability p^2.

Definition 7. *A formula of aleatoric propositional logic is in* k-block normal form *if it satisfies the following syntax for* γ*:*

$$
\begin{aligned}
\alpha_1^0 &::= \bullet \mid \circ \mid X_0 \\
\alpha_i^{j+1} &::= (A_i ? \alpha_i^j : X_0) \mid (A_i ? X_0 : \alpha_i^j) \\
\alpha_{i+1}^1 &::= (A_{i+1} ? \alpha_i^k : X_0) \mid (A_{i+1} ? X_0 : \alpha_i^k) \\
\beta_0 &::= \alpha_n^k \\
\beta_{i+1} &::= (\text{◐} ? \beta_i : \beta_i) \\
\gamma &::= \mathbb{F} X_0 \beta_\ell
\end{aligned}
$$

A representation of a formula in block form is given in Fig. 3. Here, the formula $\mathbb{F}X(\neg(A \wedge B) \rightarrow (A \wedge X))$ is converted to the 2-block normal form. As the 2-block normal form uses the conditional statements $(\alpha ? \beta : \gamma)$, where α is guaranteed to be either a propositional atom A_i or ◐, we use the convention of drawing the formulas as a tree where β is on the left branch and γ is on the right branch. The cut off branches at the α_i^j levels are shorthand for X_0. Similarly, from the definition of k-block normal form the leaves are all labelled with $\bullet$, $\circ$ or X_0.

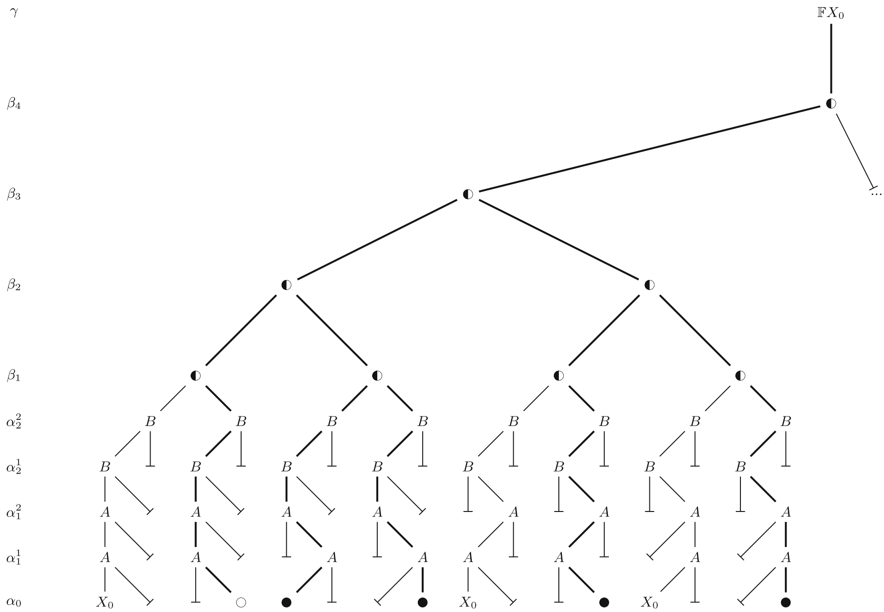


Fig. 3. A 2-block normal form representation of the formula $\mathbb{F}X(\neg(A \wedge B) \rightarrow (A \wedge X))$, Each internal node represents a proposition $(A_i ? \alpha_1 : \alpha_2)$ where α_1 is the left child, and α_2 is the right child, except for the root which is a fixed point proposition. The entire tree is repeated again as the right branch of the node at level β_3.

Lemma 2. *Every aleatoric proposition, where each distinct free variable occurs at most k times is semantically equivalent to an aleatoric proposition in k-block-form. Specifically, there is a function $\tau : \mathcal{L} \longrightarrow \mathcal{L}$, such that:*

1. *for all $\alpha \in \mathcal{L}$, $\tau(\alpha)$ is in k-block-form.*
2. *for all interpretations, $\mathcal{I}$, $\mathcal{I}(\alpha) = \mathcal{I}(\tau(\alpha))$.*

Proof. The proof is given by construction where we give a set of semantically valid transformations that: push negations down to only apply in the context of abbreviation○; modify conditional statements $(\alpha?\beta:\gamma)$ so that α is either some free variable (A_i) or ◐; modify conditional statements $(A_i?\beta:\gamma)$ so that either β or γ or both are X_0; order the atomic propositions, so in the subformulas $(A_i?\beta:\gamma)$, β and γ can only contain free variables A_j where $j < i$; and combine all fixed point operators into a single fixed point operator at the highest level.

This is achieved through the following transformations, that preserve the interpretation of the formulas. We write $\alpha \Rightarrow \beta$ to indicate that $\alpha^{\mathcal{I}} = \beta^{\mathcal{I}}$, and the form of α is a defect that needs to be corrected to move into the normal form.

1. To move negations to occur only in the context ○, we note:
   - $\neg\mathbb{F}X\alpha(X) \Rightarrow \mathbb{F}X\neg\alpha(\neg X)$ (see Fig. 1);
   - $\neg(\alpha?\beta:\gamma) \Rightarrow (\alpha?\neg\beta:\neg\gamma)$;
   - $(\neg\alpha?\beta:\gamma) \Rightarrow (\alpha?\gamma:\beta)$;
   - $(\alpha?\neg A_i:\beta) \Rightarrow (\alpha?(A_i?\bigcirc:\bullet):\beta)$.

   These can all be checked with basic algebraic reductions.
2. To ensure the internal branching nodes are only free variables or instances of ◐, we apply the following transformations:
   - $((\alpha?\beta_1:\beta_2)?\gamma_1:\gamma_2) \Rightarrow (\alpha?(\beta_1?\gamma_1:\gamma_2):(\beta_2?\gamma_1:\gamma_2))$, when β_1 and β_2 are not bound variables;
   - $((\alpha?X:\beta)?\gamma_1:\gamma_2) \Rightarrow (\alpha?X:(\beta?\gamma_1:\gamma_2))$;
   - $((\alpha?\beta:X)?\gamma_1:\gamma_2) \Rightarrow (\alpha?(\beta?\gamma_1:\gamma_2):X)$;
   - $(\mathbb{F}X\alpha(X)?\beta:\gamma) \Rightarrow \mathbb{F}X(\alpha?\beta:\gamma)$, under the assumption that X does not appear free in β or γ (or is renamed to a fresh variable if it does).
3. The previous two transformations are sufficient to give a tree structure, where subformulas $(\alpha?\beta:\gamma)$ are such that $\alpha = A_i$ or $\alpha =$ ◐. The next defect to address is fixed points appearing anywhere other than the root. To address this we apply the transformations:
   - $(\alpha?\mathbb{F}X\beta:\gamma) \Rightarrow \mathbb{F}X(\alpha?\beta:\beta[\bullet,\bigcirc\backslash\gamma])$;
   - $(\alpha?\beta:\mathbb{F}X\gamma) \Rightarrow \mathbb{F}X(\alpha?\gamma[\bullet,\bigcirc\backslash\beta]:\gamma)$.

   The idea of this transformation is to move the fixed point operator to the root of the conditional statement, so that when X is encountered (i.e. α was heads, and the evaluation of β was X), the entire statement is re-evaluated from the root. This would make favour the branch containing γ, so that branch is similarly scaled by including the evaluation of β, but with every occurrence of ● or ○ replaced by γ. Effectively, this ensures that both branches are equally likely to be reevaluated from the root, so no branch is advantaged. To show this is the case, we apply the function semantics given in Definition 5:

$$
\begin{aligned}
(\mathbb{F}X(\alpha?\beta:\beta[\bullet,\circ\backslash\gamma]))^{\mathcal{I}} &= \frac{h_\alpha \cdot h_\beta + (1-h_\alpha)\cdot(1-i_\beta^X)\cdot h_\gamma}{1-(h_\alpha \cdot i_\beta^X + (1-h_\alpha)\cdot i_\beta^X)} \\
&= \frac{h_\alpha \cdot h_\beta + (1-h_\alpha)\cdot(1-i_\beta^X)\cdot h_\gamma}{1-h_\alpha \cdot i_\beta^X - i_\beta^X + h_\alpha \cdot i_\beta^X} \\
&= \frac{h_\alpha \cdot h_\beta + (1-h_\alpha)\cdot(1-i_\beta^X)\cdot h_\gamma}{1-i_\beta^X} \\
&= h_\alpha \cdot \frac{h_\beta}{1-i_\beta^X} + (1-h_\alpha)\cdot h_\gamma \\
&= (\alpha?\mathbb{F}X\beta:\gamma)^{\mathcal{I}}
\end{aligned}
$$

The second reduction is shown in a similar manner. Note that the first line of the derivation used the property that $(\beta[\bullet,\circ\backslash\gamma])^{\mathcal{I}} = (1-i_\beta^X)\cdot h_\gamma$, which assumes that all defects have been already removed from β.

4. Having enforced a tree structure, and moved all fixed point operators to the root, the next defect to address is to ensure that, for all conditional statements, $(\alpha?\beta:\gamma)$, either $\alpha = \text{◐}$ or one of β or γ is X_0. To do this, given $\alpha = A_i$ we can apply the following transformation:

$$(A_i?\beta:\gamma) \Rightarrow \mathbb{F}X_0(\text{◐}?(A_i?\beta:X_0):(A_i?X_0:\gamma)).$$

This transformation uses a fair coin flip a fixed point operators to turn the conditional statement into a series of independent tests. Essentially, a fair coin is flipped to see whether we test the case where A_i is heads or the case where A_i is tails. In each instance we first test if the hypothesis is right (e.g. A_i is heads) and if it is not, we repeat the process. If A_i is heads, we then continue to test β, and similarly for when the fair coin lands tails, we apply a similar process to test, given that A_i is tails, γ. Using the notation of Lemma 1 we have, when $\psi = (\text{◐}?(A_i?\beta:X_0):(A_i?X_0:\gamma))$, $h_\psi = (h_\alpha \cdot h_\beta + (1-h_\alpha)\cdot h_\gamma)/2$, and $i_\psi^{X_0} = 1/2$. As $h_{\mathbb{F}X\psi} = h_\psi/(1-i_\psi)$ the result follows.

5. To order the free variables we use the identities, for all interpretations, $\mathcal{I}$:

$$(\alpha?(\beta?\gamma_1:\gamma_2):(\beta?\delta_1:\delta_2))^{\mathcal{I}} = (\beta?(\alpha?\gamma_1:\delta_1):(\alpha?\gamma_2:\delta_2))^{\mathcal{I}} \quad (2)$$

$$\alpha = (\beta?\alpha:\alpha) \quad (3)$$

These identities are proven and discussed in [3]. Using these identities we can apply the following transformations:

$$(A_j?(A_k?\beta_1:\beta_2):\gamma) \Rightarrow (A_k?(A_j?\beta_1:\gamma):(A_j?\beta_2:\gamma)) \quad \text{when } k > j \quad (4)$$

$$(A_j?\gamma:(A_k?\beta_1:\beta_2)) \Rightarrow (A_k?(A_j?\gamma:\beta_1):(A_j?\gamma:\beta_2)) \quad \text{when } k > j \quad (5)$$

$$(A_j?(\text{◐}?\beta_1:\beta_2):\gamma) \Rightarrow (\text{◐}?(A_j?\beta_1:\gamma):(A_j?\beta_2:\gamma)) \quad (6)$$

$$(A_j?\gamma:(\text{◐}?\beta_1:\beta_2)) \Rightarrow (\text{◐}?(A_j?\gamma:\beta_1):(A_j?\gamma:\beta_2)) \quad (7)$$

$$A_j\beta\gamma \Rightarrow (A_j?(A_j?\beta:\gamma):(A_j?\beta:\gamma)) \quad (8)$$

This allows us to organise the tree representation of Fig. 3 so that all paths from the leaves to the root go through exactly k instances of each free variable in order, and then the nodes labelled by ◐.
6. Finally, we combine the fixed point operators into one, noting that they have all moved to the root, and we can apply the transformation $\mathbb{F}X\mathbb{F}Y\alpha(X,Y) \Rightarrow \mathbb{F}X\alpha(X,X)$.

These transformations can be applied repeatedly until the formula is in k-block normal form. As each transformation can be shown to preserve the interpretation of the formula, this completes the proof.

From this formula we are able to define the notion of an aleatoric function:

Definition 8. *Suppose that α is a formula in k-block normal form, defined over the free atomic propositions* $\mathrm{var}(\alpha) = \mathcal{X} = \{A_1, \ldots, A_n\}$*. Let the functions $h_\alpha(A_1, \ldots, A_n)$ and $i_\alpha(A_1, \ldots, A_n)$ be defined as follows:*

$$\begin{aligned} h_{\bullet}(\mathcal{X}) &= 1 & i_{\bullet}(\mathcal{X}) &= 0 \\ h_{\bigcirc}(\mathcal{X}) &= 0 & i_{\bigcirc}(\mathcal{X}) &= 0 \\ h_{A_j}(\mathcal{X}) &= A_j & i_{A_j}(\mathcal{X}) &= 0 \\ h_{X_0}(\mathcal{X}) &= 0 & i_{X_0}(\mathcal{X}) &= 1 \\ h_{(A_j?\alpha\beta)}\mathcal{X} &= A_j \cdot h_\alpha + (1-A_j) \cdot h_\beta & i_{(A_j?\alpha\beta)}\mathcal{X} &= A_j \cdot i_\alpha + (1-A_j) \cdot i_\beta \\ h_{(◐?\alpha\beta)}\mathcal{X} &= (1/2) \cdot h_\alpha + (1/2) \cdot h_\beta & i_{(◐?\alpha\beta)}\mathcal{X} &= (1/2) \cdot i_\alpha + (1/2) \cdot i_\beta \\ h_{\mathbb{F}X_0\alpha}(\mathcal{X}) &= h_\alpha(\mathcal{X})/(1-i_\alpha(\mathcal{X})) \end{aligned}$$

Given any proposition $\alpha \in \mathcal{L}$ where k is the maximum number of times a single atomic proposition appears in α, the aleatoric function of α *is the function $f_\alpha(\mathcal{X}) = h_{\tau(\alpha)}(\mathcal{X})$, where $\tau(\alpha)$ is the k-block normal form reduction of α.*

The following corollary is just a special case of Corollary 1

Corollary 2. *Given any aleatoric proposition $\alpha \in \mathcal{L}$ and any aleatoric interpretation $\mathcal{I}$:*

$$\mathcal{I}(\alpha) = f_\alpha(1, \mathcal{I}(X_1), \ldots, \mathcal{I}(X_n)).$$

3.2 Positive Rational Functions

Here, we show that every rational function from $(0,1)^{\mathcal{X}}$ to $(0,1)$ is equivalent to some aleatoric proposition defined of the atomic propositions $\mathcal{X}$. This subsection follows the analysis of coin flipping polynomials by Mossel and Peres [13]. Particularly, the following Lemma is based on Lemma 2.7 of [13].

Definition 9. *A rational function, $f : (0,1)^{\mathcal{X}} \longrightarrow (0,1)$ is a k*-block-function *if there exists polynomials ℓ and m:*

$$\ell(\mathcal{X}) = \textstyle\sum_{a \in \rho^k_{\mathcal{X}}} \ell_a \prod_{x \in \mathcal{X}} x^{a(x,+)} \cdot (1-x)^{a(x,-)}$$
$$m(\mathcal{X}) = \textstyle\sum_{a \in \rho^k_{\mathcal{X}}} m_a \prod_{x \in \mathcal{X}} x^{a(x,+)} \cdot (1-x)^{a(x,-)}$$

where $\rho^k_{\mathcal{X}} = \{\overline{a} \in \{0, \ldots, k\}^{\mathcal{X} \times \{+,-\}} \mid \sum_{x \in \mathcal{X}} a_{(x,+)} + a_{(x,-)} = k\}$, for all $a \in \rho^k_{\mathcal{X}}$, ℓ_a and m_a are integers such that $\ell_a < m_a$ and $f(\mathcal{X}) = \ell(\mathcal{X})/m(\mathcal{X})$.

A k-block function is designed to correspond to the notion of k-block normal form aleatoric propositions, and also has an elegant correspondence to rational functions $f : [0,1]^{\mathcal{X}} \longrightarrow (0,1)$.

Lemma 3. *Given some rational function $f : [0,1]^{\mathcal{X}} \longrightarrow (0,1)$, there is some k-block function $f' : [0,1] \longrightarrow (0,1)$ such that $f(\mathcal{X}) = f'(\mathcal{X})$.*

Proof. As $f(\mathcal{X})$ is a rational function over $[0,1]$ we may assume that it may be written $L(\mathcal{X})/M(\mathcal{X})$, where L and M are relatively prime polynomials with integer coefficients. We may therefore write $L(\mathcal{X}) = \sum_{a\in A} L_a \cdot \prod_{x\in\mathcal{X}} x^{a_x}$ and $M(\mathcal{X}) = \sum_{b\in B} M_b \cdot \prod_{x\in\mathcal{X}} x^{b_x}$, where $A, B \subset \{0,\ldots,k\}^{\mathcal{X}}$, for some k. We may define homogeneous polynomials of degree $k\cdot|\mathcal{X}|$, by defining

$$L'(\mathcal{X},\mathcal{X}') = \sum_{a\in A} L_a \prod_{x\in\mathcal{X}} x^{a_x}\cdot(x+x')^{k-a_x}$$

and

$$M'(\mathcal{X},\mathcal{X}') = \sum_{b\in B} M_b \prod_{x\in\mathcal{X}} x^{b_x}\cdot(x+x')^{k-b_x}$$

so that $L'(\mathcal{X},1-\mathcal{X}) = L(\mathcal{X})$ and $M'(\mathcal{X},1-\mathcal{X}) = M(\mathcal{X})$.

Then $L'(\mathcal{X},\mathcal{X}')$ and $M'(\mathcal{X},\mathcal{X}')$ can be written as, respectively

$$L'(\mathcal{Y}) = \sum_{c\in C} L'_c \prod_{y\in\mathcal{Y}} y^{c_y} \quad \text{and} \quad M'(\mathcal{Y}) = \sum_{c\in C} M'_c \prod_{y\in\mathcal{Y}} y^{c_y},$$

where $\mathcal{Y} = \mathcal{X}\cup\mathcal{X}$ and $C = \{c\in\{0,\ldots,k\}^{\mathcal{Y}} \mid \sum_{y\in\mathcal{Y}} c_y = k\cdot|\mathcal{X}|\}$. We note that $L'(\mathcal{Y})$, $M'(\mathcal{Y})$ and $M'(\mathcal{Y}) - L'(\mathcal{Y})$ are all homogeneous positive polynomials.

From Pólya [7] we have the following result:

> Given $f : [0,1]^{\mathcal{Y}} \longrightarrow (0,1)$, a homogeneous and positive polynomial, for sufficiently large n, all the coefficients of $(\sum_{y\in\mathcal{Y}} y)^n \cdot f(\mathcal{Y})$ are positive.

It follows that for some n,

$$(\sum_{y\in\mathcal{Y}} y)^n\cdot L'(\mathcal{Y}),\ (\sum_{y\in\mathcal{Y}} y)^n\cdot M'(\mathcal{Y}),\ \text{and}\ (\sum_{y\in\mathcal{Y}} y)^n\cdot(M'(\mathcal{Y}) - L'(\mathcal{Y}))$$

all have positive coefficients. The result follows from the observations that

$$f(\mathcal{X}) = L'(\mathcal{X},1-\mathcal{X})/M'(\mathcal{X},1-\mathcal{X}),$$

and $L'(\mathcal{X},1-\mathcal{X})/M'(\mathcal{X},1-\mathcal{X})$ is a k-block function.

A correspondence can now be given for the functional representation of aleatoric propositions.

Theorem 1. *1. For every aleatoric proposition $\alpha\in\mathcal{L}$ defined over the free variables in $\mathcal{X}$, $f_\alpha(\mathcal{X})$ is a rational function from $(0,1)$ to $(0,1)$.*

2. *For k-block-function $f(\mathcal{X})$ from $(0,1)^{\mathcal{X}}$ to $(0,1)$, there is some aleatoric proposition α such that $f_\alpha(\mathcal{X}) = f(\mathcal{X})$.*
3. *For every rational function $f(\mathcal{X})$ from $[0,1]^{\mathcal{X}}$ to $(0,1)$, there is some aleatoric proposition α such that $f_\alpha(\mathcal{X}) = f(\mathcal{X})$.*

Proof. The first part is immediate from the Definition 8.
The second part follows by noting the form of the polynomials $\ell(\mathcal{X})$ and $m(\mathcal{X})$ in the proof of Lemma 3 agrees with the numerator and denominator of f_α in Definition 8 Particularly, noting the form of the tree in Fig. 3, constructing a formula for a given $\ell(\mathcal{X})$ and $m(\mathcal{X})$ can be thought of as labelling the leaves of the tree so that

1. exactlyℓ_a a-paths have the leaf labelled with $\bullet$ for each $a \in \rho^k_{\mathcal{X}}$,
2. exactly $m_a - \ell_a$ a-paths have the leaf labelled with $\circ$ for each $a \in \rho^k_{\mathcal{X}}$,
3. and all other paths are labelled with X_0,

where an a-path is a branch of the tree with exactly $a_{(x,+)}$ positive instances of x, for each $x \in \mathcal{X}$.
Finally the third part follows from the second part, and Lemma 3

We note that [1] have given a complete characterisation of relaxations of Pòlya's theorem which could further generalise this result.

4 Conclusion and Future Work

This paper has given a description of aleatoric propositions, extending the work of [3], introducing the fixed point operator $\mathbb{F}X\alpha$, and establishing a correspondence with subclasses of rational functions over $(0,1)$. While this correspondence is based on earlier work in [13], the presentation as a logical system is novel, and this provides a foundation for future work on the logical aspects of this approach.

Future work will, taking aleatoric propositions as a base, extend the formalism to first order concepts, or aleatoric predicates. In this setting, we suppose that there is a probability space of domain elements, and a set of Boolean predicates given over these domain elements. An *expectation* operator allows us to express the expectation a proposition will be true when an element is drawn randomly from the domain. The analogy is an urn of marbles, where the marbles are labelled and predicates are defined over those labels.

We will also consider axiomatisations of these logics, the satisfiability problem, and combinations with modal necessity operators.

Acknowledgements. We thank the reviewers for their helpful feedback, as well an anonymous reader who noted an error in an earlier version.

References

1. Castle, M., Powers, V., Reznik, B.: Pòlya's theorem with zeros. J. Symbolic Comput. **46**(9), 1039–1048 (2011)

2. De Finetti, B.: Theory of Probability: A Critical Introductory Treatment. John Wiley & Sons, Hoboken (1970)
3. French, T., Gozzard, A., Reynolds, M.: A modal aleatoric calculus for probabilistic reasoning. In: Khan, M.A., Manuel, A. (eds.) ICLA 2019. LNCS, vol. 11600, pp. 52–63. Springer, Heidelberg (2019). https://doi.org/10.1007/978-3-662-58771-3_6
4. Gärdenfors, P.: Qualitative probability as an intensional logic. J. Philos. Logic, 171–185 (1975)
5. Hailperin, T.: Boole's Logic and Probability: A Critical Exposition from the Standpoint of Contemporary Algebra, Logic and Probability Theory (1976)
6. Halpern, J.Y.: Reasoning about Uncertainty. MIT press, Cambridge (2017)
7. Hardy, G.H., Littlewood, J.E., Pòlya, G.: Inequalities. Cambridge University Press, Cambridge (1959)
8. Hume, D.: A Treatise of Human Nature: Volume 1: Texts (1739)
9. Keane, M.S., O'Brien, G.L.: A Bernoulli factory. ACM Trans. Model. Comput. Simul. **4**, 213–219 (1994)
10. Kolmogorov, A.N.: The theory of probability. Math. Content Methods Meaning **2**, 110–118 (1963)
11. Kozen, D.: A probabilistic PDL. J. Comput. Syst. Sci. **30**(2), 162–178 (1985)
12. Lukasiewicz, T.: Expressive probabilistic description logics. Artif. Intell. **172**(6–7), 852–883 (2008)
13. Mossel, E., Peres, Y.: New coins from old: computing with unknown bias. Combinatorica **25**(6), 707–724 (2005)
14. Ramsey, F.P.: The foundations of mathematics (1925)
15. Von Neumann, J.: various techniques used in connection with random digits. Appl. Math. Ser. **12**(36–38), 3 (1951)
16. Zadeh, L.A.: Fuzzy sets. In: Fuzzy Sets, Fuzzy Logic, And Fuzzy Systems: Selected Papers by Lotfi A Zadeh, pp. 394–432. World Scientific (1996)

Towards an Induction Principle for Nested Data Types

Peng Fu(✉) and Peter Selinger(✉)

Dalhousie University, Halifax, Canada
peng.frank.fu@gmail.com, selinger@mathstat.dal.ca

Abstract. A well-known problem in the theory of dependent types is how to handle so-called *nested data types*. These data types are difficult to program and to reason about in total dependently typed languages such as Agda and Coq. In particular, it is not easy to derive a canonical induction principle for such types. Working towards a solution to this problem, we introduce *dependently typed folds* for nested data types. Using the nested data type `Bush` as a guiding example, we show how to derive its dependently typed fold and induction principle. We also discuss the relationship between dependently typed folds and the more traditional higher-order folds.

Keywords: dependent types · nested data types · induction principles · folds

1 Introduction

Consider the following list data type and its fold function in Agda [1].

```
data List (a : Set) : Set where
  nil : List a
  cons : a -> List a -> List a

foldList : ∀ {a p : Set} -> p -> (a -> p -> p) -> List a -> p
foldList base step nil = base
foldList base step (cons x xs) = step x (foldList base step xs)
```

The keyword `Set` is a kind that classifies types. The function `foldList` has two implicitly quantified type variables `a` and `p`. In Agda, implicit arguments are indicated by braces (e.g., `{a}`), and can be omitted.

The function `foldList` is defined by structural recursion and is therefore terminating. Agda's termination checker automatically checks this. Once `foldList` is defined, we can use it to define other terminating functions such as the following `mapList` and `sumList`. This is similar to using the iterator to define terminating arithmetic functions in System **T** [6, Sect. 7].

H. H. Hansen et al. (Eds.): WoLLIC 2023, LNCS 13923, pp. 244–255, 2023.
https://doi.org/10.1007/978-3-031-39784-4_15

```
mapList : ∀ {a b : Set} -> (a -> b) -> List a -> List b
mapList f ℓ = foldList nil (λ a r -> cons (f a) r) ℓ

sumList : List Nat -> Nat
sumList ℓ = foldList zero (λ x r -> add x r) ℓ
```

When defining the `mapList` function, if the input list is empty, then we just return `nil`, so the first argument for `foldList` is `nil`. If the input list is of the form `cons a as`, then we want to return `cons (f a) (mapList f as)`, so the second argument for `foldList` is `(λ a r -> cons (f a) r)`, where `r` represents the result of the recursive call `mapList f as`. The function `sumList` is defined similarly, assuming a natural numbers type `Nat` with zero and addition.

We can generalize the type of `foldList` to obtain the following induction principle for lists.

```
indList : ∀ {a : Set} {p : List a -> Set} ->
          (base : p nil) ->
          (step : (x : a) -> (xs : List a) -> p xs -> p (cons x xs)) ->
          (ℓ : List a) -> p ℓ
indList base step nil = base
indList base step (cons x xs) = step x xs (indList base step xs)
```

We can see that the definition of `indList` is almost the same as that of `foldList`. Compared to the type of `foldList`, the type of `indList` is more general as the kind of `p` is generalized from `Set` to `List a -> Set`. We call `p` a *property* of lists. The induction principle `indList` states that to prove a property `p` for all lists, one must first prove that `nil` has the property `p`, and then assuming that `p` holds for any list `xs` as the induction hypothesis, prove that `p` holds for `cons x xs` for any `x`.

We can now use the induction principle `indList` to prove that `mapList` has the same behavior as the usual recursively defined `mapList'` function.

```
mapList' : ∀ {a b : Set} -> (a -> b) -> List a -> List b
mapList' f nil = nil
mapList' f (cons x xs) = cons (f x) (mapList' f xs)

lemma-mapList : ∀ {a b : Set} -> (f : a -> b) -> (ℓ : List a) ->
                mapList f ℓ == mapList' f ℓ
lemma-mapList f ℓ =
  indList {p = λ y -> mapList f y == mapList' f y} refl
    (λ x xs ih -> cong (cons (f x)) ih) ℓ
```

In the proof of `lemma-mapList`, we use `refl` to construct a proof by reflexivity and `cong` to construct a proof by congruence. The latter is defined such that `cong f` is a proof of `x == y -> f x == f y`. The key to using the induction principle `indList` is to specify which property of lists we want to prove. In this case the property is `(λ y -> mapList f y == mapList' f y)`.

To summarize, the fold functions for *ordinary* data types (i.e., non-nested inductive data types such as `List` and `Nat`) are well-behaved in the following sense. (1) The fold functions are defined by well-founded recursion. (2) The fold functions can be used to define a range of terminating functions (including maps). (3) The types of the fold functions can be generalized to the corresponding induction principles.

Nested data types [2] are a class of data types that one can define in most functional programming languages (OCaml, Haskell, Agda). They were initially studied by Bird and Meertens [2]. They have since been used to represent de Bruijn notation for lambda terms [3], and to give an efficient implementation of persistent sequences [7]. In this paper, we will consider the following nested data type.

```
data Bush (a : Set) : Set where
  leaf : Bush a
  cons : a -> Bush (Bush a) -> Bush a
```

According to Bird and Meertens [2], the type `Bush` a is similar to a list where at each step down the list, entries are *bushed*. For example, a value of type `Bush Nat` can be visualized as follows.

```
 bush1 = [ 4,                                          -- Nat
           [ 8, [ 5 ], [ [ 3 ] ] ],               -- Bush Nat
           [ [ 7 ], [ ], [ [ [ 7 ] ] ] ], -- Bush (Bush Nat)
           [ [ [ ], [ [ 0 ] ] ] ]   -- Bush (Bush (Bush Nat))
         ]
```

Here, for readability, we have written `[`x_1`,...,`x_n`]` instead of `cons` x_1 `(cons` x_2 `(...(cons` x_n `leaf)))`.

Unlike ordinary data types such as lists, nested data types are difficult to program with in total functional programming languages. For example, in the dependently typed proof assistant Coq, the `Bush` data type is not definable at all, since it does not pass Coq's strict positivity test. In Agda, `Bush` can be defined as a data type, but writing functions that use this type is not trivial. For example, we must use general recursion (rather than structural recursion) to define the following `hmap` function.

```
hmap : ∀ {b c : Set} -> (b -> c) -> Bush b -> Bush c
hmap f leaf = leaf
hmap f (cons x xs) = cons (f x) (hmap (hmap f) xs)
```

Note that, in contrast to the `mapList'` function for lists, this definition is not structurally recursive because the inner `hmap` is not applied to a subterm of `cons x xs`. Therefore, Agda's termination checker will reject this definition as potentially non-terminating, unless we specify the unsafe `-no-termination` flag.

The following function `hfold` for `Bush` is called a *higher-order fold* in the literature (e.g., [4,8]). Its definition uses `hmap`.

```
hfold : (b : Set -> Set) ->
        (ℓ : (a : Set) -> b a) ->
        (c : (a : Set) -> a -> b (b a) -> b a) ->
        (a : Set) -> Bush a -> b a
hfold b ℓ c a leaf = ℓ a
hfold b ℓ c a (cons x xs) =
  c a x (hfold b ℓ c (b a) (hmap (hfold b ℓ c a) xs))
```

Observe that the type variable b in hfold has kind Set -> Set, unlike the type variable p in foldList, which has type Set. The higher-order fold hfold presents the following challenges. (1) The definition of hfold requires the auxiliary function hmap, and hmap cannot easily be defined from hfold. (2) The definition of hfold, like that of hmap, is not structurally recursive and Agda's termination checker cannot prove it to be total. (3) Although it is possible (see below), it is fairly difficult to define functions such as summation on Bush. (4) Unlike the induction principle for lists, it is not clear how to obtain an induction principle for Bush from the higher-order fold hfold.

Here is the definition of a function sum that sums up all natural numbers in a data structure of type Bush Nat. Although sum is not a polymorphic function, it requires an auxiliary function that is polymorphic and utilizes an argument k that is reminiscent of continuation passing style [9].

```
sumAux : (a : Set) -> Bush a -> (k : a -> Nat) -> Nat
sumAux =
  hfold (λ a -> (a -> Nat) -> Nat)
    (λ a k -> zero) (λ a x xs k -> add (k x) (xs (λ r -> r k)))

sum : Bush Nat -> Nat
sum ℓ = sumAux Nat ℓ (λ n -> n)
```

1.1 Contributions

We present a new approach to defining fold functions for nested data types, which we call *dependently typed folds*. For concreteness, we work within the dependently typed language Agda. Dependently typed folds are defined by well-founded recursion, hence their termination is easily confirmed by Agda. Map functions and many other terminating functions can be defined directly from the dependently typed folds. Moreover, the higher-order folds (such as hfold) are definable from the dependently typed folds. In addition, the definitions of dependently typed folds can easily be generalized to corresponding induction principles. Thus we can formally reason about programs involving nested data types in a total dependently typed language. While we illustrate these ideas by focusing on the Bush example, our approach also works for other kinds of nested data types; see Sect. 5 for an example.

2 Dependently Typed Fold for Bush

Let us continue the consideration of the Bush data type. The following is the result of evaluating hmap f bush1, where bush1 is the data structure defined in the introduction, and f : Nat -> b for some type b.

```
[ f 4,                                          -- b
  [ f 8, [ f 5 ], [ [ f 3 ] ] ],                -- Bush b
  [ [ f 7 ], [ ], [ [ [ f 7 ] ] ] ],  -- Bush (Bush b)
  [ [ [ ], [ [ f 0 ] ] ] ]      -- Bush (Bush (Bush b))
]
```

To motivate the definition of the dependently typed fold below, we first consider the simpler question of how to define a map function for `Bush` by structural recursion. The reason our definition of `hmap` in the introduction was not structural is that in order to define the map function for `Bush Nat`, we need to already have the map functions defined for $\texttt{Bush}^n$ `Nat` = `Bush` (`Bush` (... (`Bush Nat`))) for all $n \geq 0$, which seems paradoxical. Our solution is to define a general map function for $\texttt{Bush}^n$, for all $n \geq 0$. First we define a type-level function `NTimes` such that `NTimes n b` = $\texttt{b}^n$:

```
NTimes : (n : Nat) -> (b : Set -> Set) -> Set -> Set
NTimes zero b a = a
NTimes (succ n) b a = b (NTimes n b a)
```

We can now define the following map function for $\texttt{Bush}^n$:

```
nmap : ∀ {a b : Set} -> (n : Nat) -> (a -> b) ->
       NTimes n Bush a -> NTimes n Bush b
nmap zero f x = f x
nmap (succ n) f leaf = leaf
nmap (succ n) f (cons x xs) =
  cons (nmap n f x) (nmap (succ (succ n)) f xs)
```

Note that `nmap 1` corresponds to the map function for `Bush a`. The recursive definition of `nmap` is well-founded because all the recursive calls are on the components of the constructor `cons`. The Agda termination checker accepts this definition of `nmap`.

We are now ready to introduce the dependently typed fold. The idea is to define the fold over the type `NTimes n Bush` simultaneously for all n.

```
nfold : (p : Nat -> Set) ->
        (ℓ : (n : Nat) -> p (succ n)) ->
        (c : (n : Nat) -> p n -> p (succ (succ n)) -> p (succ n)) ->
        (a : Set) -> (z : a -> p zero) ->
        (n : Nat) -> NTimes n Bush a -> p n
nfold p ℓ c a z zero x = z x
nfold p ℓ c a z (succ n) leaf = ℓ n
nfold p ℓ c a z (succ n) (cons x xs) =
  c n (nfold p ℓ c a z n x) (nfold p ℓ c a z (succ (succ n)) xs)
```

The dependently typed fold `nfold` captures the most general form of computing/traversal on the type `NTimes n Bush a`. Similarly to `nmap`, the definition of `nfold` is well-founded. Note that unlike the `hfold` in the introduction, this definition of fold does not require a map function to be defined first. In fact, `nmap` is definable from `nfold`:

```
nmap : ∀ {a b : Set} -> (n : Nat) -> (a -> b) ->
       NTimes n Bush a -> NTimes n Bush b
nmap {a} {b} n f ℓ =
  nfold (λ n -> NTimes n Bush b) (λ n -> leaf) (λ n -> cons) a f n ℓ
```

We can also prove that `nmap 1` satisfies the defining properties of `hmap` from the introduction. Let `hmap'` = `nmap 1`.

```
lemma-nmap : ∀ {a b : Set} -> (f : a -> b) -> (m n : Nat) ->
             (x : NTimes (add m n) Bush a) ->
             nmap (add m n) f x == nmap m (nmap n f) x
lemma-nmap f zero n x = refl
lemma-nmap f (succ m) n leaf = refl
lemma-nmap f (succ m) n (cons x xs) =
  cong2 cons (lemma-nmap f m n x) (lemma-nmap f (succ (succ m)) n xs)

hmap-leaf : ∀ {a b : Set} -> (f : a -> b) -> hmap' f leaf == leaf
hmap-leaf f = refl

hmap-cons : ∀ {a b : Set} -> (f : a -> b) -> (x : a) ->
            (xs : Bush (Bush a)) ->
            hmap' f (cons x xs) == cons (f x) (hmap' (hmap' f) xs)
hmap-cons f x xs = cong (cons (f x)) (lemma-nmap f 1 1 xs)
```

Many other terminating functions can also be conveniently defined in term of `nfold`. For example, the summation of all the entries in `Bush Nat` and the length function for `Bush` can be defined as follows:

```
sum : Bush Nat -> Nat
sum =
  nfold (λ n -> Nat) (λ n -> zero) (λ n -> add) Nat (λ x -> x) 1

length : (a : Set) -> Bush a -> Nat
length a =
  nfold (λ n -> Nat) (λ n -> zero) (λ n r1 r2 -> succ r2)
    a (λ x -> zero) 1
```

Note that this definition of `sum` is much more natural and straightforward than the one we gave in the introduction.

3 Induction Principle for Bush

While there is no obvious induction principle corresponding to the higher-order fold `hfold`, we can easily generalize the dependently typed fold `nfold` to obtain an induction principle for `Bush`. The following function `ind` is related to `nfold` in the same way that the induction principle for `List` is related to its fold function.

```
ind : ∀ {a : Set} -> {p : (n : Nat) -> NTimes n Bush a -> Set} ->
      (base : (x : a) -> p zero x) ->
      (ℓ : (n : Nat) -> p (succ n) leaf) ->
      (c : (n : Nat) -> (x : NTimes n Bush a) ->
        (xs : NTimes (succ (succ n)) Bush a) ->
        p n x -> p (succ (succ n)) xs -> p (succ n) (cons x xs)) ->
      (n : Nat) -> (xs : NTimes n Bush a) -> p n xs
ind base ℓ c zero xs = base xs
ind base ℓ c (succ n) leaf = ℓ n
ind base ℓ c (succ n) (cons x xs) =
  c n x xs (ind base ℓ c n x) (ind base ℓ c (succ (succ n)) xs)
```

Observe that `ind` follows the same structure as `nfold`. The type variable `p` is generalized to a predicate of kind `(n : Nat) -> NTimes n Bush a -> Set`. The type of `ind` specifies how to prove by induction that a property `p` holds for all members of the type `NTimes n Bush a`. More specifically, for the base case, we must show that `p` holds for any `x` of type `NTimes zero Bush a` (which equals `a`), hence `p zero x`. For the leaf case, we must show that `p` holds for `leaf` of type `NTimes (succ n) Bush a`. For the cons case, we assume as the induction hypotheses that `p` holds for some `x` of type `NTimes n Bush a` and some `xs` of type `NTimes (succ (succ n)) Bush a`, and then we must show that `p` holds for `cons x xs`.

With `ind`, we can now prove properties of `nmap`. For example, the following is a proof that `nmap` has the usual identity property of functors.

```
nmap-id : ∀ {a : Set} -> (n : Nat) -> (y : NTimes n Bush a) ->
          nmap n (id a) y == y
nmap-id {a} n y =
  ind {a} {λ n xs -> nmap n (id a) xs == xs} (λ x -> refl) (λ n -> refl)
    (λ n x xs ih1 ih2 -> cong2 cons ih1 ih2) n y
```

We note that the usual way of proving things in Agda is by recursion, relying on the Agda termination checker to prove termination. Our purpose here, of course, is to illustrate that our induction principle is strong enough to prove many properties without needing Agda's recursion. Nevertheless, the above proof is equivalent to the following proof by well-founded recursion.

```
nmap-id' : ∀ {a : Set} -> (n : Nat) -> (y : NTimes n Bush a) ->
           nmap n (id a) y == y
nmap-id' zero y = refl
nmap-id' (succ n) leaf = refl
nmap-id' (succ n) (cons x y) =
  cong2 cons (nmap-id' n x) (nmap-id' (succ (succ n)) y)
```

The first two clauses of `nmap-id'` correspond to the two arguments `(λ n -> refl)` for `nmap-id`. The recursive calls `nmap-id' n x` and `nmap-id' (succ (succ n)) y` in the definition of `nmap-id'` correspond to the inductive hypotheses `ih1` and `ih2` in `nmap-id`.

4 Higher-Order Folds and Dependently Typed Folds

Comparing `nfold`, the dependently typed fold that was defined in Sect. 2, to `hfold`, the higher-order fold defined in the introduction, we saw that `nfold` does not depend on `nmap`, and `nmap` can be defined from `nfold`. We also saw that the termination of `nfold` is obvious and that it can be used to define other terminating functions.

In this section, we will show the `hfold` is actually equivalent to `nfold` in the sense that they are definable from each other.

4.1 Defining `hfold` from `nfold`

Using `nfold`, it is straightforward to define `hfold`, because the latter is essentially the former instantiated to the case $n = 1$.

```
hfold : (b : Set -> Set) ->
        (ℓ : (a : Set) -> b a) ->
        (c : (a : Set) -> a -> b (b a) -> b a) ->
        (a : Set) -> Bush a -> b a
hfold b ℓ c a x =
  nfold (λ n -> NTimes n b a) (λ n -> ℓ (NTimes n b a))
    (λ n -> c (NTimes n b a)) a (λ x -> x) 1 x
```

We can prove that this version of `hfold` satisfies the defining properties of the version of `hfold` that was defined in the introduction (and therefore the two definitions agree). Since the proof of `hfold-cons` is rather long, we have omitted it, but the full machine-checkable proof can be found at [5].

```
hfold-leaf : (a : Set) -> (p : Set -> Set) ->
              (ℓ : (b : Set) -> p b) ->
              (c : (b : Set) -> b -> p (p b) -> p b) ->
               hfold p ℓ c a leaf == ℓ a
hfold-leaf a p ℓ c = refl

hfold-cons : (a : Set) -> (p : Set -> Set) ->
              (ℓ : (b : Set) -> p b) ->
              (c : (b : Set) -> b -> p (p b) -> p b) ->
              (x : a) -> (xs : Bush (Bush a)) ->
              hfold p ℓ c a (cons x xs)
              == c a x (hfold p ℓ c (p a) (hmap (hfold p ℓ c a) xs))
hfold-cons a p ℓ c x xs = ...
```

4.2 Defining `nfold` from `hfold`

The other direction is much trickier. In attempting to define `nfold` from `hfold`, the main difficulty is that we must supply a type function `b : Set -> Set` to `hfold`, and this `b` should somehow capture the quantification over natural numbers. Ideally, we would like to define `b` such that $\mathtt{b}^n\,\mathtt{a} = \mathtt{p}\,n$ for all n and some suitable `a`. However, this is clearly impossible, because `p` is an arbitrary type family, which can be defined so that $\mathtt{p}\,0 = \mathtt{p}\,1$ but $\mathtt{p}\,1 \neq \mathtt{p}\,2$. This would imply $\mathtt{a} = \mathtt{b}\,\mathtt{a}$ but $\mathtt{b}\,\mathtt{a} \neq \mathtt{b}^2\,\mathtt{a}$, a contradiction.

Surprisingly, it is possible to work around this by arranging things so that there is a canonical function $\mathtt{b}^n\,\mathtt{a} \to \mathtt{p}\,n$, rather than an equality. This is done by defining the following rather unintuitive type-level function `PS`.

```
PS : (p : Nat -> Set) -> Set -> Set
PS p A = (n : Nat) -> (A -> p n) -> p (succ n)
```

The type `PS p` is special because there is a map `NTimes n (PS p) a` $\to$ `p n`.

```
PS-to-P : (p : Nat -> Set) -> (a : Set) -> (z : a -> p zero) ->
          (n : Nat) -> NTimes n (PS p) a -> p n
PS-to-P p a z zero x = z x
PS-to-P p a z (succ n) hyp = hyp n ih
  where
    ih : NTimes n (PS p) a -> p n
    ih = PS-to-P p a z n
```

So if we set $\mathtt{b} = \mathtt{PS}\,\mathtt{p}$, we have the promised canonical map $\mathtt{b}^n\,\mathtt{a} \to \mathtt{p}\,n$. We can pass this `b` to `hfold` to go from `Bush a` to `PS p a`.

```
fold-PS : (p : Nat -> Set) ->
         (ℓ : (n : Nat) -> p (succ n)) ->
         (c : (n : Nat) -> p n -> p (succ (succ n)) -> p (succ n)) ->
         (a : Set) -> Bush a -> PS p a
fold-PS p ℓ c =
  hfold (PS p) (λ a n tr -> ℓ n)
    (λ a x xs n tr -> c n (tr x) (xs (succ n) (λ f -> f n tr)))
```

Now, provided that we are able to *lift* the function `Bush a -> PS p a` to its nth iteration, i.e., `NTimes n Bush a -> NTimes n (PS p) a`, then we will be able to define the dependently typed fold via the following.

```
nfold' : (p : Nat -> Set) ->
         (ℓ : (n : Nat) -> p (succ n)) ->
         (c : (n : Nat) -> p n -> p (succ (succ n)) -> p (succ n)) ->
         (a : Set) -> (z : a -> p zero) ->
         (n : Nat) -> NTimes n Bush a -> p n
nfold' p ℓ c a z n x = PS-to-P p a z n (lift n x)
  where
    lift : (n : Nat) -> NTimes n Bush a -> NTimes n (PS p) a
    lift n x =
      liftNTimes Bush (PS p) (λ a b -> hmap) n (fold-PS p ℓ c) a x
```

The `liftNTimes` function can indeed be defined by induction on natural numbers.

```
liftNTimes : (b c : Set -> Set) ->
             (∀ x y -> (x -> y) -> (b x -> b y)) ->
             (n : Nat) -> (∀ a -> b a -> c a) ->
             (a : Set) -> NTimes n b a -> NTimes n c a
liftNTimes b c m zero f a x = x
liftNTimes b c m (succ n) f a x =
  f (NTimes n c a)
    (m (NTimes n b a) (NTimes n c a) (liftNTimes b c m n f a) x)
```

Finally, we can prove that the function `nfold'` that we just defined behaves identically to the `nfold` that was defined in Sect. 2. Again, since the proof is rather long and uses several lemmas, we do not reproduce it here. The machine-checkable proof can be found at [5].

```
theorem : ∀ p ℓ c a z n x ->
          nfold p ℓ c a z n x == nfold' p ℓ c a z n x
theorem p ℓ c a z n x = ...
```

5 Nested Data Types Beyond Bush

So far, we have focused on the Bush type, but our approach works for arbitrary nested data types, including ones that are defined by mutual recursion. To illustrate this, consider the following pair of mutually recursive data types:

```
data Bob (a : Set) : Set
data Dylan (a b : Set) : Set

data Bob a where
  robert : a -> Bob a
  zimmerman : Dylan (Bob (Dylan a (Bob a))) (Bob a) -> Bob (Dylan a a) -> Bob a

data Dylan a b where
  duluth : Bob a -> Bob b -> Dylan a b
  minnesota : Dylan (Bob a) (Bob b) -> Dylan a b
```

As usual, the higher-order fold is easy to define. There are two separate such folds, one for Bob and one for Dylan:

```
hfold-bob : (bob : Set -> Set) ->
      (dylan : Set -> Set -> Set) ->
      (rob : ∀ a -> a -> bob a) ->
      (zim : ∀ a -> dylan (bob (dylan a (bob a))) (bob a) -> bob (dylan a a) -> bob a) ->
      (dul : ∀ a b -> bob a -> bob b -> dylan a b) ->
      (min : ∀ a b -> dylan (bob a) (bob b) -> dylan a b) ->
      ∀ a -> Bob a -> bob a

hfold-dylan : (bob : Set -> Set) ->
      (dylan : Set -> Set -> Set) ->
      (rob : ∀ a -> a -> bob a) ->
      (zim : ∀ a -> dylan (bob (dylan a (bob a))) (bob a) -> bob (dylan a a) -> bob a) ->
      (dul : ∀ a b -> bob a -> bob b -> dylan a b) ->
      (min : ∀ a b -> dylan (bob a) (bob b) -> dylan a b) ->
      ∀ a b -> Dylan a b -> dylan a b
```

The dependent fold requires some explanation. Recall that for Bush, the only type expressions of interest were of the form $\mathtt{Bush}^n\,\mathtt{a}$, so we used the natural number n to index these types. In the more general case, we must consider more complicated type expressions such as Dylan (Bob a) (Dylan a b). Therefore, we need to replace the natural numbers with a custom type. We define a type BobDylanIndex, which represents expressions built up from type variables and the type constructors Bob and Dylan.

```
data BobDylanIndex : Set where
  varA : BobDylanIndex
  varB : BobDylanIndex
  BobC : BobDylanIndex -> BobDylanIndex
  DylanC : BobDylanIndex -> BobDylanIndex -> BobDylanIndex
```

We can then give an interpretation function for these type expressions. This plays the role that NTimes played in the Bush case:

```
I : (Set -> Set) -> (Set -> Set -> Set) -> Set -> Set -> BobDylanIndex -> Set
I bob dylan a b varA = a
I bob dylan a b varB = b
I bob dylan a b (BobC expr) = bob (I bob dylan a b expr)
I bob dylan a b (DylanC expr1 expr2) = dylan (I bob dylan a b expr1) (I bob dylan a b expr2)
```

For example, if

```
i = DylanC (BobC varA) (DylanC varA varB),
```

then

```
I bob dylan a b i = dylan (bob a) (dylan a b).
```

The dependent fold is defined simultaneously for `Bob` and `Dylan`, and in fact for all type expressions that are built from `Bob` and `Dylan`. Its type is the following:

```
nfold : (p : BobDylanIndex -> Set) ->
        (rob : ∀ a -> p a -> p (BobC a)) ->
        (zim : ∀ a -> p (DylanC (BobC (DylanC a (BobC a))) (BobC a))
                   -> p (BobC (DylanC a a)) -> p (BobC a)) ->
        (dul : ∀ a b -> p (BobC a) -> p (BobC b) -> p (DylanC a b)) ->
        (min : ∀ a b -> p (DylanC (BobC a) (BobC b)) -> p (DylanC a b)) ->
        (a b : Set) ->
        (baseA : a -> p varA) ->
        (baseB : b -> p varB) ->
        (∀ i -> I Bob Dylan a b i -> p i)
```

Note that although the types `Bob` and `Dylan` are complicated, the corresponding `nfold` can be systematically derived from their definition. Moreover, as in the case of `Bush`, the higher-order folds and the dependent fold are definable in terms of each other. In addition, the induction principle, which generalizes `nfold`, can be easily defined. Full details can be found in the accompanying code [5].

6 Discussion

We think that the equivalence of `hfold` and `nfold` is both surprising and useful. The reason it is surprising is because it was informally believed among researchers that `hfold` is too abstract for most useful programming tasks. The reason it is potentially useful is that in the context of some dependently typed programming languages or proof assistants (such as Coq), when the user writes a data type declaration, the system should automatically derive the appropriate folds and induction principles for the data type. In the case of nested data types, there is currently no universally good way to do this (which is presumably one of the reasons Coq does not support the `Bush` type). Now on the one hand, we have `nfold`, which is a practical programming primitive, but its type is not easy to generate from a user-defined data type declaration. For example, even stating the type of `nfold` requires a reference to an ancillary data type, which is `Nat` in the case of `Bush` but can be more complicated for a general nested type. On the other hand, we have `hfold`, which is not very practical, but its type can be easily read off from a data type declaration. The fact that we have shown `nfold` to be definable in terms of `hfold` suggests a solution to this problem: given a data type declaration, the system can generate its corresponding `hfold`, and then the user can follow a generic recipe to derive the more useful `nfold`.

7 Conclusion and Future Work

Using `Bush` as an example, we showed how to define dependently typed folds for nested data types. Unlike higher-order folds, dependently typed folds can be used to define maps and other terminating functions, and they have analogous induction principles, similar to the folds for ordinary data types. We showed how to reason about programs involving nested data types in Agda. Last but not least, we also showed that dependently typed folds and higher-order folds are mutually definable. This has some potential applications in implementations of dependent type theories, because given a user-defined nested data type, the corresponding higher-order fold can be automatically generated, and then the user can derive the more useful dependent fold by following a generic recipe. All of our proofs are done in Agda, without using any unsafe flag.

Our long term goal is to derive induction principles for *any* algebraic data type (nested or non-nested). There is still a lot of work to be done. In this paper, we only showed how to get the dependently typed fold and induction principle for the single example of `Bush`. Although our approach also works for other nested data types, we have not yet given a formal characterization of dependently typed folds and their induction principles in the general case. Another research direction is to study the direct relationship between the induction principles (derived from dependently typed folds) and higher-order folds. In the `Bush` example, it corresponds to asking if we can define `ind` from `hfold`, possibly with some extra properties that can also be read off from the data type definition.

Acknowledgements. We thank the referees for their thoughtful comments. This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and by the Air Force Office of Scientific Research under Award No. FA9550-21-1-0041.

References

1. Agda documentation. https://agda.readthedocs.io/. Accessed 05 Feb 2022
2. Bird, R., Meertens, L.: Nested datatypes. In: Jeuring, J. (ed.) MPC 1998. LNCS, vol. 1422, pp. 52–67. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0054285
3. Bird, R., Paterson, R.: De Bruijn notation as a nested datatype. J. Funct. Program. **9**(1), 77–91 (1999). https://doi.org/10.1017/S0956796899003366
4. Bird, R., Paterson, R.: Generalised folds for nested datatypes. Formal Aspects Comput. **11**(2), 200–222 (1999). https://doi.org/10.1007/s001650050047
5. Fu, P., Selinger, P.: Agda code accompanying this paper (2023). https://www.mathstat.dal.ca/~selinger/papers/downloads/Wollic.agda
6. Girard, J.Y., Lafont, Y., Taylor, P.: Proofs and Types. Cambridge University Press, Cambridge (1989)
7. Hinze, R., Paterson, R.: Finger trees: a simple general-purpose data structure. J. Funct. Program. **16**(2), 197–217 (2006). https://doi.org/10.1017/S0956796805005769
8. Johann, P., Ghani, N.: Initial algebra semantics is enough! In: Della Rocca, S.R. (ed.) TLCA 2007. LNCS, vol. 4583, pp. 207–222. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-73228-0_16
9. Plotkin, G.D.: Call-by-name, call-by-value and the λ-calculus. Theor. Comput. Sci. **1**, 125–159 (1975). https://doi.org/10.1016/0304-3975(75)90017-1

A Principled Approach to Expectation Maximisation and Latent Dirichlet Allocation Using Jeffrey's Update Rule

Bart Jacobs(✉)

Institute for Computing and Information Sciences, Radboud University Nijmegen, Nijmegen, The Netherlands
bart@cs.ru.nl

Abstract. Expectation Maximisation (EM) and Latent Dirichlet Allocation (LDA) are two frequently used inference algorithms, for finding an appropriate mixture of latent variables, and for finding an allocation of topics for a collection of documents. A recent insight in probabilistic learning is that Jeffrey's update rule gives a decrease of Kullback-Leibler divergence. Its logic is error correction. It is shown that this same rule and divergence decrease logic is at the heart of EM and LDA, ensuring that successive iterations are decreasingly wrong.

1 Introduction

Learning can happen via encouragement or via discouragement, that is by reinforcing what goes well, or by slowing down what is going the wrong way. Intuitively these differences are clear. In probabilistic learning one can distinguish rules of Pearl and Jeffrey for updating (conditioning, belief revision), see [17,22] and [3,8,12,19] for comparisons. In [14] the difference between these rules has been described mathematically: Pearl's rule gives an increase of validity (expected value), whereas Jeffrey's rule gives a decrease of (Kullback-Leibler) divergence. The latter may be understood as error correction (or reduction of free energy), like in predictive coding theory [9,11,23]—where the human mind is studied as a Bayesian prediction and correction engine.

This paper demonstrates the relevance of Jeffrey's update rule — with its divergence decrease—for two fundamental inference algorithms, namely Expectation Maximisation (EM) [7] and Latent Dirichlet Allocation (LDA) [2]. EM is used for uncovering mixtures of latent variables. It has many applications, for instance in natural language processing, computer vision, and genetics. LDA is used to get a big-picture of (large) collections of documents by discovering the topics that they cover. Both are unsupervised classification algorithms.

The paper gives an abstract reformulation of these two well-known algorithms in machine learning that brings out the logic of divergence reduction (or error correction) behind them. This reformulation is inspired by categorical probability theory (see *e.g.* [4,10]), in which conditional probabilities $p(y \mid x)$ are

H. H. Hansen et al. (Eds.): WoLLIC 2023, LNCS 13923, pp. 256–273, 2023.
https://doi.org/10.1007/978-3-031-39784-4_16

reinterpreted as probabilistic functions $X \to Y$, also known as Kleisli maps or channels, with a rich structure, among others for sequential composition, parallel composition, and reversal. This paper does not assume knowledge of category theory: the relevant constructions are described concretely, especially for reversal (Bayesian inversian, dagger) since it plays a crucial role in Jeffrey's rule.

Making explicit what these algorithms EM and LDA achieve, and how, is relevant in times with a rising need that algorithms in machine learning and AI explain their outcomes. A first requirement for such explanations is a clear semantical understanding, including the underlying logic. Thus, the aim here is to analyse (two) existing algorithms, in their basic forms. The algorithms are not extended or improved, but studied as such.

This paper is organised as follows. It first introduces notation and basic terminology for multisets and (discrete probability) distributions in Sect. 2, including channels (probabilistic functions) and their reversals. Section 3 recalls Jeffrey's update rule and the associated divergence decrease. It includes a (new) strengthened version of this rule, with multiple channels, that will be used for LDA. Subsequently, Sect. 4 describes the EM algorithm using channels and shows how its correctness can be proved in just four lines, see the proof of Theorem 3. Section 5 gives a similar reformulation and proof of correctness for LDA. Simple illustrations are included for Jeffrey's rule, EM and for LDA.

2 Multisets and Distributions

A multiset (or bag) is a subset in which elements may occur multiple times. We borrow 'ket' notation $|\cdot\rangle$ from quantum theory and represent an urn with three red, two blue and one green ball as a multiset $3|R\rangle + 2|B\rangle + 1|G\rangle$. In the 'bag-of-words' model a document is understood as a multiset of words. A distribution is like a multiset but its multiplicities are not natural numbers but probabilities, from the unit interval $[0, 1]$, that add up to one, as in $\frac{1}{2}|R\rangle + \frac{1}{3}|B\rangle + \frac{1}{6}|G\rangle$.

More formally, a multiset on a set X is a function $\varphi\colon X \to \mathbb{N}$ with finite support $\mathrm{supp}(\varphi) = \{x \in X \mid \varphi(x) > 0\}$. Similarly, a distribution on X is a function $\omega\colon X \to [0, 1]$ with finite support, with $\sum_x \omega(x) = 1$. We can equivalently write them in ket notation as $\varphi = \sum_x \varphi(x)|x\rangle$ and $\omega = \sum_x \omega(x)|x\rangle$. We write $\mathcal{M}(X)$ and $\mathcal{D}(X)$ for the sets of multisets and distributions on a set X.

Notice that we do not require that the set X is finite. But when X is finite, we can say that a multiset $\varphi \in \mathcal{M}(X)$, or a distribution $\omega \in \mathcal{D}(X)$, has full support if $\mathrm{supp}(\varphi) = X$, or $\mathrm{supp}(\omega) = X$. The unit multiset $\mathbf{1}_X := \sum_x 1|x\rangle$ and the uniform distribution $\mathit{unif}_X := \sum_x \frac{1}{n}|x\rangle$, for the size $n = |X|$ of the set X, are examples with full support. A fair coin $\frac{1}{2}|H\rangle + \frac{1}{2}|T\rangle$ and a fair dice $\frac{1}{6}|1\rangle + \frac{1}{6}|2\rangle + \frac{1}{6}|3\rangle + \frac{1}{6}|4\rangle + \frac{1}{6}|5\rangle + \frac{1}{6}|6\rangle$ are examples of uniform distributions, on the set $\{H, T\}$ and on $\{1, 2, 3, 4, 5, 6\}$.

The size $\|\varphi\| \in \mathbb{N}$ of a multiset $\varphi \in \mathcal{M}(X)$ is the total number of elements, including multiplicities: $\|\varphi\| := \sum_x \varphi(x)$. We use special notation for the set of multisets of a particular size K.

$$\mathcal{M}[K](X) := \{\varphi \in \mathcal{M}(X) \mid \|\varphi\| = K\}.$$

The only multiset on X with size 0 is the constant-zero function $\mathbf{0}\colon X \to \mathbb{N}$.

For a non-zero multiset $\varphi \in \mathcal{M}(X)$ we write $\mathit{Flrn}(\varphi) \in \mathcal{D}(X)$ for the distribution obtained via 'frequentist learning', that is via counting and normalisation:

$$\mathit{Flrn}(\varphi)(x) \coloneqq \frac{\varphi(x)}{\|\varphi\|} \qquad \text{that is} \qquad \mathit{Flrn}(\varphi) = \sum_{x\in X} \frac{\varphi(x)}{\|\varphi\|}\,\big|\,x\,\big\rangle.$$

The multinomial distribution describes draws with replacement from an urn filled with coloured balls, represented as a distribution $\omega \in \mathcal{D}(X)$, where X is the set of colours. The number $\omega(x) \in [0,1]$ is the probability/fraction of balls of colour $x \in X$ in the urn. For a fixed number K, the multinomial distribution $mn[K](\omega)$ assigns a probability to a draw of K balls, represented as a multiset $\varphi \in \mathcal{M}[K](X)$. It is defined as:

$$mn[K](\omega) \coloneqq \sum_{\varphi\in\mathcal{M}[K](X)} (\varphi) \cdot \prod_{x\in X} \omega(x)^{\varphi(x)}\,\big|\,\varphi\,\big\rangle \;\in\; \mathcal{D}\Big(\mathcal{M}[K](X)\Big), \tag{1}$$

where $(\varphi) \coloneqq \frac{\|\varphi\|!}{\prod_x \varphi(x)!}$ is the multinomial coefficient of φ, see *e.g.* [13,15] for more details. For instance for a distribution $\omega = \frac{1}{8}|\,a\,\rangle + \frac{1}{4}|\,b\,\rangle + \frac{5}{8}|\,c\,\rangle$ over the set of colours $X = \{a, b, c\}$ and for draws of size $K = 2$ we get:

$$\begin{aligned} mn[2](\omega) = \tfrac{1}{64}\Big|\,2|\,a\,\rangle\Big\rangle + \tfrac{1}{16}\Big|\,1|\,a\,\rangle + 1|\,b\,\rangle\Big\rangle + \tfrac{1}{16}\Big|\,2|\,b\,\rangle\Big\rangle + \tfrac{5}{32}\Big|\,1|\,a\,\rangle + 1|\,c\,\rangle\Big\rangle \\ + \tfrac{5}{16}\Big|\,1|\,b\,\rangle + 1|\,c\,\rangle\Big\rangle + \tfrac{25}{64}\Big|\,2|\,c\,\rangle\Big\rangle. \end{aligned}$$

In Sect. 5 a distribution on words will be used to assign a multinomial probability to a document, as a multiset (bag) of words.

What we describe in (1) is the so-called multivariate case, with multiple colours. When there are just two colours, that is, when the set X has two elements, say $X = \{0, 1\}$, we are in the bivariate case. It will be used in Example 4. Via the isomorphisms $\mathcal{D}(\{0,1\}) \cong [0,1]$ and $\mathcal{M}[K](\{0,1\}) \cong \{0,1,\ldots,K\}$ one gets the binomial distribution as special case of (1), for a bias $r \in [0,1]$,

$$bn[K](r) \coloneqq \sum_{n\in\{0,\ldots,K\}} \binom{K}{n} \cdot r^n \cdot (1-r)^{K-n}\,\big|\,n\,\big\rangle \;\in\; \mathcal{D}\Big(\{0,\ldots,K\}\Big). \tag{2}$$

2.1 Channels and Their Daggers

An essential element of the principled categorical approach to probability is the use of channels, also known as Kleisli maps. For two sets X, Y, a channel from X to Y is a probabilistic function, written as $c\colon X \rightsquigarrow Y$. It is an actual function of the form $c\colon X \to \mathcal{D}(Y)$ that assigns a distribution $c(x) \in \mathcal{D}(Y)$ to each element $x \in X$. In traditional notation it is written as a conditional probability distribution $p(y \mid x)$. These channels (probabilistic functions) can be composed, both sequentially and in parallel; moreover, they can be reversed, giving what is called a dagger channel [4,5,10], also known as the Bayesian inverse $p(x \mid y)$ of $p(y \mid x)$. This gives a useful calculus of channels.

For a distribution $\omega \in \mathcal{D}(X)$ on the domain of a channel $c\colon X \rightsquigarrow Y$ we may 'push forward' (or 'transform') the distribution along the channel, giving a distribution $c \gg\!= \omega \in \mathcal{D}(Y)$, on the codomain Y of the channel. This new distribution $c \gg\!= \omega$ is also called the 'prediction'. It is defined as:

$$\big(c \gg\!= \omega\big)(y) \coloneqq \sum_{x \in X} \omega(x) \cdot c(x)(y). \tag{3}$$

Using push forward $\gg\!=$ we can define composition of channels $c\colon X \rightsquigarrow Y$ and $d\colon Y \rightsquigarrow Z$ to a new channel $d \circ c\colon X \rightsquigarrow Z$, namely as $(d \circ c)(x) \coloneqq d \gg\!= c(x)$. Notice that we use special notation $\circ$ for composition of channels.

We turn to the reversal of a channel $c\colon X \rightsquigarrow Y$ in presence of a 'prior' distribution $\omega \in \mathcal{D}(X)$. The result is a channel $c^{\dagger}_{\omega}\colon Y \rightsquigarrow X$, defined as:

$$c^{\dagger}_{\omega}(y) \coloneqq \sum_{x \in X} \frac{\omega(x) \cdot c(x)(y)}{(c \gg\!= \omega)(y)} \, \big| x \big\rangle. \tag{4}$$

For more details about this reversal we refer to the literature [4,5,10].

3 Jeffrey's Update Rule and Its Decrease of Divergence

In probabilistic learning one can distinguish two different approaches to updating, namely following Pearl [22] (and Bayes) or following Jeffrey [17], see for comparisons *e.g.* [3,8,12,16,19]. The two approaches may produce completely different outcomes, but it is poorly understood when to use which approach. The distinction between the rules is characterised mathematically in [14]: Pearl's rule increases validity (expected value) and Jeffrey's rule decreases divergence.

In the present context we need only Jeffrey's rule and refer to [8,12] for Pearl's counterpart. In the theorem below we first repeat (from [14]) the formulation of Jeffrey's rule in terms of the dagger of a channel (4), together with the associated decrease of the divergence. The second item is new and contains a generalisation of Jeffrey's rule to multiple channels and data distributions. The latter are typically obtained via frequentist learning *Flrn*, see Sects. 4 and 5. The appendix contains a proof.

Kullback-Leibler divergence D_{KL} is a standard comparison of distributions on the same set. It is defined, for $\omega, \rho \in \mathcal{D}(X)$, via the natural logarithm ln:

$$D_{KL}(\omega, \rho) \coloneqq \sum_{x \in X} \omega(x) \cdot \ln\left(\frac{\omega(x)}{\rho(x)}\right). \tag{5}$$

Jeffrey's rule reduces the divergence between data and prediction. In the cognitive context of predictive coding [9,11,23] this is called 'error correction'.

Theorem 1. *Let $\omega \in \mathcal{D}(X)$ be a distribution, used as prior.*

1. *("Jeffrey's divergence decrease") For a channel $c\colon X \rightsquigarrow Y$ and a 'data' distribution $\tau \in \mathcal{D}(Y)$,*

$$D_{KL}\big(\tau, \, c \gg\!= \omega\big) \;\geq D_{KL}\big(\tau, \, c \gg\!= \omega'\big) \quad \textit{where} \quad \omega' := c^{\dagger}_{\omega} \gg\!= \tau. \tag{6}$$

This mapping $\omega \mapsto \omega' := c^{\dagger}_{\omega} \gg\!= \tau$ is Jeffrey's update rule, giving ω' as updated, posterior distribution.

2. *("Mixture divergence decrease") Let $c_i\colon X \rightsquigarrow Y_i$ be a finite collection of channels with distributions $\tau_i \in \mathcal{D}(Y_i)$ and probabilities $r_i \in [0,1]$ satisfying $\sum_i r_i = 1$. Then:*

$$\begin{aligned} \sum\nolimits_i r_i \cdot D_{KL}\big(\tau_i, \, c_i \gg\!= \omega\big) \;\geq \sum\nolimits_i r_i \cdot D_{KL}\big(\tau_i, \, c_i \gg\!= \omega'\big) \\ \textit{where} \quad \omega' := \sum\nolimits_i r_i \cdot \Big((c_i)^{\dagger}_{\omega} \gg\!= \tau_i \Big). \end{aligned} \tag{7}$$

Proof. We refer to [14] for the details of the (non-trivial) proof of the divergence decrease for Jeffrey's update rule (6). It crucially depends on (8) below. Let $\omega \in \mathcal{D}(X)$ be a distribution with predicates $p_1, \ldots, p_n \in [0,1]^X$ satisfying $\sum_i p_i = \mathbf{1}$, pointwise, and with probabilities $r_1, \ldots, r_n \in [0,1]$ satisfying $\sum_i r_i = 1$. Assuming non-zero validities $\omega \models p_i$, for each i, one has:

$$\sum\nolimits_i \frac{r_i \cdot (\omega \models p_i)}{\sum_j r_j \cdot (\omega|_{p_j} \models p_i)} \;\leq 1. \tag{8}$$

See [14] for details about the validity (expected value) $\omega \models p$ of a predicate p w.r.t. a distribution ω, and about the updated distribution $\omega|_p$.

We will use the inequality (8) to prove the second point of the theorem. We use the disjoint union $K := \coprod_i Y_i$ as index set and with predicates and probabilities, for $(i, y) \in K$,

$$p_{(i,y)} := c_i =\!\ll \mathbf{1}_y = c_i(-)(y) \, \in \, [0,1]^X \quad s_{(i,y)} := r_i \cdot \tau_i(y) \, \in \, [0,1].$$

The proof of (7) works as follows, basically as in [14], but with an extra level of indexing, via the index set K. Recall in the mixture case the updated distribution $\omega' := \sum_i r_i \cdot \big((c_i)^{\dagger}_{\omega} \gg\!= \tau_i\big)$.

$$\begin{aligned} & \sum\nolimits_i r_i \cdot D_{KL}\big(\tau_i, \, c_i \gg\!= \omega'\big) \;-\; \sum\nolimits_i r_i \cdot D_{KL}\big(\tau_i, \, c_i \gg\!= \omega\big) \\ & \overset{(5)}{=} \sum\nolimits_i r_i \cdot \sum_{y \in Y_i} \tau_i(y) \cdot \left[\ln\left(\frac{\tau_i(y)}{(c_i \gg\!= \omega')(y)} \right) \;-\; \ln\left(\frac{\tau_i(y)}{(c_i \gg\!= \omega)(y)} \right) \right] \\ & = \sum_{(i,y) \in K} s_{(i,y)} \cdot \ln\left(\frac{(c_i \gg\!= \omega)(y)}{(c_i \gg\!= \omega')(y)} \right) \\ & = \sum_{(i,y) \in K} s_{(i,y)} \cdot \ln\left(\frac{\omega \models c_i =\!\ll \mathbf{1}_y}{\omega' \models c_i =\!\ll \mathbf{1}_y} \right) \\ & \leq \ln\left(\sum_{(i,y) \in K} s_{(i,y)} \cdot \frac{\omega \models p_{(i,y)}}{\omega' \models p_{(i,y)}} \right) \qquad \text{by Jensen's inequality} \\ & \overset{(*)}{\leq} \ln\big(1\big) \;=\; 0. \end{aligned}$$

The marked inequality $\overset{(*)}{\leq}$ uses (8). It applies since for $(i, y) \in K$,

$$\begin{aligned}
\omega' \models p_{(i,y)} &= \sum\nolimits_j r_j \cdot \big((c_j)^\dagger_\omega \gg\!= \tau_j\big) \models p_{(i,y)} \\
&= \sum\nolimits_j r_j \cdot \sum_{x \in X} \big((c_j)^\dagger_\omega \gg\!= \tau_j\big)(x) \cdot p_{(i,y)}(x) \\
&= \sum\nolimits_j r_j \cdot \sum_{x \in X} \sum_{z \in Y_j} (c_j)^\dagger_\omega(z)(x) \cdot \tau_j(z) \cdot p_{(i,y)}(x) \\
&\overset{(**)}{=} \sum_{(j,z) \in K} s_{(j,k)} \cdot \sum_{x \in X} \omega|_{c_j =\ll \mathbf{1}_z}(x) \cdot p_{(i,y)}(x) \\
&= \sum_{(j,z) \in K} s_{(j,k)} \cdot \big(\omega|_{p_{(j,z)}} \models p_{(i,y)}\big).
\end{aligned}$$

The equation $\overset{(**)}{=}$ uses that the dagger definition (4) can equivalently be described as an update: $(c_j)^\dagger_\omega(z) = \omega|_{c_j =\ll \mathbf{1}_z}$, see [14] for details. $\square$

We include an illustration of Jeffrey's rule, as in the above first item.

Example 2. The following update question is attributed to Jeffrey, and reproduced for instance in [3,6]. It involves three colors of clothes: green (g), blue (b) and violet (v), in a space $C = \{g, b, v\}$. Clothes can be sold or not, as represented by $S = \{s, s^\perp\}$. The prior sales distribution $\omega \in \mathcal{D}(S)$ is $\omega = \frac{14}{25}|s\rangle + \frac{11}{25}|s^\perp\rangle$; it tells that a bit more than half of the clothes are sold. The colour distributions for sales and non-sales are provided via a channel $c\colon S \to \mathcal{D}(C)$, of the form:

$$c(s) = \tfrac{3}{14}|g\rangle + \tfrac{3}{14}|b\rangle + \tfrac{4}{7}|v\rangle \quad c(s^\perp) = \tfrac{9}{22}|g\rangle + \tfrac{9}{22}|b\rangle + \tfrac{2}{11}|v\rangle.$$

A cloth is inspected by candlelight and the following likelihoods are reported per color: 70% certainty that it is green, 25% that it is blue, and 5% that it is violet. This gives a data/evidence distribution $\tau = \frac{7}{10}|g\rangle + \frac{1}{4}|b\rangle + \frac{1}{20}|v\rangle \in \mathcal{D}(C)$. We ask: what is the likelihood that the observed cloth will be sold?

The push-forward colour distribution $c \gg\!= \omega$ with its prior divergence from the data are:

$$c \gg\!= \omega \overset{(3)}{=} \tfrac{3}{10}|g\rangle + \tfrac{3}{10}|b\rangle + \tfrac{2}{5}|v\rangle \quad D_{KL}\big(\tau,\, c \gg\!= \omega\big) \overset{(5)}{=} 0.444.$$

The formula (4) determines the dagger channel $d := c^\dagger_\omega\colon C \to \mathcal{D}(S)$ as:

$$d(g) = \tfrac{2}{5}|s\rangle + \tfrac{3}{5}|s^\perp\rangle \quad d(b) = \tfrac{2}{5}|s\rangle + \tfrac{3}{5}|s^\perp\rangle \quad d(v) = \tfrac{4}{5}|s\rangle + \tfrac{1}{5}|s^\perp\rangle.$$

We then get as updated (posterior) sales distribution $\omega' := d \gg\!= \tau \in \mathcal{D}(S)$ with decreased divergence:

$$\omega' := d \gg\!= \tau = \tfrac{21}{50}|s\rangle + \tfrac{29}{50}|s^\perp\rangle \qquad \text{now with } D_{KL}\big(\tau,\, c \gg\!= \omega'\big) = 0.368.$$

The posterior sale probability $\frac{21}{50}$ for the inspected cloth is lower than the prior probability $\frac{14}{25} = \frac{28}{50}$. This outcome also occurs in [3, Ex. 1], [6, p.41] (as marginal), but without the above dagger-channel and the divergence decrease.

4 Expectation Maximisation (EM)

Expectation Maximisation (EM) is an algorithm where two steps, called E-step and M-step are alternated and iterated, as in E-M-E-M-E-M-$\cdots$, until some fixed point is reached. Its first general formulation occurs in [7], but it was used in more specialised forms before, see [18, 1.8] for historical details. In general, EM seeks an appropriate mixture of hidden/latent variables together with appropriate parameter values in a statistical model, see [20].

Here we describe the model and algorithm in channel-based form, where the divergence between data and predictions decreases with every iteration. The setting involves a channel, with a 'mixture' distribution on its domain and a 'data' multiset on its codomain. The channel will have type $Z \rightsquigarrow Y$, where Z is the space of classifications, and Y is the data space. Typically, the channel is determined by a parameter θ, which we write as $c[\theta]\colon Z \rightsquigarrow Y$. This θ may be a single number, a list of numbers, or even a matrix, of some dimension.

Theorem 3. *Let a 'data' multiset $\psi \in \mathcal{M}(Y)$ be given. We consider an initial 'mixture' distribution $\omega^{(0)} \in \mathcal{D}(Z)$ and a family of channels $c[\theta]\colon Z \rightsquigarrow Y$, with parameter θ, having an initial value $\theta^{(0)}$.*

Consider the following two steps at stage $n \in \mathbb{N}$, to produce new distributions and channels, assuming that we already have a distribution $\omega^{(n)} \in \mathcal{D}(Z)$ and channel $c^{(n)} := c[\theta^{(n)}]\colon Z \rightsquigarrow Y$, for parameter value $\theta^{(n)}$.

E-step. *Using Jeffrey's update rule, from Theorem 1 (1), we obtain a next mixture distribution as:*

$$\omega^{(n+1)} := c[\theta^{(n)}]^{\dagger}_{\omega^{(n)}} \gg\!= Flrn(\psi) \;\in\; \mathcal{D}(Z). \tag{9}$$

M-step. *We pick as next channel-parameter value the one with minimal Kullback-Leibler divergence in:*

$$\theta^{(n+1)} \;\in\; \underset{\theta}{\operatorname{argmin}}\, D_{KL}\Big(Flrn(\psi),\, c[\theta] \gg\!= \omega^{(n)}\Big). \tag{10}$$

Take $c^{(n+1)} := c[\theta^{(n+1)}]$ as next channel.

These two steps result in decreasing divergences.

1. *Each iteration yields a decrease of Kullback-Leibler divergence:*

$$D_{KL}\Big(Flrn(\psi),\, c^{(n+1)} \gg\!= \omega^{(n+1)}\Big) \leq D_{KL}\Big(Flrn(\psi),\, c^{(n)} \gg\!= \omega^{(n)}\Big). \tag{11}$$

This means that the predicted data distribution is decreasingly wrong.

2. *A next parameter $\theta^{(n+1)}$ can (often) be found as solution to the equation:*

$$\sum_{z\in Z,\, y\in Y} \psi(y) \cdot \big(c^{(n)}\big)^{\dagger}_{\omega^{(n)}}(y)(z) \cdot \frac{\mathrm{d}}{\mathrm{d}\theta} \ln\Big(c[\theta](z)(y)\Big) = 0. \tag{12}$$

This solution is not the minimal one in (10), but it still yields the relevant decrease of divergence in (11).

The word 'often' is inserted because finding a minimal parameter value via a solution of (12) only works when the channel has suitable (partial) derivatives, see Example 4 below.

Proof 1. The claimed decrease of divergence arises as follows.

$$\begin{aligned}
& D_{KL}\Big(Flrn(\psi),\, c[\theta^{(n+1)}] \gg\!= \omega^{(n+1)}\Big) \\
& \leq D_{KL}\Big(Flrn(\psi),\, c[\theta^{(n)}] \gg\!= \omega^{(n+1)}\Big) && \text{since } \theta^{(n+1)} \text{ is argmin} \\
& \leq D_{KL}\Big(Flrn(\psi),\, c[\theta^{(n)}] \gg\!= \big(c[\theta^{(n)}]^{\dagger}_{\omega^{(n)}} \gg\!= Flrn(\psi)\big)\Big) && \text{by definition of } \omega^{(n+1)} \\
& \leq D_{KL}\Big(Flrn(\psi),\, c[\theta^{(n)}] \gg\!= \omega^{(n)}\Big) && \text{by Theorem 1 (1).}
\end{aligned}$$

2. The minimum parameter value θ in the expression $D_{KL}\Big(Flrn(\psi),\, c[\theta] \gg\!= \omega^{(n)}\Big)$ in (10) is located where the derivative $\frac{\mathrm{d}}{\mathrm{d}\theta}$ is zero. We thus calculate:

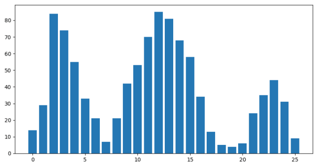
$$\begin{aligned}
& \frac{\mathrm{d}}{\mathrm{d}\theta} D_{KL}\Big(Flrn(\psi),\, c[\theta] \gg\!= \omega^{(n)}\Big) \\
& \overset{(5)}{=} \frac{\mathrm{d}}{\mathrm{d}\theta} \sum_{y\in Y} Flrn(\psi)(y) \cdot \ln\left(\frac{Flrn(\psi)(y)}{(c[\theta] \gg\!= \omega^{(n)})(y)}\right) \\
& = \frac{\mathrm{d}}{\mathrm{d}\theta} \sum_{y\in Y} Flrn(\psi)(y) \cdot \ln\Big(Flrn(\psi)(y)\Big) - \sum_{y\in Y} Flrn(\psi)(y) \cdot \ln\Big((c[\theta] \gg\!= \omega^{(n)})(y)\Big) \\
& = \frac{-1}{\|\psi\|} \cdot \sum_{y\in Y} \psi(y) \cdot \frac{\mathrm{d}}{\mathrm{d}\theta} \ln\Big((c[\theta] \gg\!= \omega^{(n)})(y)\Big) \\
& = \frac{-1}{\|\psi\|} \cdot \sum_{y\in Y} \frac{\psi(y)}{(c[\theta] \gg\!= \omega^{(n)})(y)} \cdot \frac{\mathrm{d}}{\mathrm{d}\theta} (c[\theta] \gg\!= \omega^{(n)})(y) \\
& = \frac{-1}{\|\psi\|} \cdot \sum_{y\in Y} \frac{\psi(y)}{(c[\theta] \gg\!= \omega^{(n)})(y)} \cdot \frac{\mathrm{d}}{\mathrm{d}\theta} \sum_{z\in Z} c[\theta](z)(y) \cdot \omega^{(n)}(z) \\
& = \frac{-1}{\|\psi\|} \cdot \sum_{z\in Z,\, y\in Y} \frac{\psi(y) \cdot \omega^{(n)}(z)}{(c[\theta] \gg\!= \omega^{(n)})(y)} \cdot \frac{\mathrm{d}}{\mathrm{d}\theta} c[\theta](z)(y) \\
& = \frac{-1}{\|\psi\|} \cdot \sum_{z\in Z,\, y\in Y} \frac{\psi(y) \cdot \omega^{(n)}(z) \cdot c[\theta](z)(y)}{(c[\theta] \gg\!= \omega^{(n)})(y)} \cdot \frac{\mathrm{d}}{\mathrm{d}\theta} \ln\Big(c[\theta](z)(y)\Big) \\
& \overset{(4)}{=} \frac{-1}{\|\psi\|} \cdot \sum_{z\in Z,\, y\in Y} \psi(y) \cdot c[\theta]^{\dagger}_{\omega^{(n)}}(y)(z) \cdot \frac{\mathrm{d}}{\mathrm{d}\theta} \ln\Big(c[\theta](z)(y)\Big). \qquad (*)
\end{aligned}$$

At this stage we need two more observations to see why it suffices to solve the Eq. (12).

(a) The leading factor $\frac{-1}{\|\psi\|}$ can be dropped from the above last line $(*)$ when we seek a solution via setting it to zero; because of the minus sign -1, we are not looking for a minimum, but for a maximum.

(b) The first θ in the dagger expression $c[\theta]^{\dagger}_{\omega^{(n)}}$ in $(*)$ can be replaced by $\theta^{(n)}$, which turns $c[\theta]^{\dagger}_{\omega^{(n)}}$ into $\big(c^{(n)}\big)^{\dagger}_{\omega^{(n)}}$, as in (12). This is a subtle point. As we can see in the four line proof of Theorem 3 (1), we only need that the solution $\theta^{(n+1)}$ yields a divergence that is less than the divergence for $\theta^{(n)}$. Hence if one of the θ's in $(*)$ equals $\theta^{(n)}$, we do not get the real minimum divergence for $\theta^{(n+1)}$, but we still get a divergence that is less than the one for $\theta^{(n)}$. $\square$

Example 4. Consider the histogram of 1000 data elements on the right, on the space $\{0, 1, \ldots, N\}$ for $N = 25$. The shape of the data suggests that we have a mixture of three binomials at hand. Indeed, we have obtained these data by sampling 1000 times from the mixture of binomials:



$$\tfrac{1}{2} \cdot bn[N]\big(\tfrac{1}{2}\big) \,+\, \tfrac{1}{3} \cdot bn[N]\big(\tfrac{1}{8}\big) \,+\, \tfrac{1}{6} \cdot bn[N]\big(\tfrac{9}{10}\big). \tag{13}$$

Our aim in this example is to see if we can recover the mixture weights $(\frac{1}{2}, \frac{1}{3}, \frac{1}{6})$ and the biases $(\frac{1}{2}, \frac{1}{8}, \frac{9}{10})$ in (13) from these sampled data, via EM as described in Theorem 3. Formally, the above plot is used as a multiset $\psi = 14|0\rangle + 29|1\rangle + \cdots + 9|25\rangle \in \mathcal{M}[1000]\big(\{0, \ldots, 25\}\big)$.

We take a three element latent space, say $Z = \{1, 2, 3\}$, together with a parameterised channel $c[\boldsymbol{\theta}] \colon Z \rightsquigarrow \{0, 1, \ldots, N\}$, in this situation with $N = 25$. The channel $c[\boldsymbol{\theta}]$ consists of three binomial distributions, with a 3-tuple $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3) \in [0, 1]^3$ as parameter, via $c[\boldsymbol{\theta}](i) := bn[N](\theta_i)$.

In this situation we illustrate how to solve Eq. (12), where, for convience we abbreviate the dagger channel as $d_n := \big(c^{(n)}\big)^{\dagger}_{\omega^{(n)}} \colon \{0, \ldots, N\} \rightsquigarrow Z$. We look at the solution for partial derivatives, for each $i \in Z$, using the familiar equation $\frac{\partial}{\partial x} \ln(x) = \frac{1}{x}$, plus the fact that the logarithm turns products into sums:

$$\begin{aligned} 0 &\overset{(12)}{=} \sum_{z \in \{1,2,3\}} \sum_{k \in \{0,\ldots,N\}} \psi(k) \cdot d_n(k)(z) \cdot \frac{\partial}{\partial \theta_i} \ln\Big(c[\boldsymbol{\theta}](z)(k)\Big) \\ &\overset{(2)}{=} \sum_{z \in \{1,2,3\}} \sum_{k \in \{0,\ldots,N\}} \psi(k) \cdot d_n(k)(z) \cdot \frac{\partial}{\partial \theta_i} \ln\Big(\tbinom{N}{k} \cdot \theta_z^k \cdot (1 - \theta_z)^{N-k}\Big) \\ &= \sum_{k \in \{0,\ldots,N\}} \psi(k) \cdot d_n(k)(i) \cdot \left[\frac{k}{\theta_i} - \frac{N-k}{1-\theta_i}\right]. \end{aligned}$$

Via some elementary arithmethic we now get as solution:

$$\theta_i = \frac{\sum_k \psi(k) \cdot k \cdot d_n(k)(i)}{N \cdot \sum_k \psi(k) \cdot d_n(k)(i)}. \tag{*}$$

At this stage we can put things together and give a concrete description of the EM-algorithm (for the current example).

1. Pick arbitrary $\omega^{(0)} \in \mathcal{D}(Z) = \mathcal{D}(\{1, 2, 3\})$ and $\boldsymbol{\theta}^{(0)} \in [0, 1]^3$;
2. Assume $\omega^{(n)} \in \mathcal{D}(Z)$ and $\boldsymbol{\theta}^{(n)} \in [0, 1]^3$ are already computed and use them first to form the dagger channel $d_n := c\big[\boldsymbol{\theta}^{(n)}\big]^{\dagger}_{\omega^{(n)}} \colon \{0, \ldots, N\} \rightsquigarrow Z$ as in (4).

(E). Take the next mixture distribution $\omega^{(n+1)} \in \mathcal{D}(Z)$ via Jeffrey's rule:

$$\omega^{(n+1)}(i) \overset{(9)}{=} \Big(d_n \gg\!= \mathit{Flrn}(\psi)\Big)(i) = \frac{1}{\|\psi\|} \cdot \sum_{k \in \{0,\ldots,N\}} d_n(k)(i) \cdot \psi(k).$$

(M). Take the next parameters $\boldsymbol{\theta}^{(n+1)} \in [0,1]^3$ as:

$$\theta_i^{(n+1)} \overset{(*)}{=} \frac{\sum_k \psi(k) \cdot k \cdot d_n(k)(i)}{N \cdot \sum_k \psi(k) \cdot d_n(k)(i)} = \frac{\sum_k \mathit{Flrn}(\psi)(k) \cdot k \cdot d_n(k)(i)}{N \cdot \omega^{(n+1)}(i)}.$$

The table below gives an overview of five runs of this algorithm, starting from arbitrary values. Clearly, the divergences are decreasing, as prescribed in (11).

round	**KL-div**	**mixtures** $\omega^{(n)}$	**biases** $\boldsymbol{\theta}^{(n)}$
0	0.853	$0.477|1\rangle + 0.354|2\rangle + 0.169|3\rangle$	0.235, 0.389, 0.691
1	0.326	$0.353|1\rangle + 0.35|2\rangle + 0.297|3\rangle$	0.159, 0.46, 0.754
2	0.132	$0.321|1\rangle + 0.454|2\rangle + 0.225|3\rangle$	0.128, 0.478, 0.812
3	0.029	$0.311|1\rangle + 0.515|2\rangle + 0.174|3\rangle$	0.122, 0.488, 0.872
4	0.011	$0.309|1\rangle + 0.535|2\rangle + 0.156|3\rangle$	0.121, 0.493, 0.898

We see that in five rounds we already get quite close to the original mixture and biases in (13). The order is different, but this is because the classification labels in $Z = \{1, 2, 3\}$ are meaningless and cannot be distinguished by the algorithm.

5 Latent Dirichlet Allocation (LDA)

The second model and inference algorithm in this paper was introduced in [2] under the name Latent Dirichlet Allocation, commonly abbreviated as LDA. It is used for what is called topic modeling: classifying documents according to their topics. The set-up of the algorithm is more complicated then EM and involves continuous Dirichlet distributions. In our analysis we show that LDA is essentially about divergence reduction via Jeffrey's rule—in multi-channel form, as in Theorem 1 (2). The Dirichlet distributions introduce a certain level of complexity, but turn out to play a limited role in the algorithm itself. We cover the essentials and refer to the literature for further information (see *e.g.* [20,21]).

Dirichlet is a continuous distribution on discrete distributions. Writing $\mathcal{G}$ for the Giry monad of continuous distributions, we have $\mathit{Dir}(\alpha) \in \mathcal{G}\big(\mathcal{D}(X)\big)$, where X is a finite set and $\alpha \in \mathcal{M}(X)$ is a multiset with full support[1]. This $\mathit{Dir}(\alpha)$ is

[1] In the current paper we use multisets with natural numbers as multiplicities; this can be generalised to non-negative real numbers as multiplicities. The Dirichlet distribution $\mathit{Dir}(\alpha)$ can be defined for such more general multisets. But we shall not do so here since it does not affect the LDA algorithm.

defined via a probability density function (pdf) $dir(\alpha)\colon \mathcal{D}(X) \to \mathbb{R}_{\geq 0}$, namely:

$$dir(\alpha)(\omega) := \frac{(\|\alpha\| - 1)!}{\prod_x (\alpha(x) - 1)!} \cdot \prod_{x \in X} \omega(x)^{\alpha(x)-1}.$$

The continuous Dirichlet distribution $Dir(\alpha) \in \mathcal{G}\big(\mathcal{D}(X)\big)$ is the function that assigns to a measurable subset $M \subseteq \mathcal{D}(X)$ the probability $\int_{\omega \in M} dir(\alpha)(\omega)\, \mathrm{d}\omega$.

We assume a finite set W of words and use the bag-of-words model for documents, so that a document is a multiset $\psi \in \mathcal{M}(W)$ over words. As data we use a collection of such documents/multisets, written as $\boldsymbol{\psi} = \big(\psi_i\big)_{i \in I}$, for some finite index set I. We shall write it as $\boldsymbol{\psi} \in \mathcal{M}(W)^I$ and call it a corpus.

We also assume a finite set T of topics. This may simply be a set $\boldsymbol{n} := \{0, 1, \ldots, n-1\}$, since topics do not have an interpretation.

We shall use multisets $\alpha \in \mathcal{M}(T)$ and $\beta \in \mathcal{M}(W)$ as parameters for Dirichlet distributions $Dir(\alpha) \in \mathcal{G}\big(\mathcal{D}(T)\big)$ and $Dir(\beta) \in \mathcal{G}\big(\mathcal{D}(W)\big)$, where $\alpha \in \mathcal{M}(T)$ and $\beta \in \mathcal{M}(W)$ are multisets with full support. We put them in parallel, using the tensor $\otimes$ for continuous distributions, and thus get:

$$Dir(\alpha)^I := \underbrace{Dir(\alpha) \otimes \cdots \otimes Dir(\alpha)}_{|\,I\,|\text{ times}} \in \mathcal{G}\Big(\underbrace{\mathcal{D}(T) \times \cdots \times \mathcal{D}(T)}_{|\,I\,|\text{ times}}\Big) = \mathcal{G}\big(\mathcal{D}(T)^I\big).$$

Similarly, we use $Dir(\beta)^T \in \mathcal{G}\big(\mathcal{D}(W)^T\big)$. These parallel products $\otimes$ of continuous distributions work via the multiplication of the pdf's involved, see *e.g.* [21].

These parallel Dirichlet's are used as (continuous) distributions on $\boldsymbol{\theta} \in \mathcal{D}(T)^I$ and $\boldsymbol{\zeta} \in \mathcal{D}(W)^T$, that is on a document-topic channel $\boldsymbol{\theta}\colon I \to \mathcal{D}(T)$ and a topic-word channel $\boldsymbol{\zeta}\colon T \to \mathcal{D}(W)$. This $\boldsymbol{\theta}$ sends a document (index) $i \in I$ to the topic distribution $\boldsymbol{\theta}(i) \in \mathcal{D}(T)$ for document $\psi_i \in \mathcal{M}(W)$. Similarly, $\boldsymbol{\zeta}$ sends a topic $t \in T$ to the distribution $\boldsymbol{\zeta}(t) \in \mathcal{D}(W)$ of words, for the topic t.

The LDA model consists of the following composite, where *mn* is multinomial.

$$\mathcal{D}(T)^I \times \mathcal{D}(W)^T \xrightarrow{comp} \mathcal{D}(W)^I \xrightarrow{mn^I} \mathcal{D}\big(\mathcal{M}(W)^I\big)$$

The function *comp* performs channel composition: $comp\big(\boldsymbol{\theta}, \boldsymbol{\zeta}\big) = \boldsymbol{\zeta} \circ\!\!\cdot\, \boldsymbol{\theta}\colon I \to \mathcal{D}(W)$. The likelihood for document data $\boldsymbol{\psi} \in \mathcal{M}(W)^I$, given hyperparameters α, β is expressed by the (continuous) push forward:

$$\Big((mn^I \circ comp) \gg\!\!= \big(Dir(\alpha)^I \otimes Dir(\beta)^T\big)\Big)(\boldsymbol{\psi}) \in [0, 1]. \qquad (14)$$

We can write the expression (14) in terms of integrals:

$$\int_{\boldsymbol{\theta}\in\mathcal{D}(T)^I}\int_{\boldsymbol{\zeta}\in\mathcal{D}(W)^T}\prod_{i\in I} dir(\alpha)\big(\boldsymbol{\theta}(i)\big)\cdot\prod_{t\in T} dir(\beta)\big(\boldsymbol{\zeta}(t)\big)\cdot\prod_{i\in I} mn\big(\boldsymbol{\zeta} \gg\!= \boldsymbol{\theta}(i)\big)(\psi_i)\,\mathrm{d}\boldsymbol{\zeta}\,\mathrm{d}\boldsymbol{\theta}.$$

We are interested in the likelihood expression with $\boldsymbol{\theta}$ and $\boldsymbol{\zeta}$ as free variables:

$$\mathcal{L}_{\alpha,\beta,\boldsymbol{\psi}}\big(\boldsymbol{\theta},\boldsymbol{\zeta}\big) \coloneqq \prod_{i\in I} dir(\alpha)\big(\boldsymbol{\theta}(i)\big)\cdot\prod_{t\in T} dir(\beta)\big(\boldsymbol{\zeta}(t)\big)\cdot\prod_{i\in I} mn\big(\boldsymbol{\zeta} \gg\!= \boldsymbol{\theta}(i)\big)(\psi_i). \tag{15}$$

The LDA aim is to find the document-topic channel $\boldsymbol{\theta}\colon I \to \mathcal{D}(T)$ and the topic-word $\boldsymbol{\zeta}\colon T \to \mathcal{D}(W)$ that maximise this likelihood expression (15).

We shall use the (natural) logarithm ln of this expression, commonly called the log-likelihood; it turns the above products $\prod$ into sums $\sum$. Since ln is monotone, we might as well maximise the log-likelihood. A crucial observation is that this log-likelihood can be formulated in terms of Kullback-Leibler divergence. This opens the door to applying Jeffrey's update rule.

Lemma 5. *Let $\alpha \in \mathcal{M}(T)$ and $\beta \in \mathcal{M}(W)$ be multisets with full support and let $\boldsymbol{\psi} \in \mathcal{M}(W)^I$ be corpus of documents. The log-likelihood $\ln \mathcal{L}_{\alpha,\beta,\boldsymbol{\psi}}\big(\boldsymbol{\theta},\boldsymbol{\zeta}\big)$ of the expression (15) can be written as:*

$$\begin{aligned}
&\ln \mathcal{L}_{\alpha,\beta,\boldsymbol{\psi}}\big(\boldsymbol{\theta},\boldsymbol{\zeta}\big)\\
&= C - \sum_{i\in I}\big(\|\alpha-\mathbf{1}\| + \|\psi_i\|\big)\cdot\Bigg(\frac{\|\alpha-\mathbf{1}\|}{\|\alpha-\mathbf{1}\|+\|\psi_i\|}\cdot D_{KL}\Big(\mathit{Flrn}(\alpha-\mathbf{1}),\,\boldsymbol{\theta}(i)\Big)\\
&\qquad\qquad + \frac{\|\psi_i\|}{\|\alpha-\mathbf{1}\|+\|\psi_i\|}\cdot D_{KL}\Big(\mathit{Flrn}(\psi_i),\,\boldsymbol{\zeta} \gg\!= \boldsymbol{\theta}(i)\Big)\Bigg)\\
&\qquad - \sum_{t\in T}\|\beta-\mathbf{1}\|\cdot D_{KL}\Big(\mathit{Flrn}(\beta-\mathbf{1}),\,\boldsymbol{\zeta}(t)\Big),
\end{aligned} \tag{16}$$

where C is a constant depending on the parameters $\alpha, \beta, \boldsymbol{\psi}$ but not on the variables $\boldsymbol{\theta}, \boldsymbol{\zeta}$. Recall that $\mathbf{1}$ is the multiset of singletons, so that $(\alpha-\mathbf{1})(x) = \alpha(x)-1$. This subtraction is allowed since α has full support. The same holds for β.

Proof (of Lemma 5). We apply the logarithm ln to (15), expand the Dirichlet and multinomial expressions, and write C for some constant, not depending on

$\boldsymbol{\theta}, \boldsymbol{\zeta}$.

$$\begin{aligned}
&\ln \mathcal{L}_{\alpha,\beta,\psi}\left(\boldsymbol{\theta}, \boldsymbol{\zeta}\right) \\
&= \sum_{i \in I} \ln\left(\frac{(\|\alpha\|-1)!}{\prod_t (\alpha(t)-1)!}\right) + \sum_{t \in T} (\alpha(t)-1) \cdot \ln\big(\boldsymbol{\theta}(i)(t)\big) \\
&\qquad + \sum_{t \in T} \ln\left(\frac{(\|\beta\|-1)!}{\prod_w (\beta(w)-1)!}\right) + \sum_{w \in W} (\beta(w)-1) \cdot \ln\big(\boldsymbol{\zeta}(t)(w)\big) \\
&\qquad + \sum_{i \in I} \ln\big((\psi_i)\big) + \sum_{w \in W} \psi_i(w) \cdot \ln\Big(\big(\boldsymbol{\zeta} \gg\!= \boldsymbol{\theta}(i)\big)(w)\Big). \\
&= C + \sum_{i \in I} \|\alpha - \mathbf{1}\| \cdot \sum_{t \in T} \mathit{Flrn}(\alpha - \mathbf{1})(t) \cdot \ln\big(\boldsymbol{\theta}(i)(t)\big) \\
&\qquad\quad - \|\alpha - \mathbf{1}\| \cdot \sum_{t \in T} \mathit{Flrn}(\alpha - \mathbf{1})(t) \cdot \ln\big(\mathit{Flrn}(\alpha - \mathbf{1})(t)\big) \\
&\qquad + \sum_{t \in T} \|\beta - \mathbf{1}\| \cdot \sum_{w \in W} \mathit{Flrn}(\beta - \mathbf{1})(w) \cdot \ln\big(\boldsymbol{\zeta}(t)(w)\big) \\
&\qquad\quad - \|\beta - \mathbf{1}\| \cdot \sum_{w \in W} \mathit{Flrn}(\beta - \mathbf{1})(w) \cdot \ln\big(\mathit{Flrn}(\beta - \mathbf{1})(w)\big) \\
&\qquad + \sum_{i \in I} \|\psi_i\| \cdot \sum_{w \in W} \mathit{Flrn}(\psi_i)(w) \cdot \ln\Big(\big(\boldsymbol{\zeta} \gg\!= \boldsymbol{\theta}(i)\big)(w)\Big) \\
&\qquad\quad - \|\psi_i\| \cdot \sum_{w \in W} \mathit{Flrn}(\psi_i)(w) \cdot \ln\big(\mathit{Flrn}(\psi_i)(w)\big) \\
&= C - \sum_{i \in I} \|\alpha - \mathbf{1}\| \cdot D_{KL}\Big(\mathit{Flrn}(\alpha - \mathbf{1}),\, \boldsymbol{\theta}(i)\Big) + \|\psi_i\| \cdot D_{KL}\Big(\mathit{Flrn}(\psi_i),\, \boldsymbol{\zeta} \gg\!= \boldsymbol{\theta}(i)\Big) \\
&\qquad - \sum_{t \in T} \|\beta - \mathbf{1}\| \cdot D_{KL}\Big(\mathit{Flrn}(\beta - \mathbf{1}),\, \boldsymbol{\zeta}(t)\Big) \\
&= C - \sum_{i \in I} r_i \cdot \Big(r_{i,1} \cdot D_{KL}\Big(\mathit{Flrn}(\alpha - \mathbf{1}),\, \boldsymbol{\theta}(i)\Big) + r_{i,2} \cdot D_{KL}\Big(\mathit{Flrn}(\psi_i),\, \boldsymbol{\zeta} \gg\!= \boldsymbol{\theta}(i)\Big)\Big) \\
&\qquad - \sum_{t \in T} \|\beta - \mathbf{1}\| \cdot D_{KL}\Big(\mathit{Flrn}(\beta - \mathbf{1}),\, \boldsymbol{\zeta}(t)\Big).
\end{aligned}$$

In the last line we use the abbreviations (21). □

This lemma tells us that in order to *maximise* the log-likelihood we have to *minimise* the three Kullback-Leibler divergences in (16), because of the minus sign $-$ before the D_{KL} expressions.

Theorem 6. *Consider the LDA situation as described above, with multiset parameters* $\alpha \in \mathcal{M}(T)$ *and* $\beta \in \mathcal{M}(W)$ *and a corpus of documents* $\boldsymbol{\psi} = \big(\psi_i\big)_{i \in I}$.

An infinite series of channels $\boldsymbol{\theta}^{(n)} \in \mathcal{D}(T)^I$ *and* $\boldsymbol{\zeta}^{(n)} \in \mathcal{D}(W)^T$ *with increasing likelihoods:*

$$\mathcal{L}_{\alpha,\beta,\psi}\left(\boldsymbol{\theta}^{(n+1)}, \boldsymbol{\zeta}^{(n+1)}\right) \geq \mathcal{L}_{\alpha,\beta,\psi}\left(\boldsymbol{\theta}^{(n)}, \boldsymbol{\zeta}^{(n)}\right) \tag{17}$$

is obtained in the following manner.

At stage 0*, arbitrary channels* $\boldsymbol{\theta}^{(0)} \in \mathcal{D}(T)^I$ *and* $\boldsymbol{\zeta}^{(0)} \in \mathcal{D}(W)^T$ *are chosen. Subsequent stages are handled as follows.*

1. *The next document-topic channel* $\boldsymbol{\theta}^{(n+1)} \in \mathcal{D}(T)^I$ *is defined via the mixture version of Jeffrey's rule in Theorem 1 (2), as convex combination, at* $i \in I$*:*

$$\begin{aligned}\boldsymbol{\theta}^{(n+1)}(i) \coloneqq \frac{\|\alpha - \mathbf{1}\|}{\|\alpha - \mathbf{1}\| + \|\psi_i\|} \cdot \mathit{Flrn}(\alpha - \mathbf{1}) \\ + \frac{\|\psi_i\|}{\|\alpha - \mathbf{1}\| + \|\psi_i\|} \cdot \Big(\big(\boldsymbol{\zeta}^{(n)}\big)^{\dagger}_{\boldsymbol{\theta}^{(n)}(i)} \gg\!= \mathit{Flrn}(\psi_i) \Big).\end{aligned} \tag{18}$$

This rule is used with the identity channel together with the channel $\boldsymbol{\zeta}^{(n)}$*.*

2. *The next topic-word channel* $\boldsymbol{\zeta}^{(n+1)} \in \mathcal{D}(W)^T$ *at* $t \in T$ *and* $w \in W$ *is:*

$$\begin{aligned}\boldsymbol{\zeta}^{(n+1)}(t)(w) \coloneqq \operatorname*{argmin}_{\zeta \in \mathcal{D}(W)^T} \sum_{i \in I} D_{KL}\Big(\mathit{Flrn}(\psi_i),\, \boldsymbol{\zeta} \gg\!= \boldsymbol{\theta}^{(n+1)}(i)\Big) \\ + D_{KL}\Big(\mathit{Flrn}(\beta - \mathbf{1}),\, \boldsymbol{\zeta}(t)\Big).\end{aligned} \tag{19}$$

Concretely, it can be chosen as:

$$\boldsymbol{\zeta}^{(n+1)}(t)(w) = \frac{\beta(w) - 1 + \sum_{i \in I} \psi_i(w) \cdot \big(\boldsymbol{\zeta}^{(n)}\big)^{\dagger}_{\boldsymbol{\theta}^{(n)}(i)}(w)(t)}{\|\beta - \mathbf{1}\| + \sum_{i \in I} \|\psi_i\| \cdot \Big(\big(\boldsymbol{\zeta}^{(n)}\big)^{\dagger}_{\boldsymbol{\theta}^{(n)}(i)} \gg\!= \mathit{Flrn}(\psi_i)\Big)(t)}. \tag{20}$$

Proof. For the first point it sufficies to prove this for the log-likelihood $\ln \mathcal{L}$. We drop the subscripts $\alpha, \beta, \boldsymbol{\psi}$ for convenience. Also, we abbreviate:

$$r_i \coloneqq \|\alpha - \mathbf{1}\| + \|\psi_i\| \quad r_{i,1} \coloneqq \frac{\|\alpha - \mathbf{1}\|}{r} \quad r_{i,2} \coloneqq \frac{\|\psi_i\|}{r} \tag{21}$$

Thus $r_{i,1} + r_{i,2} = 1$. Using the reformulation in Lemma 5 we get:

$$\begin{aligned}\ln \mathcal{L}\big(\boldsymbol{\theta}^{(n+1)}, \boldsymbol{\zeta}^{(n+1)}\big) &= C - \sum_{i \in I} r_i \cdot \Big(r_{i,1} \cdot D_{KL}\Big(\mathit{Flrn}(\alpha - \mathbf{1}),\, \boldsymbol{\theta}^{(n+1)}(i)\Big) \\ &\qquad + r_{i,2} \cdot D_{KL}\Big(\mathit{Flrn}(\psi_i),\, \boldsymbol{\zeta}^{(n+1)} \gg\!= \boldsymbol{\theta}^{(n+1)}(i)\Big)\Big) \\ &\quad - \sum_{t \in T} \|\beta - \mathbf{1}\| \cdot D_{KL}\Big(\mathit{Flrn}(\beta - \mathbf{1}),\, \boldsymbol{\zeta}^{(n+1)}(t)\Big) \\ &\geq C - \sum_{i \in I} r_i \cdot \Big(r_{i,1} \cdot D_{KL}\Big(\mathit{Flrn}(\alpha - \mathbf{1}),\, \boldsymbol{\theta}^{(n+1)}(i)\Big) \\ &\qquad + r_{i,2} \cdot D_{KL}\Big(\mathit{Flrn}(\psi_i),\, \boldsymbol{\zeta}^{(n)} \gg\!= \boldsymbol{\theta}^{(n+1)}(i)\Big)\Big) \\ &\quad - \sum_{t \in T} \|\beta - \mathbf{1}\| \cdot D_{KL}\Big(\mathit{Flrn}(\beta - \mathbf{1}),\, \boldsymbol{\zeta}^{(n)}(t)\Big) \\ &\geq C - \sum_{i \in I} r_i \cdot \Big(r_{i,1} \cdot D_{KL}\Big(\mathit{Flrn}(\alpha - \mathbf{1}),\, \boldsymbol{\theta}^{(n)}(i)\Big) \\ &\qquad + r_{i,2} \cdot D_{KL}\Big(\mathit{Flrn}(\psi_i),\, \boldsymbol{\zeta}^{(n)} \gg\!= \boldsymbol{\theta}^{(n)}(i)\Big)\Big) \\ &\quad - \sum_{t \in T} \|\beta - \mathbf{1}\| \cdot D_{KL}\Big(\mathit{Flrn}(\beta - \mathbf{1}),\, \boldsymbol{\zeta}^{(n)}(t)\Big) \\ &= \ln \mathcal{L}\big(\boldsymbol{\theta}^{(n)}, \boldsymbol{\zeta}^{(n)}\big).\end{aligned}$$

The first inequality $\geq$ holds because $\boldsymbol{\zeta}^{(n+1)}$ is defined as argmin in (19). The second inequality $\geq$ follows from Jeffrey's divergence reduction, in mixture form, see Theorem 1 (2). We apply it with prior distribution $\omega \coloneqq \boldsymbol{\theta}^{(n)}(i)$, with two channels, namely the identity $c_1 \coloneqq \mathrm{id}\colon T \rightsquigarrow T$ and $c_2 \coloneqq \boldsymbol{\zeta}^{(n)}\colon T \rightsquigarrow W$, with two distributions $\tau_1 \coloneqq \mathit{Flrn}(\alpha-\mathbf{1}) \in \mathcal{D}(T)$ and $\tau_2 \coloneqq \mathit{Flrn}(\psi_i) \in \mathcal{D}(W)$, and with two probabilities $r_{1,1} \coloneqq \frac{\|\alpha-\mathbf{1}\|}{\|\alpha-\mathbf{1}\|+\|\psi_i\|}$ and $r_{i,2} \coloneqq \frac{\|\psi_i\|}{\|\alpha-\mathbf{1}\|+\|\psi_i\|}$. The dagger of the identity channel is the identity, so that $(c_1)^{\dagger}_{\omega} \gg\!= \tau_1 = \mathit{Flrn}(\alpha-\mathbf{1})$. The updated state ω' in Theorem 1 (2) is then $\boldsymbol{\theta}^{(n+1)}(i)$ as defined above.

We turn to formula (20). In order to find the argmin in (19) we use Lagrange's multiplier method, see *e.g.* [1, §2.2]. This method ensures that in the solution gives convex combinations

Thus, we first extend the relevant equation with additional parameters κ_t, for $t \in T$, in the function H defined as the log-likelihood plus an extra expression—typical for Lagrange:

$$H\big(\boldsymbol{\zeta}, \boldsymbol{\kappa}\big) = \ln \mathcal{L}_{\alpha,\beta,\boldsymbol{\psi}}\big(\boldsymbol{\theta}, \boldsymbol{\zeta}\big) + \sum_{t \in T} \kappa(t) \cdot \left(1 - \sum_{w \in W} \boldsymbol{\zeta}(t)(w)\right).$$

Thus we keep the hyperparameters $\alpha, \beta, \boldsymbol{\psi}$ and also the channel $\boldsymbol{\theta}$ fixed. We then consider the partial derivatives, for $s \in T$ and $v \in W$.

$$\begin{aligned}
\frac{\partial H}{\partial \boldsymbol{\zeta}(s)(v)}\big(\boldsymbol{\zeta}, \boldsymbol{\kappa}\big) &= \frac{\big(\beta(v) - 1\big)}{\boldsymbol{\zeta}(s)(v)} + \sum_{i \in I} \frac{\psi_i(v) \cdot \boldsymbol{\theta}(i)(s)}{\big(\boldsymbol{\zeta} \gg\!= \boldsymbol{\theta}(i)\big)(v)} - \kappa(s) \\
&= \frac{1}{\boldsymbol{\zeta}(s)(v)} \cdot \left(\beta(v) - 1 + \sum_{i \in I} \frac{\psi_i(v) \cdot \boldsymbol{\theta}(i)(s) \cdot \boldsymbol{\zeta}(s)(v)}{\big(\boldsymbol{\zeta} \gg\!= \boldsymbol{\theta}(i)\big)(v)}\right) - \kappa(s) \\
&= \frac{\beta(v) - 1 + \sum_{i \in I} \psi_i(v) \cdot \boldsymbol{\zeta}^{\dagger}_{\boldsymbol{\theta}(i)}(v)(s)}{\boldsymbol{\zeta}(s)(v)} - \kappa(s) \\
\frac{\partial H}{\partial \kappa(s)}\big(\boldsymbol{\zeta}, \boldsymbol{\kappa}\big) &= 1 - \sum_{w \in W} \boldsymbol{\zeta}(s)(w).
\end{aligned}$$

Setting all of these to zero yields:

$$\boldsymbol{\zeta}(s)(v) = \frac{\beta(v) - 1 + \sum_{i \in I} \psi_i(v) \cdot \boldsymbol{\zeta}^{\dagger}_{\boldsymbol{\theta}(i)}(v)(s)}{\kappa(s)}.$$

Thus:

$$\begin{aligned}
1 \,=\, \sum_{v \in W} \boldsymbol{\zeta}(s)(v) &= \sum_{v \in W} \frac{\beta(v) - 1 + \sum_{i \in I} \psi_i(v) \cdot \boldsymbol{\zeta}^{\dagger}_{\boldsymbol{\theta}(i)}(v)(s)}{\kappa(s)} \\
&= \frac{\|\beta-\mathbf{1}\| + \sum_{i \in I} \sum_{v \in W} \psi_i(v) \cdot \boldsymbol{\zeta}^{\dagger}_{\boldsymbol{\theta}(i)}(v)(s)}{\kappa(s)} \\
&= \frac{\|\beta-\mathbf{1}\| + \sum_{i \in I} \|\psi_i\| \cdot \big(\boldsymbol{\zeta}^{\dagger}_{\boldsymbol{\theta}(i)} \gg\!= \mathit{Flrn}(\psi_i)\big)(s)}{\kappa(s)}.
\end{aligned}$$

But then:

$$\boldsymbol{\zeta}(s)(v) = \frac{\beta(v) - 1 + \sum_{i\in I} \psi_i(v) \cdot \boldsymbol{\zeta}^{\dagger}_{\boldsymbol{\theta}(i)}(v)(s)}{\|\beta - \mathbf{1}\| + \sum_{i\in I} \|\psi_i\| \cdot \big(\boldsymbol{\zeta}^{\dagger}_{\boldsymbol{\theta}(i)} \gg\!= \mathit{Flrn}(\psi_i)\big)(s)}.$$

We may now use $\theta^{(n)}$ and $\zeta^{(n)}$ in the expression on the right-hand-side for the next-stage choice of $\zeta^{(n+1)}$, as in (20). □

We include a very simple example to illustrate LDA.

Example 7. We take a set $W = \{a, b, c, d, e, f\}$ with the first six letters of the alphabet as the set of words, and two topics: $T = \{1, 2\}$. We consider a corpus with 3 multisets of words, in the middle column in the table below. We see that the words b, d, f occur frequently in the first document, whereas the other words a, c, e occur often in the second one. The frequencies of letters in the third document is roughly equal. Hence we expect document 1 to be mostly on one topic, and document 2 on the other topic, and document 3 on both.

data	**document multiset**	**topic distribution**
1	$1\|a\rangle + 6\|b\rangle + 1\|c\rangle + 7\|d\rangle + 2\|e\rangle + 8\|f\rangle$	$0.831\|1\rangle + 0.169\|2\rangle$
2	$10\|a\rangle + 1\|b\rangle + 8\|c\rangle + 2\|d\rangle + 9\|e\rangle + 1\|f\rangle$	$0.132\|1\rangle + 0.868\|2\rangle$
3	$4\|a\rangle + 3\|b\rangle + 4\|c\rangle + 5\|d\rangle + 2\|e\rangle + 3\|f\rangle$	$0.512\|1\rangle + 0.488\|2\rangle$

The hyperparameter $\alpha \in \mathcal{M}(T)$ and $\beta \in \mathcal{M}(W)$ are chosen as constants, with multiplicity 2 for alpha and 1 for β. Running the LDA algorithm, as described in Theorem 6, 25 times yields a document-topic channel $\boldsymbol{\theta}$ with topic distributions for each document, in the column on the right in the above table. As expected, documents 1 and 2 are about different (opposite) topics.

The LDA-algorithm also produces a topic-word channel $\boldsymbol{\zeta} : T \rightsquigarrow W$. It assigns in this simple example the following word probabilities to topics:

$$\begin{aligned} 1 &\mapsto 0.0000665|a\rangle + 0.278|b\rangle + 0.000707|c\rangle + 0.362|d\rangle + 0.0223|e\rangle + 0.337|f\rangle \\ 2 &\mapsto 0.362|a\rangle + 0.00277|b\rangle + 0.313|c\rangle + 0.0274|d\rangle + 0.295|e\rangle + 0.000435|f\rangle. \end{aligned}$$

This is consistent with what we saw above: topic 1 makes makes words b, d, f most likely, and topic 2 makes the other words a, c, e most likely.

6 Conclusions

EM and LDA are based on Jeffrey's update rule. Even if in actual implementations the formulations in terms of channels and their daggers may not be directly useful—for instance when results are approximated, typically via Gibbs sampling—having a crisp description of the mathematical essentials may be useful for understanding and reasoning about these fundamental EM and LDA algorithms in machine learning.

Acknowledgements. Thanks to Dario Stein for many discussions about LDA.

References

1. Bishop, C.: Pattern Recognition and Machine Learning. Information Science and Statistics. Springer, New York (2006)
2. Blei, D., Ng, A., Jordan, M.: Latent Dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003). https://doi.org/10.5555/944919.944937
3. Chan, H., Darwiche, A.: On the revision of probabilistic beliefs using uncertain evidence. Artif. Intell. **163**, 67–90 (2005). https://doi.org/10.1016/j.artint.2004.09.005
4. Cho, K., Jacobs, B.: Disintegration and Bayesian inversion via string diagrams. Math. Struct. in Comp. Sci. **29**(7), 938–971 (2019). https://doi.org/10.1017/s0960129518000488
5. Clerc, F., Danos, V., Dahlqvist, F., Garnier, I.: Pointless learning. In: Esparza, J., Murawski, A.S. (eds.) FoSSaCS 2017. LNCS, vol. 10203, pp. 355–369. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-662-54458-7_21
6. Darwiche, A.: Modeling and Reasoning with Bayesian Networks. Cambridge University Press, Cambridge (2009)
7. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm. J. Roy. Stat. Soc. **39**(1), 1–38 (1977)
8. Dietrich, F., List, C., Bradley, R.: Belief revision generalized: a joint characterization of Bayes' and Jeffrey's rules. J. Econ. Theory **162**, 352–371 (2016). https://doi.org/10.1016/j.jet.2015.11.006
9. Friston, K.: The free-energy principle: a unified brain theory? Nat. Rev. Neurosci. **11**(2), 127–138 (2010). https://doi.org/10.1038/nrn2787
10. Fritz, T.: A synthetic approach to Markov kernels, conditional independence, and theorems on sufficient statistics. Adv. Math. **370**, 107239 (2020). https://doi.org/10.1016/J.AIM.2020.107239
11. Hohwy, J.: The Predictive Mind. Oxford University Press, Oxford (2013). https://doi.org/10.1093/acprof:oso/9780199682737.001.0001
12. Jacobs, B.: The mathematics of changing one's mind, via Jeffrey's or via Pearl's update rule. J. Artif. Intell. Res. **65**, 783–806 (2019). https://doi.org/10.1613/jair.1.11349
13. Jacobs, B.: From multisets over distributions to distributions over multisets. In: Logic in Computer Science. IEEE, Computer Science Press (2021). https://doi.org/10.1109/lics52264.2021.9470678
14. Jacobs, B.: Learning from what's right and learning from what's wrong. In: Sokolova, A (ed.) Math. Found. of Programming Semantics, number 351 in Elect. Proc. in Theor. Comp. Sci., pp. 116–133 (2021). https://doi.org/10.4204/EPTCS.351.8
15. Jacobs, B.: Urns & tubes. Compositionality, **4**(4) (2022). https://doi.org/10.32408/compositionality-4-4
16. Jacobs, B., Stein, D.: Pearl's and Jeffrey's update as modes of learning in probabilistic programming. In: Kerjean, M., Levy, P., (eds.) Math. Found. of Programming Semantics, Electronic Notes in Theoretical Computer Science (2023)
17. Jeffrey, R.: The Logic of Decision, 2nd edn. The University of Chicago Press, Chicago (1983)
18. McLachlan, G., Krishnan, T.: The EM Algorithm and Extensions. Probability and Statistics. Wiley, Hoboken 2 edn. (2007). https://doi.org/10.1002/9780470191613

19. Mrad, A.B., Delcroix, V., Piechowiak, S., Leicester, P., Abid, M.: An explication of uncertain evidence in Bayesian networks: likelihood evidence and probabilistic evidence. Appl. Intell. **43**(4), 802–824 (2015). https://doi.org/10.1007/s10489-015-0678-6
20. Murphy, K.: Machine Learning. A Probabilistic Perspective. MIT Press, Cambridge (2012)
21. Panangaden, P.: Labelled Markov Processes. Imperial College Press, London (2009)
22. Pearl, J.: Jeffrey's rule, passage of experience, and neo-Bayesianism. In: Kyburg, Jr. H. (ed.), Knowledge Representation and Defeasible Reasoning, pp. 245–265. Kluwer Academic Publishers (1990). https://doi.org/10.1007/978-94-009-0553-5_10
23. Rao, R., Ballard, D.: Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. Nat. Neurosci. **2**, 79–87 (1999). https://doi.org/10.1038/4580

Parameterized Complexity of Propositional Inclusion and Independence Logic

Yasir Mahmood[1](✉) and Jonni Virtema[2]

[1] DICE Group, Department of Computer Science, Paderborn University, Paderborn, Germany
yasir.mahmood@uni-paderborn.de
[2] Department of Computer Science, University of Sheffield, Sheffield, UK
j.t.virtema@sheffield.ac.uk

Abstract. We give a comprehensive account on the parameterized complexity of model checking and satisfiability of propositional inclusion and independence logic. We discover that for most parameterizations the problems are either in FPT or paraNP-complete.

Keywords: Propositional Logic · Team Semantics · Model checking · Satisfiability · Parameterized Complexity

1 Introduction

The research program on team semantics was conceived in the early 2000s to create a unified framework to study logical foundations of different notions of dependence between variables. Soon after the introduction of first-order dependence logic [28], the framework was extended to cover propositional and modal logic [29]. In this context, a significant step was taken in [5], where the focus shifted to study dependencies between formulas instead of variables. The framework of team semantics has been proven to be remarkably malleable. During the past decade the framework has been re-adapted for the needs of an array of disciplines. In addition to the modal variant, team semantics has been generalized to temporal [19] and probabilistic [4] frameworks, and fascinating connections to fields such as database theory [11], statistics [1], real valued computation [9], verification [20], and quantum information theory [16] have been identified.

Boolean satisfiability problem (SAT) and quantified Boolean formula problem (QBF) have had a widespread influence in diverse research communities. In particular, QBF solving techniques are important in application domains such as planning, program synthesis and verification, adversary games, and nonmonotonic reasoning, to name a few [27]. Further generalizations of QBF are the dependency quantified Boolean formula problem (DQBF) and alternating DQBF which allow richer forms of variable dependence [10,24,25]. Propositional logics with team semantics offer a fresh perspective to study enrichments of SAT and

H. H. Hansen et al. (Eds.): WoLLIC 2023, LNCS 13923, pp. 274–291, 2023.
https://doi.org/10.1007/978-3-031-39784-4_17

Table 1. (Left) An example database with 5 attributes and universe size 15. (Right) An encoding with $3 \cdot \lceil \log_2(3) \rceil + \lceil \log_2(5) \rceil + \lceil \log_2(4) \rceil$ many propositional variables.

`Instructor`	`Time`	`Room`	`Course`	`Responsible`
Antti	09:00	A.10	Genetics	Antti
Antti	11:00	A.10	Chemistry	Juha
Antti	15:00	B.20	Ecology	Antti
Jonni	10:00	C.30	Bio-LAB	Jonni
Juha	10:00	C.30	Bio-LAB	Jonni
Juha	13:00	A.10	Chemistry	Juha

i_1i_2	$t_1t_2t_3$	r_1r_2	c_1c_2	p_1p_2
00	110	11	11	00
00	111	11	00	10
00	000	00	01	00
01	001	01	10	01
10	001	01	10	01
10	010	11	00	10

QBF. Indeed, the so-called *propositional dependence logic* ($\mathcal{PDL}$) is known to coincide with DQBF, whereas quantified propositional logics with team semantics have a close connection to alternating DQBF [10,30].

Propositional dependency logics extend propositional logic with atomic dependency statements describing various forms of variable dependence. In this setting, formulas are evaluated over propositional teams (i.e., sets of propositional assignments with common variable domain). An *inclusion atom* $\boldsymbol{x} \subseteq \boldsymbol{y}$ is true in a team T, if $\forall s \in T\, \exists t \in T$ such that $s(\boldsymbol{x}) = t(\boldsymbol{y})$. An *independence atom* $\boldsymbol{x} \bot_{\boldsymbol{z}} \boldsymbol{y}$ expresses that in a team T, for any fixed value for the variables in $\boldsymbol{z}$ the values for $\boldsymbol{x}$ and $\boldsymbol{y}$ are informationally independent. The extension of propositional logic with inclusion and independence atoms yield inclusion ($\mathcal{PINC}$) and independence ($\mathcal{PIND}$) logics, respectively.

Example 1. Table 1 illustrates an example from relational databases. The set of records corresponds to a team, that satisfies the dependency `Responsible` $\subseteq$ `Instructor`. Moreover, it violates the independence `Instructor`$\bot_{\texttt{Course}}$`Time` as witnessed by tuples (Antti, 11:00, A.10, Chemistry, Juha) and (Juha, 13:00, A.10, Chemistry, Juha). In propositional logic setting, datavalues can be represented as bit strings of appropriate length (as depicted in Table 1).

The complexity landscape of the classical (non-parameterized) decision problems — satisfiability, validity, and model checking — is well mapped in the propositional and modal team semantics setting (see [14, page 627] for an overview). Parameterized complexity theory, pioneered by Downey and Fellows [2], is a widely studied subarea of complexity theory. The motivation being that it provides a deeper analysis than the classical complexity theory by providing further insights into the source of intractability. The idea here is to identify meaningful parameters of inputs such that fixing those makes the problem tractable. One example of a fruitful parameter is the treewidth of a graph. A parameterized problem (PP) is called fixed parameter tractable, or in **FPT** for short, if for a given input x with parameter k, the membership of x in PP can be decided in time $f(k) \cdot p(|x|)$ for some computable function f and polynomial p. That is, for each fixed value of k the problem is tractable in the classical sense of tractability (in **P**), and the degree of the polynomial is independent of the parameter.

Table 2. Overview of parameterized complexity results with pointers to the results. The **paraNP**-cases are complete, except for only membership in the first row. MC_s denotes model checking for strict semantics whereas MC/SAT refer to both semantics.

Parameter	$\mathcal{PIND}$		$\mathcal{PINC}$	
	MC	SAT	MC_s	SAT
formula-tw	**paraNP**[18]	**FPT**[19]	**paraNP**[17]	in **paraNP**[13]
formula-team-tw	**FPT**[8]	-	**FPT**[8]	-
team-size	**FPT**[7]	-	**FPT**[7]	-
formula-size	**FPT**[8]	Trivial	**FPT**[8]	Trivial
formula-depth	**FPT**[8]	**FPT**[9]	**FPT**[8]	**FPT**[9]
#variables	**FPT**[8]	**FPT**[9]	**FPT**[8]	**FPT**[9]
#splits	**paraNP**[18]	**FPT**[19]	**paraNP**[17]	**P**[14] if #splits=0
arity	**paraNP**[18]	**paraNP**[19]	**paraNP**[17]	**paraNP**[10]

The class **paraNP** consists of problems decidable in time $f(k) \cdot p(|x|)$ on a non-deterministic machine.

In the propositional team semantics setting, the study of parameterized complexity was initiated by Meier and Reinbold [23] in the context of parameterized enumeration problems, and by Mahmood and Meier [22] in the context of classical decision problems, for $\mathcal{PDL}$. In the first-order team semantics setting, Kontinen et al. [18] studied parameterized model checking of dependence and independence logic, and in [17] introduced the *weighted-definability* problem for dependence, inclusion and independence logic thereby establishing a connection with the parameterized complexity classes in the well-known **W**-*hierarchy*.

We focus on the parameterized complexity of model checking (MC) and satisfiability (SAT) of propositional inclusion and independence logic. We consider both *lax* and *strict* semantics. The former is the prevailing semantics in the team semantics literature. The past rejection of strict semantics was based on the fact that it does not satisfy locality [8] (the locality principle dictates that satisfaction of a formula should be invariant on the truth values of variables that do not occur in the formula). Recent works have revealed that locality of strict semantics can be recovered by moving to multiteam semantics (here teams are multisets) [3]. Since, in propositional team semantics, the shift from teams to multiteams has no complexity theoretic implications, we stick with the simpler set based semantics for our logics. In the model checking problem, one is given a team T and a formula ϕ, and the task is to determine whether $T \models \phi$. In the satisfiability problem, one is given a formula ϕ, and the task is to decide whether there exists a non-empty satisfying team T for ϕ. Table 2 gives an overview of our results. We consider only strict semantics for MC of $\mathcal{PINC}$, since for lax semantics the problem is tractable already in the non-parameterized setting [14, Theorem 3.5].

2 Preliminaries

We assume familiarity with standard notions in complexity theory such as classes **P**, **NP** and **EXP** [26]. We give a short exposition of relevant concepts from parameterized complexity theory. For a broader introduction consider the textbook of Downey and Fellows [2], or that of Flum and Grohe [7].

A *parameterized problem* (PP) $\Pi \subseteq \Sigma^* \times \mathbb{N}$ consists of tuples (x, k), where x is called an instance and k is the (value of the) parameter.

FPT and paraNP. Let Π be a PP over $\Sigma^* \times \mathbb{N}$. Then Π is *fixed parameter tractable* (**FPT** for short) if it can be decided by a deterministic algorithm $\mathcal{A}$ in time $f(k) \cdot p(|x|)$ for any input (x, k), where f is a computable function and p is a polynomial. If the algorithm $\mathcal{A}$ is non-deterministic instead, then Π belongs to the class **paraNP**.

The notion of hardness in parameterized setting is employed by fpt-reductions.

fpt-reductions. Let $\Pi \subseteq \Sigma^* \times \mathbb{N}$ and $\Theta \subseteq \Gamma^* \times \mathbb{N}$ be two PPs. Then Π is *fpt-reducible* to Θ, if there exists an fpt-computable function $g\colon \Sigma^* \times \mathbb{N} \to \Gamma^* \times \mathbb{N}$ such that (1) for all $(x, k) \in \Sigma^* \times \mathbb{N}$ we have that $(x, k) \in \Pi \Leftrightarrow g(x, k) \in \Theta$ and (2) there exists a computable function $h\colon \mathbb{N} \to \mathbb{N}$ such that for all $(x, k) \in \Sigma^* \times \mathbb{N}$ and $g(x, k) = (x', k')$ we have that $k' \leq h(k)$.

We will use the following result to prove **paraNP**-hardness. Let Π be a PP over $\Sigma^* \times \mathbb{N}$. Then the *$\ell$-slice of Π*, for $\ell \geq 0$, is the set $\Pi_\ell := \{\, x \mid (x, \ell) \in \Pi \,\}$.

Proposition 2 ([7, Theorem 2.14]). *Let Π be a PP in* **paraNP**. *If there exists an $\ell \geq 0$ such that Π_ℓ is* **NP***-complete, then Π is* **paraNP**-complete.

Moreover, we will use the following folklore result to get several upper bounds.

Proposition 3. *Let Q be a problem such that (Q, k) is* **FPT** *and let ℓ be a parameter with $k \leq f(\ell)$ for some computable function f. Then (Q, ℓ) is* **FPT**.

Propositional Team Based Logics. Let Var be a countably infinite set of variables. The syntax of propositional logic ($\mathcal{PL}$) is defined via the following grammar: $\phi ::= x \mid \neg x \mid \phi \vee \phi \mid \phi \wedge \phi$, where $x \in \mathrm{Var}$. Observe that we allow only atomic negations. As usual $\top := x \vee \neg x$ and $\bot := x \wedge \neg x$. Propositional dependence logic $\mathcal{PDL}$ is obtained by extending $\mathcal{PL}$ by atomic formulas of the form $=(\boldsymbol{x}; \boldsymbol{y})$, where $\boldsymbol{x}, \boldsymbol{y} \subset \mathrm{Var}$ are finite tuples of variables. Similarly, adding inclusion atoms $\boldsymbol{x} \subseteq \boldsymbol{y}$ where $(|\boldsymbol{x}| = |\boldsymbol{y}|)$ and independence atoms $\boldsymbol{x} \perp_{\boldsymbol{z}} \boldsymbol{y}$ gives rise to propositional inclusion ($\mathcal{PINC}$) and independence ($\mathcal{PIND}$) logic, respectively. When we wish to talk about any of the three considered logics, we simply write $\mathcal{L}$. That is, unless otherwise stated, $\mathcal{L} \in \{\mathcal{PDL}, \mathcal{PINC}, \mathcal{PIND}\}$. For an assignment s and a tuple $\boldsymbol{x} = (x_1, \ldots, x_n)$, $s(\boldsymbol{x})$ denotes $(s(x_1), \ldots, s(x_n))$.

Team Semantics. Let ϕ, ψ be $\mathcal{L}$-formulas and $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \subset \mathrm{Var}$ be finite tuples of variables. A *team* T is a set of assignments $t\colon \mathrm{Var} \to \{\, 0, 1 \,\}$. The satisfaction relation $\models$ is defined as follows:

$$
\begin{array}{lll}
T \models x & \text{iff} & \forall t \in T : t(x) = 1, \\
T \models \neg x & \text{iff} & \forall t \in T : t(x) = 0, \\
T \models \phi \wedge \psi & \text{iff} & T \models \phi \text{ and } T \models \psi, \\
T \models \phi \vee \psi & \text{iff} & \exists T_1, T_2 (T_1 \cup T_2 = T) : T_1 \models \phi \text{ and } T_2 \models \psi, \\
T \models \boldsymbol{x} \subseteq \boldsymbol{y} & \text{iff} & \forall t \in T \, \exists t' \in T : t(\boldsymbol{x}) = t'(\boldsymbol{y}), \\
T \models \boldsymbol{x} \perp_{\boldsymbol{z}} \boldsymbol{y} & \text{iff} & \forall t, t' \in T : t(\boldsymbol{z}) = t'(\boldsymbol{z}),\ \exists t'' : t''(\boldsymbol{xzy}) = t(\boldsymbol{xz}) t'(\boldsymbol{y}).
\end{array}
$$

Intuitively, an inclusion atom $\boldsymbol{x} \subseteq \boldsymbol{y}$ is true if the value taken by $\boldsymbol{x}$ under an assignment t is also taken by $\boldsymbol{y}$ under some assignment t'. Moreover, the independence atom $\boldsymbol{x} \perp_{\boldsymbol{z}} \boldsymbol{y}$ has the meaning that whenever the value for $\boldsymbol{z}$ is fixed under two assignments t and t', then there is an assignment t'' which maps $\boldsymbol{x}$ and $\boldsymbol{y}$ according to t and t', respectively. We can interpret the dependence atom $=(\boldsymbol{x}; \boldsymbol{y})$ as the independence atom $\boldsymbol{y} \perp_{\boldsymbol{x}} \boldsymbol{y}$. The operator $\vee$ is also called a split-junction in the context of team semantics. Note that in the literature there exist two semantics for the split-junction: *lax* and *strict* semantics (e.g., Hella et al. [14]). Strict semantics requires the "splitting of the team" to be a partition whereas lax semantics allows an "overlapping" of the team. Regarding $\mathcal{PDL}$ and $\mathcal{PIND}$, the complexity for SAT and MC is the same irrespective of the considered semantics. However, the picture is different for MC in $\mathcal{PINC}$ as depicted in [14, page 627]. For any logic $\mathcal{L}$, we denote MC under strict (respectively, lax) semantics by $\text{MC}_s(\text{MC}_l)$. Moreover, MC_l is in **P** for $\mathcal{PINC}$ and consequently, we have only MC_s in Table 2.

3 Graph Representation of the Input

In order to consider specific structural parameters, we need to agree on a representation of an input instance. We follow the conventions given in [22]. Well-formed $\mathcal{L}$-formulas, for every $\mathcal{L} \in \{\mathcal{PDL}, \mathcal{PINC}, \mathcal{PIND}\}$, can be seen as binary trees (the syntax tree) with leaves as atomic subformulas (variables and dependency atoms). Similarly to $\mathcal{PDL}$ [22], we take the syntax structure (defined below) rather than syntax tree as a graph structure in order to consider treewidth as a parameter. We use the same graph representation for each logic $\mathcal{L}$. That is, when an atom $=(\boldsymbol{x}; \boldsymbol{y})$ is replaced by either $\boldsymbol{x} \subseteq \boldsymbol{y}$ or $\boldsymbol{x} \perp_{\emptyset} \boldsymbol{y}$, the graph representation, and hence, the treewidth of this graph remains the same. Also, in the case of MC, we include assignments in the graph representation. In the latter case, we consider the Gaifman graph of the structure that models both, the team and the input formula.

Syntax Structure. Let ϕ be an $\mathcal{L}$-formula with propositions $\{x_1, \ldots, x_n\}$ and $T = \{ s_1, \ldots s_m \}$ a team. The *syntax structure* $\mathcal{A}_{T,\phi}$ has the vocabulary, $\tau_{T,\phi} := \{ \mathsf{VAR}^1, \mathsf{SF}^1, \succcurlyeq^2, \mathsf{DEP}^2, \mathsf{isTrue}^2, \mathsf{isFalse}^2, r, c_1, \ldots, c_m \}$, where the superscript denote the arity of each relation. The universe of $\mathcal{A}_{T,\phi}$ is $A := \text{SF}(\phi) \cup \text{Var}(\phi) \cup \{ c_1^{\mathcal{A}}, \ldots, c_m^{\mathcal{A}} \}$, where $\text{SF}(\phi)$ and $\text{Var}(\phi)$ denote the set of subformulas and variables appearing in ϕ, respectively.

– SF and VAR are unary relations: 'is a subformula of ϕ' and 'is a variable in ϕ'.

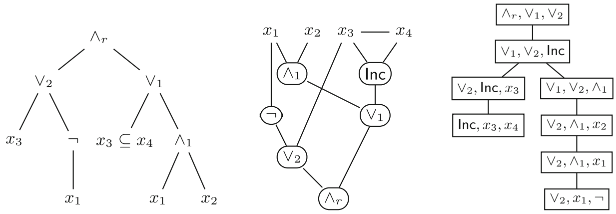


Fig. 1. An example syntax tree (left) with the corresponding Gaifman graph (middle) and a tree decomposition (right) for $(x_3 \vee \neg x_1) \wedge \big(x_3 \subseteq x_4 \vee (x_1 \wedge x_2)\big)$. We abbreviated subformulas in the inner vertices of the Gaifman graph for better presentation.

- $\succcurlyeq$ is a binary relation such that $\psi \succcurlyeq^{\mathcal{A}} \alpha$ iff α is an immediate subformula of ψ. That is, either $\psi = \neg\alpha$ or there is a $\beta \in \mathrm{SF}(\phi)$ such that $\psi = \alpha \oplus \beta$ where $\oplus \in \{\wedge, \vee\}$. Moreover, r is a constant symbol representing ϕ.
- DEP is a binary relation connecting $\mathcal{L}$-atoms and its parameters. For example, if $\alpha = \boldsymbol{x} \subseteq \boldsymbol{y}$ and $x, y \in \boldsymbol{x} \cup \boldsymbol{y}$, then $\mathsf{DEP}(\alpha, x)$ and $\mathsf{DEP}(x, y)$ are true.
- The set $\{ c_1, \ldots, c_m \}$ encodes the team T. Each $c_i \in \tau_{T,\phi}$ corresponds to an assignment $s_i \in T$ for $i \leq m$, interpreted as $c_i^{\mathcal{A}} \in A$.
- isTrue and $\mathsf{isFalse}$ relate VAR with $c_1, \ldots, c_m$. $\mathsf{isTrue}(c, x)$ (resp., $\mathsf{isFalse}(c, x)$) is true iff x is mapped to 1 (resp., 0) by the assignment in T interpreted by c.

The *syntax structure* $\mathcal{A}_\phi$ over a vocabulary τ_ϕ is defined analogously. Here τ_ϕ neither contains the team related relations nor the constants $c_i^{\mathcal{A}}$ for $1 \leq i \leq m$.

Gaifman Graph. Let T be a team, ϕ an $\mathcal{L}$-formula, $\mathcal{A}_{T,\phi}$ and A as above. The *Gaifman graph* $G_{T,\phi} = (A, E)$ of the $\tau_{T,\phi}$-structure $\mathcal{A}_{T,\phi}$ is defined as $E := \big\{ \{u, v\} \bigm| u, v \in A,$ such that there is an $R \in \tau_{T,\phi}$ with $(u, v) \in R \big\}$. Analogously, we let G_ϕ to be the *Gaifman graph* for the τ_ϕ-structure $\mathcal{A}_\phi$.

Note that for G_ϕ we have $E = \mathsf{DEP} \cup \succcurlyeq$ and for $G_{T,\phi}$ we have that $E = \mathsf{DEP} \cup \succcurlyeq \cup \mathsf{isTrue} \cup \mathsf{isFalse}$.

Treewidth. A *tree decomposition* of a graph $G = (V, E)$ is a tree $T = (B, E_T)$, where the vertex set $B \subseteq \mathcal{P}(V)$ is a collection of *bags* and E_T is the edge relation such that (1) $\bigcup_{b \in B} b = V$, (2) for every $\{ u, v \} \in E$ there is a bag $b \in B$ with $u, v \in b$, and (3) for all $v \in V$ the restriction of T to v (the subset with all bags containing v) is connected. The *width* of a tree decomposition $T = (B, E_T)$ is the size of the largest bag minus one: $\max_{b \in B} |b| - 1$. The *treewidth* of a graph G is the minimum over widths of all tree decompositions of G. The treewidth of a tree is one. Intuitively, it measures the tree-likeness of a given graph.

Example 4 (Adapted from [22]*).* Figure 1 represents the Gaifman graph of the syntax structure $\mathcal{A}_\phi$ (in middle) with a tree decomposition (on the right). Since the largest bag is of size 3, the treewidth of the given decomposition is 2. Figure 2 presents the Gaifman graph of $\mathcal{A}_{T,\phi}$, that is, when the team $T = \{s_1, s_2\} = \{0011, 1110\}$ is part of the input (an assignment s is denoted as $s(x_1 \ldots x_4)$).

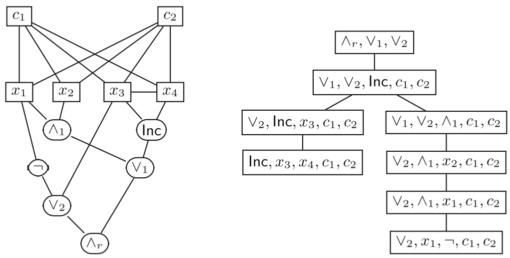


Fig. 2. The Gaifman graph for $\langle T, \Phi\rangle$ (Example 4) with a possible tree decomposition.

Parameterizations. We consider eight different parameters for MC and six for SAT. For MC, we include formula-tw, formula-team-tw, team-size, formula-size, #variables, formula-depth, #splits and arity. However, for SAT, formula-team-tw and team-size are not meaningful. All these parameters arise naturally in problems we study. Let T be a team and ϕ an $\mathcal{L}$-formula. #splits denotes the number of times a split-junction ($\vee$) appears in ϕ and #variables denotes the number of distinct propositional variables. formula-depth is the depth of the syntax tree of ϕ, that is, the length of the longest path from root to any leaf in the syntax tree. team-size is the cardinality of the team T, and formula-size is $|\phi|$. For a dependence atom $=(\boldsymbol{x};\boldsymbol{y})$ and inclusion atom $\boldsymbol{x} \subseteq \boldsymbol{y}$, the arity is defined as $|\boldsymbol{x}|$ (recall that $|\boldsymbol{x}| = |\boldsymbol{y}|$ for an inclusion atom), whereas, for an independence atom $\boldsymbol{x} \perp_{\boldsymbol{z}} \boldsymbol{y}$, it is the number of distinct variables appearing in $\boldsymbol{x} \perp_{\boldsymbol{z}} \boldsymbol{y}$. Finally, arity denotes the maximum arity of any $\mathcal{L}$-atom in ϕ. Regarding treewidth, recall that for MC, we also include the assignment-variable relation in the graph representation. This yields two graphs: G_ϕ for ϕ, and $G_{T,\phi}$ for $\langle T, \phi\rangle$. Consequently, there are two treewidth notions. formula-tw is the treewidth of G_ϕ and formula-team-tw is the treewidth of $G_{T,\phi}$. The name emphasises whether the team is also part of the graph. As we pointed out formula-team-tw and team-size are both only relevant for MC because an instance of SAT does not contain a team.

Given an instance $\langle T, \phi\rangle$ and a parameterisation κ, then $\kappa(T, \phi)$ denotes the parameter value of $\langle T, \phi\rangle$. The following relationship between several of the aforementioned parameters was proven for $\mathcal{PDL}$. It is easy to observe that the lemma also applies to $\mathcal{PINC}$ and $\mathcal{PIND}$.

Lemma 5 **([22]).** *Let $\mathcal{L} \in \{\mathcal{PDL}, \mathcal{PINC}, \mathcal{PIND}\}$, ϕ an $\mathcal{L}$-formula and T be a team. Then,* team-size$(T, \phi) \leq 2^{\#\text{variables}(T,\phi)}$*,* team-size$(T, \phi) \leq 2^{\text{formula-size}(T,\phi)}$*, and* formula-size$(T, \phi) \leq 2^{2\cdot\text{formula-depth}(T,\phi)}$.

Moreover, recall that we use the same graph representation for $\mathcal{PDL}, \mathcal{PINC}$ and $\mathcal{PIND}$. As a consequence, the following result also applies.

Corollary 6 ([22]). *Let $\mathcal{L} \in \{\mathcal{PDL}, \mathcal{PINC}, \mathcal{PIND}\}$, ϕ an $\mathcal{L}$-formula and T be a team. Then* formula-team-tw(T, ϕ) *bounds* team-size(T, ϕ).

4 Complexity of Inclusion and Independence Logic

We start with general complexity results that hold for any team based logic whose atoms are **P**-checkable. An atom α is **P**-checkable if given a team T, $T \models \alpha$ can be checked in polynomial time. It is immediate that each atom considered in this paper is **P**-checkable.

Theorem 7. *Let $\mathcal{L}$ be a team based logic such that $\mathcal{L}$-atoms are* **P***-checkable, then* MC *for $\mathcal{L}$ when parameterized by* team-size *is* **FPT***.*

Proof. We claim that the bottom up (brute force) algorithm for the model checking of $\mathcal{PDL}$ [22, Theorem 17] works for any team based logic $\mathcal{L}$ such that $\mathcal{L}$-atoms are **P**-checkable. The algorithm begins by checking the satisfaction of atoms against each subteam. This can be achieved in **FPT**-time since teamsize and consequently the number of subteams is bounded. Moreover, by taking the union of subteams for split-junction and keeping the same team for conjunction the algorithm can find subteams for each subformula in **FPT**-time. Lastly, it checks that the team T is indeed a satisfying team for the formula ϕ. For any team based logic $\mathcal{L}$, the **FPT** runtime is guaranteed if $\mathcal{L}$-atoms are **P**-checkable. Finally, the proof works for both strict and lax semantics. □

The following corollary to Theorem 7 is derived using Lemma 5 and Proposition 3.

Corollary 8. *Let $\mathcal{L}$ be a team based logic such that $\mathcal{L}$-atoms are* **P***-checkable, then* MC *for $\mathcal{L}$ when parameterized by k is* **FPT***, if $k \in \{$*formula-team-tw, formula-depth, #variables, formula-size*$\}$.*

The following theorem states results for satisfiability.

Theorem 9. *Let $\mathcal{L}$ be a team based logic s.t. $\mathcal{L}$-atoms are* **P***-checkable, then* SAT *for $\mathcal{L}$ when parameterized by k is* **FPT***, if $k \in \{$*formula-depth, #variables*$\}$.*

Proof. Notice first that the case for formula-size is trivial because any problem parameterized by input size is **FPT**. Moving on, bounding formula-depth also bounds formula-size, this yields **FPT**-membership for formula-depth in conjunction with Proposition 3. Finally, for #variables, one can enumerate all of the $2^{2^{\#\text{variables}}}$-many teams in **FPT**-time and determine whether any of these satisfies the input formula. The last step requires that the model checking parameterized by team-size is **FPT**, which is true due to Theorem 7. This completes the proof. □

Our main technical contributions are the following two theorems which establish that the satisfiability problem of $\mathcal{PINC}$ parameterized by arity is **paraNP**-complete, and that SAT of $\mathcal{PINC}$ without disjunctions is tractable. We start with the former. The hardness follows from the **NP**-completeness of $\mathcal{PL}$. For membership, we give a non-deterministic algorithm $\mathbb{A}$ solving SAT.

Theorem 10. *There is a non-deterministic algorithm* $\mathbb{A}$ *that, given a* $\mathcal{PINC}$*-formula* ϕ *with arity* k*, runs in* $O(2^k \cdot p(|\phi|))$*-time and outputs a non-empty team* T *such that* $T \models \phi$ *if and only if* ϕ *is satisfiable.*

Proof. We present the proof for lax semantics first, towards the end we describe some modifications that solve the case for strict semantics. Given an input $\mathcal{PINC}$-formula ϕ, the algorithm $\mathbb{A}$ operates on the syntax tree of ϕ and constructs a sequence of teams $f_i(\psi)$ for each $\psi \in \mathrm{SF}(\phi)$ as follows. We let $f_0(\psi) := \emptyset$ for each $\psi \in \mathrm{SF}(\phi)$. Then, $\mathbb{A}$ begins by non-deterministically selecting a singleton team $f_1(\phi)$ for ϕ. For $i \geq 1$, $\mathbb{A}$ implements the following steps recursively.

For odd $i \in \mathbb{N}$, $f_i(\psi)$ is defined in a *top-down* fashion as follows.

1. $f_i(\phi) := f_{i-1}(\phi)$ for $i \geq 3$.
2. If $\psi = \psi_0 \wedge \psi_1$, let $f_i(\psi_0) := f_i(\psi)$ and $f_i(\psi_1) := f_i(\psi)$.
3. If $\psi = \psi_0 \vee \psi_1$, then non-deterministically select two teams P_0, P_1 such that $P_0 \cup P_1 = f_i(\psi) \setminus f_{i-1}(\psi)$ and set $f_i(\psi_j) := f_{i-1}(\psi_j) \cup P_j$ for $j = 0, 1$.
   For even i, $f_i(\psi)$ is defined in a *bottom-up* fashion as follows.
4. If $\psi \in \mathrm{SF}(\phi)$ is an atomic literal, then immediately reject if $f_{i-1}(\psi) \not\models \psi$ and set $f_i(\psi) := f_{i-1}(\psi)$ otherwise. If $\psi \in \mathrm{SF}(\phi)$ is an inclusion atom, then construct $f_i(\psi) \supseteq f_{i-1}(\psi)$ such that $f_i(\psi) \models \psi$. For $\psi := \boldsymbol{x} \subseteq \boldsymbol{y}$, this is done by (I) adding partial assignments $t(\boldsymbol{y}) := s(\boldsymbol{x})$ whenever an assignment s is a cause for the *failure* of ψ, and (II) non-deterministically selecting extensions of these assignments to the other variables.
5. If $\psi = \psi_0 \wedge \psi_1$, or $\psi = \psi_0 \vee \psi_1$ let $f_i(\psi) := f_i(\psi_0) \cup f_i(\psi_1)$.

$\mathbb{A}$ terminates by accepting when a fixed point is reached. That is, we obtain $j \in \mathbb{N}$ such that $f_i(\psi) = f_{i+1}(\psi)$ for each $\psi \in \mathrm{SF}(\phi)$ when $i \geq j$. Moreover, $\mathbb{A}$ rejects if Step 4 triggers a rejection. Notice that the only step when new assignments are added is at the atomic level. Whereas the split in Step 3 concerns those assignments which arise from other subformulas through union in Step 5. We first prove the following claim regarding the overall runtime for $\mathbb{A}$.

Claim I. $\mathbb{A}$ runs in at most $O(2^k \cdot p(|\phi|))$ steps for some polynomial p, where k is the arity of ϕ. That is, a fixed point or rejection is reached in this time.

Proof of Claim. In each iteration i, either $\mathbb{A}$ rejects, or keeps adding new assignments. Furthermore, new assignments are added only in the cases for inclusion atoms. As a result, if $\mathbb{A}$ has not yet reached a fixed point the reason is that some inclusion atom has generated new assignments. Since we take union of subteams in the bottom-up step, the following top-down iteration in Steps 2 and 3 may also add assignments in a subteam. That is, the subteams from each $\psi \in \mathrm{SF}(\phi)$ are propagated to other subformulas during each iteration. Now, each inclusion atom of arity $l \leq k$ can generate at most 2^l new assignments due to Step 4 in the algorithm. Let n denote the number of inclusion atoms in ϕ and k be their maximum arity. Then $\mathbb{A}$ iterates at most $2^k \cdot cn$ times, where c is some constant due to the propagation of teams to other subformulas. This implies that, if no rejection has occured, there is some $j \leq 2^k \cdot cn$ such that

$f_i(\psi) = f_j(\psi)$ for each subformula $\psi \in \mathrm{SF}(\phi)$ and $i \geq j$. We denote this fixed point by $f_\infty(\psi)$ for each $\psi \in \mathrm{SF}(\phi)$.

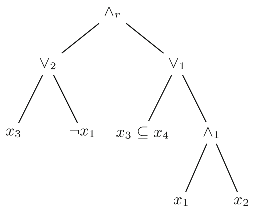
Now, we analyze the time it takes to compute each iteration. For odd $i \geq 1$, Steps 1 and 2 set the same team for each subformula and therefore take linear time. Notice that the size of team in each iteration is bounded by $2^k \cdot n$. This holds because new assignments are added only in the case of inclusion atoms and $\mathbb{A}$ starts with a singleton team. Consequently, Step 3 non-deterministically splits the teams of size $2^k \cdot n$ in each iteration i for odd $i \geq 1$. Moreover, Step 4 for even i requires (1) polynomial time in $|f_i(\psi)|$, if ψ is an atomic literal, and (2) non-deterministic polynomial time in $2^l \cdot |f_i(\psi)|$ if ψ is an inclusion atom of arity $l \leq k$. Finally, the union in Step 5 again requires linear time. This implies that each iteration takes at most a runtime of $2^k \cdot p(|\phi|)$ for some polynomial p. This completes the proof of Claim 1.

We now prove that $\mathbb{A}$ accepts the input formula ϕ if and only if ϕ is satisfiable. Suppose that $\mathbb{A}$ accepts and let $f_\infty(\phi)$ denote the fixed point. We first prove by induction that $f_\infty(\psi) \models \psi$ for each subformula ψ of ϕ. Notice that there is some i such that $f_\infty(\psi) = f_i(\psi)$. The case for atomic subformulas is clear due to the Step 4 of $\mathbb{A}$. For conjunction, observe that the team remains the same for each conjunct. That is, when $\psi = \psi_0 \wedge \psi_1$ and the claim holds for $f_\infty(\psi_i)$ and ψ_i, then $f_\infty(\psi) \models \psi_0 \wedge \psi_1$ is true. For disjunction, if $\psi = \psi_0 \vee \psi_1$ and $f_\infty(\psi_i)$ are such that $f_\infty(\psi_i) \models \psi_i$ for $i = 0, 1$, then we have that $f_\infty(\psi) \models \psi$ where $f_\infty(\psi) = f_\infty(\psi_0) \cup f_\infty(\psi_1)$. In particular $f_\infty(\phi) \models \phi$ and the correctness of our algorithm follows.

For the other direction, suppose ϕ is satisfiable and T is a witnessing team. Then there exists a labelling function for T and ϕ, given as follows.

I. The label for ϕ is T.
II. For every subformula $\psi = \psi_0 \oplus \psi_1$ with subteam label $P \subseteq T$, the subteam label for ψ_i is P_i $(i = 0, 1)$ such that we have $P_0 = P_1 = P$, if $\oplus = \wedge$, and $P_0 \cup P_1 = P$ if $\oplus = \vee$,
III. $P_\psi \models \psi$ for every $\psi \in \mathrm{SF}(\phi)$ with label P_ψ.

Then we prove that there exists an accepting path when the non-deterministic algorithm $\mathbb{A}$ operates on ϕ. We claim that when initiated on a subteam $\{s\} \subseteq T$, $\mathbb{A}$ constructs a fixed point $f_\infty(\phi)$ and halts by accepting ϕ. Recall that $\mathbb{A}$ propagates teams back and forth until a fixed point is reached. Moreover, the new assignments are added only at the atomic level. Let $\alpha := \boldsymbol{x} \subseteq \boldsymbol{y}$ be an inclusion atom such that $f_i(\alpha) \neq \emptyset$ for odd i, then $\mathbb{A}$ constructs a subteam $f_{i+1}(\alpha) \supseteq f_i(\alpha)$ (on a non-deterministic branch) containing assignments t from P_α such that $f_{i+1}(\alpha) \models \alpha$. Since, there are at most $2^{|\boldsymbol{y}|}$-many different assignments for $\boldsymbol{y}$, we know that Step 4 applies to α at most $2^{|\boldsymbol{y}|}$ times. That is, once all the different assignments for $\boldsymbol{y}$ have been checked in some iteration i: Step 4 does not add any further assignments to $f_{i'}(\alpha)$ for $i' \geq i + 1$. Finally, since there is a non-empty team T such that $T \models \phi$, this implies that $\mathbb{A}$ does not reject ϕ in any iteration (because there is a choice for $\mathbb{A}$ to consider subteams guaranteed by the labelling function). Consequently, $\mathbb{A}$ accepts by constructing a fixed point in



ψ	$f_1(\psi)$	$f_2(\psi)$	$f_3(\psi)$
ϕ	0110	0110,1111	0110,1111
$\vee_2$	0110	0110	0110,1111
$\vee_1$	0110	0110,1111	0110,1111
$x_3 \subseteq x_4$	0110	0110,1111	0110,1111
$x_1 \wedge x_2$			1111
x_3	0110	0110	0110,1111
$\neg x_1$	0110	0110	0110

Fig. 3. The table (Right) indicates subteams for each $\psi \in \mathrm{SF}(\phi)$ (Left). The teams $f_1(\psi)$ and $f_3(\psi)$ are propagated top-down whereas $f_2(\psi)$ is propagated bottom-up. For brevity we omit subformulas x_1 and x_2 of $x_1 \wedge x_2$.

at most $O(2^k \cdot p(|\phi|))$-steps (follows from Claim I). This completes the proof and establishes the correctness.

A minor variation in the algorithm $\mathbb{A}$ solves SAT for the strict semantics. When moving downwards, $\mathbb{A}$ needs to ensure that an assignment goes to only one side of the split. Moreover, since the subteams are selected non-deterministically for atomic subformulas, (in the bottom-up iteration) only subteams which can split according to the strict semantics are considered. $\square$

Example 11. We include an example to explain how $\mathbb{A}$ from the proof of Theorem 10 operates. Figure 3 depicts the steps of $\mathbb{A}$ on a formula ϕ. An assignment over $\{x_1, \ldots, x_4\}$ is seen as a tuple of length four. It is easy to observe that the third iteration already yields a fixed point and that $f_3(\psi) = f_4(\psi)$ for each $\psi \in \mathrm{SF}(\phi)$. In this example, the initial guess made by $\mathbb{A}$ is the team $\{0110\}$.

The following corollaries follow immediately from the proof of Theorem 10.

Corollary 12. *Given a $\mathcal{PINC}$-formula ϕ with arity k, then ϕ is satisfiable if and only if there is a team T of size at most $O(2^k \cdot p(|\phi|))$ such that $T \models \phi$.*

Proof. Simulate the algorithm $\mathbb{A}$ from the proof of Theorem 10. Since ϕ is satisfiable, $\mathbb{A}$ halts in at most $O(2^k \cdot p(|\phi|))$-steps and thereby yields a team (namely, $f_\infty(\phi)$) of the given size. $\square$

Corollary 13. SAT *for $\mathcal{PINC}$, when parameterized by* formula-tw *of the input formula is in* **paraNP**.

Proof. Recall the Graph structure where we allow edges between variables within an inclusion atom. This implies that for each inclusion atom α, there is a bag in the tree decomposition that contains all variables of α. As a consequence, a formula ϕ with treewidth k has inclusion atoms of arity at most k. Consequently, SAT parameterized by treewidth of the input formula can be solved using the **paraNP**-time algorithm from the proof of Theorem 10. $\square$

Regarding the parameter #splits, the precise parameterized complexity is still open for now. However, we prove that if there is no split in the formula, then SAT can be solved in polynomial time. This case is interesting in its own right because it gives rise to the so-called Poor Man's $\mathcal{PINC}$, similar to the case of Poor Man's $\mathcal{PDL}$ [6,21,23]. The model checking for this fragment is in **P**; this follows from the fact that MC for $\mathcal{PINC}$ with lax semantics is in **P**. In the following, we prove that SAT for Poor Man's $\mathcal{PINC}$ is also in **P**.

Theorem 14. *There is a deterministic algorithm* $\mathbb{B}$ *that given a* $\mathcal{PINC}$*-formula* ϕ *with no splits runs in* **P***-time and accepts if and only if* ϕ *is satisfiable.*

Proof. We give a recursive labelling procedure ($\mathbb{B}$) that runs in polynomial time and accepts if and only if ϕ is satisfiable. The labelling consists of assigning a value $c \in \{0, 1\}$ to each variable x.

1. Begin by labelling all $\mathcal{PL}$-literals in ϕ by the value that satisfies them, namely $x = 1$ for x and $x = 0$ for $\neg x$.
2. For each inclusion atom $\boldsymbol{p} \subseteq \boldsymbol{q}$ and a labelled variable $q_i \in \boldsymbol{q}$, label the variable $p_i \in \boldsymbol{p}$ with same value c as for q_i. Where p_i appears in $\boldsymbol{p}$ at the same position, as q_i in $\boldsymbol{q}$.
3. Propagate the label for p_i from the previous step. That is, consider p_i as a labelled variable and repeat Step 2 for as long as possible.
4. If some variable x is labelled with two opposite values, then reject. Otherwise, accept.

The fact that $\mathbb{B}$ works in polynomial time is clear because each variable is labelled at most once. If a variable is labelled to two different values, then it gives a contradiction and the procedure stops.

For the correctness, notice first that if $\mathbb{B}$ accepts then we have a partition of $\mathrm{Var}(\phi)$ into a set X of labelled variables and a set $Y = \mathrm{Var}(\phi) \setminus X$. When $\mathbb{B}$ stops, due to step 3, ϕ does not contain an inclusion atom $\boldsymbol{p} \subseteq \boldsymbol{q}$ such that $q_i \in \boldsymbol{q}$ and $p_i \in \boldsymbol{p}$ for some $q_i \in X, p_i \in Y$, where p_i appears in $\boldsymbol{p}$ at the same position as q_i in $\boldsymbol{q}$. Let $T = \{s \in 2^{\mathrm{Var}(\phi)} \mid x \text{ is labelled with } s(x), \text{ for each } x \in X\}$. Since $\mathbb{B}$ accepts, each variable $x \in X$ has exactly one label and therefore assignments in T are well-defined. Moreover T includes all possible assignments over Y. One can easily observe that $T \models \phi$. T satisfies each literal because each $s \in T$ satisfies it. Let $\boldsymbol{p} \subseteq \boldsymbol{q}$ be an inclusion atom and $s \in T$ be an assignment. We know that for each $x \in \boldsymbol{q}$ that is fixed by s, the corresponding variable $y \in \boldsymbol{p}$ is also fixed, whereas, T contains every possible value for variables in $\boldsymbol{q}$ which are not fixed. This makes the inclusion atom true.

To prove the other direction, suppose that $\mathbb{B}$ rejects. Then there are three cases under which a variable contains contradictory labels. Either both labels of the variable are caused by a literal (Case 1), or inclusion atoms are involved in one (Case 2), or both (Case 3) labels. In other words, either ϕ contains $x \wedge \neg x$ as a subformula, or it contains $x \wedge \neg y$ and there is a sequence of inclusion atoms, such that keeping $x = 1$ and $y = 0$ contradicts some inclusion atoms in ϕ (see Fig. 4).

Case 1 Both labels of a variable x are caused by a literal. In this case, x takes two labels because ϕ contains $x \wedge \neg x$. The proof is trivial since ϕ is unsatisfiable.

$$x - z_1 \text{ —— } z_2 \text{ —— } \cdots - z_{n-1} - y \qquad \begin{array}{l} x - z_1^x - z_2^x - \cdots - z_{n-1}^x - v \\ y - z_1^y \text{ —— } z_2^y - \cdots \text{ - } z_{m-1}^y - v \end{array}$$

Fig. 4. Intuitive explanation of two cases in the proof. (Left) x and $\neg y$ propagate a conflicting value to eachother. (Right) x and $\neg y$ propagate conflicting values to v.

Var(ϕ)	x_1	x_2	x_3	x_4	x_5
Labels for Var(ϕ)	1	1		0	
Propagation due to $x_5 \subseteq x_4$	1	1		0	0
Propagation due to $x_1x_3 \subseteq x_5x_2$	1/0	1	1	0	0

Fig. 5. Labels for literals and their propagation to inclusion atoms (See Example 15).

Case 2 One label of a variable y is caused by a literal ($\neg y$ or y) and the other by inclusion atoms. Then, there are inclusion atoms $\boldsymbol{p}_j \subseteq \boldsymbol{q}_j$ and variables z_j for $j \leq n$ such that: $z_0 = x$, $z_n = y$, and z_j and z_{j+1} occur in the same position in $\boldsymbol{q}_j$ and $\boldsymbol{p}_j$, respectively, for $0 \leq j < n$. This implies that ϕ is not satisfiable since for any team T such that $T \models x \wedge \neg y$, T does not satisfy the subformula $\bigwedge_j \boldsymbol{p}_j \subseteq \boldsymbol{q}_j$ of ϕ. A similar reasoning applies if ϕ contains $\neg x \wedge y$ instead.

Case 3 Both labels of a variable v are caused by inclusion atoms. Then, there are two collections of inclusion atoms $\boldsymbol{p}_j \subseteq \boldsymbol{q}_j$ for $j \leq n$, and $\boldsymbol{r}_k \subseteq \boldsymbol{s}_k$ for $k \leq m$. Moreover, there are two sequences of variables z_j^x for $j \leq n$ and z_k^y for $k \leq m$, and a variable v such that, $z_0^x = x$, $z_0^y = y$, $z_n^x = v = z_m^y$, and
1. for each $j \leq n$, z_j^x appears in $\boldsymbol{q}_j$ at the same position, as z_{j+1}^x in $\boldsymbol{p}_j$,
2. for each $k \leq m$, z_k^y appears in $\boldsymbol{s}_k$ at the same position, as z_{k+1}^y in $\boldsymbol{r}_k$.

This again implies that ϕ is not satisfiable since for any T such that $T \models x \wedge \neg y$, it does not satisfy the subformula $\bigwedge_j \boldsymbol{p}_j \subseteq \boldsymbol{q}_j \wedge \bigwedge_k \boldsymbol{r}_k \subseteq \boldsymbol{s}_k$ of ϕ.

Consequently, the correctness follows. This completes the proof. □

Example 15. We include an example to highlights how $\mathbb{B}$ operates. Let $\phi := (x_1 \wedge x_2 \wedge \neg x_4) \wedge (x_1x_3 \subseteq x_5x_2) \wedge (x_5 \subseteq x_4)$. The table in Fig. 5 illustrates the steps of $\mathbb{B}$ on ϕ. Clearly, $\mathbb{B}$ rejects ϕ since the variable x_1 has conflicting labels.

The **FPT** cases for SAT of $\mathcal{PINC}$ follow from Theorem 9. Regarding MC, recall that we consider strict semantics alone. The results of Theorem 17 are obtained from the reduction for proving **NP**-hardness of MC_s for $\mathcal{PINC}$ [14]. Here we confirm that their reduction is indeed an *fpt*-reduction with respect to considered parameters. The following lemma is essential for proving Theorem 17 and we include it for self containment.

Lemma 16 **([14]).** MC *for* $\mathcal{PINC}$ *under strict semantics is* **NP**-*hard.*

Proof Idea. The hardness is achieved through a reduction from the set splitting problem to the model checking problem for $\mathcal{PINC}$ with strict semantics. An instance of set splitting problem consists of a family $\mathcal{F}$ of subsets of a finite set S. The problem asks if there are $S_1, S_2 \subseteq S$ such that $S_1 \cup S_2 = S$, $S_1 \cap S_2 = \emptyset$ and for each $A \in \mathcal{F}$ there exists $a_1, a_2 \in \mathcal{A}$ such that $a_1 \in S_1, a_2 \in S_2$. Let $\mathcal{F} = \{B_1, \ldots, B_n\}$ and $\bigcup \mathcal{F} = S = \{a_1, \ldots, a_k\}$. Let p_i and q_j denote fresh variables for each $a_i \in S$ and $B_j \in \mathcal{F}$. Moreover, let $V_{\mathcal{F}} = \{p_1, \ldots, p_k, q_1, \ldots, q_n, p_\top, p_c, p_d\}$. Then define $T_{\mathcal{F}} = \{s_1, \ldots, s_k, s_c, s_d\}$, where each assignment s_i is defined as follows:

$$s_i(p) := \begin{cases} 1, & \text{if } p = p_i \text{ or } p = p_\top, \\ 1, & \text{if } p = q_j \text{ and } a_i \in B_j \text{ for some } j, \\ 0, & \text{otherwise.} \end{cases}$$

That is, $T_{\mathcal{F}}$ includes an assignment s_i for each $a_i \in S$. The reduction also yields the following $\mathcal{PINC}$-formula.

$$\phi_{\mathcal{F}} := (\neg p_c \wedge \bigwedge_{i \leq n} p_\top \subseteq q_i) \vee (\neg p_d \wedge \bigwedge_{i \leq n} p_\top \subseteq q_i)$$

Clearly, the split of $T_{\mathcal{F}}$ into T_1, T_2 ensures the split of S into S_1 and S_2 and vice versa. Whereas, s_c and s_d ensure that none of the split is empty. □

Theorem 17. MC$_s$ *for* $\mathcal{PINC}$ *when parameterized by* k *is* **paraNP**-*complete if* $k \in \{\#\mathsf{splits}, \mathsf{arity}, \mathsf{formula\text{-}tw}\}$. *Whereas, it is* **FPT** *in other cases.*

Proof. Consider the $\mathcal{PINC}$-formula $\phi_{\mathcal{F}}$ from Lemma 16, which includes only one split-junction and the inclusion atoms have arity one. This gives the desired **paraNP**-hardness for MC$_s$ when parameterized by #splits and arity.

The proof for formula-tw is more involved and we prove the following claim. □

Claim. $\phi_{\mathcal{F}}$ has fixed formula-tw. That is, the treewidth of $\phi_{\mathcal{F}}$ is independent of the input instance $\mathcal{F}$ of the set-spliting problem. Moreover formula-tw$(\phi_{\mathcal{F}}) \leq 4$.

Proof of Claim. The $\mathcal{PINC}$-formula $\phi_{\mathcal{F}}$ is related to an input instance $\mathcal{F}$ of the set splitting problem only through its input size, which is n. Therefore the formula structure remains unchanged when we vary an input instance, only the size of two big conjunctions vary. To prove the claim, we give a tree decomposition for the formula with formula-tw$(\phi_{\mathcal{F}}) = 4$. Since the treewidth is minimum over all tree decompositions, this proves the claim. We rewrite the formula as below.

$$\phi_{\mathcal{F}} := (\neg p_c \wedge_l \bigwedge_{i \leq n} p_\top \subseteq_i^l q_i) \vee (\neg p_d \wedge_r \bigwedge_{i \leq n} p_\top \subseteq_i^r q_i)$$

That is, each subformula is renamed so that it is easy to identify as to which side of the split it appears (e.g., $p_\top \subseteq_i^l q_i$ denotes the ith inclusion atom in the big conjunction on the left, denoted as I_i^l in the graph). The graphical representation

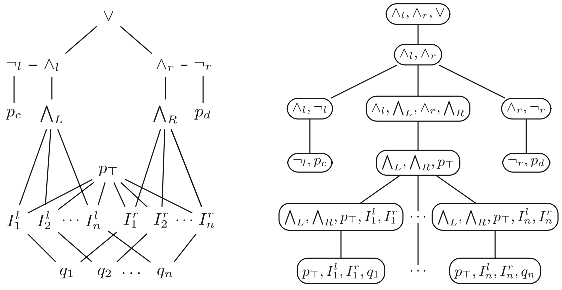


Fig. 6. The Gaifman graph (Left) and a tree decomposition (Right) for $\phi_{\mathcal{F}}$. Note that we abbreviated subformulas in the inner vertices of the Gaifman graph for presentation reasons. Also, edges between $p_\top$ and variables q_i are omitted for better presentation, but those are covered in the decomposition on the right.

of $\phi_{\mathcal{F}}$ with $V = \mathrm{SF}(\phi_{\mathcal{F}}) \cup \mathrm{Var}(\phi_{\mathcal{F}})$, as well as, a tree decomposition, is given in Fig. 6. Notice that there is an edge between x and y in the Gaifman graph if and only if either y is an immediate subformula of x, or y is a variable appearing in the inclusion atom x. It is easy to observe that the decomposition presented in Fig. 6 is indeed a valid tree decomposition in which each node is labelled with its corresponding bag. Moreover, since the maximum bag size is 5, the treewidth of this decomposition is 4. This proves the claim. ■

The remaining **FPT**-cases for MC_s follow from Theorem 7 and Corollary 8. This completes the proof to our theorem. □

Recall that a dependence atom $=(\boldsymbol{x};\boldsymbol{y})$ is equivalent with the independence atom $\boldsymbol{y}\perp_{\boldsymbol{x}}\boldsymbol{y}$. As a consequence, (in the classical setting) hardness results for $\mathcal{PDL}$ immediately translate to those for $\mathcal{PIND}$. Nevertheless, in the parameterized setting, one has to further check whether this translation 'respects' the parameter value of the two instances. This concerns the parameter arity and formula-tw. This is due to the reason that, a dependence atom $=(\boldsymbol{x};\boldsymbol{y})$ has arity $|\boldsymbol{x}|$, whereas, the equivalent independence atom $\boldsymbol{y}\perp_{\boldsymbol{x}}\boldsymbol{y}$ has arity $|\boldsymbol{x}\cup\boldsymbol{y}|$.

Theorem 18. MC *for $\mathcal{PIND}$, when parameterized by k is* **paraNP***-complete if $k \in \{$#splits, arity, formula-tw$\}$. Whereas, it is* **FPT** *in other cases.*

Proof. Notice that MC for $\mathcal{PDL}$ when parameterized by $k \in \{$arity, #splits, formula-tw$\}$ is also **paraNP**-complete. We argue that in reductions for $\mathcal{PDL}$, replacing dependence atoms by the equivalent independence atoms yield *fpt*-reduction for the above mentioned cases. Moreover, this holds for both strict and lax semantics.

For formula-tw and arity, when proving **paraNP**-hardness of $\mathcal{PDL}$, the resulting formula has treewidth of one [22, Cor. 16] and the arity is zero [22, Theorem 15]. Moreover, only dependence atoms of the form $=(;p)$ where p is a propositional variable, are used and the syntax structure of the $\mathcal{PDL}$-formula is already a tree. Consequently, replacing $=(;p)$ with $p\perp_{\emptyset}p$ implies that only independence atoms of arity 1 are used. Notice also that replacing dependence atoms by independence atoms does not increase the treewidth of the input formula. This is because when translating dependence atoms into independence atoms, no new variables are introduced. As a result, the reduction also preserves the treewidth. This proves the claim as 1-slice regarding both parameters arity and formula-tw, is **NP**-hard.

Regarding the #splits, the claim follows due to Mahmood and Meier [22, Theorem 18] because the reduction from the colouring problem uses only 2 splits.

Finally, the **FPT** cases follow from Theorem 7 and Corollary 8. □

Theorem 19. SAT *for* $\mathcal{PIND}$*, parameterized by* arity *is* **paraNP**-*complete. Whereas, it is* **FPT** *in other cases.*

Proof. Recall that $\mathcal{PL}$ is a fragment of $\mathcal{PIND}$. This immediately gives **paraNP**-hardness when parameterized by arity, because SAT for $\mathcal{PL}$ is **NP**-complete. The **paraNP**-membership is clear since SAT for $\mathcal{PIND}$ is also **NP**-complete [12, Theorem 1]. The **FPT** cases for $k \in \{$formula-depth, #variables$\}$ follow because of Theorem 9. The cases for #splits and formula-tw follow due to a similar reasoning as in $\mathcal{PDL}$ [22] because it is enough to find a singleton satisfying team [13, Lemma 4.2]. This completes the proof. □

5 Concluding Remarks

We presented a parameterized complexity analysis for $\mathcal{PINC}$ and $\mathcal{PIND}$. The problems we considered were satisfiability and model checking. Interestingly, the parameterized complexity results for $\mathcal{PIND}$ coincide with that of $\mathcal{PDL}$ [22] in each case. Moreover, the complexity of model checking under a given parameter remains the same for all three logics. We proved that for a team based logic $\mathcal{L}$ such that $\mathcal{L}$-atoms can be evaluated in **P**-time, MC for $\mathcal{L}$ when parameterized by team-size is always **FPT**.

It is interesting to notice that for $\mathcal{PDL}$ and $\mathcal{PIND}$, SAT is easier than MC when parameterized by formula-tw. This is best explained by the fact that $\mathcal{PDL}$ is downwards closed and a formula is satisfiable iff some singleton team satisfies it. Moreover, $\mathcal{PIND}$ also satisfies this 'satisfiable under singleton team' property. The parameters team-size and formula-team-tw are not meaningful for SAT due to the reason that we do not impose a size restriction for the satisfying team in SAT. Furthermore, arity is quite interesting because SAT for all three logics is **paraNP**-complete. This implies that while the fixed arity does not lower the complexity of SAT in $\mathcal{PDL}$ and $\mathcal{PIND}$, it does lower it from **EXP**-completeness to **NP**-completeness for $\mathcal{PINC}$. As a byproduct, we obtain that the complexity of satisfiability for the fixed arity fragment of $\mathcal{PINC}$ is **NP**-complete. Thereby, we answer an open question posed by Hella and Stumpf [15, P.13]. The **paraNP**-membership of SAT when parameterized by arity implies that one can encode the problem into classical satisfiability and employ a SAT-solver to solve satisfiability for the fixed arity fragment of $\mathcal{PINC}$. We leave as a future work the suitable

SAT-encoding for $\mathcal{PINC}$ that runs in **FPT**-time and enables one to use SAT-solvers. Further future work involves finding the precise complexity of SAT for $\mathcal{PINC}$ when parameterized by #splits and formula-tw.

Acknowledgement. This work was supported by the European Union's Horizon Europe research and innovation programme within project ENEXA (101070305) and by the German Research Foundation (DFG), project VI 1045/1-1.

References

1. Corander, J., Hyttinen, A., Kontinen, J., Pensar, J., Väänänen, J.: A logical approach to context-specific independence. Ann. Pure Appl. Logic **170**(9), 975–992 (2019). https://doi.org/10.1016/j.apal.2019.04.004
2. Downey, R.G., Fellows, M.R.: Fundamentals of Parameterized Complexity. TCS, Springer, London (2013). https://doi.org/10.1007/978-1-4471-5559-1
3. Durand, A., Hannula, M., Kontinen, J., Meier, A., Virtema, J.: Approximation and dependence via multiteam semantics. Ann. Math. Artif. Intell. **83**(3–4), 297–320 (2018). https://doi.org/10.1007/s10472-017-9568-4
4. Durand, A., Hannula, M., Kontinen, J., Meier, A., Virtema, J.: Probabilistic team semantics. In: Ferrarotti, F., Woltran, S. (eds.) FoIKS 2018. LNCS, vol. 10833, pp. 186–206. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-90050-6_11
5. 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). https://doi.org/10.1007/978-3-642-39992-3_13
6. Ebbing, J., Lohmann, P.: Complexity of model checking for modal dependence logic. In: Bieliková, M., Friedrich, G., Gottlob, G., Katzenbeisser, S., Turán, G. (eds.) SOFSEM 2012. LNCS, vol. 7147, pp. 226–237. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-27660-6_19
7. Flum, J., Grohe, M.: Parameterized Complexity Theory. TTCSAES, Springer, Heidelberg (2006). https://doi.org/10.1007/3-540-29953-X
8. Galliani, P.: Inclusion and exclusion dependencies in team semantics - on some logics of imperfect information. Ann. Pure Appl. Logic **163**(1), 68–84 (2012). https://doi.org/10.1016/j.apal.2011.08.005
9. Hannula, M., Kontinen, J., Van den Bussche, J., Virtema, J.: Descriptive complexity of real computation and probabilistic independence logic. In: LICS 2020, pp. 550–563. ACM (2020). https://doi.org/10.1145/3373718.3394773
10. Hannula, M., Kontinen, J., Lück, M., Virtema, J.: On quantified propositional logics and the exponential time hierarchy. In: GandALF. EPTCS, vol. 226, pp. 198–212 (2016)
11. Hannula, M., Kontinen, J., Virtema, J.: Polyteam semantics. J. Log. Comput. **30**(8), 1541–1566 (2020). https://doi.org/10.1093/logcom/exaa048
12. Hannula, M., Kontinen, J., Virtema, J., Vollmer, H.: Complexity of propositional independence and inclusion logic. In: Italiano, G.F., Pighizzini, G., Sannella, D.T. (eds.) MFCS 2015. LNCS, vol. 9234, pp. 269–280. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48057-1_21
13. Hannula, M., Kontinen, J., Virtema, J., Vollmer, H.: Complexity of propositional logics in team semantic. ACM Trans. Comput. Log. **19**(1), 2:1–2:14 (2018). https://doi.org/10.1145/3157054

14. Hella, L., Kuusisto, A., Meier, A., Virtema, J.: Model checking and validity in propositional and modal inclusion logics. J. Log. Comput. **29**(5), 605–630 (2019). https://doi.org/10.1093/logcom/exz008
15. Hella, L., Stumpf, J.: The expressive power of modal logic with inclusion atoms. In: Proceedings of the 6th GandALF, pp. 129–143 (2015)
16. Hyttinen, T., Paolini, G., Väänänen, J.: A logic for arguing about probabilities in measure teams. Arch. Math. Logic **56**(5–6), 475–489 (2017). https://doi.org/10.1007/s00153-017-0535-x
17. Kontinen, J., Mahmood, Y., Meier, A., Vollmer, H.: Parameterized complexity of weighted team definability (2023). https://doi.org/10.48550/arXiv.2302.00541
18. Kontinen, J., Meier, A., Mahmood, Y.: A parameterized view on the complexity of dependence and independence logic. J. Log. Comput. **32**(8), 1624–1644 (2022). https://doi.org/10.1093/logcom/exac070
19. Krebs, A., Meier, A., Virtema, J.: A team based variant of CTL. In: TIME 2015, pp. 140–149 (2015). https://doi.org/10.1109/TIME.2015.11
20. Krebs, A., Meier, A., Virtema, J., Zimmermann, M.: Team semantics for the specification and verification of hyperproperties. In: MFCS 2018, vol. 117, pp. 10:1–10:16. Dagstuhl, Germany (2018). https://doi.org/10.4230/LIPIcs.MFCS.2018.10
21. Lohmann, P., Vollmer, H.: Complexity results for modal dependence logic. Stud. Logica **101**(2), 343–366 (2013). https://doi.org/10.1007/s11225-013-9483-6
22. Mahmood, Y., Meier, A.: Parameterised complexity of model checking and satisfiability in propositional dependence logic. Ann. Math. Artif. Intell. **90**(2), 271–296 (2021). https://doi.org/10.1007/s10472-021-09730-w
23. Meier, A., Reinbold, C.: Enumeration complexity of poor man's propositional dependence logic. In: Ferrarotti, F., Woltran, S. (eds.) FoIKS 2018. LNCS, vol. 10833, pp. 303–321. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-90050-6_17
24. Peterson, G., Reif, J., Azhar, S.: Lower bounds for multiplayer noncooperative games of incomplete information. Comput. Math. Appl. **41**(7), 957–992 (2001)
25. Peterson, G.L., Reif, J.H.: Multiple-person alternation, pp. 348–363. IEEE Computer Society (1979). https://doi.org/10.1109/SFCS.1979.25
26. Pippenger, N.: Theories of Computability. Cambridge University Press, Cambridge (1997)
27. Shukla, A., Biere, A., Pulina, L., Seidl, M.: A survey on applications of quantified Boolean formulas. In: ICTAI 2019 (2019). https://doi.org/10.1109/ICTAI.2019.00020
28. Väänänen, J.: Dependence Logic. Cambridge University Press, Cambridge Cambridge (2007)
29. Väänänen, J., Hodges, W.: Dependence of variables construed as an atomic formula (2008)
30. Virtema, J.: Complexity of validity for propositional dependence logics. Inf. Comput. **253**, 224–236 (2017). https://doi.org/10.1016/j.ic.2016.07.008

Parallelism in Realizability Models

Satoshi Nakata(✉)

Research Institute for Mathematical Sciences, Kyoto University, Kyoto, Japan
nakata@kurims.kyoto-u.ac.jp

Abstract. Study of parallel operations such as Plotkin's parallel-or has promoted the development of the theory of programming languages. In this paper, we consider parallel operations in the framework of categorical realizability. Given a partial combinatory algebra A equipped with an "abstract truth value" Σ (called predominance), we introduce the notions of Σ-or and Σ-and combinators in A. By choosing a suitable A and Σ, a form of parallel-or may be expressed as a Σ-or combinator. We then investigate the relationship between these combinators and the realizability model $\mathbf{Ass}(A)$ (the category of assemblies over A) and show the following: under a natural assumption on Σ, (i) A admits Σ-and combinator iff for any assembly $X \in \mathbf{Ass}(A)$ the Σ-subsets (canonical subassemblies) of X form a poset with respect to inclusion. (ii) A admits both Σ-and and Σ-or combinators iff for any $X \in \mathbf{Ass}(A)$ the Σ-subsets of X form a lattice with respect to intersection and union.

Keywords: Realizability · Partial combinatory algebra · Parallel-or function

1 Introduction

Traditionally, the *realizability interpretation* has been introduced as semantics of intuitionistic arithmetic. It rigorously defines "what it means to justify a proposition by an algorithm." While it is originally formulated in terms of recursive functions [8], it is later generalized to a framework based on *Partial Combinatory Algebras* (PCAs), which include various computational models. The interpretation itself has been given a categorical generalization, such as the *realizability topos* and the category of *assemblies*. In particular, in the category $\mathbf{Ass}(A)$ of assemblies over PCA A, we can discuss implementation of mathematical structures and functions by algorithms [19]. Moreover, $\mathbf{Ass}(A)$ provide effective models to higher-order programming languages such as PCF [1,9,14].

In this paper, we will consider how the structure of the realizability model $\mathbf{Ass}(A)$ is affected by the choice of a computational model A. More specifically, we focus on the following two concepts.

I. Parallel operations in PCA:
Comparing Kleene's first algebra $\mathcal{K}_1$ and term models of lambda calculus as PCA, there is a difference in the degree of parallelism. For example, term

H. H. Hansen et al. (Eds.): WoLLIC 2023, LNCS 13923, pp. 292–304, 2023.
https://doi.org/10.1007/978-3-031-39784-4_18

models exclude Plotkin's *parallel-or* function [16], whereas $\mathcal{K}_1$ does not. While such a parallel operation has received a lot of attention in the theory of programming languages, it also plays an implicit role in elementary recursion theory. For example, the union of two semi-decidable sets $U, V \subseteq \mathbb{N}$ is again semi-decidable precisely because a Turing machine can check whether input $n \in \mathbb{N}$ belongs to U or to V *in parallel*. In this paper, we first consider a pair of nonempty subsets $\Sigma = (T, F)$ of PCA A as an "abstract truth value" and define combinators Σ-or and Σ-and in A. In a suitable A, these notions may express a form of parallel operations.

II. Σ-subsets in $\mathbf{Ass}(A)$:
It is known that such a pair $\Sigma = (T, F)$ may be identified with a *predominance* $t : 1 \to \Sigma$ in $\mathbf{Ass}(A)$, which is a morphism obtained by weakening the condition for being a subobject classifier [9]. An important feature is that, for every assembly X over A, Σ induces a certain class of "canonical" subassemblies of X. It is called the class of *Σ-subsets* of X and is written $\mathrm{Sub}_\Sigma(X)$. Unlike the subobject lattice $\mathrm{Sub}(X)$, $\mathrm{Sub}_\Sigma(X)$ does not form a poset in general. When it does, Σ is called *dominance* and used to construct a subcategory of (internal) domains in the context of *Synthetic domain theory* [7,9,13,14,18].
Interestingly, considering a suitable Σ in $\mathbf{Ass}(\mathcal{K}_1)$, the Σ-subsets of a natural number object exactly correspond to the semi-decidable subsets of $\mathbb{N}$ [11]. That is, the notion of Σ-subset can be regarded as a generalization of semi-decidable set. From the discussion in I., we can expect that if A admits Σ-or, then $\mathrm{Sub}_\Sigma(X)$ is closed under union.

The purpose of this paper is to give a precise correspondence between these two concepts. In particular, we prove the following results. Under a natural assumption on a predominance Σ, A admits Σ-and combinator if and only if, for every assembly X, the Σ-subsets of X form a poset with respect to inclusion (Theorem 24). Furthermore, A admits both Σ-and and Σ-or combinators if and only if, for every assembly X, the Σ-subsets of X form a lattice with respect to intersection and union (Theorem 28).

Outline

The structure of this paper is as follows. In Sect. 2, we give some basic definitions and properties about PCAs. In Sect. 3, we introduce the notions of Σ-or and Σ-and combinators in a PCA relative to an "abstract truth value" (predominance) Σ. In Sect. 4, we proceed to the category $\mathbf{Ass}(A)$ of assemblies over A and the notion of Σ-subset. Lastly, in Sect. 5, we discuss the relationship between Σ-or and Σ-and combinators in A and the structure of the Σ-subsets in $\mathbf{Ass}(A)$.

2 Preliminary

We review some basic concepts and notations in realizability theory.

Definition 1 **([9]).** *A* partial combinatory algebra (PCA) *is a set A equipped with a partial binary operation $\cdot : A \times A \rightharpoonup A$ such that there exist elements* k, s $\in A$ *satisfying the conditions*

$$\mathrm{k} \cdot x \downarrow, \quad (\mathrm{k} \cdot x) \cdot y = x, \quad (\mathrm{s} \cdot x) \cdot y \downarrow, \quad ((\mathrm{s} \cdot x) \cdot y) \cdot z \cong (x \cdot z) \cdot (y \cdot z)$$

for any $x, y, z \in A$. Here $\downarrow$ is to be read as "defined" (and $\uparrow$ as "undefined") and $\cong$ means that if one side is defined, then so is the other and they are equal. We often write xy instead of $x \cdot y$, and axy instead of $(ax)y$. A PCA is called total *if its operation is total. Obviously, a singleton forms a total PCA, that is called a* trivial *PCA.*

PCA is often regarded as an "abstract machine" and there are many interesting examples: Turing machines, λ-calculus, the continuous functions of type $\omega^\omega \to \omega$, a reflexive object in any cartesian-closed category [19]. A common feature of PCAs is that they can imitate untyped λ-calculus as follows.

Notation 2. *Let $T(A)$ denote the set of terms generated by constants a, b, $\cdots \in A$, variables x, y, $\cdots$ and binary function symbol $\cdot$. We write $FV(t)$ for the set of free variables occurring in $t \in T(A)$.*

Given a term $t \in T(A)$ and a variable x, we define a new term $\lambda^* x.\, t$ by induction on the structure of t. For instance, $\lambda^* x.\, x$ is defined by skk, $\lambda^* x.\, t$ by kt if t is either a variable $y \neq x$ or a constant a, and $\lambda^* x.\, tt'$ by $\mathrm{s}(\lambda^* x.\, t)(\lambda^* x.\, t')$. By repetition, we obtain an element $\lambda^* \boldsymbol{x}.\, t(\boldsymbol{x})$ in A for any $\boldsymbol{x} = x_1, \cdots, x_n$.

Theorem 3 **([9,19]).** *Let A be a PCA and $t(\boldsymbol{x}) \in T(A)$. Then, for any $a_1, \cdots, a_n \in A$, $(\lambda^* \boldsymbol{x}.\, t(\boldsymbol{x}))a_1 \cdots a_{n-1}$ is defined and $(\lambda^* \boldsymbol{x}.\, t(\boldsymbol{x}))a_1 \cdots a_n \cong t(a_1, \cdots, a_n)$ holds.*

Remark 4. In particular, $\lambda^* x.\, (ab) := \mathrm{s}(\mathrm{k}a)(\mathrm{k}b) \in A$ is always defined even if $a \cdot b \uparrow$. This dummy λ-abstraction is useful to lock the evaluation. It may be later unlocked by applying it to an arbitrary element c in A:

$$(\lambda^* x.\, ab) \cdot c \cong a \cdot b.$$

This technique is used in Sects. 3 and 5.

Notation 5. *We use the following notations:* i $:= \lambda^* x.\, x$, true $:= \lambda^* xy.\, x$, false $:= \lambda^* xy.\, y$, (if b then x else y) $:= bxy$, $\langle x, y \rangle := \lambda^* z.\, zxy$, fst $:= \lambda^* p.\, p$(true), snd $:= \lambda^* p.\, p$(false).

In this paper, we are mainly interested in the following examples.

Example 6. (i) **Kleene's first algebra** $\mathcal{K}_1$: Consider the set of natural numbers $\mathbb{N}$ with a partial operation $\cdot : \mathbb{N} \times \mathbb{N} \rightharpoonup \mathbb{N}$ defined by $n \cdot m := [\![n]\!](m)$, where $[\![n]\!]$ is the n-th partial computable function (with respect to a fixed effective numbering of Turing machines). This PCA is called *Kleene's first algebra* and is denoted by $\mathcal{K}_1$. The undefinedness $\uparrow$ of $a \cdot b$ can be regarded as divergence of computation.

(ii) **λ-term models**: Let Λ^0 be the set of *closed* λ-terms and T a *λ-theory*, that is, a congruence relation on λ-terms which contains β-equivalence. Considering the quotient modulo T, we obtain a total PCA Λ^0/T equipped with the application operation.
Another variation of λ-term model is given based on the *call-by-value* reduction strategy on Λ^0. A *value* is either an abstraction $\lambda x. M$ or a variable x. Values are denoted by V, W and the set of closed values by Λ^0_v. According to [5, Definition 7], we define $\to_{cbv}$ by the following binary relation (where $\overline{N} \equiv N_1, \cdots, N_n$ with $n \geq 0$):

$$\frac{}{(\lambda x. M)V\overline{N} \to_{cbv} M[V/x]\overline{N}} \qquad \frac{M \to_{cbv} M'}{VM\overline{N} \to_{cbv} VM'\overline{N}}$$

That is, one reduces a term from left to right with the constraint that the β-reduction can be applied only when the argument is a value. The transitive reflexive closure of $\to_{cbv}$ is denoted by $\twoheadrightarrow_{cbv}$. Note that the above reduction is called the left reduction in Plotkin's seminal work [15].
Define a partial operation $\cdot : \Lambda^0_v \times \Lambda^0_v \rightharpoonup \Lambda^0_v$ by:

$$V_1 \cdot V_2 \ := W \ \text{ if } V_1 V_2 \twoheadrightarrow_{cbv} W \text{ and } W \in \Lambda^0_v.$$

Otherwise, $V_1 \cdot V_2$ is undefined. Together with combinators $S := \lambda xyz. xz(yz)$ and $K := \lambda xy. x$, we obtain a non-total PCA $(\Lambda^0_v, \cdot)$.

3 Parallel Combinators in PCA

Recall that Plotkin's parallel-or function $\mathrm{por}^{\mathrm{p}}$, originally introduced in the context of PCF [16], behaves as follows:

$$\begin{array}{ll} \mathrm{por}^{\mathrm{p}} MN \Downarrow \mathrm{true} & \text{if } M \Downarrow \mathrm{true} \text{ or } N \Downarrow \mathrm{true}, \\ \mathrm{por}^{\mathrm{p}} MN \Downarrow \mathrm{false} & \text{if } M \Downarrow \mathrm{false} \text{ and } N \Downarrow \mathrm{false}, \\ \mathrm{por}^{\mathrm{p}} MN \Uparrow & \text{otherwise} \end{array}$$

(where M, N are terms and $M \Downarrow V$ means that M evaluates to a value V). The point is that evaluation of a term may diverge. Hence one has to evaluate the arguments M, N in parallel to check if $\mathrm{por}^{\mathrm{p}} MN \Downarrow \mathrm{true}$. Given $\mathrm{por}^{\mathrm{p}}$, we may define a term por such that

$$(1) \qquad \mathrm{por} MN \Downarrow \quad \text{iff} \quad M \Downarrow \text{ or } N \Downarrow,$$

that may be seen as a weaker form of parallel-or. We now consider such operations in a PCA A. To make things as general as possible, we define them relative to two nonempty subsets (T, F) of A, which stand for "true/termination" and "false/failure", respectively.

The idea of dealing with two nonempty subsets of A is due to Longley. Actually he considered a more general notion of divergence in [9,10]. As he pointed out, these data correspond to a *predominance* in the category **Ass**(A) of assemblies.

Definition 7. *Given S_0, $S_1 \subseteq A$, we define $S_0 \times S_1 := \{\, \langle a_0, a_1 \rangle \in A \mid a_0 \in S_0$ and $a_1 \in S_1 \,\}$.*

We call a pair $\Sigma = (T, F)$ of nonempty subsets of A, which need not be disjoint, a predominance *on A. An element $\mathrm{or}_\Sigma \in A$ is called a Σ-*or *combinator if it satisfies*

$$\mathrm{or}_\Sigma(T \times T) \subseteq T, \quad \mathrm{or}_\Sigma(T \times F) \subseteq T,$$
$$\mathrm{or}_\Sigma(F \times T) \subseteq T, \quad \mathrm{or}_\Sigma(F \times F) \subseteq F.$$

*To be precise, $\mathrm{or}_\Sigma(T \times T) \subseteq T$ means that for every $f, g \in T$, $\mathrm{or}_\Sigma\langle f, g \rangle$ is defined and belongs to T. Dually, an element $\mathrm{and}_\Sigma \in A$ is called a Σ-*and *combinator if it satisfies*

$$\mathrm{and}_\Sigma(T \times T) \subseteq T, \quad \mathrm{and}_\Sigma(T \times F) \subseteq F,$$
$$\mathrm{and}_\Sigma(F \times T) \subseteq F, \quad \mathrm{and}_\Sigma(F \times F) \subseteq F.$$

We say that A admits Σ-or *if there exists or_Σ in A, and similarly for Σ-and.*

Example 8. Let $\Sigma_{\mathrm{d}} := (\{\,\mathrm{true}\,\}, \{\,\mathrm{false}\,\})$. Then, every PCA admits Σ_{d}-or and Σ_{d}-and because $\mathrm{or}_{\Sigma_{\mathrm{d}}}$ can be defined as

$$\lambda^* p.\,(\text{if fst} \cdot p \text{ then true else (if snd} \cdot p \text{ then true else false)}),$$

and similarly for $\mathrm{and}_{\Sigma_{\mathrm{d}}}$.

Example 9. Berry showed the following sequentiality theorem. Consider a λ-theory $T_{\mathcal{BT}}$ that identifies λ-terms which have the same Böhm tree. In the PCA $\Lambda^0/T_{\mathcal{BT}}$, there is no term M such that

$$M\langle \mathrm{i}, \Omega \rangle = M\langle \Omega, \mathrm{i} \rangle = \mathrm{i}, \qquad M\langle \Omega, \Omega \rangle = \Omega,$$

where $\Omega := (\lambda x.\, xx)(\lambda x.\, xx)$ (See [2]). Hence $\Lambda^0/T_{\mathcal{BT}}$ does not admit Σ-or with respect to $\Sigma = (\{\,\mathrm{i}\,\}, \{\,\Omega\,\})$.

We next introduce an important predominance, which works uniformly for all non-total PCAs. This example is essentially due to Mulry.

Definition 10 ([11]). *For a non-total A, define a predominance $\Sigma_{\mathrm{sd}} := (T_{\mathrm{sd}}, F_{\mathrm{sd}})$ by*

$$T_{\mathrm{sd}} := \{\, a \in A \mid a \cdot \mathrm{i} \downarrow \,\}, \quad F_{\mathrm{sd}} := \{\, a \in A \mid a \cdot \mathrm{i} \uparrow \,\}.$$

By definition, every Σ_{sd}-or combinator satisfies $\mathrm{or}_{\Sigma_{\mathrm{sd}}}\langle f, g \rangle \downarrow$ and

$$\mathrm{or}_{\Sigma_{\mathrm{sd}}}\langle f, g \rangle \cdot \mathrm{i} \downarrow \quad \text{iff} \quad f \cdot \mathrm{i} \downarrow \text{ or } g \cdot \mathrm{i} \downarrow$$

for every $f, g \in A$. In analogy with (1), we simply call $\mathrm{or}_{\Sigma_{\mathrm{sd}}}$ a *parallel-or* combinator and dually call $\mathrm{and}_{\Sigma_{\mathrm{sd}}}$ a *parallel-and*. We have chosen i as the “key” to “unlock” the evaluation, but actually it can be anything.

Proposition 11. *For every non-total PCA A, A admits parallel-or* $\mathrm{or}_{\Sigma_{\mathrm{sd}}}$ *if and only if A has a combinator* $\mathrm{por}^{\mathrm{u}}$ *that satisfies* $\mathrm{por}^{\mathrm{u}}\langle f, g\rangle \downarrow$ *and*

$$\mathrm{por}^{\mathrm{u}}\langle f, g\rangle \cdot a \downarrow \quad \text{iff} \quad f \cdot a \downarrow \;\; \text{or } g \cdot a \downarrow$$

for any $f, g, a \in A$.

Let $\mathrm{dom}(f)$ denote the set $\{\, a \in A \mid f \cdot a \downarrow \,\}$. Then we have $\mathrm{dom}(\mathrm{por}^{\mathrm{u}}\langle f, g\rangle) = \mathrm{dom}(f) \cup \mathrm{dom}(g)$. Since subsets of the form $\mathrm{dom}(f)$ are precisely the semi-decidable sets (computably enumerable sets) in $\mathcal{K}_1$, we may claim that our parallel-or combinator has a generalized ability to take the union of two semi-decidable sets.

Let us now examine which PCA admits parallel-and (resp. parallel-or). We may expect that any PCA has a combinator which behaves as follows: "evaluate $f \cdot \mathrm{i}$ first; if it terminates, evaluate $g \cdot \mathrm{i}$ next." If we try to express this by a λ-term, we get a Σ_{sd}-and combinator (parallel-and).

Theorem 12. *Every non-total PCA admits parallel-and.*

On the other hand, parallel-or is more subtle. It is certainly true that Turing machines can perform a computation like: "evaluate $f \cdot \mathrm{i}$ and $g \cdot \mathrm{i}$ in parallel until one of them terminates." However, such a computation cannot be performed in λ-calculus due to its sequential nature. Consequently,

Proposition 13. *$\mathcal{K}_1$ admits both parallel-and and parallel-or, while Λ_v^0 admits parallel-and but not parallel-or.*

4 Predominances in the Category of Assemblies

In the modern theory of realizability, one builds a category over a given PCA A, in such a way that elements of A are used to implement a function or to justify a proposition in the constructive sense. There are several examples such as the *realizability topos* $\mathbf{RT}(A)$, the category $\mathbf{Ass}(A)$ of *assemblies* and the category $\mathbf{Mod}(A)$ of *modest sets* [19]. In particular, considering $\mathbf{RT}(\mathcal{K}_1)$, we can obtain the *effective topos* of Hyland [6] and the standard interpretation of first-order number theory in $\mathbf{RT}(\mathcal{K}_1)$ precisely corresponds to Kleene's traditional realizability interpretation [8]. In this sense, such categories are called "realizability models" in the literature.

We here focus on $\mathbf{Ass}(A)$, a full subcategory of $\mathbf{RT}(A)$. Notably, the latter can be obtained from the former by the *exact completion* [3,9]. $\mathbf{Ass}(A)$ is more primitive than $\mathbf{RT}(A)$ and is sufficiently rich as semantics of programming languages [1,9,14].

Definition 14. *An* assembly *over A is a pair $X = (|X|, ||\cdot||_X)$, where $|X|$ is a set and $||\cdot||_X : |X| \to \mathcal{P}(A)$ is a function such that $||x||_X$ is nonempty for any $x \in |X|$. An element $a \in A$ is called a* realizer *of x if $a \in ||x||_X$. A morphism of assemblies $f : (|X|, ||\cdot||_X) \to (|Y|, ||\cdot||_Y)$ is a function $f : |X| \to |Y|$ which has a* realizer *$r_f \in A$, that is, for any $x \in |X|$ and $a \in ||x||_X$, $r_f a$ is defined and in $||f(x)||_Y$. We say that r_f* realizes *f.*

One can verify that the assemblies and morphisms over A form a category $\mathbf{Ass}(A)$ (whose composition and identity are inherited from the category of sets). It has a terminal object given by $1 := (\{ * \}, ||\cdot||_1)$ with $||*||_1 := A$. Furthermore, $\mathbf{Ass}(A)$ always has a natural number object (NNO) N. For example, a canonical NNO in $\mathbf{Ass}(\mathcal{K}_1)$ is given by $N := (\mathbb{N}, ||\cdot||_N)$ with $||n||_N := \{ n \}$. The hom-set on N exactly corresponds the set of total computable functions on $\mathbb{N}$.

$\mathbf{Ass}(A)$ is a finitely complete locally cartesian-closed category [9,19]. This is a common feature of toposes such as the category of sets and realizability toposes. Every topos, in addition, has a subobject classifier, while $\mathbf{Ass}(A)$ does not unless A is trivial. Nevertheless, as one can see in [9,14], there is a useful concept of a "restricted classifier". Recall that a morphism $t : 1 \rightarrowtail \Sigma$ in a finitely complete category is a *subobject classifier* if for every monomorphism $m : U \rightarrowtail X$ there is *exactly one* morphism $\chi_m : X \to \Sigma$ which gives a pullback diagram

$$\begin{array}{ccc} U & \xrightarrow{\;!\;} & 1 \\ {\scriptstyle m}\big\downarrow & \lrcorner & \big\downarrow{\scriptstyle t} \\ X & \xrightarrow[\chi_m]{} & \Sigma. \end{array}$$

χ_m is called the *characteristic map* of m. By slightly weakening the condition, we obtain the concept of predominance.

Definition 15 ([17]). *Let $\mathcal{C}$ be a finitely complete category and Σ an object of $\mathcal{C}$. A monomorphism $t : 1 \rightarrowtail \Sigma$ is a* predominance *if every monomorphism $m : U \rightarrowtail X$ has* at most one *characteristic map χ_m in the above sense.*

*A subobject $[m]$ of X (that is the equivalence class of a monomorphism $m : U \rightarrowtail X$) is called Σ-*subset *of X and written $U \subseteq_\Sigma X$ if m arises as a pullback of $1 \rightarrowtail \Sigma$. Let $\mathrm{Sub}_\Sigma(X)$ denote the set of Σ-subsets of X.*

By definition, $\mathrm{Sub}_\Sigma(X)$ is a subclass of $\mathrm{Sub}(X)$, the class of subobjects of X. If $t : 1 \rightarrowtail \Sigma$ is a subobject classifier, we have $\mathrm{Sub}_\Sigma(X) = \mathrm{Sub}(X)$ for every X. One can easily show that a predominance $t : 1 \rightarrowtail \Sigma$ is an isomorphism iff $\mathrm{Sub}_\Sigma(X)$ consists of the equivalence class of isomorphisms. Such a predominance is called *trivial*.

Longley discussed the above notions in $\mathbf{Ass}(A)$ [9]. Suppose that a monomorphism $t : 1 \rightarrowtail \Sigma$ in $\mathbf{Ass}(A)$ is a predominance. Then we can observe that the cardinality of the underlying set $|\Sigma|$ is no more than two. Further if $\mathrm{card}|\Sigma| = 1$, Σ is a terminal object in $\mathbf{Ass}(A)$, hence t is trivial. Thus the non-triviality of t implies that Σ has a doubleton $|\Sigma| = \{ t, f \}$ as the underlying set, so it determines a predominance $(||t||_\Sigma, ||f||_\Sigma)$ on A. Conversely, each predominance (T, F) on A induces a non-trivial predominance $t : 1 \rightarrowtail \Sigma$ with $|\Sigma| := \{ t, f \}$, $||t||_\Sigma := T$ and $||f||_\Sigma := F$. To sum up:

Theorem 16 ([9, Subsection 4.2]). *The non-trivial predominances in $\mathbf{Ass}(A)$ are in bijective correspondence with the predominances on A.*

Moreover, every monomorphism $m : U \rightarrowtail X$ that arises as a pullback of $t : 1 \rightarrowtail \Sigma$ is isomorphic to the inclusion $U' \rightarrowtail X$ whose domain is a canonical subassembly defined below.

Definition 17. *Let X be an assembly in* $\mathbf{Ass}(A)$*. An assembly* $U = (|U|, ||\cdot||_U)$ *is a* canonical subassembly of X *if* $|U| \subseteq |X|$ *and* $||x||_U = ||x||_X$ *for any* $x \in |U|$.

As a convention, we identify each element of $\mathrm{Sub}_\Sigma(X)$ with the associated canonical subassembly of X and Σ-subset relation $U \subseteq_\Sigma X$ with the inclusion $|U| \subseteq |X|$.

Here we give two examples.

Example 18. 1. $\Sigma_{\mathrm{d}} = (\{\,\mathrm{true}\,\}, \{\,\mathrm{false}\,\})$: In this case, for a Σ_{d}-subset U of X and its characteristic map $\chi : X \to \Sigma_{\mathrm{d}}$, we have

$$x \in |U| \iff \chi(x) = t \iff \forall a \in ||x||_X \;\; r_\chi \cdot a = \mathrm{true},$$

where r_χ is a realizer of χ. When $A = \mathcal{K}_1$ and X is the canonical NNO N given above, $|U|$ is nothing but a decidable subset of $\mathbb{N}$. That is, $\mathrm{Sub}_{\Sigma_{\mathrm{d}}}(N)$ is equal to the set of decidable subsets of $\mathbb{N}$.

2. $\Sigma_{\mathrm{sd}} = (T_{\mathrm{sd}}, F_{\mathrm{sd}})$: Similarly to (1), we obtain

$$x \in |U| \iff \forall a \in ||x||_X \;\; r_\chi a \cdot \mathrm{i} \downarrow.$$

Thus when $A = \mathcal{K}_1$ and X is the canonical NNO, $|U|$ is the domain of a partial computable function $e_U := \lambda^* n.\,(r_\chi n) \cdot \mathrm{i}$. Hence $\mathrm{Sub}_{\Sigma_{\mathrm{sd}}}(N)$ coincides with the set of semi-decidable subsets of $\mathbb{N}$.

It is obvious that $\subseteq_\Sigma$ is a reflexive, antisymmetric relation on $\mathrm{Sub}_\Sigma(X)$ with the greatest element X and the least element $\emptyset$ (the empty assembly). But $\subseteq_\Sigma$ is not an order in general.

Definition 19. ([7,17]). *A* dominance *on A is a predominance Σ such that $\subseteq_\Sigma$ is transitive.*

Longley gave the following characterization of being a dominance in $\mathbf{Ass}(A)$.

Theorem 20 ([9, Proposition 4.2.7]). *Let $\Sigma = (T, F)$ be a predominance on A. The following are equivalent.*

1. *Σ is a dominance.*
2. *There exists a combinator $r_\mu \in A$ such that*

$$r_\mu(T \times (A \Rightarrow T)) \subseteq T, \quad r_\mu(T \times (A \Rightarrow F)) \subseteq F, \quad r_\mu(F \times A) \subseteq F,$$

where $S_0 \Rightarrow S_1$ denotes $\{\, e \in A \mid$ whenever $a \in S_0$, $ea \in S_1 \,\}$.

Remark 21. The notion of predominance has been studied in the context of *Synthetic domain theory (SDT).* It is one of the necessary pieces to construct a subcategory of "abstract domains" in a suitable category $\mathcal{C}$ (such as $\mathbf{Ass}(A)$, $\mathbf{Mod}(A)$). Various axioms for predominance have been investigated by Hyland, Phoa, Taylor and others, and being dominance is the first step towards SDT [7,13,14,18]. In fact, when a predominance t is a dominance, it induces a *lifting monad* $\perp$ on $\mathbf{Ass}(A)$. By using this monad, Longley concretely demonstrated how to construct a model of an extension of PCF. In this process, he showed that the predominance Σ_{sd} on an arbitrary non-total A is a dominance [9, Example 4.2.9 (ii)].

5 Parallel Combinators with Respect to Σ and Σ-Subsets

In this section, we will make clear the correspondence between the parallel combinators on A considered in Sect. 3 and the structure of Σ-subsets in Sect. 4. Interestingly, under a natural assumption on a predominance, our notion of Σ-and and the condition (2) of Theorem 20 correspond perfectly, thus we obtain that if A admits Σ-and then the Σ-subsets form a poset with respect to inclusion. In addition, we show that A admits Σ-or iff the Σ-subsets are closed under union. This is a generalization of the correspondence between parallel-or and union of semi-decidable sets discussed in Sect. 1.

Lemma 22. *Let $\Sigma = (T, F)$ be a predominance on A. If Σ is a dominance, then A admits Σ-and.*

Proof. By Theorem 20, A has a combinator r_μ that satisfies

$$r_\mu(T \times (A \Rightarrow T)) \subseteq T, \quad r_\mu(T \times (A \Rightarrow F)) \subseteq F, \quad r_\mu(F \times A) \subseteq F.$$

Defining $\mathrm{and}_\Sigma \coloneqq \lambda^* p.\, r_\mu \langle \mathrm{fst}\, p, \mathrm{k}(\mathrm{snd}\, p)\rangle$, we obtain a Σ-and in A.

The converse holds under an additional assumption and we obtain the first characterization theorem:

Definition 23. *Given $a, b \in A$, we write $a \cong b$ if $a \cdot x \cong b \cdot x$ for every $x \in A$. A predominance $\Sigma = (T, F)$ is a called* Rice partition *of A if T is closed under $\cong$ and $F = A \setminus T$.*

Theorem 24. *Let $\Sigma = (T, F)$ be a Rice partition of A. Then A admits Σ-and iff Σ is a dominance iff $(\mathrm{Sub}_\Sigma(X), \subseteq_\Sigma)$ is a poset for every $X \in \mathbf{Ass}(A)$.*

Proof. We only need to show the forward direction of the first equivalence. Suppose that A admits Σ-and. Letting $l \coloneqq \lambda^* xy.(x \cdot \mathrm{i} \cdot y)$, lb is always defined and $(b\mathrm{i}) \cdot y \cong (lb) \cdot y$ for any $b, y \in A$. Since (T, F) is a Rice partition, we have

$$\begin{cases} b \in (A \Rightarrow T) \implies b\mathrm{i} \in T \implies lb \in T \\ b \in (A \Rightarrow F) \implies b\mathrm{i} \in F \implies lb \in F \\ b \in A \implies lb \in T \cup F \end{cases}$$

for any $b \in A$. We thus have the following implications:

$$\begin{aligned} a \in T \text{ and } b \in (A \Rightarrow T) &\implies a \in T \text{ and } lb \in T \\ &\implies \mathrm{and}_\Sigma\langle a, lb\rangle \in T, \end{aligned}$$

$$\begin{aligned} a \in T \text{ and } b \in (A \Rightarrow F) &\implies a \in T \text{ and } lb \in F \\ &\implies \mathrm{and}_\Sigma\langle a, lb\rangle \in F, \end{aligned}$$

$$\begin{aligned} a \in F \text{ and } b \in A &\implies a \in F \text{ and } (lb \in T \text{ or } lb \in F) \\ &\implies \mathrm{and}_\Sigma\langle a, lb\rangle \in F. \end{aligned}$$

Therefore $r_\mu \coloneqq \lambda^* p.\, \mathrm{and}_\Sigma\langle \mathrm{fst}\, p, l\,(\mathrm{snd}\, p)\rangle$ satisfies condition (2) of Theorem 20.

Notice that if A is non-total, A naturally has a Rice partition, that is, $\Sigma_{\mathrm{sd}} = (T_{\mathrm{sd}}, F_{\mathrm{sd}})$. In conjunction with Theorem 12, we obtain Longley's result that Σ_{sd} is a dominance (See Remark 21).

Now suppose that Σ is a dominance. Then for every object X, $(\mathrm{Sub}_\Sigma(X), \subseteq_\Sigma)$ is a poset with the least and greatest elements. Moreover, it is automatically equipped with binary meets (intersections).

Definition 25. *Let U and V be canonical subassemblies of X. $U \cap V$ denotes the canonical subassembly of X such that $|U \cap V| := |U| \cap |V|$ and $||x||_{U\cap V} := ||x||_X$ for any $x \in |U| \cap |V|$. Similarly for $U \cup V$.*

It is well-known that the set $\mathrm{Sub}(X)$ of subobjects of X forms a lattice in $\mathbf{Ass}(A)$. On the other hand:

Lemma 26. *If Σ is a dominance, then, for every assembly X, $\mathrm{Sub}_\Sigma(X)$ is closed under intersection $\cap$ and $(\mathrm{Sub}_\Sigma(X), \subseteq_\Sigma, \cap)$ forms a meet-semilattice.*

Proof. Let U, V be canonical subassemblies of X and $m : U \rightarrowtail X$, $n : V \rightarrowtail X$ the inclusions, respectively. Then $U \cap V$ can be obtained as in the following pullback diagram:

$$\begin{array}{ccc} U \cap V & \rightarrowtail & U \\ {\scriptstyle n^{-1}(m)} \downarrow & \lrcorner & \downarrow {\scriptstyle m} \\ V & \underset{n}{\rightarrowtail} & X. \end{array}$$

If both U and V are Σ-subsets of X, then $U \cap V$ is a Σ-subset of V since $\mathrm{Sub}_\Sigma(X)$ is closed under pullback. Hence $U \cap V$ is a Σ-subset of X. Recalling the structure of the subobject lattice $\mathrm{Sub}(X)$, the binary meet appears as a pullback. Thus $\cap$ behaves as a meet with respect to $\subseteq_\Sigma$.

This means that $(\mathrm{Sub}_\Sigma(X), \subseteq_\Sigma)$ is a sub-meet-semilattice of $\mathrm{Sub}(X)$ when Σ is a dominance.

Let us finally discuss the effect of having a Σ-or combinator in A. As we have already seen in Sect. 3, a parallel-or in $\mathcal{K}_1$ has the ability to take the join of two semi-decidable subsets. This fact can be generalized and refined as follows. Notice that the assumption of Rice partition implies that U is a Σ-subset of X iff there exists a characteristic map $\chi_U : X \to \Sigma$ with a realizer r_{χ_U} satisfying

$$x \in |U| \iff \chi_U(x) = t \iff r_{\chi_U}(||x||_X) \subseteq T.$$

The second equivalence is ensured by $T \cap F = \emptyset$. We are now ready to prove the second characterization theorem.

Theorem 27. *Let $\Sigma = (T, F)$ be a predominance with $T \cap F = \emptyset$. Then A admits Σ-or if and only if $\mathrm{Sub}_\Sigma(X)$ is closed under union $\cup$ for every assembly X.*

Proof. We first show the forward direction. Let U, V be Σ-subsets of X, χ_U, χ_V their characteristic maps and r_{χ_U}, r_{χ_V} their realizers, respectively. Then the canonical subassembly $U \cup V$ naturally induces a function $\chi_{U \cup V} : |X| \to |\Sigma|$ such that

$$x \in |U| \cup |V| \iff \chi_{U \cup V}(x) = t.$$

Since A admits Σ-or, we can define $r_{\chi_{U \cup V}}$ as $\lambda^* x. \operatorname{or}_\Sigma \langle r_{\chi_U}\, x, r_{\chi_V}\, x\rangle$ in A. Then $r_{\chi_{U \cup V}}$ behaves as follows:

$$\begin{aligned} r_{\chi_{U \cup V}}(||x||_X) \subseteq T &\iff r_{\chi_U}(||x||_X) \subseteq T \text{ or } r_{\chi_V}(||x||_X) \subseteq T \\ &\iff x \in |U| \text{ or } x \in |V| \\ &\iff x \in |U| \cup |V|. \end{aligned}$$

Thus $r_{\chi_{U \cup V}}$ is a realizer of $\chi_{U \cup V}$ and $U \cup V$ is a Σ-subset of X.

To show the backward direction, let us note the following two facts:

– Given two assemblies X and Y, the product $X \times Y$ in $\mathbf{Ass}(A)$ can be concretely described as

$$|X \times Y| := |X| \times |Y|, \qquad ||(x, y)||_{X \times Y} := ||x||_X \times ||y||_Y.$$

– Every subset S of A induces an assembly $\overline{S}$ such that

$$|\overline{S}| := S, \qquad ||a||_{\overline{S}} := \{\, a \,\}.$$

For example, there is an assembly $\overline{T \cup F} \times \overline{T \cup F}$ that corresponds to the set $\{\, \langle a, b\rangle \in A \mid a, b \in T \cup F \,\}$.

Let $H := T \cup F$. Then we have $\overline{T} \subseteq_\Sigma \overline{H}$ because there is a characteristic map $\chi_T : \overline{H} \to \Sigma$ such that $\chi_T(a) = t$ iff $a \in T$, and it is realized by i. Similarly, one can easily verify the following relations:

$$\overline{T} \times \overline{H} \subseteq_\Sigma \overline{H} \times \overline{H}, \qquad \overline{H} \times \overline{T} \subseteq_\Sigma \overline{H} \times \overline{H}.$$

Lastly, since $\operatorname{Sub}_\Sigma(\overline{H} \times \overline{H})$ is closed under union, we obtain

$$\overline{T} \times \overline{H} \cup \overline{H} \times \overline{T} \subseteq_\Sigma \overline{H} \times \overline{H}.$$

This induces a characteristic map $\chi : \overline{H} \times \overline{H} \to \Sigma$ and a realizer r_χ such that for any $a, b \in T \cup F$,

$$a \in T \text{ or } b \in T \iff \chi((a, b)) = t \iff r_\chi(||(a, b)||_{\overline{H} \times \overline{H}}) \subseteq T.$$

Note that $||(a, b)||_{\overline{H} \times \overline{H}} = \{\, \langle a, b\rangle \,\}$. Hence r_χ satisfies the following property: for any $a, b \in T \cup F$,

– $r_\chi \cdot \langle a, b\rangle$ belongs to T if $a \in T$ or $b \in T$.
– Otherwise, $r_\chi \cdot \langle a, b\rangle$ belongs to F.

Thus r_χ is nothing but a Σ-or combinator.

By restricting to the case of Rice partition, we can summarize the role of Σ-and and Σ-or as follows.

Theorem 28. *Suppose that $\Sigma = (T, F)$ is a Rice partition of A. Then A admits both Σ-and and Σ-or if and only if $(\mathrm{Sub}_\Sigma(X), \subseteq_\Sigma, \cap, \cup)$ forms a lattice for every assembly X.*

Proof. The backward direction is obvious by Theorem 24 and Theorem 27.

For the forward direction, it remains to check that $\cup$ behaves as a join with respect to $\subseteq_\Sigma$. It is sufficient to verify the following claims: if $U, V \subseteq_\Sigma X$ then

$$U \subseteq_\Sigma U \cup V, \qquad U \cup V \subseteq_\Sigma X.$$

The latter is just closure under union, that is already established by Theorem 27. For the former, let $\chi_U : X \to \Sigma$ be the characteristic map of $U \rightarrowtail X$, which exists by $U \subseteq_\Sigma X$. Then $\chi_U|_{|U \cup V|} : U \cup V \to \Sigma$ is the characteristic map of $U \rightarrowtail U \cup V$, which is realized by any realizer of χ_U.

By recalling that a non-total PCA always has Rice partition Σ_{sd} that is a dominance, we finally conclude:

Corollary 29. *Let A be a non-total PCA. Then A admits parallel-or in A if and only if $(\mathrm{Sub}_{\Sigma_{\mathrm{sd}}}(X), \subseteq_{\Sigma_{\mathrm{sd}}}, \cap, \cup)$ forms a lattice for every object X in $\mathbf{Ass}(A)$.*

As we have stated in Proposition 13, Λ_v^0 is an example of a non-total PCA that does not admit parallel-or. Therefore, one cannot always take a union of Σ_{sd}-subsets in $\mathbf{Ass}(\Lambda_v^0)$ unlike in $\mathbf{Ass}(\mathcal{K}_1)$.

6 Future Work

In this paper we have focused on $\mathbf{Ass}(A)$ among other realizability models. In $\mathbf{Ass}(A)$, (non-trivial) predominances Σ are exactly those that arise from pairs (T, F) of nonempty subsets of A. This simplicity has led to a handy description of Σ-subsets as canonical subassemblies, and consequently a clear correspondence between Σ-and/or combinators and the structure of $\mathrm{Sub}_\Sigma(X)$. All the results in this paper hold for the category $\mathbf{Mod}(A)$ of modest sets over A too, that is a full subcategory of $\mathbf{Ass}(A)$.

On the other hand, the situation is entirely different if we consider the realizability topos $\mathbf{RT}(A)$, that is the exact completion of $\mathbf{Ass}(A)$. The predominances in $\mathbf{RT}(A)$ include the subobject clasifier as well as those associated with a *local operator* j (a.k.a. *Lawvere-Tierney topology*) such as the predominance classifying j-dense subobjects and the one classifying j-closed subobjects. Studying parallel operations in relation to these predominances could be interesting, since local operators in $\mathbf{RT}(A)$ correspond to subtoposes of $\mathbf{RT}(A)$ on one hand, and can be seen as "generalized Turing degrees" on the other [4,6,12]. It is left to future work.

Acknowledgment. I am grateful to my supervisor, Kazushige Terui, and Naohiko Hoshino for many useful discussions. This work is supported by JST Grant Number JPMJFS2123.

References

1. Abramsky, S., Lenisa, M.: Linear realizability and full completeness for typed lambda-calculi. Ann. Pure Appl. Logic **134**(2–3), 122–168 (2005)
2. Barendregt, H.P.: The Lambda Calculus: Its Syntax and Semantics, vol. 103. Studies in Logic and the Foundations of Mathematics. Elsevier, Amsterdam (1984)
3. Carboni, A., Freyd, P.J., Scedrov, A.: A categorical approach to realizability and polymorphic types. In: Main, M., Melton, A., Mislove, M., Schmidt, D. (eds.) MFPS 1987. LNCS, vol. 298, pp. 23–42. Springer, Heidelberg (1988). https://doi.org/10.1007/3-540-19020-1_2
4. Faber, E., van Oosten, J.: More on geometric morphisms between realizability toposes. Theory Appl. Categories **29**, 874–895 (2014)
5. Guerrieri, G.: Head reduction and normalization in a call-by-value lambda-calculus. In Chiba, Y., Escobar, S., Nishida, N., Sabel, D., Schmidt-Schauß, M. (eds.) 2nd International Workshop on Rewriting Techniques for Program Transformations and Evaluation (WPTE 2015), vol. 46 of OpenAccess Series in Informatics (OASIcs), pp. 3–17, Dagstuhl, Germany (2015). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik
6. Hyland, J.M.E.: The effective topos. In: The L E.J. Brouwer Centenary Symposium, vol. 110 of Stud. Logic Foundations Math. North-Holland, pp. 165–216 (1982)
7. Hyland, J.M.E.: First steps in synthetic domain theory. In *Category Theory*, LNM, vol. 1488, pp. 131–156 (1991)
8. Kleene, S.C.: On the interpretation of intuitionistic number theory. J. Symb. Log. **10**, 109–124 (1945)
9. Longley, J.R.: Realizability Toposes and Language Semantics. PhD thesis, University of Edinburgh (1994)
10. Longley, J.R., Simpson, A.K.: A uniform approach to domain theory in realizability models. Math. Struct. Comput. Sci. **7**(5), 469–505 (1997)
11. Mulry, P.S.: Generalized Banach-Mazur functionals in the topos of recursive sets. J. Pure Appl. Algebra **26**(1), 71–83 (1982)
12. Phoa, W.: Relative computability in the effective topos. Math. Proc. Cambridge Philos. Soc. **106**, 419–422 (1989)
13. Phoa, W.: Domain Theory in Realizability Toposes. PhD thesis, University of Cambridge (1990)
14. Phoa, W.: From term models to domains. Inf. Comput. **109**(1), 211–255 (1994)
15. Plotkin, G.D.: Call-by-name, call-by-value and the λ-calculus. Theoret. Comput. Sci. **1**(2), 125–159 (1975)
16. Plotkin, G.D.: LCF considered as a programming language. Theoret. Comput. Sci. **5**(3), 223–255 (1977)
17. Rosolini, G.: Continuity and Effectiveness in Topoi. PhD thesis, University of Oxford (1986)
18. Taylor, P.: The fixed point property in synthetic domain theory. In: Proceedings Sixth Annual IEEE Symposium on Logic in Computer Science, pp. 152–160 (1991)
19. van Oosten, J.: Realizability: an introduction to its categorical side, vol. 152 of Studies in Logic and the Foundations of Mathematics. Elsevier (2008)

Bisimulations Between Verbrugge Models and Veltman Models

Tin Perkov(✉)

Faculty of Teacher Education, University of Zagreb, Zagreb, Croatia
tin.perkov@ufzg.hr

Abstract. Veltman semantics is the basic Kripke-like semantics for interpretability logic. Verbrugge semantics is a generalization of Veltman semantics. An appropriate notion of bisimulation between a Verbrugge model and a Veltman model is developed in this paper. We show that a given Verbrugge model can be transformed into a bisimilar Veltman model.

Keywords: Modal logic · Interpretability logic · Bisimulations

1 Introduction

Interpretability logic is an extension of provability logic, which formalizes the notion of relative interpretability between arithmetical first-order theories. Intuitively, we say that such a theory T interprets another theory T' if there is a translation from the language of T' to the language of T such that all translations of axioms of T' are provable in T. Intepretability logic is a modal logic which, together with usual unary modality $\Box$, whose intended interpretation in this context is provability, has another modality $\rhd$, which is binary. Formulas of the form $A \rhd B$ are intended to mean that some base theory T extended with the formula A interprets the theory obtained by extending the same base theory T with the formula B. In this paper we will only deal with modal semantics of interpretability logic in general, so we omit overviewing axiomatic systems of interpretability logic (cf. e.g. [9] for this, and also for more details on arithmetical aspects).

The basic semantics of interpretability logic is defined on Veltman models, Kripke-like structures built over standard Kripke models of provability logic, which means that the accessibility relation is transitive and converse well founded, by adding a family of relations S_w between worlds R-accessible from w, for each world w in the model, satisfying certain properties, e.g. reflexivity and transitivity (a precise definition is given in the next section). Verbrugge semantics ([8], cf. also [1]) is a generalization in which relations S_w are no longer between worlds, but between worlds and sets of worlds. This semantics proved to be useful in showing some independence results which could not be proved using Veltman semantics ([11]), in proving some completeness results in cases of

H. H. Hansen et al. (Eds.): WoLLIC 2023, LNCS 13923, pp. 305–317, 2023.
https://doi.org/10.1007/978-3-031-39784-4_19

incompleteness w.r.t. Veltman semantics ([2,5]), and it also enabled using filtration technique, which could not be used on Veltman models, in order to prove finite model property and consequently some decidability results ([4,6]).

This paper addresses the following question: for a given Verbrugge model, can we obtain a Veltman model which would be closely related with the initial Verbrugge model, preferably by some appropriately defined notion of bisimulation, or at least by modal equivalence. This question is natural, since Veltman models are still appealing due to their relative simplicity, so we would like to keep them as the basic semantics and it would be nice to have a bridge by which we could possibly transfer some results from Verbrugge semantics back to Veltman semantics, or better understand why some of them cannot be transferred.

It is not surprising that this question was being addressed from the very beginning of work on Verbrugge semantics: already in [8] (cf. also [7]) a transformation of a given Verbrugge model to a modally equivalent Veltman model was provided, but using different notion of Verbrugge model than the one used in the present paper (different notions of Verbrugge models come from various possible ways to define so-called quasi-transitivity, a property of Verbrugge models which corresponds to transitivity of relations S_w in Veltman models). Similar attempt in [10] resulted in Veltman model bisimilar to a given Verbrugge model in a certain sense, but under additional conditions of image-finiteness and inverse image-finiteness. Also, this was an indirect result: bisimilarity was observed between the same kind of structures, after a transformation. In the present paper we will work with a directly defined notion of bisimulation between different kinds of structures, namely between a Verbrugge model and a Veltman model, and we will be able to avoid additional constraints on these structures.

In Sect. 2 we recall basic definitions and we define the notion of bisimulation between a Verbrugge model and a Veltman model. In Sect. 3 we provide arguments in favour of thus defined notion, e.g. we prove an analogue of Hennessy-Milner theorem. In Sect. 4 we obtain a Veltman model bisimilar to a given Verbrugge model. In Sect. 5 we conclude with some remarks on future work.

2 Bisimulation Between Verbrugge and Veltman Model

The alphabet of interpretability logic consists of countably many propositional variables and symbols $\bot$, $\rightarrow$ and $\rhd$. Formulas are given by

$$\varphi ::= p \,|\, \bot \,|\, \varphi_1 \rightarrow \varphi_2 \,|\, \varphi_1 \rhd \varphi_2,$$

where p ranges over the set of propositional variables. We use usual abbreviations $\top := \neg\bot$, $\neg\varphi := \varphi \rightarrow \bot$, $\varphi_1 \vee \varphi_2 := \neg\varphi_1 \rightarrow \varphi_2$, $\varphi_1 \wedge \varphi_2 := \neg(\neg\varphi_1 \vee \neg\varphi_2)$, $\varphi_1 \leftrightarrow \varphi_2 := (\varphi_1 \rightarrow \varphi_2) \wedge (\varphi_2 \rightarrow \varphi_1)$, $\Box\varphi := \neg\varphi \rhd \bot$, $\Diamond\varphi := \neg\Box\neg\varphi$.

A *Veltman model* is a tuple $(W, R, \{S_w : w \in W\}, \Vdash)$ such that:

- $W \neq \emptyset$ is a set called the *domain*, whose elements are called *worlds*
- $R \subseteq W \times W$ is a transitive and converse well-founded relation called the *accessibility relation*

- $S_w \subseteq R[w] \times R[w]$, where $R[w] = \{u \in W : wRu\}$, is a reflexive and transitive relation such that $wRuRv$ always implies uS_wv, for each $w \in W$
- $\Vdash$ is a relation between worlds and formulas such that for all $w \in W$ we have $w \nVdash \bot$, $w \Vdash \varphi_1 \rightarrow \varphi_2$ if and only if $w \nVdash \varphi_1$ or $w \Vdash \varphi_2$, and $w \Vdash \varphi_1 \rhd \varphi_2$ if and only of for all $u \in W$ such that wRu and $u \Vdash \varphi_1$ there is $v \in W$ such that uS_wv and $v \Vdash \varphi_2$[1]

A *Verbrugge model* is a tuple $(W, R, \{S_w : w \in W\}, \Vdash)$, where W and R are as in the definition of Veltman models, while $S_w \subseteq R[w] \times (\mathcal{P}(R[w]) \setminus \{\emptyset\})$ such that:

- if wRu, then $uS_w\{u\}$ (*quasi-reflexivity*)
- if uS_wV and vS_wZ_v for all $v \in V$, then $uS_w \bigcup_{v \in V} Z_v$ (*quasi-transitivity*)
- if $wRuRv$, then $uS_w\{v\}$
- if uS_wV and $V \subseteq Z \subseteq R[w]$, then uS_wZ (*monotonicity*),

while $\Vdash$ is defined similarly as in the definition of Veltman model, except the following: $w \Vdash \varphi_1 \rhd \varphi_2$ if and only if for all u such that wRu and $u \Vdash \varphi_1$ there is V such that uS_wV and for all $v \in V$ we have $v \Vdash \varphi_2$ (we write $V \Vdash \varphi_2$).

When we need to emphasize that $w \Vdash \varphi$ is observed in the context of a structure $\mathfrak{M}$, we will write $\mathfrak{M}, w \Vdash \varphi$.

As aforementioned, there are other variants of Verbrugge models in the literature, which differ from the above one only in the definition of quasi-transitivity and in some cases in omitting monotonicity. In this paper we work only with the above definition, since it is predominant in the literature (cf. a recent overview [1], which includes a discussion on other possibilities).

Bisimulation is the basic equivalence between modal models. It has three defining conditions: atomic equivalence between related worlds (at), the condition describing how the first model is simulated in the second one (forth), and the condition describing how the second model is simulated in the first one (back). When we work with the same kind of structures, (forth) and (back) are mutually symmetric. But now we will define the notion of bisimulation between different kinds of structures, which will therefore lack this symmetry. In fact, the direction from Verbrugge model to Veltman model (forth) will be much more complex than the opposite one.

Definition 1. *Let $\mathfrak{M} = (W, R, \{S_w : w \in W\}, \Vdash)$ be a Verbrugge model and let $\mathfrak{M}' = (W', R', \{S'_{w'} : w' \in W'\}, \Vdash)$ be a Veltman model. A* **bisimulation** *between $\mathfrak{M}$ and $\mathfrak{M}'$ is any non-empty relation $Z \subseteq W \times W'$ such that:*

(at) *$\mathfrak{M}, w \Vdash p$ if and only if $\mathfrak{M}', w' \Vdash p$ for all $w \in W, w' \in W'$ such that wZw', for each propositional variable p*

(forth) *if wZw' and wRu, then there exists a non-empty $U' \subseteq W'$ such that $w'R'u'$ and uZu' for all $u' \in U'$ and for any $F : U' \rightarrow W'$ such that $u'S'_{w'}F(u')$ for all $u' \in U'$, there is V such that uS_wV and for all $v \in V$ there is $u' \in U'$ such that $vZF(u')$*

[1] Equivalently, we can define a Veltman model to be $(W, R, \{S_w : w \in W\}, V)$, where V maps each propositional variable to a subset of W, and then define satisfaction relation $\Vdash$ recursively, but this is non-essential and just a matter of style.

(back) *if* wZw' *and* $w'R'u'$, *then there exists* $u \in W$ *such that* wRu, uZu' *and for each* $V \subseteq W$ *such that* uS_wV *there are* $v \in V$ *and* $v' \in W'$ *such that* $u'S'_{w'}v'$ *and* vZv'.[2]

Consider an example of thus defined bisimulation.

Example 1. Consider a Verbrugge model $\mathfrak{M}$ such that:

- $W = \{0, 1, 2, 3\}$, $R = \{(0, 1), (0, 2), (0, 3)\}$, $1S_0\{2, 3\}$
- $1 \Vdash p$, $2 \Vdash q$, $3 \Vdash r$

Now, consider a Veltman model $\mathfrak{M}'$ as follows:

- $W' = \{0', 1', 1'', 2', 3'\}$, $R' = \{(0', 1'), (0', 1''), (0', 2'), (0', 3')\}$, $1'S'_{0'}2'$, $1''S'_{0'}3'$
- $1' \Vdash p$, $1'' \Vdash p$, $2' \Vdash q$, $3' \Vdash r$

Note that we omitted some pairs in S_0 and $S'_{0'}$, namely those enforced by (quasi)-reflexivity and monotonicity.

It is easy to verify that $Z = \{(0, 0'), (1, 1'), (1, 1''), (2, 2'), (3, 3')\}$ is a bisimulation.

The following proposition shows that the necessary requirement on any notion of bisimulation is satisfied: bisimilar worlds are modally equivalent.

Proposition 1. *Let* $\mathfrak{M} = (W, R, \{S_w : w \in W\}, \Vdash)$ *be a Verbrugge model,* $\mathfrak{M}' = (W', R', \{S'_{w'} : w' \in W'\}, \Vdash)$ *a Veltman model and* $Z \subseteq W \times W'$ *a bisimulation between* $\mathfrak{M}$ *and* $\mathfrak{M}'$. *Then for all* w, w' *such that* wZw' *we have that* w *and* w' *are modally equivalent, i.e.* $\mathfrak{M}, w \Vdash \varphi$ *if and only if* $\mathfrak{M}', w' \Vdash \varphi$, *for each formula* φ.

Proof. The claim is proved by induction on the complexity of a formula. We only present the inductive step in the case of a formula of the form $\varphi_1 \rhd \varphi_2$.

Assume $\mathfrak{M}, w \Vdash \varphi_1 \rhd \varphi_2$ and wZw'. We need to prove $\mathfrak{M}', w' \Vdash \varphi_1 \rhd \varphi_2$. Let $u' \in W'$ such that $w'R'u'$ and $u' \Vdash \varphi_1$. Then (back) implies there is u such that wRu and uZu'. By the induction hypothesis $u \Vdash \varphi_1$. Since $w \Vdash \varphi_1 \rhd \varphi_2$, there is V such that uS_wV and $V \Vdash \varphi_2$. But (back) also implies that for any

[2] As pointed out by a reviewer, one could alternatively generalize these conditions to develop an analogous notion of bisimulation between Verbrugge models, and then establish a connection between a Verbrugge and a Veltman model by composing a bisimulation between Verbrugge models and a simple transformation from a Veltman to a Verbrugge model described at the beginning of Sect. 4. The present approach has an advantage that the already complex (forth) condition has an additional quantifier in case of two Verbrugge models, and (back) condition would be symmetric to (forth) when observed between two Verbrugge models, while in the present paper it is much simpler. Nevertheless, an analogous notion of bisimulation between Verbrugge models is of independent interest and is thoroughly studied in a near future paper [3].

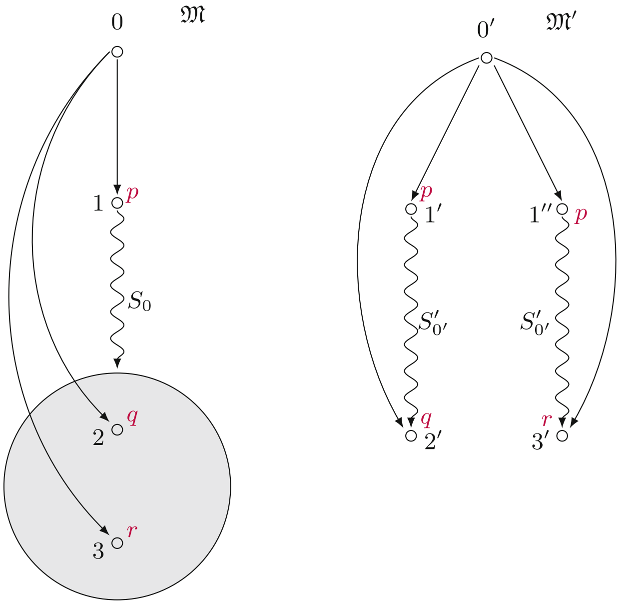


Fig. 1. Illustration of Example 1

S_w-successor of u, thus also for V, there are $v \in V$ and $v' \in W'$ such that vZv' and $u'S_{w'}v'$. Again by the induction hypothesis $v' \Vdash \varphi_2$, as desired.

Conversely, assume $\mathfrak{M}', w' \Vdash \varphi_1 \rhd \varphi_2$ and wZw' and prove $\mathfrak{M}, w \Vdash \varphi_1 \rhd \varphi_2$. Let $u \in W$ such that wRu and $u \Vdash \varphi_1$. Then by (forth) there is $U' \neq \emptyset$ such that $w'R'u'$ and uZu', and thus by the induction hypothesis $u' \Vdash \varphi_1$, for all $u' \in U'$, such that for any choice of one $S'_{w'}$-successor for each world in U', there is V such that uS_wV and each world in V is bisimilar to some of those $S'_{w'}$-successors. Now for all $u' \in U'$, since $w' \Vdash \varphi_1 \rhd \varphi_2$ and $u' \Vdash \varphi_1$, there is v' such that $u'S_{w'}v'$ and $v' \Vdash \varphi_2$. For such a world v', put $F(u') = v'$. Since the above holds for any choice F of one $S'_{w'}$-successor for each $u' \in U'$, it holds in particular for the choice F. Thus there is V such that uS_wV and each $v \in V$ is bisimilar to some $F(u')$, so by the induction hypothesis $v \Vdash \varphi_2$ for all $v \in V$, as desired. ■

Example 2. Since Z defined in Example 1 is a bisimulation, the previous proposition implies that 0 and $0'$ are modally equivalent (as are all pairs in Z).

3 Hennessy-Milner Theorem

As aforementioned, the first requirement of any notion of bisimulation is that it implies modal equivalence. That requirement shows that the definition is not too weak, i.e. structural relation between two models is strong enough to ensure modal formulas cannot distinguish them. But on the other hand, some proposed relation between models can be too strong. For example, isomorphism of course implies modal equivalence, but it is obviously unnecessarily strong, i.e. much weaker structural relations can imply modal equivalence. In other words, one would like to have a converse of the previous proposition, to see that the defined notion of bisimulation is just enough strong. Unfortunately, it is well known that the direct converse never holds (a counterexample can easily be constructed from some of the known counterexamples for basic modal logic). But, there are some approximations, notably Hennessy-Milner-like theorems, which say that the converse holds in case of image-finite models. If Hennessy-Milner analogue holds for some proposed notion, then it is a good sign that the notion is just about as strong as bisimulation should be.

Theorem 1. *Let* $\mathfrak{M} = (W, R, \{S_w : w \in W\}, \Vdash)$ *be a Verbrugge model and let* $\mathfrak{M}' = (W', R', \{S'_{w'} : w' \in W'\}, \Vdash)$ *be a Veltman model such that relations* R *and* R' *are image-finite, i.e. for all* $w \in W, w' \in W'$ *we have that* $R[w]$ *and* $R'[w']$ *are finite.*

Then any $w \in W$ *and* $w' \in W'$ *are modally equivalent if and only if there is a bisimulation* Z *between* $\mathfrak{M}$ *and* $\mathfrak{M}'$ *such that* wZw'*.*

Proof. Let $Z \subseteq W \times W'$ be the modal equivalence between worlds of $\mathfrak{M}$ and $\mathfrak{M}'$, i.e. wZw' if and only if w and w' satisfy exactly the same formulas. We will prove that Z is a bisimulation between $\mathfrak{M}$ and $\mathfrak{M}'$. Together with the previous proposition, this clearly implies the claim.

Obviously (at) holds. Assume (back) does not hold, i.e. there are w, w', u' such that wZw' and $w'R'u'$ and for all u such that wRu and uZu' there is V such that uS_wV and for all $v \in V$ and v' such that $u'S_{w'}v'$ we have that v and v' are not modally equivalent.

For any x such that wRx which is not modally equivalent to u' there is a formula φ_x such that $u' \Vdash \varphi_x$ and $x \nVdash \varphi_x$. Since there are only finitely many such worlds x, there is a finite conjunction φ of one such formula for each x, so $u' \Vdash \varphi$. Observe now that for any u such that wRu we have uZu' if and only if $u \Vdash \varphi$.

Now, let $u \in W$ such that wRu and uZu'. By the assumption, there is $V_u \subseteq R[w]$ such that uS_wV_u and no $v \in V_u$ is modally equivalent to any v' such that $u'S_{w'}v'$. For each $y \in R[w]$ which is not modally equivalent to any v' such that $u'S_{w'}v'$, and for each such v', there is a formula $\psi_{y,v'}$ which is satisfied at y, but not at v'. Since S_w-successors of u' are R-successors of w, there are only finitely many of them, so there is a finite conjunction ψ_y of one such formula for each v', and we have $y \Vdash \psi_y$ and $v' \nVdash \psi_y$. But now we clearly have $V_u \Vdash \psi$, where ψ is the disjunction of all ψ_y, where $y \in R[w]$ is not modally equivalent to u'. Hence, $w \Vdash \varphi \rhd \psi$. Since wZw', we have $w' \Vdash \varphi \rhd \psi$. Since $u' \Vdash \varphi$, there is v' such that $u'S_{w'}v'$ and $v' \Vdash \psi$, so $v' \Vdash \psi_y$ for some y and thus $v' \Vdash \psi_{y,v'}$, which is a contradiction.

It remains to prove (forth). Assume it does not hold, i.e. there are w, w', u such that wZw', wRu and for any $U' \neq \emptyset$ such that $w'R'u'$ and uZu' for all $u' \in U'$, there is a choice $F : U' \to W'$ of one $S'_{w'}$-successor for each $u' \in U'$ such that for all V such that uS_wV there is $v \in V$ not equivalent to $F(u')$ for any $u' \in U'$.

In particular, this holds if we take U' to be the set of all $u' \in R'[w']$ such that uZu'. Further in this proof U' will denote that set.

Similarly as in the proof of (back), we can show that there is a formula φ such that $u \Vdash \varphi$ and for all $u' \in R'[w']$ we have uZu' if and only if $u' \Vdash \varphi$. Furthermore, for any $u' \in U'$ and any V such that uS_wV, there is a formula $\psi_{u',V}$ which is satisfied at $F(u')$, but not at some $v \in V$. For each $u' \in U'$, let $\psi_{u'}$ be the conjunction of all $\psi_{u',V}$, ranging over all V such that uS_wV. Again, this is clearly a finite conjunction, and we have $F(u') \Vdash \psi_{u'}$.

Let ψ be the disjunction of all $\psi_{u'}$, where $u' \in U'$. Clearly $w' \Vdash \varphi \rhd \psi$. Now wZw' implies $w \Vdash \varphi \rhd \psi$. Since $u \Vdash \varphi$, there is V such that uS_wV and $V \Vdash \psi$. Hence, $V \Vdash \psi_{u'}$ for some $u' \in U'$. But then $V \Vdash \psi_{u',V}$, which is a contradiction, since there is $v \in V$ not satisfying $\psi_{u',V}$. ■

The reader may be surprised to see that, in the definition of bisimulations between Verbrugge models and Veltman models, the condition (forth) demands the existence of a set of worlds U' instead of just the existence of at least one world as usual. But without this, the notion of bisimulation would not be useful. To see this, note that a seemingly more natural (forth) would demand: if wZw' and wRu, then there is u' such that $w'R'u'$, uZu' and for all v' such that $u'S'_{w'}v'$ there is V such that uS_wV and vZv' for all $v \in V$. It is easily checked that this does imply modal equivalence, but nevertheless it is too restrictive, since it has a consequence that all worlds in V are mutually modally equivalent, which practically collapses Verbrugge semantics to Veltman semantics.

The following example illustrates why we need (forth) to be as complex as it is, and also provides an idea how to proceed with the main goal of the paper: find a bisimilar Veltman model for a given Verbrugge model.

Example 3. To illustrate usefulness of seemingly too complicated (forth), consider again the bisimulation Z defined in Example 1. Let us consider just one part of the verification that Z is a bisimulation, namely (forth) for $0R1$ and $0Z0'$. Then the good choice for U' is $\{1', 1''\}$. Then e.g. for F defined by $F(1') = 2'$, $F(1'') = 3'$, we have $1S_0\{2, 3\}$, $2Z2'$, $3Z3'$.

With the aforementioned more restrictive definition of bisimulation, we would not have a bisimulation in this example, thus we can use it as a counterexample for Hennessy-Milner analogue in that case. Namely, in the situation illustrated above, for $0R1$ and $0Z0'$ the restrictive (forth) would force us to choose just one R'-successor of $0'$ bisimilar to 1. Then for both possible choices we would not be able to satisfy the remaining requirement of (forth), e.g. for $1'$ and its $S'_{0'}$-successor $2'$, there is no V such that $1S_0V$ and all elements of V are bisimilar to $2'$.

4 Obtaining a Veltman Model Bisimilar to a Given Verbrugge Model

It is straightforward to obtain a Verbrugge model from a given Veltman model $\mathfrak{M} = (W, R, \{S_w : w \in W\}, \Vdash)$: we use the same W and R, and define uS'_wV if and only if uS_wv for some $v \in V$. It is very easy to see that $Z = \{(w, w) : w \in W\}$ is a bisimulation between thus obtained Verbrugge model and $\mathfrak{M}$.

Although it is very simple, our running example already illustrates that the opposite direction is much more involved. The basic idea is that each world from a given Verbrugge model will have multiple copies in the associated Veltman model, to make it possible for S_w-connections with sets of worlds to be simulated by connections with worlds which are representatives of these sets.

So, let $\mathfrak{M} = (W, R, \{S_w : w \in W\}, \Vdash)$ be a Verbrugge model. We will define a Veltman model associated with $\mathfrak{M}$, which we will denote by $Vel(\mathfrak{M}) = (W', R', \{S'_{w'} : w' \in W'\}, \Vdash)$.

First we introduce some notation and terminology.

4.1 $\overline{S_w}$-Paths

When we consider subsets of W, it will often be essential that they are represented as certain unions. We will keep track of such information in the following way: instead of some $X \subseteq W$, we will consider a family $\overline{X} = \{X_i : i \in I\}$ such that $X = \bigcup_{i \in I} X_i$. We will use the notation $\overline{X}$ for the sake of simplicity, although, of course, $\overline{X}$ is not uniquely determined by X. It will, however, always be clear form the context what we mean by $\overline{X}$.

For non-empty $U, V \subseteq W$ such that $U = \bigcup_{i \in I} U_i$ and $V = \bigcup_{u \in U} V_u$, we write $\overline{U}\,\overline{S_w}\,\overline{V}$ if uS_wV_u for all $u \in U$. Observe that the quasi-transitivity can now be expressed in the following way: if uS_wV and $\overline{V}\,\overline{S_w}\,\overline{Z}$, then uS_wZ.

Observe also that uS_wV is equivalent to $\overline{\{u\}}\,\overline{S_w}\,\overline{V}$, where $\overline{\{u\}} = \{\{u\}\}$ and $\overline{V} = \{V\}$ are singleton families, which is of course more complicated notation, but useful for considering some sequences as $\overline{S_w}$-paths, as follows.

Definition 2. *Consider a finite path of the form* $\overline{\{u\}}\,\overline{S_w}\,\overline{V_1}\,\overline{S_w}\,\overline{V_2 S_w} \ldots \overline{S_w}\,\overline{V_k}$. *We call the sequence* $\overline{\{u\}}, \overline{V_1}, \overline{V_2}, \ldots, \overline{V_k}$ *an* $\overline{S_w}$**-path starting with** u*, or simply an* $\overline{S_w}$**-path** *if it is clear from the context what it starts with.*

In what follows, to avoid repeating all properties each time we mention such paths, and since we will not consider any other kind of paths, when we shortly say that something is an $\overline{S_w}$- path, we will always mean that it is a finite path starting from a singleton family which has a singleton set as its only element. If wRu, we will consider just $\overline{\{u\}}$ to be an $\overline{S_w}$-path (of length zero).

If $\overline{\{u\}}, \overline{V_1}, \overline{V_2}, \ldots, \overline{V_k}, \overline{V_{k+1}}, \ldots, \overline{V_{k+l}}$ is an $\overline{S_w}$-path, then for a given $v \in V_k$ we denote by $\overline{\{v\}}, \overline{V'_1}, \overline{V'_2}, \ldots, \overline{V'_l}$ the $\overline{S_w}$-path such that $\overline{V'_i} \subseteq \overline{V_{k+i}}$, $i = 1, 2, \ldots, l$, which is uniquely determined in the way that V'_1 is the element of the family $\overline{V_{k+1}}$ which is determined by v, i.e. obtained as an S_w-successor of v, and $\overline{V'_{i+1}}$ is the subfamily of $\overline{V_{k+i+1}}$ consisting exactly of elements determined by elements of V'_i, for $i = 1, 2, \ldots, l-1$. We say that thus obtained $\overline{S_w}$-path is *induced* by v.

4.2 Well Defined Choice of Representatives

For an $\overline{S_w}$-path $\{u\}, \overline{V_1}, \overline{V_2}, \ldots, \overline{V_k}$, we say that $(u, v_1, v_2, \ldots, v_k)$ is a *well defined sequence of representatives* if $v_i \in V_i$ for all $i = 1, \ldots, k$, and v_{i+1} is a world from the element of the family $\overline{V_{i+1}}$ which is determined by v_i, i.e. obtained as an S_w-successor of v_i, for $i = 1, \dots, k-1$.

Definition 3. *Let $w \in W$ and $x \in W$ such that xRw, and let f_x be a function which maps each $\overline{S_x}$-path starting with w to a well defined sequence of representatives (when needed, if $f_x(\overline{\{w\}}, \overline{V_1}, \dots, \overline{V_n}) = (w, v_1, \dots, v_n)$, we write $f_x(\overline{\{w\}}, \overline{V_1}, \dots, \overline{V_n})_i = v_i$, $i = 1, \dots, n$, and $f_x(\overline{\{w\}}, \overline{V_1}, \dots, \overline{V_k})_0 = w$). Then we say that f_x is a* **well defined choice of representatives of $\overline{S_x}$-paths starting with** *w, if the following conditions hold:*

- *$f_x(\overline{\{w\}}, \overline{V_1}) = (w, v_1)$, where $v_1 \in V_1$ is arbitrarily chosen, for each $\overline{S_x}$-path of length 1*
- *if $f_x(\overline{\{w\}}, \overline{V_1}, \dots, \overline{V_n})$ is defined for each $\overline{S_x}$-path of length n, then for each $\overline{S_x}$-put of length $n+1$ we have the following cases:*
  - *$f_x(\overline{\{w\}}, \overline{V_1}, \overline{V_2}, \dots, \overline{V_{n+1}})_{i+1} = f_x(\overline{\{w\}}, \overline{V_2}, \dots, \overline{V_{n+1}})_i$ for $i = 1, \dots, n$, if $V_1 = \{u\}$ is singleton and wRu*
  - *$f_x(\overline{\{w\}}, \overline{V_1}, \dots, \overline{V_{n+1}}) = (w, v_1, \dots, v_{n+1})$, otherwise, where we have that $f_x(\overline{\{w\}}, \overline{V_1}, \dots, \overline{V_n}) = (w, v_1, \dots, v_n)$, and v_{n+1} is arbitrarily chosen so that $(w, v_1, \dots, v_{n+1})$ is a well defined sequence of representatives and $f_x(\overline{\{w\}}, \overline{V_1}, \dots, \overline{V_n}, \overline{V_{n+1}})_{n+1} = f_x(\overline{\{w\}}, \overline{V_1}, \dots, \overline{V_n}, \overline{V_{n+1}}')_{n+1}$ whenever the element of the family $\overline{V_{n+1}}$ determined by v_n equals the element of the family $\overline{V_{n+1}}'$ determined by v_n*

4.3 A Veltman Model Associated with a Given Verbrugge Model

Now we are ready to define $Vel(\mathfrak{M})$ and to prove that it is indeed a Veltman model.

Definition 4. *Let $\mathfrak{M} = (W, R, \{S_w : w \in W\}, \Vdash)$ be a Verbrugge model. By $Vel(\mathfrak{M}) = (W', R', \{S'_{w'} : w' \in W'\}, \Vdash)$ we denote a structure* **associated with** *$\mathfrak{M}$, defined as follows:*

- *W' consists of all ordered pairs (w, f), where $w \in W$, and f is a function which maps each $x \in W$ such that xRw to a function f_x which is a well defined choice of representatives of $\overline{S_x}$-paths starting with w*
- *$(w, f)R'(u, g)$ if and only if wRu and for all x such that xRw, for each $\overline{S_x}$-path $\overline{\{u\}}, \overline{V_1}, \dots, \overline{V_k}$ (observe that then $\overline{\{w\}}, \overline{\{u\}}, \overline{V_1}, \dots, \overline{V_k}$ is an $\overline{S_x}$-path starting with w) we have*

$$f_x(\overline{\{w\}}, \overline{\{u\}}, \overline{V_1}, \dots, \overline{V_k})_{i+1} = g_x(\overline{\{u\}}, \overline{V_1}, \dots, \overline{V_k})_i, \quad i = 1, \dots, k$$

- *$(u, g)S'_{(w,f)}(v, h)$ if and only if $(w, f)R'(u, g)$, $(w, f)R'(v, h)$ and there is an $\overline{S_w}$-path $\overline{\{u\}}, \overline{V_1}, \dots, \overline{V_k}$ such that $v = g_w(\overline{\{u\}}, \overline{V_1}, \dots, \overline{V_k})_k$, for some $k \geqslant 0$, and for any continuation, i.e. for any $\overline{S_w}$-path $\overline{\{u\}}, \overline{V_1}, \dots, \overline{V_k}, \overline{V_{k+1}}, \dots, \overline{V_{k+l}}$ we have*

$$g_w(\overline{\{u\}}, \overline{V_1}, \dots, \overline{V_{k+l}})_{k+i} = h_w(\overline{\{v\}}, \overline{V'_1}, \dots, \overline{V'_l})_i, \quad i = 1, \dots, l$$

where $\overline{S_w}$-path $\overline{\{v\}}, \overline{V'_1}, \dots, \overline{V'_l}$ is induced by v

– *$Vel(\mathfrak{M}), (w, f) \Vdash p$ if and only if $\mathfrak{M}, w \Vdash p$, for all $(w, f) \in W'$, for each propositional variable p*

Proposition 2. *Let $\mathfrak{M}$ be a Verbrugge model. Then $Vel(\mathfrak{M})$ is a Veltman model.*

Proof. Obviously R' is converse well founded. To show that it is also transitive, let $(w, f)R'(u, g)R'(v, h)$. Then $wRuRv$ and therefore wRv. Furthermore, for an arbitrary x such that xRw and an arbitrary $\overline{S_x}$-path $\overline{\{v\}}, \overline{V_1}, \ldots, \overline{V_k}$ we have

$$h_x(\overline{\{v\}}, \overline{V_1}, \ldots, \overline{V_k})_i = g_x(\overline{\{u\}}, \overline{\{v\}}, \overline{V_1}, \ldots, \overline{V_k})_{i+1}$$

$$= f_x(\overline{\{w\}}, \overline{\{u\}}, \overline{\{v\}}, \overline{V_1}, \ldots, \overline{V_k})_{i+2} = f_x(\overline{\{w\}}, \overline{\{v\}}, \overline{V_1}, \ldots, \overline{V_k})_{i+1}, \quad i = 1, \ldots, k$$

(the last equality holds since f_x is a well defined choice of representatives).

Now we verify properties of the relation $S'_{(w,f)}$ for an arbitrary (w, f). The reflexivity trivially follows from the convention that $\overline{\{u\}}$ is considered to be an $\overline{S_w}$-path of length 0. To prove the transitivity, assume $(u, g)S'_{(w,f)}(v, h)S'_{(w,f)}(z, s)$. Since $(u, g)S'_{(w,f)}(v, h)$, there is an $\overline{S_w}$-path $\overline{\{u\}}, \overline{V_1}, \ldots, \overline{V_k}$ such that $v = v_k = g_w(\overline{\{u\}}, \overline{V_1}, \ldots, \overline{V_k})_k$ and other properties from the definition of the relation $S'_{(w,f)}$ hold. Also, since $(v, h)S'_{(w,f)}(z, s)$, there is an $\overline{S_w}$-path $\overline{\{v\}}, \overline{V'_1}, \ldots, \overline{V'_l}$ such that $z = z_l = h_w(\overline{\{v\}}, \overline{V'_1}, \ldots, \overline{V'_l})_l$, with other properties from the definition of $S'_{(w,f)}$.

Put $V_{k+j} = (V_k \setminus \{v\}) \cup V'_j$ and $\overline{V_{k+j}} = \{\{x\} : x \in V_k \setminus \{v\}\} \cup \overline{V'_j}$, $j = 1, \ldots, l$. Since $vS_wV'_1$ i $xS_w\{x\}$ for all $x \in V_k \setminus \{v\}$, we have $\overline{V_k}\,\overline{S_w}\,\overline{V_{k+1}}$. Similarly, since $\overline{V'_j}\,\overline{S_w}\,\overline{V'_{j+1}}$ for $j = 1, \ldots, l$, we have that $\overline{\{u\}}, \overline{V_1}, \ldots, \overline{V_{k+l}}$ is an $\overline{S_w}$-path. Then $(u, g)S'_{(w,f)}(v, h)$ implies $g_w(\overline{\{u\}}, \overline{V_1}, \ldots, \overline{V_{k+l}})_{k+l} = h_w(\overline{\{v\}}, \overline{V'_1}, \ldots, \overline{V'_l})_l = z$.

To check the remaining condition needed to conclude $(u, g)S'_{(w,f)}(z, s)$, take any finite sequence $\overline{Z_1}, \ldots, \overline{Z_m}$ such that $\overline{\{u\}}, \overline{V_1}, \ldots, \overline{V_{k+l}}, \overline{Z_1}, \ldots, \overline{Z_m}$ is an $\overline{S_w}$-path. Consider the $\overline{S_w}$-path $\overline{\{v\}}, \overline{V'_1}, \ldots, \overline{V'_l}, \overline{Z'_1}, \ldots, \overline{Z'_m}$ induced by v and the $\overline{S_w}$-path $\overline{\{z\}}, \overline{Z''_1}, \ldots, \overline{Z''_m}$ induced by z. Then $(u, g)S'_{(w,f)}(v, h)S'_{(w,f)}(z, s)$ implies $g_w(\overline{\{u\}}, \overline{V_1}, \ldots, \overline{V_{k+l}}, \overline{Z_1}, \ldots, \overline{Z_m})_{k+l+j} = h_w(\overline{\{v\}}, \overline{V'_1}, \ldots, \overline{V'_l}, \overline{Z'_1}, \ldots, \overline{Z'_m})_{l+j} = s_w(\overline{\{z\}}, \overline{Z''_1}, \ldots, \overline{Z''_m})_j$, for all $j = 1, \ldots, m$.

Finally, assume $(w, f)R'(u, g)R'(v, h)$ and show $(u, g)S'_{(w,f)}(v, h)$. First, we have $wRuRv$, so $uS_w\{v\}$, i.e. $\overline{\{u\}}, \overline{\{v\}}$ is an $\overline{S_w}$-path and obviously it must be $g_w(\overline{\{u\}}, \overline{\{v\}})_1 = v$. Note that for any continuation $\overline{\{u\}}, \overline{\{v\}}, \overline{V_2}, \ldots, \overline{V_{l+1}}$, the $\overline{S_w}$-path induced by v is actually $\overline{\{v\}}, \overline{V_2}, \ldots, \overline{V_{l+1}}$. Furthermore, since $(u, g)R'(v, h)$, by the definition of the relation R' applied to the path $\overline{\{v\}}, \overline{V_2}, \ldots, \overline{V_{l+1}}$, we have $g_w(\overline{\{u\}}, \overline{\{v\}}, \overline{V_2}, \ldots, \overline{V_{l+1}})_{i+1} = h_w(\overline{\{v\}}, \overline{V_2}, \ldots, \overline{V_{l+1}})_i$, $i = 1, \ldots, l$, which is exactly what we need to conclude $(u, g)S'_{(w,f)}(v, h)$. ■

4.4 The Main Result

Theorem 2. *Let $\mathfrak{M} = (W, R, \{S_w : w \in W\}, \Vdash)$ be a Verbrugge model. Put $wZ(x, f)$ if and only if $w = x$. Then Z is a bisimulation between $\mathfrak{M}$ and $Vel(\mathfrak{M})$.*

Proof. The condition (at) holds by the definition of satisfaction in $Vel(\mathfrak{M})$.

To show (back), choose any $(w, f) \in W'$ and suppose $(w, f)R'(u, g)$. Then wRu and $uZ(u, g)$. Let $V \subseteq W$ such that uS_wV. Then for $v = g_w(\overline{\{u\}}, \overline{V})_1$ we have $v \in V$. It remains to define some h such that $(u, g)S'_{(w,f)}(v, h)$. First, to ensure $(w, f)R'(v, h)$, for all x such that xRw put $h_x(\overline{\{v\}}, \overline{V_1}, \overline{V_2}, \dots)_i = f_x(\overline{\{w\}}, \overline{\{v\}}, \overline{V_1}, \overline{V_2}, \dots)_{i+1}$, $i = 1, 2, \dots$, for each $\overline{S_w}$-path $\overline{\{v\}}, \overline{V_1}, \overline{V_2}, \dots$

Now, we define h_w as follows: for each $\overline{S_w}$-path $\overline{\{v\}}, \overline{V_1}, \overline{V_2}, \dots$, if $\overline{S_w}$-path $\overline{\{u\}}, \overline{V}, \overline{V_1}', \overline{V_2}', \dots$ is such that $\overline{\{v\}}, \overline{V_1}, \overline{V_2}, \dots$ is induced by v with respect to it, put $h_w(\overline{\{v\}}, \overline{V_1}, \overline{V_2}, \dots)_i = g_w(\overline{\{u\}}, \overline{V}, \overline{V_1}', \overline{V_2}', \dots)_{i+1}$, $i = 1, 2, \dots$, which is not ambiguous, i.e. does not depend on a choice of $\overline{S_w}$-path $\overline{\{u\}}, \overline{V}, \overline{V_1}', \overline{V_2}', \dots$, due to the definition of well defined choice of representatives.

To be more precise, to conclude that h_w is well defined, we need to show that for any $\overline{S_w}$-paths $\overline{\{u\}}, \overline{V}, \overline{V_1'}, \dots, \overline{V_k'}$ and $\overline{\{u\}}, \overline{V}, \overline{V_1''}, \dots, \overline{V_k''}$ such that the $\overline{S_w}$-path $\overline{\{v\}}, \overline{V_1}, \dots, \overline{V_k}$ is induced by v with respect to both of those paths, we have $g_w(\overline{\{u\}}, \overline{V}, \overline{V_1'}, \dots, \overline{V_k'}) = g_w(\overline{\{u\}}, \overline{V}, \overline{V_1''}, \dots, \overline{V_k''})$. We prove this by induction on k. For $k = 1$, since $\overline{\{v\}}, \overline{V_1}$ is induced by v with respect to both $\overline{\{u\}}, \overline{V}, \overline{V_1'}$ and $\overline{\{u\}}, \overline{V}, \overline{V_1''}$, the element of $\overline{V_1'}$ determined by v equals the element of $\overline{V_1''}$ determined by v.

So, by definition of well defined choice of representatives, $g_w(\overline{\{u\}}, \overline{V}, \overline{V_1'})_2 = g_w(\overline{\{u\}}, \overline{V}, \overline{V_1''})_2$. Also, of course $g_w(\overline{\{u\}}, \overline{V}, \overline{V_1'})_1 = g_w(\overline{\{u\}}, \overline{V}, \overline{V_1''})_1 = v$, so $g_w(\overline{\{u\}}, \overline{V}, \overline{V_1'}) = g_w(\overline{\{u\}}, \overline{V}, \overline{V_1''})$. Assume now that we have proved the claim for $k = n$ and let us prove it for $k = n + 1$. Let $\overline{\{v\}}, \overline{V_1}, \dots, \overline{V_{n+1}}$ be induced by v with respect both to $\overline{\{u\}}, \overline{V}, \overline{V_1'}, \dots, \overline{V_{n+1}'}$ and $\overline{\{u\}}, \overline{V}, \overline{V_1'}, \dots, \overline{V_{n+1}'}$. By induction hypothesis, we have $g_w(\overline{\{u\}}, \overline{V}, \overline{V_1'}, \dots, \overline{V_n'}) = g_w(\overline{\{u\}}, \overline{V}, \overline{V_1''}, \dots, \overline{V_n''})$. Since $\overline{\{v\}}, \overline{V_1}, \dots, \overline{V_{n+1}}$ is induced by v with respect to both paths, and since $g_w(\overline{\{u\}}, \overline{V})_1 = v$, the elements of families $\overline{V_{n+1}'}$ and $\overline{V_{n+1}''}$ determined by the world $g_w(\overline{\{u\}}, \overline{V}, \overline{V_1'}, \dots, \overline{V_n'})_{n+1}$ must be equal, because they are determined in the same way by the induced path. Hence, by the definition of well defined choice of representatives, $g_w(\overline{\{u\}}, \overline{V}, \overline{V_1'}, \dots, \overline{V_{n+1}'})_{n+2} = g_w(\overline{\{u\}}, \overline{V}, \overline{V_1''}, \dots, \overline{V_{n+1}''})_{n+2}$, as needed.

For all other x such that xRv we can choose h_x arbitrarily. It is easy to see that for thus defined h we have $(u, g)S'_{(w,f)}(v, h)$.

It remains to show (forth). Let $w \in W$, $(w, f) \in W'$ and u such that wRu. Let $U' = \{(x, g) \in W' : x = u \text{ and } (w, f)R'(x, g)\}$. It is easy to see that $U' \neq \emptyset$. We claim that this is a good choice of U' which shows that (forth) holds, i.e. that for any choice of one $S'_{(w,f)}$-successor for each world in U' there is V such that uS_wV and each $v \in V$ is bisimilar to some of those $S'_{(w,f)}$-successors, i.e. the first component of some of them equals v (we will shortly say that such v is *covered*). Assume the opposite, i.e. there exists a choice of one $S'_{(w,f)}$-successor for each world in U' such that for any V such that uS_wV there is $v \in V$ which is not bisimilar to any of those $S'_{(w,f)}$-successors. Let $F : U' \to W'$ be such a choice of $S'_{(w,f)}$-successors.

We will show that there exists a well defined choice of representatives of $\overline{S_w}$-paths starting with u such that each representative on any path is not covered, i.e. does not equal the first component of any $(v, h) \in F(U')$. For each path $\overline{\{u\}}, \overline{V_1}, \ldots, \overline{V_n}$, denote by V_1' the set of all uncovered elements in V_1, and by V_1'' the set of all covered elements in V_1. Then denote by V_2' the set of all uncovered worlds belonging to those elements of the union $\overline{V_2}$ which are determined by elements from V_1', i.e. they are their S_w-successors, and denote by V_2'' the set of all such worlds which are covered. Analogously define $V_3', V_3'', \ldots, V_n', V_n''$. Obviously, a desired well defined choice of representatives will exist if for any path all sets $V_1', V_2', \ldots, V_n'$ are non-empty. Now, the assumption implies $V_1' \neq \emptyset$. Assuming $V_2' = \emptyset$, by quasi-transitivity $vS_w\{v\}$ for all $v \in V_1''$ and $\overline{V_1'}\,\overline{S_w}\,\overline{V_2''}$ would imply $uS_w(V_1'' \cup V_2'')$, hence we would find an S_w-successor of u with all of its elements covered, contrary to the assumption. Similarly, for any k we can see that, if $V_k' = \emptyset$ while all before it are non-empty, we would have $uS_w(V_1'' \cup \cdots \cup V_k'')$, where $V_1'' \cup \cdots \cup V_k''$ is covered, which contradicts the assumption.

Thus we proved that there is a well defined choice of representatives g_w such that all representatives on each path are uncovered. For all x such that xRw we can choose g_x such that the condition from the definition of R' holds, and for all other $x \in W$ we can choose g_x arbitrarily. In this way we obtain g such that $(w, f)R'(u, g)$. But, by the definition of $S'_{(w,f)}$, the world $F(u, g)$ is obtained as a representative determined by g_w, which is impossible, since all representatives determined by g_w are uncovered. ■

Corollary 1. *For any formula φ and for all $(w, f) \in W'$ we have:*
$\mathfrak{M}, w \Vdash \varphi$ *if and only if* $Vel(\mathfrak{M}), (w, f) \Vdash \varphi$.

5 Further Work

By analogy to other notions of bisimulation, it is to be expected that finite approximations of bisimulation, so-called n-bisimulations, where n is a natural number, can be defined, as well as bisimulation games and n-games, with desirable properties: (n-)bisimilarity is equivalent to the existence of Defender's winning strategy in bisimulation (n-)game, and n-bisimilarity implies n-modal equivalence, i.e. the equivalence w.r.t. formulas of modal depth at most n. Furthermore, we conjecture that the converse in the case of finite alphabet would also hold. Together with Hennessy-Milner analogue we proved in Sect. 3, these results would round up arguments in favour of the definition of bisimulation between Verbrugge and Veltman models presented in this paper, but this exceeds limits and purpose of this paper, the main purpose being to show how we can transform a Verbrugge model to a bisimilar Veltman model.

More important further line of research, closely related to this purpose, is to explore how this transformation behaves with respect to particular classes of Verbrugge models and Veltman models, with additional constraints related to various principles of interpretability, which are used as additional axioms of many systems of interpretability logic in the literature. For example, if a given

Verbrugge model $\mathfrak{M}$ is an ILM-model, i.e. belongs to the characteristic class of Verbrugge models related to so-called Montagna's principle, does $Vel(\mathfrak{M})$ belong to the corresponding characteristic class of Veltman models? An analogous question may be addressed system by system, or more generally, if possible, conditions may be provided under which such a preservation works. Or if it does not work, can we modify the construction of $Vel(\mathfrak{M})$ for a particular principle or set of principles to make it work? Certainly, there is no general positive answer, since obviously for systems complete w.r.t. Verbrugge semantics but incomplete w.r.t. Veltman semantics, a transformation from a Verbrugge model to a modally equivalent Veltman model does not exist (cf. [5] for the case of the system ILP_0).

Acknowledgment. I am grateful to Mladen Vuković and the reviewers for careful reading and valuable suggestions. I thank Sebastijan Horvat for allowing me to adapt a picture from a manuscript of his PhD thesis to obtain Fig. 1 much easier than from scratch.

References

1. Joosten, J., Mas Rovira, J., Mikec, L., Vuković, M.: An overview of Verbrugge semantics, a.k.a. generalised Veltman semantics. In: Hansson, S.O., et al. (eds.) Dick de Jongh on Intuitionistic and Provability Logic. Outstanding Contributions to Logic, Springer. To appear (2023) Preprint available at https://arxiv.org/abs/2007.04722. Accessed 16 Feb 2023
2. Kurahashi, T., Okawa, Y.: Modal completeness of sublogics of the interpretability logic IL. Math. Log. Q. **67**, 164–185 (2021)
3. Horvat, S., Perkov, T., Vuković, M.: Bisimulations and bisimulation games between Verbrugge models. Mathematical Logic Quarterly. To appear (2023)
4. Mikec, L., Perkov, T., Vuković, M.: Decidability of interpretability logics ILM_0 i ILW*. Log. J. IGPL **25**, 758–772 (2017)
5. Mikec, L., Vuković, M.: Interpretability logics and generalized Veltman semantics. J. Symb. Log. **85**, 749–772 (2020)
6. Perkov, T., Vuković, M.: Filtrations of generalized Veltman models. Math. Log. Q. **62**, 412–419 (2016)
7. Mas Rovira, J.: Interpretability logics and generalized Veltman semantics in Agda. Master's thesis. University of Barcelona (2020). http://diposit.ub.edu/dspace/handle/2445/173054. Accessed 16 Feb 2023
8. Verbrugge, R.: Verzamelingen-Veltman frames en modellen (Set-Veltman frames and models). Unpublished manuscript (original scan included in [7]). Amsterdam (1992)
9. Visser, A.: An overview of interpretability logic. In: Kracht, M., et al. (eds.) Advances in Modal Logic, vol. 1, pp. 307–359. CSLI Publications (1998)
10. Vuković, M.: Bisimulations between generalized Veltman models and Veltman models. Math. Log. Q. **54**, 368–373 (2008)
11. Vuković, M.: The principles of interpretability. Notre Dame J. Form. Log. **40**, 227–235 (1999)

Focus-Style Proofs for the Two-Way Alternation-Free μ-Calculus

Jan Rooduijn(✉) and Yde Venema

ILLC, University of Amsterdam, Amsterdam, The Netherlands
j.m.w.rooduijn@uva.nl

Abstract. We introduce a cyclic proof system for the two-way alternation-free modal μ-calculus. The system manipulates one-sided Gentzen sequents and locally deals with the backwards modalities by allowing analytic applications of the cut rule. The global effect of backwards modalities on traces is handled by making the semantics relative to a specific strategy of the opponent in the evaluation game. This allows us to augment sequents by so-called trace atoms, describing traces that the proponent can construct against the opponent's strategy. The idea for trace atoms comes from Vardi's reduction of alternating two-way automata to deterministic one-way automata. Using the multi-focus annotations introduced earlier by Marti and Venema, we turn this trace-based system into a path-based system. We prove that our system is sound for all sequents and complete for sequents not containing trace atoms.

Keywords: two-way modal μ-calculus · alternation-free · cyclic proof theory

1 Introduction

The modal μ-calculus, introduced in its present form by Kozen [10], is an extension of modal logic by least and greatest fixed point operators. It retains many of the desirable properties of modal logic, such as bisimulation invariance, and relatively low complexity of the model-checking and satisfiability problems. Nevertheless, the modal μ-calculus achieves a great gain in expressive power, as the fixed point operators can be used to capture a form of recursive reasoning. This is illustrated by the fact that the modal μ-calculus embeds many well-known extensions of modal logic, such as Common Knowledge Logic, Linear Temporal Logic and Propositional Dynamic Logic.

A natural further extension is to add a converse modality $\breve{a}$ for each modality a. The resulting logic, called *two-way modal μ-calculus*, can be viewed as being able to reason about the past. As such, it can interpret the past operator of Tense Logic, and moreover subsumes PDL with converse. In this paper we are concerned with the proof theory of the two-way modal μ-calculus.

The research of this author has been made possible by a grant from the Dutch Research Council NWO, project number 617.001.857.

A version of this paper including an appendix with full proofs can be found on arXiv.

H. H. Hansen et al. (Eds.): WoLLIC 2023, LNCS 13923, pp. 318–335, 2023.
https://doi.org/10.1007/978-3-031-39784-4_20

Developing good proof systems for the modal μ-calculus is notoriously difficult. In [10], Kozen introduced a natural Hilbert-style axiomatisation, which was proven to be complete only more than a decade later by Walukiewicz [24]. Central to this proof is the use of tableau systems introduced by Niwiński and Walukiewicz in [17]. One perspective on these tableau systems is that they are cut-free Gentzen-style sequent systems allowing infinite branches. A proof in such a system, called a *non-well-founded proof*, is accepted whenever every infinite branch satisfies a certain progress condition. In case this progress condition is ω-regular (as it is in the case of the modal μ-calculus), automata-theoretic methods show that for every non-well-founded proof there is a *regular* proof, *i.e.* a proof tree containing only finitely many non-isomorphic subtrees. Since these kind of proofs can be naturally presented as finite trees with back edges, they are called *cyclic proofs*. As an alternative to non-well-founded proofs, one can use proof rules with infinitely many premisses. We will not take this route, but note that it has been applied to the two-way modal μ-calculus by Afshari, Jäger and Leigh in [2].

In [12] Lange and Stirling, for the logics LTL and CTL, annotate formulas in sequents with certain automata-theoretic information. This makes it possible to directly construct cyclic proof systems, without the detour through automata theory. This technique has been further developed by Jungteerapanich and Stirling [7,21] for the modal μ-calculus. Moreover, certain fragments of the modal μ-calculus, such as the alternation-free fragment [14] and modal logic with the master modality [19] have received the same treatment. Encoding automata-theoretic information in cyclic proofs, through annotating formulas, makes them more amenable to proof-theoretic applications, such as the extraction of interpolants from proofs [3,13].

The logic at hand, the two-way modal μ-calculus, poses additional difficulties. Already without fixed point operators, backwards modalities are known to require more expressivity than offered by a cut-free Gentzen system [18]. A common solution is to add more structure to sequents, as *e.g.* the nested sequents of Kashima [8]. This approach, however, does not combine well with cyclic proofs, as the number of possible sequents in a given proof becomes unbounded. We therefore opt for the alternative approach of still using ordinary sequents, but allowing analytic applications of the cut rule (see [6] for more on the history of this approach). The combination of analytic cuts and cyclic proofs has already been shown to work well in the case of Common Knowledge Logic [20]. Choosing analytic cuts over sequents with extended structure has recently also been gaining interest in the proof theory of logics without fixed point operators [4].

Although allowing analytic cuts handles the backwards modalities on a local level, further issues arise on a global level in the combination with non-well-founded branches. The main challenge is that the progress condition should not just hold on infinite branches, but also on paths that can be constructed by moving both up and down a proof tree. Our solution takes inspiration from Vardi's reduction of alternating two-way automata to deterministic one-way automata [22]. Roughly, the idea is to view these paths simply as upwards paths, only interrupted by several detours, each returning to the same state as where it departed. One of the main insights of the present research is that such detours have a natural interpretation in terms of the game semantics of the modal μ-calculus. We exploit this by extending the syntax with so-called *trace atoms*, whose semantics corresponds with this interpretation. Our sequents will then be one-sided Gentzen sequents containing annotated formulas, trace atoms, and negations of trace atoms.

For the sake of simplicity we will restrict ourselves to the alternation-free fragment of the modal μ-calculus. This roughly means that we will allow no entanglement of least and greatest fixed point operators. In this setting it suffices to annotate formulas with just a single bit of information, distinguishing whether the formula is *in focus* [14]. This is a great simplification compared to the full language, where annotations need to be strings and a further global annotation, called the *control*, is often used [7,21]. Despite admitting simple annotations, the trace structure of the alternation-free modal μ-calculus remains intricate. This is mainly caused by the fact that disjunctions may still appear in the scope of greatest fixed point operators, causing traces to split.

While this paper was under review, the preprint [1] by Enqvist et al. appeared, in which a proof system is presented for the two-way modal μ-calculus (with alternation). Like our system, their system is cyclic. Moreover, they also extend the syntax in order to apply the techniques from Vardi in a proof-theoretical setting. However, their extension, which uses so-called *ordinal variables*, is substantially different from ours, which uses trace atoms. It would be interesting to see whether the two approaches are intertranslatable.

In Sect. 2 we define the two-way alternation-free modal μ-calculus. Section 3 is devoted to introducing the proof system, after which in Sect. 4 we show that proofs correspond to winning strategies in a certain parity game. In Sect. 5 we prove soundness and completeness. The concluding Sect. 6 contains a short summary and some ideas for further research.

2 The (Alternation-Free) Two-Way Modal μ-Calculus

For the rest of this paper we fix the countably infinite sets P of *propositional variables* and D of *actions*. Since we want our modal logic to be *two-way*, we define an involution operation $\breve{\cdot} : \mathsf{D} \to \mathsf{D}$ such that for every $a \in \mathsf{D}$ it holds that $\breve{a} \neq a$ and $\breve{\breve{a}} = a$. We work in negation normal form, where the language $\mathcal{L}_{2\mu}$ of the *two-way modal μ-calculus* is generated by the following grammar:

$$\varphi ::= p \mid \overline{p} \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \langle a \rangle \varphi \mid [a]\varphi \mid \mu x \varphi \mid \nu x \varphi$$

where $p, x \in \mathsf{P}$, $a \in \mathsf{D}$ and in the formation of $\eta x \varphi$ ($\eta \in \{\mu, \nu\}$) the formula $\overline{x}$ does not occur in φ. The language $\mathcal{L}_{2\mu}$ expresses $\top$ and $\bot$, *e.g.* as $\nu x.x$ and $\mu x.x$. For the reader familiar with the ordinary modal μ-calculus, note that the only distinctive feauture of $\mathcal{L}_{2\mu}$ is the assumed involution operator on D.

We use standard terminology for the binding of variables by a fixpoint operator η. In particular, we write $FV(\varphi)$ for the set of variables $x \in \mathsf{P}$ that occur freely in φ and $BV(\varphi)$ for the set of those that are bound by some fixpoint operator. Note that for every $\overline{x}$ occurring in φ, we have $x \in FV(\varphi)$. For technical convenience, we assume that each formula φ is *tidy*, *i.e.* that $FV(\varphi) \cap BV(\varphi) = \emptyset$. The *unfolding* of a formula $\psi = \eta x \varphi$ is the formula $\varphi[\psi/x]$, obtained by substituting every free occurrence of x in φ by ψ. No free variables of ψ are captured by this procedure, because $FV(\psi) \cap BV(\varphi) \subseteq FV(\varphi) \cap BV(\varphi) = \emptyset$. The *closure* of a formula $\xi \in \mathcal{L}_{2\mu}$ is the least set $\mathsf{Clos}(\xi) \subseteq \mathcal{L}_{2\mu}$ such that $\xi \in \mathsf{Clos}(\xi)$ and:

(i) $\varphi \circ \psi \in \mathsf{Clos}(\xi)$ implies $\varphi, \psi \in \mathsf{Clos}(\xi)$ for each $\circ \in \{\vee, \wedge\}$;
(ii) $\triangle\varphi \in \mathsf{Clos}(\xi)$ implies $\varphi \in \mathsf{Clos}(\xi)$ for every $\triangle \in \{\langle a \rangle, [a] \mid a \in \mathsf{D}\}$;
(iii) $\eta x \varphi \in \mathsf{Clos}(\xi)$ implies $\varphi[\eta x \varphi/x] \in \mathsf{Clos}(\xi)$ for every $\eta \in \{\mu, \nu\}$.

It is well known that $\mathsf{Clos}(\xi)$ is always finite and that all formulas in $\mathsf{Clos}(\xi)$ are tidy if ξ is so (see *e.g.* [23]).

Formulas of $\mathcal{L}_{2\mu}$ are interpreted in *Kripke models* $\mathbb{S} = (S, (R_a)_{a \in \mathsf{D}}, V)$, where S is a set of *states*, for each $a \in \mathsf{D}$ we have an *accessibility relation* $R_a \subseteq S \times S$, and $V : \mathsf{P} \to \mathcal{P}(S)$ is a *valuation function*. We assume that each model is *regular*, *i.e.* that R_a is the converse relation of $R_{\breve{a}}$ for every $a \in \mathsf{D}$. Recall that the converse relation of a relation R consists of those (y, x) such that $(x, y) \in R$.

We set $R_a[s] := \{t \in S : sR_at\}$ and let $\mathbb{S}[x \mapsto X]$ be the model obtained from $\mathbb{S}$ by replacing the valuation function V by $V[x \mapsto X]$, defined by setting $V[x \mapsto X](x) = X$ and $V[x \mapsto X](p) = V(p)$ for every $p \neq x$. The *meaning* $[\![\varphi]\!]^{\mathbb{S}} \subseteq S$ of a formula $\xi \in \mathcal{L}_{2\mu}$ in $\mathbb{S}$ is inductively on the complexity of ξ:

$$
\begin{array}{llll}
[\![p]\!]^{\mathbb{S}} & := V(p) & [\![\overline{p}]\!]^{\mathbb{S}} & := S \setminus V(p) \\
[\![\varphi \vee \psi]\!]^{\mathbb{S}} & := [\![\varphi]\!]^{\mathbb{S}} \cup [\![\psi]\!]^{\mathbb{S}} & [\![\varphi \wedge \psi]\!]^{\mathbb{S}} & := [\![\varphi]\!]^{\mathbb{S}} \cap [\![\psi]\!]^{\mathbb{S}} \\
[\![\langle a \rangle \varphi]\!]^{\mathbb{S}} & := \{s \in S \mid R_a[s] \cap [\![\varphi]\!]^{\mathbb{S}} \neq \emptyset\} & [\![[a]\varphi]\!]^{\mathbb{S}} & := \{s \in S \mid R_a[s] \subseteq [\![\varphi]\!]^{\mathbb{S}}\} \\
[\![\mu x \varphi]\!]^{\mathbb{S}} & := \bigcap \{X \subseteq S \mid [\![\varphi]\!]^{\mathbb{S}[x \mapsto X]} \subseteq X\} & [\![\nu x \varphi]\!]^{\mathbb{S}} & := \bigcup \{X \subseteq S \mid X \subseteq [\![\varphi]\!]^{\mathbb{S}[x \mapsto X]}\}
\end{array}
$$

We will use the definable (see [23]) negation operator $\bar{\cdot}$ on $\mathcal{L}_{2\mu}$, for which it holds that $[\![\overline{\xi}]\!]^{\mathbb{S}} = S \setminus [\![\xi]\!]^{\mathbb{S}}$.

In this paper we shall only work with an alternative, equivalent, definition of the semantics, given by the *evaluation game* $\mathcal{E}(\xi, \mathbb{S})$. We refer the reader to the appendix below for the basic notions of (parity) games. The game $\mathcal{E}(\xi, \mathbb{S})$ is played on the board $\mathsf{Clos}(\xi) \times S$, and its ownership function and admissible moves are given in the following table.

Position	Owner	Admissible moves
$(p, s), s \in V(p)$	$\forall$	$\emptyset$
$(p, s), s \notin V(p)$	$\exists$	$\emptyset$
$(\varphi \vee \psi, s)$	$\exists$	$\{(\varphi, s), (\psi, s)\}$
$(\varphi \wedge \psi, s)$	$\forall$	$\{(\varphi, s), (\psi, s)\}$
$(\langle a \rangle \varphi, s)$	$\exists$	$\{\varphi\} \times R_a[s]$
$([a]\varphi, s)$	$\forall$	$\{\varphi\} \times R_a[s]$
$(\eta x \varphi, s)$	$-$	$\{(\varphi[\eta x \varphi/x], s)\}$

The following proposition is standard in the literature on the modal μ-calculus. See [11, Proposition 6.7] for a proof.

Proposition 1. *For every infinite $\mathcal{E}(\xi, \mathbb{S})$-match $\mathcal{M} = (\varphi_n, s_n)_{n\in\omega}$, there is a unique fixpoint formula $\eta x \chi$ which occurs infinitely often in $\mathcal{M}$ and is a subformula of φ_n for cofinitely many n.*

The winner of an infinite match $\mathcal{E}(\xi, \mathbb{S})$-match is $\exists$ if in the previous proposition $\eta = \nu$, and $\forall$ if $\eta = \mu$. It is well known that $\mathcal{E}(\xi, \mathbb{S})$ can be realised as a parity game by defining a suitable priority function on $\mathsf{Clos}(\xi) \times S$ (we again refer the reader to [11] for a detailed proof of this fact). Because of this we may, by Theorem 1 in Appendix A, assume that winning strategies are optimal and positional. Finally, we state the known fact that the two approaches provide the same meaning to formulas. For every $\varphi \in \mathsf{Clos}(\xi)$: $(\varphi, s) \in \mathrm{Win}_\exists(\mathcal{E}(\xi, \mathbb{S}))@(\varphi, s)$ if and only if $s \in [\![\varphi]\!]^{\mathbb{S}}$. If either is side of the bi-implication holds, we say that φ *is satisfied* in $\mathbb{S}$ at s and write $\mathbb{S}, s \Vdash \varphi$.

In this paper we are concerned with a fragment of $\mathcal{L}_{2\mu}$ containing only those formulas ξ which are *alternation free*, *i.e.* such that for every subformula $\eta x \varphi$ of ξ it holds that no free occurrence of x in φ is in the scope of an $\overline{\eta}$-operator in φ (where $\overline{\eta}$ denotes the opposite fixed point operator of η). This fragment is called *the alternation-free two-way modal μ-calculus* and denoted by $\mathcal{L}^{af}_{2\mu}$. We close this section by stating some typical properties of the alternation-free fragment. For $\eta \in \{\mu, \nu\}$ we use the term *η-formula* for a formula of the form $\eta x \varphi$.

Proposition 2. *Let $\xi \in \mathcal{L}^{af}_{2\mu}$ be an alternation-free formula. Then:*

- *Every formula $\varphi \in \mathsf{Clos}(\xi)$ is alternation free.*
- *The negation $\overline{\xi}$ is alternation free.*
- *An infinite $\mathcal{E}(\xi, \mathbb{S})$-match is won by $\exists$ precisely if it contains infinitely many ν-formulas, and by $\forall$ precisely if it contains infinitely many μ-formulas.*

3 The Proof System

We will call a set Σ of formulas *negation-closed* if for every $\xi \in \Sigma$ it holds that $\overline{\xi} \in \Sigma$ and $\mathsf{Clos}(\xi) \subseteq \Sigma$. For the remainder of this paper we fix a finite and negation-closed set Σ of $\mathcal{L}^{af}_{2\mu}$-formulas. For reasons of technical convenience, we will assume that every formula is drawn from Σ. This does not restrict the scope of our results, as any formula is contained in some finite negation-closed set.

3.1 Sequents

Syntax. Inspired by [14], we annotate formulas by a single bit of information.

Definition 1. *An* annotated formula *is a formula with an annotation in* $\{\circ, \bullet\}$.

The letters $b, c, d, \ldots$ are used as variables ranging over the annotations $\circ$ and $\bullet$. An annotated formula φ^b is said to be *out of focus* if $b = \circ$, and *in focus* if $b = \bullet$. The focus annotations will keep track of so-called *traces* on paths through proofs.

Roughly, a trace on a path is a sequence of formulas, such that the i-th formula occurs in the i-th sequent on the path, and the $i+1$-th formula 'comes from' the i-th formula in a way which we will define later. In Sect. 4 we will construct a game in which the winning strategies of one player correspond precisely to the proofs in our proof system. The focus mechanism enables us to formulate this game as a parity game. This is essentially also the approach taken in [14].

Where traces usually only moves upwards in a proof, the backwards modalities of our language will be enable them to go downwards as well. We will handle this in our proof system by further enriching our sequents with the following additional information.

Definition 2. *For any two formulas φ, ψ, there is a* trace atom $\varphi \rightsquigarrow \psi$ *and a* negated trace atom $\varphi \not\rightsquigarrow \psi$.

The idea for trace atoms will become more clear later, but for now one can think of $\varphi \rightsquigarrow \psi$ as expressing that there is some kind of trace going from φ to ψ, and of $\varphi \not\rightsquigarrow \psi$ as its negation. Finally, our sequents are built from the above three entities.

Definition 3. *A* sequent *is a finite set consisting of annotated formulas, trace atoms, and negated trace atoms.*

Whenever we want to refer to general elements of a sequent Γ, without specifying whether we mean annotated formulas or (negated) trace atoms, we will use the capital letters $A, B, C, \ldots$.

Semantics. We will now define the semantics of sequents. Unlike annotations, which do not affect the semantics but only serve as bookkeeping devices, the trace atoms have a well-defined interpretation. We will work with a refinement of the usual satisfaction relation that is defined with respect to a strategy for $\forall$ in the evaluation game. Most of the time, this strategy will be both *optimal* and *positional* (see Appendix A for the precise definition of these terms). Because we will frequently need to mention such optimal positional strategies, we will refer to them by the abbreviation *ops*. We first define the interpretation of annotated formulas. Note that the focus annotations play no role in this definition.

Definition 4. *Let $\mathbb{S}$ be a model, let f be an ops for $\forall$ in $\mathcal{E}@(\bigwedge \Sigma, \mathbb{S})$ and let φ^b be an annotated formula. We write $\mathbb{S}, s \Vdash_f \varphi^b$ if f is* not *winning for $\forall$ at (φ, s).*

The following proposition, which is an immediate consequence of Theorem 1 of the appendix, relates $\Vdash_f$ to the usual satisfaction relation $\Vdash$.

Proposition 3. $\mathbb{S}, s \Vdash \varphi$ *iff for every ops f for $\forall$ in $\mathcal{E}(\bigwedge \Sigma, \mathbb{S})$: $\mathbb{S}, s \Vdash_f \varphi^b$.*

The semantics of trace atoms is also given relative to an ops for $\forall$ in the game $\mathcal{E}(\bigwedge \Sigma, \mathbb{S})$ (in the following often abbreviated to $\mathcal{E}$).

Definition 5. *Given an ops f for $\forall$ in $\mathcal{E}$, we say that $\varphi \rightsquigarrow \psi$ is* satisfied *in $\mathbb{S}$ at s with respect to f (and write $\mathbb{S}, s \Vdash_f \varphi \rightsquigarrow \psi$) if there is an f-guided match*

$$(\varphi, s) = (\varphi_0, s_0) \cdot (\varphi_1, s_1) \cdots (\varphi_n, s_n) = (\psi, s) \quad (n \geq 0)$$

such that for no $i < n$ the formula φ_i is a μ-formula. We say that $\mathbb{S}$ satisfies $\varphi \not\rightsquigarrow \psi$ *at s with respect to f (and write $\mathbb{S}, s \Vdash_f \varphi \not\rightsquigarrow \psi$) iff $\mathbb{S}, s \nVdash_f \varphi \rightsquigarrow \psi$.*

The idea behind the satisfaction of a trace atom $\varphi \rightsquigarrow \psi$ at a state s is that $\exists$ can take the match from (φ, s) to (ψ, s) without passing through a μ-formula. This is good for the player $\exists$. For instance, if $\varphi \rightsquigarrow \psi$ and $\psi \rightsquigarrow \varphi$ are satisfied at s with respect to f for some $\varphi \neq \psi$, then f is necessarily losing for $\forall$ at the position (φ, s). We will later relate trace atoms to traces in infinitary proofs.

We interpret sequents disjunctively, that is: $\mathbb{S}, s \Vdash_f \Gamma$ whenever $\mathbb{S}, s \Vdash_f A$ for some $A \in \Gamma$. The sequent Γ is said to be *valid* whenever $\mathbb{S}, s \Vdash_f \Gamma$ for every model $\mathbb{S}$, state s of $\mathbb{S}$, and ops f for $\forall$ in $\mathcal{E}$.

Remark 1. There is another way in which one could interpret sequents, which corresponds to what one might call *strong validity*, and which the reader should note is different from our notion of validity. Spelling it out, we say that Γ is *strongly valid* if for every model $\mathbb{S}$ and state s there is an A in Γ that such that for every ops f for $\forall$ in $\mathcal{E}$ it holds that $\mathbb{S}, s \Vdash_f A$. While these two notions coincide for sequents containing only annotated formulas, the sequent given by $\{\varphi \wedge \psi \rightsquigarrow \varphi, \varphi \wedge \psi \rightsquigarrow \psi\}$ shows that they do not in general.

We finish this subsection by defining three operations on sequents that, respectively, extract the formulas contained annotated in some sequent, take all annotated formulas out of focus, and put all formulas into focus.

$$
\begin{aligned}
\Gamma^- &:= \{\chi \mid \chi^b \in \Gamma \text{ for some } b \in \{\circ, \bullet\}\},\\
\Gamma^\circ &:= \{\varphi \rightsquigarrow \psi \mid \varphi \rightsquigarrow \psi \in \Gamma\} \cup \{\varphi \not\rightsquigarrow \psi \mid \varphi \not\rightsquigarrow \psi \in \Gamma\} \cup \{\chi^\circ \mid \chi \in \Gamma^-\},\\
\Gamma^\bullet &:= \{\varphi \rightsquigarrow \psi \mid \varphi \rightsquigarrow \psi \in \Gamma\} \cup \{\varphi \not\rightsquigarrow \psi \mid \varphi \not\rightsquigarrow \psi \in \Gamma\} \cup \{\chi^\bullet \mid \chi \in \Gamma^-\}.
\end{aligned}
$$

3.2 Proofs

In this subsection we give the rules of our proof system. Because the rule for modalities is quite involved, its details are given in a separate definition.

Definition 6. *Let Γ be a sequent and let $[a]\varphi^b$ be an annotated formula. The* jump $\Gamma^{[a]\varphi^b}$ *of Γ with respect to $[a]\varphi^b$ consists of:*

1. (a) $\varphi^{s([a]\varphi, \Gamma)}$;
   (b) $\psi^{s(\langle a \rangle \psi, \Gamma)}$ *for every* $\langle a \rangle \psi^c \in \Gamma$;
   (c) $[\breve{a}]\chi^\circ$ *for every* $\chi^d \in \Gamma$ *such that* $[\breve{a}]\chi \in \Sigma$;

2. (a) $\varphi \rightsquigarrow \langle \breve{a} \rangle \chi$ *for every* $[a]\varphi \rightsquigarrow \chi \in \Gamma$ *such that* $\langle \breve{a} \rangle \chi \in \Sigma$;
   (b) $\langle \breve{a} \rangle \chi \not\rightsquigarrow \varphi$ *for every* $\chi \not\rightsquigarrow [a]\varphi \in \Gamma$ *such that* $\langle \breve{a} \rangle \chi \in \Sigma$;
   (c) $\psi \rightsquigarrow \langle \breve{a} \rangle \chi$ *for every* $\langle a \rangle \psi \rightsquigarrow \chi \in \Gamma$ *such that* $\langle \breve{a} \rangle \chi \in \Sigma$;
   (d) $\langle \breve{a} \rangle \chi \not\rightsquigarrow \psi$ *for every* $\chi \not\rightsquigarrow \langle a \rangle \psi \in \Gamma$ *such that* $\langle \breve{a} \rangle \chi \in \Sigma$,

where $s(\xi, \Gamma)$ *is defined by:*

$$s(\xi, \Gamma) = \begin{cases} \bullet & \text{if } \xi^\bullet \in \Gamma, \\ \bullet & \text{if } \theta \not\rightsquigarrow \xi \in \Gamma \text{ for some } \theta^\bullet \in \Gamma, \\ \circ & \text{otherwise.} \end{cases}$$

Before we go on to provide the rest of the proof system, we will give some intuition for the modal rule, by proving the lemma below. This lemma essentially expresses that the modal rule is sound. Since the annotations play no role in the soundness of an individual rule, we suppress the annotations in the proof below for the sake of readability. Intuition for the annotations in the modal rule, and in particular for the function s, is given later.

Lemma 1. *Given a model* $\mathbb{S}$*, a state* s *of* $\mathbb{S}$*, and an ops* f *for* $\forall$ *in* $\mathcal{E}$ *such that* $\mathbb{S}, s \not\Vdash_f [a]\varphi^b, \Gamma$*, there is an* a*-successor* t *of* s*, such that* $\mathbb{S}, t \not\Vdash_f \Gamma^{[a]\varphi^b}$*.*

Proof. Let $\mathbb{S}, s \not\Vdash_f [a]\varphi$ be the state t chosen by $f([a]\varphi, s)$. We claim that $\mathbb{S}, t \not\Vdash_f \Gamma^{[a]\varphi^b}$. To start with, since f is winning, we have $\mathbb{S}, t \not\Vdash_f \varphi$. Moreover, if $\langle a\rangle\psi$ belongs to Γ, then $\mathbb{S}, s \not\Vdash_f \langle a\rangle\psi$ and thus $\mathbb{S}, s \not\Vdash_f \psi$. Thirdly , if χ belongs to Γ and $[\breve{a}]\chi \in \Sigma$, then, by optimality, it holds that $\mathbb{S}, t \not\Vdash_f [\breve{a}]\chi$.

The above shows all conditions under item 1. For the conditions under item 2, suppose that $\langle\breve{a}\rangle\chi \in \Sigma$. We only show 2(d), because the others are similar. Suppose that $\chi \not\rightsquigarrow \langle a\rangle\psi \in \Gamma$. Then $\mathbb{S}, s \not\Vdash_f \chi \not\rightsquigarrow \langle a\rangle\psi$, whence $\mathbb{S}, s \Vdash_f \chi \rightsquigarrow \langle a\rangle\psi$. That means that there is an f-guided $\mathcal{E}$-match

$$(\chi, s) = (\varphi_0, s_0) \cdot (\varphi_1, s_1) \cdots (\varphi_n, s_n) = (\langle a\rangle\psi, s) \quad (n \geq 0)$$

such that none of the φ_i's is a μ-formula. But then the f-guided $\mathcal{E}$-match

$$(\langle\breve{a}\rangle\chi, t) \cdot (\varphi_0, s_0) \cdots (\varphi_n, s_n) \cdot (\psi, t)$$

witnesses that $\mathbb{S}, t \not\Vdash_f \langle\breve{a}\rangle\chi \not\rightsquigarrow \psi$, as required.

The rules of the system Focus^2 are given in Fig. 1. In each rule, the annotated formulas occurring in the set Γ are called *side formulas*. Moreover, the rules in $\{\mathsf{R}_\vee, \mathsf{R}_\wedge, \mathsf{R}_\mu, \mathsf{R}_\nu, \mathsf{R}_{[a]}\}$ have precisely one *principal formula*, which by definition is the annotated formula appearing to the left of Γ in the conclusion. Note that, due to the fact that sequents are taken to be sets, an annotated formula may at the same time be both a principal formula and a side formula.

We will now define the relation of *immediate ancestry* between formulas in the conclusion and formulas in the premisses of some arbitrary rule application. For any side formula in the conclusion of some rule, we let its *immediate ancestors* be the corresponding side formulas in the premisses. For every rule except $\mathsf{R}_{[a]}$, if some formula in the conclusion is a principal formula, its *immediate ancestors* are the annotated formulas occurring to the left of Γ in the premisses. Finally, for the *modal rule* $\mathsf{R}_{[a]}$, we stipulate that $\varphi^{s([a]\varphi,\Gamma)}$ is an *immediate ancestor* of the principal formula $[a]\varphi^b$, and that each $\psi^{s(\langle a\rangle\psi,\Gamma)}$ contained in $\Gamma^{[a]\varphi^b}$ due to clause 1(b) of Definition 6 is an *immediate ancestor* of $\langle a\rangle\psi^b \in \Gamma$.

$$\dfrac{}{\varphi^b, \overline{\varphi}^c, \Gamma}\ \mathsf{Ax1} \qquad \dfrac{}{\varphi \rightsquigarrow \psi, \varphi \not\rightsquigarrow \psi, \Gamma}\ \mathsf{Ax2} \qquad \dfrac{}{\varphi \rightsquigarrow \varphi, \Gamma}\ \mathsf{Ax3}$$

$$\dfrac{(\varphi \vee \psi) \not\rightsquigarrow \varphi, (\varphi \vee \psi) \not\rightsquigarrow \psi, \varphi^b, \psi^b, \Gamma}{\varphi \vee \psi^b, \Gamma}\ \mathsf{R}_\vee \qquad \dfrac{\varphi^\circ, \Gamma \qquad \overline{\varphi}^\circ, \Gamma}{\Gamma}\ \mathsf{cut}$$

$$\dfrac{(\varphi \wedge \psi) \not\rightsquigarrow \varphi, \varphi^b, \Gamma \qquad (\varphi \wedge \psi) \not\rightsquigarrow \psi, \psi^b, \Gamma}{\varphi \wedge \psi^b, \Gamma}\ \mathsf{R}_\wedge \qquad \dfrac{\varphi[\mu x \varphi / x]^\circ, \Gamma}{\mu x \varphi^b, \Gamma}\ \mathsf{R}_\mu$$

$$\dfrac{\nu x \varphi \not\rightsquigarrow \varphi[\nu x \varphi / x], \varphi[\nu x \varphi / x] \rightsquigarrow \nu x \varphi, \varphi[\nu x \varphi / x]^b, \Gamma}{\nu x \varphi^b, \Gamma}\ \mathsf{R}_\nu \qquad \dfrac{\Gamma^{[a]\varphi^b}}{[a]\varphi^b, \Gamma}\ \mathsf{R}_{[a]}$$

$$\dfrac{\Gamma^\bullet}{\Gamma^\circ}\ \mathsf{F} \qquad \dfrac{\varphi \not\rightsquigarrow \psi, \psi \not\rightsquigarrow \chi, \varphi \not\rightsquigarrow \chi, \Gamma}{\varphi \not\rightsquigarrow \psi, \psi \not\rightsquigarrow \chi, \Gamma}\ \mathsf{trans} \qquad \dfrac{\varphi \rightsquigarrow \psi, \Gamma \qquad \varphi \not\rightsquigarrow \psi, \Gamma}{\Gamma}\ \mathsf{tc}$$

Fig. 1. The proof rules of the system Focus^2.

As mentioned before, the purpose of the focus annotations is to keep track of *traces* of formulas on branches. Usually, a trace is a sequence of formulas $(\varphi_n)_{n<\omega}$ such that each φ_k is an immediate ancestor of φ_{k+1}. The idea is then that whenever an infinite branch has cofinitely many sequents with a formula in focus, this branch contains a trace on which infinitely many formulas are ν-formulas. Disregarding the backwards modalities for now, this can be seen as follows. As long as the focus rule is not applied, any focussed formula is an immediate ancestor of some earlier focussed formula. Since the principal formula of R_μ loses focus, while the principal formula of R_ν preserves focus, a straightforward application of Kőnig's Lemma shows that every infinite branch contains a trace with infinitely many ν-formulas. We refer the reader to [14] for more details.

Our setting is slightly more complicated, because the function s in Definition 6 additionally allows the focus to transfer along negated trace atoms, rather than just from a formula to one of its immediate ancestors. This is inspired by [22], as are the conditions in the second part of Definition 6. The main idea is that, because of the backwards modalities, traces may move not only up, but also down a proof tree. To get a grip on these more complex traces, we cut them up in segments consisting of upward paths, which are the same as ordinary traces, and loops, which are captured by the negated trace atoms. This intuitive idea will become explicit in the proof of completeness in Sect. 5.

We are now ready to define a notion of infinitary proofs in Focus^2.

Definition 7. *A* Focus^2_∞-proof *is a (possibly infinite) derivation in* Focus^2 *with:*

1. *All leaves are axioms.*
2. *On every infinite branch cofinitely many sequents have a formula in focus.*
3. *Every infinite branch has infinitely many applications of* $\mathsf{R}_{[a]}$.

As mentioned above, conditions 2 and 3 are meant to ensure that every infinite trace contains infinitely many ν-formulas. We will use this in Sect. 5 to show that infinitary proofs are sound. The key idea is to relate the traces in a proof to matches in a purported countermodel of its conclusion.

We leave it to the reader to verify that each rule, apart from the modal rule, is truth-preserving with respect to a given model $\mathbb{S}$, state s of $\mathbb{S}$, and ops f for Refuter in $\mathcal{E}(\bigwedge \Sigma, \mathbb{S})$. Since Lemma 1 already showed the soundness of the modal rule, we obtain:

Proposition 4. *Well-founded* Focus^2_∞*-proofs are sound.*

We close this section with two examples of Focus^2_∞-proofs. The first example demonstrates cut and item 1(c) of Definition 6. The second example demonstrates trace atoms.

Example 1. Define the following two formulas:

$$\varphi := \mu x(\langle \breve{a} \rangle x \vee p), \qquad \psi := \nu y([a]x \wedge \varphi).$$

The formula φ expresses 'there is a backwards a-path to some state where p holds'. The formula ψ expresses 'φ holds at every state reachable by a forwards a-path'. As our context Σ we take least negation-closed set containing φ and ψ:

$$\{\varphi, \langle \breve{a} \rangle \varphi \vee p, \langle \breve{a} \rangle \varphi, p, \psi, [a]\psi \wedge \varphi, [a]\psi, \overline{\varphi}, [\breve{a}]\overline{\varphi} \wedge \overline{p}, p, [\breve{a}]\overline{\varphi}, \overline{\psi}, \langle a \rangle \overline{\psi} \vee \overline{\varphi}, \langle a \rangle \overline{\psi}\}.$$

The implication $p \to \psi$ is valid, and below we give a Focus^2_∞-proof. As this particular proof does not rely on trace atoms, we omit them for readability.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\overline{p}^\bullet, \psi^\bullet, \langle \breve{a} \rangle \varphi^\circ, p^\circ}\,\mathsf{Ax1}}{\overline{p}^\bullet, \psi^\bullet, \langle \breve{a} \rangle \varphi \vee p^\circ}\,\mathsf{R}_\vee}{\overline{p}^\bullet, \psi^\bullet, \varphi^\circ}\,\mathsf{R}_\mu \qquad \dfrac{\dfrac{\dfrac{\begin{matrix}\pi \\ \psi^\bullet, [\breve{a}]\overline{\varphi}^\circ\end{matrix}}{\overline{p}^\bullet, [a]\psi^\bullet, \overline{\varphi}^\circ}\,\mathsf{R}_{[a]} \qquad \dfrac{}{\overline{p}^\bullet, \varphi^\bullet, \overline{\varphi}^\circ}\,\mathsf{Ax1}}{\overline{p}^\bullet, [a]\psi \wedge \varphi^\bullet, \overline{\varphi}^\circ}\,\mathsf{R}_\wedge}{\overline{p}^\bullet, \psi^\bullet, \overline{\varphi}^\circ}\,\mathsf{R}_\nu}{\overline{p}^\bullet, \psi^\bullet}\,\mathsf{cut}$$

In the above proof, the proof π is given by

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\overline{\varphi}^\circ, \varphi^\circ}\,\mathsf{Ax1}}{[a]\psi^\bullet, [\breve{a}]\overline{\varphi}^\circ, \langle \breve{a} \rangle \varphi^\circ, p^\circ}\,\mathsf{R}_{[\breve{a}]}}{[a]\psi^\bullet, [\breve{a}]\overline{\varphi}^\circ, \langle \breve{a} \rangle \varphi \vee p^\circ}\,\mathsf{R}_\vee}{[a]\psi^\bullet, [\breve{a}]\overline{\varphi}^\circ, \varphi^\circ}\,\mathsf{R}_\mu \qquad \dfrac{\begin{matrix}\vdots \\ \psi^\bullet, [\breve{a}]\overline{\varphi}^\circ\end{matrix}}{[a]\psi^\bullet, [\breve{a}]\overline{\varphi}^\circ, \overline{\varphi}^\circ}\,\mathsf{R}_{[a]}}{[a]\psi^\bullet, [\breve{a}]\overline{\varphi}^\circ}\,\mathsf{cut} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\varphi^\circ, \overline{\varphi}^\circ}\,\mathsf{Ax1}}{\langle \breve{a} \rangle \varphi^\circ, p^\circ, [\breve{a}]\overline{\varphi}^\circ}\,\mathsf{R}_{[\breve{a}]}}{\langle \breve{a} \rangle \varphi \vee p^\circ, [\breve{a}]\overline{\varphi}^\circ}\,\mathsf{R}_\vee}{\varphi^\bullet, [\breve{a}]\overline{\varphi}^\circ}\,\mathsf{R}_\mu}{[a]\psi \wedge \varphi^\bullet, [\breve{a}]\overline{\varphi}^\circ}\,\mathsf{R}_\wedge}{\psi^\bullet, [\breve{a}]\overline{\varphi}^\circ}\,\mathsf{R}_\nu$$

where the vertical dots indicate that the proof continues by repeating what happens at the root of π. The resulting proof of $\overline{p}^\bullet, \psi^\bullet$ has a single infinite branch, which can easily be seen to satisfy Condition 2 of Definition 7.

Example 2. Define $\varphi := \nu x\langle a\rangle\langle\breve{a}\rangle x$, *i.e.* φ expresses that there is an infinite path of alternating a and $\breve{a}$ transitions. Clearly this holds at every state with an a-successor. Hence the implication $\langle a\rangle p \to \varphi$ is valid. As context Σ we consider the least negation-closed set containing both $\langle a\rangle p$ and φ, *i.e.*,

$$\{\langle a\rangle p, p, \varphi, \langle a\rangle\langle\breve{a}\rangle\varphi, \langle\breve{a}\rangle\varphi, [a]\overline{p}, \overline{p}, \overline{\varphi}, [a][\breve{a}]\overline{\varphi}, [\breve{a}]\overline{\varphi}\}.$$

The following is a Focus^2_∞-proof of $\langle a\rangle p \to \varphi$.

$$\dfrac{\dfrac{\dfrac{}{\overline{p}^\bullet, \langle\breve{a}\rangle\varphi^\bullet, \langle\breve{a}\rangle\varphi \not\rightarrowtail \langle\breve{a}\rangle\varphi, \langle\breve{a}\rangle\varphi \rightsquigarrow \langle\breve{a}\rangle\varphi}\ \mathsf{Ax2}}{[a]\overline{p}^\bullet, \langle a\rangle\langle\breve{a}\rangle\varphi^\bullet, \varphi \not\rightarrowtail \langle a\rangle\langle\breve{a}\rangle\varphi, \langle a\rangle\langle\breve{a}\rangle\varphi \rightsquigarrow \varphi}\ \mathsf{R}_{[a]}}{[a]\overline{p}^\bullet, \varphi^\bullet}\ \mathsf{R}_\nu$$

Note that it is also possible to use $\mathsf{Ax3}$ instead of $\mathsf{Ax2}$ in the above proof.

4 The Proof Search Game

We will define a proof search game $\mathcal{G}(\Sigma)$ for the proof system Focus^2_∞ in the standard way. First, we require a slightly more formal definition of the notion of a rule instance.

Definition 8. *A* rule instance *is a triple* $(\Gamma, \mathsf{r}, \langle\Delta_1, \ldots, \Delta_n\rangle)$ *such that*

$$\dfrac{\Delta_1 \cdots \Delta_n}{\Gamma}\ \mathsf{r}$$

is a valid rule application in Focus^2.

The set of positions of $\mathcal{G}(\Sigma)$ is $\mathsf{Seq}_\Sigma \cup \mathsf{Inst}_\Sigma$, where Seq_Σ is the set of sequents and Inst_Σ is the set of valid rule instances (containing only formulas in Σ). Since Σ is finite, the game $\mathcal{G}(\Sigma)$ has only finitely many positions. The ownership function and admissible moves of $\mathcal{G}(\Sigma)$ are as in the following table:

Position	Owner	Admissible moves
$\Gamma \in \mathsf{Seq}_\Sigma$	Prover	$\{i \in \mathsf{Inst}_\Sigma \mid \mathsf{conc}(i) = \Gamma\}$
$(\Gamma, \mathsf{r}, \langle\Delta_1, \ldots, \Delta_n\rangle) \in \mathsf{Inst}_\Sigma$	Refuter	$\{\Delta_i \mid 1 \leq i \leq n\}$

In the above table, the expression $\mathsf{conc}(i)$ stands for the conclusion (*i.e.* the first element of the triple) of the rule instance i. As usual, a finite match is lost by the player who got stuck. An infinite $\mathcal{G}(\Sigma)$-match is won by Prover if and only it has a final segment

$$\Gamma_0 \cdot i_0 \cdot \Gamma_1 \cdot i_1 \cdots$$

on which each Γ_k has at least one formula in focus and the instance i_k is an application of $\mathsf{R}_{[a]}$ for infinitely many k. The two main observations about $\mathcal{G}(\Sigma)$ that we will use are the following:

1. A Focus^2_∞-proof of Γ is the same as a winning strategy for Prover in $\mathcal{G}(\Sigma)@\Gamma$.
2. $\mathcal{G}(\Sigma)$ is a parity game, whence positionally determined.

The first observation is immediate when viewing a winning strategy as a subtree of the full game tree. To make the second observation more explicit, we give the parity function Ω for $\mathcal{G}(\Sigma)$. On Seq_Σ, we simply set $\Omega(\Gamma) := 0$ for every $\Gamma \in \mathsf{Seq}_\Sigma$. On Inst_Σ, we define:

$$\Omega(\Gamma, \mathsf{r}, \langle \Delta_1, \ldots, \Delta_n \rangle) := \begin{cases} 3 & \text{if } \Gamma \text{ has no formula in focus,} \\ 2 & \text{if } \Gamma \text{ has a formula in focus and } \mathsf{r} = \mathsf{R}_{[a]}, \\ 1 & \text{if } \Gamma \text{ has a formula in focus and } \mathsf{r} \neq \mathsf{R}_{[a]}. \end{cases}$$

As a result we immediately obtain a method to reduce general non-well-founded proofs to cyclic proofs. Indeed, if Prover has a winning strategy, she also has *positional* winning strategy, which clearly corresponds to a *regular* Focus^2_∞-proof (that is, a proof containing only finitely many non-isomorphic subtrees.)

5 Soundness and Completeness

In this section we will prove the soundness and completeness of the system Focus^2_∞. More specifically, for soundness we will show that if Γ is invalid, then Refuter has a winning strategy in $\mathcal{G}(\Sigma)@\Gamma$. Our completeness result is slightly less wide in scope, showing only that if Refuter has a winning strategy in $\mathcal{G}(\Sigma)@\Gamma$, then Γ^- is invalid.

5.1 Soundness

For soundness, we assume an ops f for $\forall$ in $\mathcal{E} := \mathcal{E}(\bigwedge \Sigma, \mathbb{S})$ for some $\mathbb{S}$ and s such that $\mathbb{S}, s \not\Vdash_f \Gamma$. The goal is to construct from f a strategy T_f for Refuter in $\mathcal{G} := \mathcal{G}(\Sigma)$. The key idea is to assign to each position p reached in $\mathcal{G}$ a state s such that whenever $p = \Delta \in \mathsf{Seq}_\Sigma$ it holds that $\mathbb{S}, s \not\Vdash_f \Delta$. For $p \in \mathsf{Inst}_\Sigma$, the choice of T_f is then based on $f(\varphi, s)$ where φ is a formula determined by the rule instance p. The existence of such an s implies that p cannot be an axiom and thus that Refuter never gets stuck. For infinite matches, the proof works by showing that a T_f-guided $\mathcal{G}@\Gamma$-match lost by Refuter induces an f-guided $\mathcal{E}@\varphi$-match lost by $\forall$. As mentioned above, the key idea here is to relate an f-guided $\mathcal{E}@\varphi$-match to a trace through the T_f-guided $\mathcal{G}@\Gamma$-match. If the $\mathcal{G}@\Gamma$-match is losing for Refuter, it must contain a trace with infinitely many ν-formulas, which gives us an $\mathcal{E}@\varphi$-match lost by $\forall$. A novel challenge here is that not all steps in a trace necessarily go from a formula to one of its immediate ancestors, but may instead transfer along a negated trace atom. When this happens, say from φ_n to φ_{n+1}, it holds for Δ as above that both $\varphi_n^\bullet$ and $\varphi_n \not\rightsquigarrow \varphi_{n+1}$ belong to Δ. Since, by the above, it holds that $\mathbb{S}, s \not\Vdash_f \Delta$, we use the fact that $\mathbb{S}, s \Vdash_f \varphi_n \rightsquigarrow \varphi_{n+1}$ to take the $\mathcal{E}@\varphi$-match from (φ_n, s) to (φ_{n+1}, s). In the end, we obtain:

Proposition 5. *If Γ is the conclusion of a Focus^2_∞-proof, then Γ is valid.*

5.2 Completeness

For completeness we conversely show that from a winning strategy T for Refuter in $\mathcal{G}@\Gamma$, we can construct a model $\mathbb{S}^T$ and a positional strategy f_T for $\forall$ in $\mathcal{E}(\bigwedge \Sigma, \mathbb{S}^T)$ such that $\mathbb{S}^T$ falsifies Γ^- with respect to f_T. The strategy f_T we construct will not necessarily be optimal, but by Theorem 1 of Appendix A it follows that there must also be an ops $\overline{f_T}$ such that $\mathbb{S}^T \not\Vdash_f \Gamma^-$. We will view T as a tree, and restrict attention a certain subtree. We first need to define two relevant properties of rule applications.

Definition 9. *A rule application is* cumulative *if all of the premisses are supersets of the conclusion. A rule application is* productive *if all of the premisses are distinct from the conclusion.*

Without renaming T, we restrict T to its subtree where Prover adheres to the following (non-deterministic) strategy:

1. Exhaustively apply productive instances of cut and tc.
2. If applicable, apply the focus rule.
3. Exhaustively take applications of $\mathsf{R}_\vee$, $\mathsf{R}_\wedge$, R_μ, R_ν, trans that are both cumulative and productive.
4. If applicable, apply an axiom.
5. If applicable, apply a modal rule and loop back to stage (1).

It is not hard to see that each of the above phases terminates. More precisely, phases (2), (4) and (5) either terminate immediately or after applying a single rule. By the productivity requirement and the finiteness of Σ, phases (1) and (3) must terminate after a finite number of rule applications as well. Note also that non-cumulative rule applications can only happen in phases (2) or (5).

We will now define the model $\mathbb{S}^T$. The set S^T of states consists of maximal paths in T not containing a modal rule. We write $\Gamma(\rho)$ for $\bigcup\{\Gamma : \Gamma \text{ occurs in } \rho\}$. Note that, since the only possibly non-cumulative rule application in ρ is the focus rule, $\Gamma(\rho)^\bullet = \mathsf{last}(\rho)^\bullet$ for every state ρ of $\mathbb{S}^T$. Moreover, we write $\rho_1 \xrightarrow{a} \rho_2$ if ρ_2 is directly above ρ_1 in T, separated only by an application of $\mathsf{R}_{[a]}$ (we assume that trees grow upwards). We write $\rightarrow$ for the union $\bigcup\{\xrightarrow{a}: a \in \mathsf{D}\}$. Clearly, under the relation $\rightarrow$ the states of $\mathbb{S}^T$ form a forest (not necessarily a tree!). We write $\rho \leq \tau$ if τ is a descendant of ρ in this forest, *i.e.* $\leq$ is the reflexive-transitive closure of $\rightarrow$. The relations R_a^T of $\mathbb{S}^T$ are defined as follows:

$$\rho_1 R_a^T \rho_2 \text{ if and only if } \rho_1 \xrightarrow{a} \rho_2 \text{ or } \rho_2 \xrightarrow{\breve{a}} \rho_1.$$

Note that $\mathbb{S}^T$ is clearly regular. We define the valuation $V^T : S^T \rightarrow \mathcal{P}(\mathsf{P})$ by

$$V^T(\rho) := \{p : \overline{p} \in \Gamma(\rho)^-\}.$$

The restriction on T, together with the fact that it is winning for Refuter, guarantees that each $\Gamma(\rho)$ satisfies certain saturation properties, which are spelled out in the following lemma. We will later use these saturation conditions to construct our positional strategy f_T for $\forall$ in $\mathcal{E}(\bigwedge \Sigma, \mathbb{S}^T)$ and to show that $\mathbb{S}^T$ falsifies Γ with respect to f_T.

Lemma 2. *For every state ρ of $\mathbb{S}^T$, the set $\Gamma(\rho)$ is* saturated. *That is, it satisfies all of the following conditions:*

- *For no φ it holds that $\varphi, \overline{\varphi} \in \Gamma(\rho)^-$.*
- *For all φ it holds that $\varphi^\circ \in \Gamma(\rho)$ if and only if $\overline{\varphi}^\circ \notin \Gamma(\rho)$*
- *For all φ it holds that $\varphi \rightsquigarrow \psi \in \Gamma(\rho)$ if and only if $\varphi \not\rightsquigarrow \psi \notin \Gamma(\rho)$.*
- *For no φ it holds that $\varphi \rightsquigarrow \varphi \in \Gamma(\rho)$.*
- *If $\psi_1 \vee \psi_2 \in \Gamma(\rho)^-$, then for both i: $\psi_1 \vee \psi_2 \not\rightsquigarrow \psi_i \in \Gamma(\rho)$ and $\psi_i \in \Gamma(\rho)^-$.*
- *If $\psi_1 \wedge \psi_2 \in \Gamma(\rho)^-$, then for some i: $\psi_1 \wedge \psi_2 \not\rightsquigarrow \psi_i \in \Gamma(\rho)$ and $\psi_i \in \Gamma(\rho)^-$.*
- *If $\mu x \varphi \in \Gamma(\rho)^-$, then $\varphi[\mu x \varphi / x] \in \Gamma(\rho)^-$.*
- *If $\nu x \varphi \in \Gamma(\rho)^-$, then $\nu x \varphi \not\rightsquigarrow \varphi[\nu x \varphi / x] \in \Gamma(\rho)$ and $\varphi[\nu x \varphi / x] \in \Gamma(\rho)^-$.*
- *If $\nu x \varphi \in \Gamma(\rho)^-$, then $\varphi[\nu x \varphi / x] \rightsquigarrow \nu x \varphi \in \Gamma(\rho)$.*
- *If $\varphi \not\rightsquigarrow \psi, \psi \not\rightsquigarrow \chi \in \Gamma(\rho)$, then $\varphi \not\rightsquigarrow \chi \in \Gamma(\rho)$.*

Now let ρ_0 be a state of $\mathbb{S}^T$ containing the root Γ and let φ_0 be some formula such that $\varphi_0 \in \Gamma^-$. We wish to show that φ_0 is *not* satisfied at ρ_0 in $\mathbb{S}^T$. To this end, we will construct a winning strategy f_T for $\forall$ in the game $\mathcal{E} := \mathcal{E}(\bigwedge \Sigma, \mathbb{S}^T)$ initialised at (φ_0, ρ_0). The strategy f_T is defined as follows:

- At $(\psi_1 \wedge \psi_2, \rho)$, pick a conjunct $\psi_i \in \Gamma(\rho)^-$ such that $\psi_1 \wedge \psi_2 \not\rightsquigarrow \psi_i \in \Gamma(\rho)$.
- At $([a]\varphi, \rho)$, choose (φ, τ) for some τ such that $\rho \xrightarrow{a} \tau$ by virtue of some application of $\mathsf{R}_{[a]}$ with $[a]\varphi^b$ principal for some $b \in \{\circ, \bullet\}$.

Before we show that f_T is winning for $\forall$, we must first argue that it is well defined. By saturation, for every formula $\psi_1 \wedge \psi_2$ contained in $\Gamma(\rho)^-$, there is a $\psi_i \in \Gamma(\rho)^-$ with $\psi_1 \wedge \psi_2 \not\rightsquigarrow \psi_i \in \Gamma(\rho)$. Likewise, for every formula $[a]\varphi^b \in \Gamma(\rho)$, there is a τ directly above ρ in T, separated only by an application of $\mathsf{R}_{[a]}$ with $[a]\varphi^b$ principal. The following lemma therefore suffices. Its proof is by induction on the length of $\mathcal{M}$ and heavily relies on the saturation properties of Lemma 2.

Lemma 3. *Let $\mathcal{M}$ be an f_T-guided $\mathcal{E}$-match initialised at (φ_0, ρ_0). Then for any position (φ, ρ) occurring in $\mathcal{M}$ it holds that $\varphi \in \Gamma(\rho)^-$. Moreover, if (φ, ρ) comes directly after a modal step and the focus rule is applied in ρ, then $\varphi^\bullet \in \Gamma(\rho)$.*

The following lemma is key to the completeness proof. It shows that if an f_T-guided $\mathcal{E}@(\varphi_0, \rho_0)$-match loops from some state ρ to itself, without passing through a μ-formula, then this information is already contained in ρ in the form of a negated trace atom. The proof goes by induction on the number of distinct states of S^T occurring in $\mathcal{N}$. The base case, where only ρ is visited, can be shown by applying several instances of Lemma 2. For the inductive step, we crucially rely on the conditions 2(a) – 2(d) of Definition 6 to relate the trace atoms in two states τ and τ' such that $\tau R_a^T \tau'$.

Lemma 4. *Let $\rho \in S^T$. Suppose that an f_T-guided $\mathcal{E}@(\varphi_0, \rho_0)$-match $\mathcal{M}$ has a segment $\mathcal{N}$ of the form:*

$$(\varphi, \rho) = (\psi_0, s_0) \cdot (\psi_1, s_1) \cdots (\psi_n, s_n) = (\psi, \rho) \quad (n \geq 0)$$

such that for no $i < n$ the formula φ_i is a μ-formula. Then $\varphi \not\rightsquigarrow \psi \in \Gamma(\rho)$.

With the above lemmata in place, we are ready to prove that $\forall$ wins every full f_T-guided $\mathcal{E}@(\varphi_0, \rho_0)$-match $\mathcal{M}$. If $\mathcal{M}$ is finite, it is not hard to show that it must be $\exists$ who got stuck. If $\mathcal{M}$ is infinite, the proof depends on whether $\mathcal{M}$ visits some single state infinitely often. If it does, one can show that if $\exists$ would win the match $\mathcal{M}$, then $\mathcal{M}$ would visit some state ρ with $\nu x\varphi, \varphi[\nu x\varphi/x] \not\rightsquigarrow \varphi \in \Gamma(\rho)^-$, contradicting saturation. If, on the other hand, $\mathcal{M}$ visits each state at most finitely often, the proof works by showing that a win for $\exists$ in $\mathcal{M}$ would imply that T contains an infinite branch won by Prover, which is also a contradiction. In the end, we obtain the following proposition.

Proposition 6. *The strategy f_T is winning for $\forall$ in $\mathcal{E}@(\varphi_0, \rho_0)$.*

Since φ_0 was chosen arbitrarily from Γ^-, we find that $\mathbb{S}^T \not\Vdash_{f_T} \Gamma^-$. Hence, by Theorem 1 of Appendix A, we obtain completeness for the formulas in a sequent.

Proposition 7. *If Γ^- is valid, then Γ has a Focus^2_∞-proof.*

6 Conclusion

We have constructed a non-well-founded proof system Focus^2_∞ for the two-way alternation-free modal μ-calculus $\mathcal{L}^{af}_{2\mu}$. This system naturally reduces to a cyclic system when restricting to positional strategies in the proof search game.

Using the proof search game and the game semantics for the modal μ-calculus, we have shown that the system is sound for all sequents, and complete for those sequents not containing trace atoms. A natural first question for future research is to see if a full completeness result can be obtained. For this, a logic of trace atoms would have to be developed. One could for instance think of a rule like

$$\frac{\varphi \rightsquigarrow \chi, \Gamma \qquad \psi \rightsquigarrow \chi, \Gamma}{\varphi \wedge \psi \rightsquigarrow \chi, \Gamma} \; \mathsf{R}_\wedge$$

Following on this, we think it would be interesting to properly include trace atoms in the syntax by allowing the Boolean, modal and perhaps even the fixed point operators to apply to trace atoms. An example of a valid formula in this syntax is given by $((\varphi \rightsquigarrow \langle a\rangle\psi) \wedge [a](\psi \rightsquigarrow \langle \breve{a}\rangle\varphi)) \to \varphi$.

Another pressing question is whether our system could be used to prove interpolation, as has been done for language without backwards modalities in [14]. To the best of our knowledge it is currently an open question whether $\mathcal{L}^{af}_{2\mu}$ has interpolation. At the same time, it is known that analytic applications of the cut rule do not necessarily interfere with the process of extracting interpolants from proofs [9,16].

Finally, it would be interesting to see if our system can be extended to the full language $\mathcal{L}_{2\mu}$. The main challenge would be to keep track of the most important fixed point variable being unfolded on a trace. Perhaps this could be done by employing an annotation system such as the one by Jungteerapanich and Stirling [7,21], together with trace atoms that record the most important fixed point variable unfolded on a loop.

Acknowledgement. We thank Johannes Marti for insightful conversations at the outset of the present research. We also thank the anonymous reviewers for their helpful comments.

A Parity games

Definition 10. *A* (two-player) game *is a structure* $\mathcal{G} = (B_0, B_1, E, W)$ *where* E *is a binary relation on* $B := B_0 + B_1$*, and* W *is a map* $B^\omega \to \{0, 1\}$*.*

The set B is called the *board* of $\mathcal{G}$, and its elements are called *positions*. Whether a position belongs to B_0 or B_1 determines which player *owns* that position. If a player $\Pi \in \{0, 1\}$ owns a position q, it is their turn to play and the set of their *admissible moves* is given by the image $E[q]$.

Definition 11. *A* match *in* $\mathcal{G} = (B_0, B_1, E, W)$ *(or simply a* $\mathcal{G}$-match*) is a path* $\mathcal{M}$ *through the graph* (B, E)*. A match is said to be* full *if it is a maximal path.*

Note that a full match $\mathcal{M}$ is either finite, in which case $E[\mathsf{last}(\mathcal{M})] = \emptyset$, or infinite. For a $\Pi \in \{0, 1\}$, we write $\overline{\Pi}$ for the other player $\Pi + 1 \mod 2$.

Definition 12. *A full match* $\mathcal{M}$ *in* $\mathcal{G} = (B_0, B_1, E, W)$ *is* won *by player* Π *if either* $\mathcal{M}$ *is finite and* $\mathsf{last}(\mathcal{M}) \in B_{\overline{\Pi}}$*, or* $\mathcal{M}$ *is infinite and* $W(\mathcal{M}) = \Pi$*.*

If a full match $\mathcal{M}$ is finite, and $\mathsf{last}(\mathcal{M})$ belongs to B_Π for $\Pi \in \{0, 1\}$, we say that the player Π got *stuck*. A *partial match* is a match which is not full.

Definition 13. *In the context of a game* $\mathcal{G}$*, we denote by* PM_Π *the set of partial* $\mathcal{G}$*-matches* $\mathcal{M}$ *such that* $\mathsf{last}(\mathcal{M})$ *belongs to the player* Π*.*

Definition 14. *A strategy for* Π *in a game* $\mathcal{G}$ *is a map* $f : \mathrm{PM}_\Pi \to B$*. Moreover, a* $\mathcal{G}$*-match* $\mathcal{M}$ *is said to be* f-guided *if for any* $\mathcal{M}_0 \sqsubset \mathcal{M}$ *with* $\mathcal{M}_0 \in \mathrm{PM}_\Pi$ *it holds that* $\mathcal{M}_0 \cdot f(\mathcal{M}_0) \sqsubseteq \mathcal{M}$*.*

For a position q, the set $\mathrm{PM}_\Pi(q)$ contains all $\mathcal{M} \in \mathrm{PM}_\Pi$ such that $\mathsf{first}(\mathcal{M}) = q$.

Definition 15. *A strategy* f *for* Π *in* $\mathcal{G}$ *is* surviving *at a position* q *if* $f(\mathcal{M})$ *is admissible for every* $\mathcal{M} \in \mathrm{PM}_\Pi(q)$*, and* winning *at* q *if in addition all full* f*-guided matches starting at* q *are won by* Π*. A position* q *is said to be* winning *for* Π *if* Π *has a strategy winning at* q*. We denote the set of all positions in* $\mathcal{G}$ *that are winning for* Π *by* $\mathrm{Win}_\Pi(\mathcal{G})$*.*

We write $\mathcal{G}@q$ for the game $\mathcal{G}$ *initialised* at the position q of $\mathcal{G}$. A strategy f for Π is *surviving* (*winning*) in $\mathcal{G}@q$ if it is surviving (winning) in $\mathcal{G}$ at q.

Definition 16. *A strategy* f *is* positional *if it only depends on the last move,* i.e. *if* $f(\mathcal{M}) = f(\mathcal{M}')$ *for all* $\mathcal{M}, \mathcal{M}' \in \mathrm{PM}_\Pi$ *with* $\mathsf{last}(\mathcal{M}) = \mathsf{last}(\mathcal{M}')$*.*

We will often present a positional strategy for Π as a map $f : B_\Pi \to B$.

Definition 17. *A* priority map *on some board B is a map $\Omega : B \to \omega$ of finite range. A* parity game *is a game of which the winning condition is given by $W_\Omega(\mathcal{M}) = \max(Inf_\Omega(\mathcal{M})) \mod 2$, where $Inf_\Omega(\mathcal{M})$ is the set of positions occuring infinitely often in $\mathcal{M}$.*

The following theorem captures the key property of parity games: they are *positionally determined*. In fact, each player Π has a positional strategy f_Π that is *optimal*, in the sense that f_Π is winning for Π in $\mathcal{G}@q$ for *every* $q \in \mathrm{Win}_\Pi(\mathcal{G})$.

Theorem 1 ([5,15]). *For any parity game $\mathcal{G}$, there are positional strategies f_Π for each player $\Pi \in \{0, 1\}$, such that for every position q one of the f_Π is a winning strategy for Π in $\mathcal{G}@q$.*

References

1. Afshari, B., Enqvist, S., Leigh, G.E., Marti, J., Venema, Y.: Proof Systems for Two-way Modal mu-Calculus (2023). ILLC Prepublication Series, PP-2023-03
2. Afshari, B., Jäger, G., Leigh, G.E.: An infinitary treatment of full mu-calculus. In: Iemhoff, R., Moortgat, M., de Queiroz, R. (eds.) WoLLIC 2019. LNCS, vol. 11541, pp. 17–34. Springer, Heidelberg (2019). https://doi.org/10.1007/978-3-662-59533-6_2
3. Afshari, B., Leigh, G.E., Menéndez Turata, G.: Uniform interpolation from cyclic proofs: the case of modal mu-calculus. In: Das, A., Negri, S. (eds.) TABLEAUX 2021. LNCS (LNAI), vol. 12842, pp. 335–353. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86059-2_20
4. Ciabattoni, A., Lang, T., Ramanayake, R.: A theory of cut-restriction: first steps (2022). arXiv: 2203.01600
5. Emerson, E.A., Jutla, C.S.: Tree automata, mu-calculus and determinacy (extended abstract). In: 32nd Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, 1–4 October 1991, pp. 368–377. IEEE Computer Society (1991)
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. Springer, Netherlands, Dordrecht (1999). https://doi.org/10.1007/978-94-017-1754-0_6
7. Jungteerapanich, N.: A tableau system for the modal μ-calculus. In: Giese, M., Waaler, A. (eds.) TABLEAUX 2009. LNCS (LNAI), vol. 5607, pp. 220–234. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02716-1_17
8. Kashima, R.: Cut-free sequent calculi for some tense logics. Stud. Logica **53**(1), 119–136 (1994)
9. Kowalski, T., Ono, H.: Analytic cut and interpolation for bi-intuitionistic logic. Rev. Symbol. Logic **10**(2), 259–283 (2017)
10. Kozen, D.: Results on the propositional mu-calculus. Theoret. Comput. Sci. **27**, 333–354 (1983)
11. Kupke, C., Marti, J., Venema, Y.: Size matters in the modal μ-calculus (2020). arXiv: 2010.14430
12. Lange, M., Stirling, C.: Focus games for satisfiability and completeness of temporal logic. In: LICS 2001, pp. 357–365 (2001)

13. Marti, J., Venema, Y.: Focus-style proof systems and interpolation for the alternation-free μ-calculus (2021). arXiv: 2103.01671
14. Marti, J., Venema, Y.: A focus system for the alternation-free μ-calculus. In: Das, A., Negri, S. (eds.) TABLEAUX 2021. LNCS (LNAI), vol. 12842, pp. 371–388. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86059-2_22
15. Mostowski, A.: Games with forbidden positions. Technical report, Instytut Matematyki, Uniwersytet Gdański, Poland (1991)
16. Nguyen, L.A.: Analytic tableau systems and interpolation for the modal logics KB, KDB, K5, KD5. Stud. Logica **69**(1), 41–57 (2001)
17. Niwiński, D., Walukiewicz, I.: Games for the mu-calculus. Theoret. Comput. Sci. **163**(1&2), 99–116 (1996)
18. Ohnishi, M., Matsumoto, K.: Gentzen method in modal calculi. Osaka Math. J. **9**(2), 113–130 (1957)
19. Rooduijn, J.: Cyclic hypersequent calculi for some modal logics with the master modality. In: Das, A., Negri, S. (eds.) TABLEAUX 2021. LNCS (LNAI), vol. 12842, pp. 354–370. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86059-2_21
20. Rooduijn, J.M.W., Zenger, L.: An analytic proof system for common knowledge logic over S5. In: Fernández-Duque, A.P.D., Pinchinat, S. (eds.) AiML 2022, pp. 659–680. College Publications (2022)
21. Stirling, C.: A tableau proof system with names for modal μ-calculus. In: Voronkov, A., Korovina, M.V. (eds.) HOWARD-60: A Festschrift on the Occasion of Howard Barringer's 60th Birthday, EPiC Series in Computing, vol. 42, pp. 306–318 (2014)
22. Vardi, M.Y.: Reasoning about the past with two-way automata. In: Larsen, K.G., Skyum, S., Winskel, G. (eds.) ICALP 1998. LNCS, vol. 1443, pp. 628–641. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0055090
23. Venema, Y.: Lectures on the modal μ-calculus. Language and Computation, University of Amsterdam, Lecture notes. Institute for Logic (2020)
24. Walukiewicz, I.: Completeness of Kozen's axiomatisation of the propositional μ-calculus. Inf. Comput. **157**(1–2), 142–182 (2000)

Relevant Reasoning and Implicit Beliefs

Igor Sedlár[1(✉)] and Pietro Vigiani[2]

[1] Institute of Computer Science, Czech Academy of Sciences, Prague, Czech Republic
sedlar@cs.cas.cz

[2] Department of Philosophy, Scuola Normale Superiore, Pisa, Italy
pietro.vigiani@sns.it

Abstract. Combining relevant and classical modal logic is an approach to overcoming the logical omniscience problem and related issues that goes back at least to Levesque's well known work in the 1980s. The present authors have recently introduced a variant of Levesque's framework where explicit beliefs concerning conditional propositions can be formalized. However, our framework did not offer a formalization of implicit belief in addition to explicit belief. In this paper we provide such a formalization. Our main technical result is a modular completeness theorem.

Keywords: Epistemic logic · explicit belief · implicit belief · knowledge representation · modal logic · relevant logic

1 Introduction

Formal models of epistemic notions such as belief are often based on some form of modal logic and possible-worlds semantics [4]. In this approach, beliefs of an agent are modelled by a set of accessible possible worlds, and they are expressed by means of a modal operator quantifying over possible worlds: a proposition is believed if it is true in every accessible possible world. This endows the model with many closure principles allowing to make predictions about an agent's beliefs given information about their prior beliefs. For instance, if a conjunction is believed, then so are both conjuncts since every possible world satisfying the conjunction satisfies both conjuncts as well. However, such predictions are often inaccurate when it comes to real-life agents. Such agents frequently fail to realize consequence relations occurring between pieces of information (e.g. if they do not have sufficient resources at their disposal, such as time and memory), or they prioritize relevance over consequence (when the consequences at hand are not relevant to the prior beliefs or the context in general).

The possible-worlds model provides a good rendering of what has to be true given what is believed by the agent, or what is *implicitly believed*, but it fails to model what is actively held to be true by the agent, or what is *explicitly believed*. Many adjustments of the model exist that address the issue. Hector Levesque [9] famously provided a model of explicit belief based on the logic of First Degree Entailment, FDE, the implication-free fragment of Anderson and

H. H. Hansen et al. (Eds.): WoLLIC 2023, LNCS 13923, pp. 336–350, 2023.
https://doi.org/10.1007/978-3-031-39784-4_21

Belnap's relevant logic of entailment E [1]. In Levesque's model, explicit beliefs are modelled by a set of situations; unlike possible worlds, situations may be incomplete or inconsistent [2], and so information supported by a situation is not closed under consequences valid in classical propositional logic. This means that, in Levesque's model, explicit belief is not closed under classical consequence, but it is closed under consequence valid in FDE. While closure under FDE is a source of some criticism [3], it makes Levesque's framework a simple model of agents who prioritize relevance over consequence[1]. An important aspect of Levesque's model is that it combines an account of explicit belief with an account of implicit belief: a proposition is believed implicitly if it holds in every possible world satisfying the agent's explicit beliefs.

Levesque's model has been extended to allow for nesting of epistemic operators [8], which makes it possible to articulate various assumptions about the interplay of explicit and implicit belief. However, the model fails to provide a satisfactory account of explicit belief concerning *conditional propositions*. This is related to the absence of a sensible conditional connective in FDE. In a recent paper [13], we offered an extension of Levesque's model using fully-fledged relevant logic instead of the implication-free fragment. However, while our framework represented explicit beliefs (truth in all accessible situations), it did not account for implicit belief (truth in all accessible worlds). In this paper we extend the framework of [13] with an account of implicit belief. Our main technical result is a modular completeness theorem applying to a range of relevant epistemic logics with implicit and explicit belief operators.

The rest of the paper is structured as follows. In Sect. 2 we introduce the semantic framework for relevant epistemic logic of explicit and implicit belief, and in Sect. 3 we provide sound and complete axiomatisations for several logics based on the semantic framework. In the concluding Sect. 4 we summarise the paper and point to interesting further lines of research.

2 Relevant Epistemic Logic with Classical Worlds

In this section we introduce our semantic framework, based on so-called W-models introduced in [13]. These models combine the standard semantics for relevant modal logic based on *situations* [6] with a representation of *classical possible worlds*. The point of this combination is to represent agents as reasoning according to relevant logic while being situated in classical possible worlds. In our framework, a possible world is a special kind of situation where relevant negation and implication turn out to behave like their Boolean counterparts. We define validity as satisfaction in all possible worlds, and so logics based on our framework extend classical propositional logic CPC.

When it comes to modelling explicit belief in this framework, it is crucial that any situation (not only possible worlds) can be accessible from possible worlds.

[1] Such agents can be seen as reasoning according to Harman's *clutter avoidance principle* [7] in that they do not clutter their minds with trivial but unrelated consequences of the given information.

Consequently, explicit beliefs as modelled by a relevant epistemic logic $\mathsf{C.L}$ are closed under the underlying relevant logic L:

$$\frac{\vdash_{\mathsf{L}} \varphi_1 \wedge \cdots \wedge \varphi_n \to \psi}{\vdash_{\mathsf{C.L}} \Box\varphi_1 \wedge \cdots \wedge \Box\varphi_n \to \Box\psi}$$

Hence, relevant epistemic logics $\mathsf{C.L}$ model agents reasoning according to a relevant logic L while being situated in classical possible worlds. In this paper we add to the framework of [13] a representation of implicit belief using an additional epistemic accessibility relation on situations to obtain relevant epistemic logics $\mathsf{CI.L}$. Our semantics for implicit belief is set up with an eye to two crucial principles concerning the properties of implicit belief, namely, that implicit belief extends explicit belief and that it is closed under classical consequence:

$$\vdash_{\mathsf{CI.L}} \Box\varphi \to \Box_I\varphi \qquad \frac{\vdash_{\mathsf{CPC}} \varphi_1 \wedge \cdots \wedge \varphi_n \to \psi}{\vdash_{\mathsf{CI.L}} \Box_I\varphi_1 \wedge \cdots \wedge \Box_I\varphi_n \to \Box_I\psi}$$

Consequently, implicit belief is the classical closure of explicit belief (see Proposition 2):

$$\frac{\vdash_{\mathsf{CPC}} \varphi_1 \wedge \cdots \wedge \varphi_n \to \psi}{\vdash_{\mathsf{CI.L}} \Box\varphi_1 \wedge \cdots \wedge \Box\varphi_n \to \Box_I\psi}$$

Definition 1 (Language). *Let $\mathcal{L}$ be generated from a countable set of atomic propositions At via the following grammar:*

$$\varphi \in \mathcal{L} ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \to \varphi \mid \Box\varphi \mid \Box_I\varphi \mid \Box_L\varphi$$

where $p \in At$. We abbreviate $(\varphi \to \psi) \wedge (\psi \to \varphi)$ as $\varphi \leftrightarrow \psi$. Moreover, $\forall, \exists, \Rightarrow, \Leftrightarrow$ will denote, respectively, universal quantification, existential quantification, implication and equivalence in the meta-language.

The modal operators $\Box$ and $\Box_I$ have a clear epistemic interpretation, formalising explicit and implicit belief, respectively. On the other hand, $\Box_L$ has a technical role in our framework, namely, internalising in the object language validity in relevant logic. The role of $\Box_L$ becomes clear after the semantics is set up.[2]

Definition 2 (L-model). *Let a* L-model *be the tuple $(S, L, \leq, {}^*, R, Q, Q_I, Q_L, V)$ such that $(S, \leq)$ is a poset; * is an anti-monotonic function on S with respect to $\leq$; R is a ternary relation on S which is downward (upward) monotone in its first and second (third) argument; Q, Q_L are binary relations on S which are downward (upward) monotone in their first (second) argument; Q_I is a binary relation on S which is downward monotone in its first argument; and $L, V(p)$ are upward-closed subsets of S, for all $p \in At$. Moreover,*

$$\forall s \exists x (x \in L \quad \& \quad Rxss) \tag{1}$$

$$s \in L \quad \& \quad Rstu \Rightarrow t \leq u \tag{2}$$

[2] $\Box_L$ can be seen as a sort of provability operator; see Lemma 7.

The definition of L-models is virtually the standard definition of models for relevant modal logic (see e.g. [6]). L-models consist of a partially ordered set of situations, or information states, ordered by the amount of information they contain (support), and each component of the L-model satisfies the usual monotonicity condition with respect to $\leq$. We say that $\leq$ models an information order on situations in that $s \leq t$ means that t contains (supports) at least as much information as s. The unary operation * is the "Routley star", mapping each state s to its maximally compatible state, i.e. the state which is maximal with respect to the information order $\leq$ among those states that do not support the negation of any formula supported by s. R is the usual ternary relation interpreting $\rightarrow$, where $Rstu$ means that the result of combining the information contained in s with that contained in t contains at least as much information as that contained in u. L is the designated set of *logical situations*, containing situations carrying logical information, with Conditions (1–2) enforcing, as usual, the semantic deduction theorem with respect to relevant implication $\rightarrow$. As in Levesque's semantics, explicit beliefs are modelled by a set of situations. In particular, Q is the epistemic accessibility relation associated with explicit belief, associating with each state s the *epistemic state* $Q(s)$ of the (contextually fixed) agent according to the (information contained in) situation s. More specifically, $Q(s)$ consists of the situations that contain the information that is explicitly believed by the agent according to the information in s. In comparison to standard epistemic relevant models, L-models feature two further accessibility relations, Q_L and Q_I, associated with $\Box_L$ and $\Box_I$, respectively.

Definition 3 (W-model). *Let a* W-model *be the following tuple.*

$$\mathfrak{M} = (S, W, L, 0, 1, \leq, {}^*, R, Q, Q_I, Q_L, V)$$

- $(S, L, \leq, {}^*, R, Q, Q_I, Q_L, V)$ *is a L-model;*
- $W \subseteq S$ *such that for all* $w \in W$ *and* $s, t \in S$

$$w^* = w \tag{3}$$
$$Rwww \tag{4}$$
$$Rwst \Rightarrow s = 0 \textit{ or } w \leq t \tag{5}$$
$$Rwst \Rightarrow t = 1 \textit{ or } s \leq w \tag{6}$$
$$Q_I ws \Rightarrow s \in W \tag{7}$$
$$Q_I(w) \subseteq Q(w) \tag{8}$$
$$Q_L(W) = L \tag{9}$$

- $0 \notin V(p)$, $1 \in V(p)$ *for all* $p \in At$, *such that for all* $s, t \in S$, $Q_{(LI)} \in \{Q, Q_L, Q_I\}$

$$0 \leq s \leq 1 \tag{10}$$
$$1^* = 0 \quad \& \quad 0^* = 1 \tag{11}$$
$$Q_{(LI)}00 \tag{12}$$

$$Q_{(LI)}1s \Rightarrow s = 1 \tag{13}$$
$$R010 \tag{14}$$
$$R1st \Rightarrow (s = 0 \textit{ or } t = 1) \tag{15}$$

where for any relation A, $A(x) = \{y \mid Axy\}$ and $A(B) = \{y \mid \exists x \in B(Axy)\}$.

As already mentioned, possible worlds are seen as a special kind of situations; see Conditions (3–8). Conditions (3–6) enforce classical behaviour of negated and implicative formulas when evaluated at possible worlds, as clarified by Lemma 3. Conditions (7–8) concerning Q_I yield the intended interpretation of $Q_I(w)$, which represents implicit beliefs of the agent in possible world w. In particular, by Condition (7), $Q_I(w)$ contains only possible worlds and so implicit beliefs are closed under classical consequence, while by Condition (8) the "implicit" epistemic state of the agent at w is a subset of the "explicit" epistemic state, and so every explicit belief is an implicit belief. Note also that, contrary to Q and Q_L, we do not assume that Q_I is upward monotone in its second argument[3]. Finally, Condition (9) plays a fundamental role in connecting the classical and the relevant layers of our semantics. Stipulating that the set of logical states L is exactly the set of Q_L-accessible states from W yields a modified version of the semantic deduction theorem, as clarified by Lemma 4 (item 1).

The last component of W-models are the bounds $0, 1$, which represent the empty situation and the full situation, respectively (the terminology is clarified by Lemma 2). The bounds were used in [14] to provide a general frame semantics for relevant modal logic. In our setting the bounds play a technical role that will be clarified in the completeness proof; see also their discussion in [13].

Definition 4 (Satisfaction). *Let the* satisfaction *relation in a W-model $\mathfrak{M}$ (notation $\models$) be a binary relation between states of $\mathfrak{M}$ and formulas of $\mathfrak{L}$ defined recursively (on $\mathfrak{L}$) as follows.*

$$\begin{array}{lcl}
\mathfrak{M}, s \models p & \iff & s \in V(p) \\
\mathfrak{M}, s \models \neg\varphi & \iff & \mathfrak{M}, s^* \not\models \varphi \\
\mathfrak{M}, s \models \varphi \wedge \psi & \iff & \mathfrak{M}, s \models \varphi \quad \& \quad \mathfrak{M}, s \models \psi \\
\mathfrak{M}, s \models \varphi \vee \psi & \iff & \mathfrak{M}, s \models \varphi \textit{ or } \mathfrak{M}, s \models \psi \\
\mathfrak{M}, s \models \varphi \to \psi & \iff & Rstu, \mathfrak{M}, t \models \varphi \Rightarrow \mathfrak{M}, u \models \psi \\
\mathfrak{M}, s \models \Box\varphi & \iff & Qst \Rightarrow \mathfrak{M}, t \models \varphi \\
\mathfrak{M}, s \models \Box_I\varphi & \iff & Q_Ist \Rightarrow \mathfrak{M}, t \models \varphi \\
\mathfrak{M}, s \models \Box_L\varphi & \iff & Q_Lst \Rightarrow \mathfrak{M}, t \models \varphi
\end{array}$$

Let the proposition expressed by a formula φ in a W-model $\mathfrak{M}$ be $[\![\varphi]\!]^{\mathfrak{M}} = \{s \mid \mathfrak{M}, s \models \varphi\}$. Let a formula φ be valid *in a W-model $\mathfrak{M}$, written $\mathfrak{M} \models \varphi$, iff for all $w \in W$ we have that $\mathfrak{M}, w \models \varphi$. Let a formula φ be* entailed *by a set of formulas*

[3] This condition has to do with the canonical model construction (see Sect. 3), since in the canonical model Q_I^c will not be upward monotone.

Γ in a W-model $\mathfrak{M}$, written $\Gamma \models_{\mathfrak{M}} \varphi$ iff for all $s \in S$, $\mathfrak{M}, s \models \varphi$ if $\mathfrak{M}, s \models \psi$ for all $\psi \in \Gamma$. Let a formula φ be classically entailed *by a set of formulas Γ in a W-model $\mathfrak{M}$, written $\Gamma \models^{c}_{\mathfrak{M}} \varphi$ iff for all $w \in W$, $\mathfrak{M}, w \models \varphi$ if $\mathfrak{M}, w \models \psi$ for all $\psi \in \Gamma$.*

The intended properties of the semantics are highlighted in the following series of lemmas. We omit reference to $\mathfrak{M}$ whenever it is clear from the context.

Lemma 1 (Heredity). *For every W-model $\mathfrak{M}$, $s, t \in S$ and $\varphi \in \mathfrak{L}$: $s \in [\![\varphi]\!]^{\mathfrak{M}} \quad \& \quad s \leq t \Rightarrow t \in [\![\varphi]\!]^{\mathfrak{M}}$.*

Proof. By induction on the structure of φ. The base case holds by the fact that $V(p)$ is upward monotone. The cases involving $\wedge, \vee$ are trivial, while the cases involving $\neg, \rightarrow, \Box, \Box_I, \Box_L$ hold thanks to monotonicity properties of the corresponding accessibility relations (i.e., $^*, R, Q, Q_I, Q_L$, respectively). $\Box$

Lemma 2 (Full empty). *For every W-model $\mathfrak{M}$ and $\varphi \in \mathfrak{L}$: $\mathfrak{M}, 1 \models \varphi$ and $\mathfrak{M}, 0 \not\models \varphi$.*

Proof. The proof is by induction on the structure of φ, as given in [13]. The new case of $\varphi = \Box_I \psi$ is established as follows. Assuming $Q_I 1 s$, we have by (13) that $s = 1$, hence by induction hypothesis (IH) $s \models \psi$, by which we conclude that $1 \models \Box_I \psi$. Moreover, by (12) $Q_I 00$, hence there is s, namely 0, such that $Q_I 0 s$ and (by IH) $s \not\models \psi$, by which we conclude that $0 \not\models \Box_I \psi$. $\Box$

Lemma 3 (Worlds extensionality). *For every W-model $\mathfrak{M}$, $w \in W$ and $\varphi, \psi \in \mathfrak{L}$:*

$$\mathfrak{M}, w \models \neg\varphi \iff \mathfrak{M}, w \not\models \varphi$$
$$\mathfrak{M}, w \models \varphi \rightarrow \psi \iff \mathfrak{M}, w \not\models \varphi \text{ or } \mathfrak{M}, w \models \psi$$

Proof. The first claim follows from (3). The second claim follows by (4) in one direction, while the other is established by case distinction, assuming $Rwst$ and $s \models \varphi$. If $w \not\models \varphi$, by (6) either $t = 1$ (by which we conclude by Lemma 2 that $t \models \psi$), or $s \leq w$, (by which we conclude by Lemma 1 that $w \models \varphi$, which is a contradiction). If $w \models \psi$, by (5) $w \leq t$, hence by Lemma 1 we conclude that $t \models \psi$. $\Box$

Lemma 4 ((Classical) entailment). *For every W-model $\mathfrak{M}$ and $\varphi, \psi \in \mathfrak{L}$:*

1. $\varphi \models_{\mathfrak{M}} \psi \Leftrightarrow \mathfrak{M} \models \Box_L(\varphi \rightarrow \psi)$;
2. $\varphi \models^{c}_{\mathfrak{M}} \psi \Leftrightarrow \mathfrak{M} \models \varphi \rightarrow \psi$.

Proof. The first item follows from (1–2, 9) and Lemma 1, while the second from Lemma 3.

Distinguishing between explicit and implicit beliefs has interesting applications to the problem of logical omniscience. W-models help to identify the origin of logical omniscience and circumvent the problem to some extent. To recall, the logical omniscience problem for an epistemic logic extending classical propositional logic lies in the fact that whenever a set of formulas Γ classically entails φ and the agent believes each formula in Γ, then the agent automatically believes φ [5].

In the spirit of Levesque's [9], omniscience is avoided since it is possible that φ classically entails ψ in a model $\mathfrak{M}$ without $[\![\varphi]\!]^{\mathfrak{M}} \subseteq [\![\psi]\!]^{\mathfrak{M}}$. Crucially, Q is allowed to "reach out" to non-worldly situations from possible worlds, thus providing counterexamples to classically valid entailments. On the other hand, the situation with implicit belief is different: since Q_I connects possible worlds only with possible worlds by Condition (7), it cannot reach counterexamples to classically valid entailments. Thus, logical omniscience is restored, as clarified by Proposition 1. We stress that this is a welcome result, since implicit belief captures the (classical) consequences of explicit belief, i.e. what an ideal, unbounded agent would explicitly believe; see Proposition 2 at the end of the section.

Proposition 1 (Logical omniscience). *For all $\Gamma, \{\varphi\} \subseteq \mathfrak{L}$ and all W-models $\mathfrak{M}$:*

1. $\Gamma \models^c_{\mathfrak{M}} \varphi \not\Rightarrow \Box\Gamma \models^c_{\mathfrak{M}} \Box\varphi$;
2. $\Gamma \models^c_{\mathfrak{M}} \varphi \Rightarrow \Box_I\Gamma \models^c_{\mathfrak{M}} \Box_I\varphi$.

where $\Box_{(I)}\Gamma = \{\Box_{(I)}\psi \mid \psi \in \Gamma\}$ *for* $\Box_{(I)} \in \{\Box, \Box_I\}$.

Proof. Item (1) follows from the fact that, for $\Gamma = \{\psi_i \mid i \in K\}$, $\bigcap_{i\in K}([\![\psi_i]\!] \cap W) \subseteq [\![\varphi]\!] \cap W$ does not in general imply $\bigcap_{i\in K}([\![\psi_i]\!] \cap Q(W)) \subseteq [\![\varphi]\!] \cap Q(W)$. For example, consider the formulas $\neg p \vee q$ and $p \to q$, which are true in the same possible worlds for all W-model $\mathfrak{M}$, hence $\neg p \vee q \models^c_{\mathfrak{M}} p \to q$ but the two formulas may not be true in the same situations. In particular, take the W-model $\mathfrak{M}$ with $S = \{s, t\}$ such that $s^* = t$, $t \notin V(p), s \in V(p), s \notin V(q), Qss$ and $Rsss$ (the remaining components can be specified so that $\mathfrak{M}$ is indeed a W-model). In $\mathfrak{M}$, we have $s \models \Box(\neg p \vee q)$ but $s \not\models \Box(p \to q)$. Item (2) follows from the fact that, thanks to (7) we have that $Q_I(W) \subseteq W$, by which we conclude that $\bigcap_{i\in K}([\![\psi_i]\!] \cap W) \subseteq [\![\varphi]\!] \cap W$ does imply $\bigcap_{i\in K}([\![\psi_i]\!] \cap Q_I(W)) \subseteq [\![\varphi]\!] \cap Q_I(W)$. □

We note that, thanks to the above proposition, the logic of W-models is hyperintensional, in that agents can distinguish between logically equivalent propositions. The fact that in W-models agents are not logically omniscient with respect to explicit belief has other interesting consequences. Most notably, agents' belief bases are not *cluttered* by irrelevant information. That is, explicit belief is not closed under some implications valid in classical logic where the consequent introduces information that is unrelated to the information expressed in the antecedent. In our framework, "irrelevant" is seen simply as "not following by relevant logic". For instance, the following clutter principles fail for explicit belief, but they do hold for implicit belief:

$$\Box\varphi \to \Box(\psi \to \varphi) \tag{16}$$

$$\Box\varphi \to \Box(\psi \vee \neg\psi) \tag{17}$$

$$\Box(\varphi \wedge \neg\varphi) \to \Box\psi \tag{18}$$

Avoidance of epistemic clutter in our framework is mediated by the fact that relevant logics satisfy the *variable sharing principle*: an implication $\varphi \to \psi$ is provable only if φ and ψ share at least one propositional variable. This means that cases where φ and ψ are "totally unrelated" have counterexamples which can then be exploited in our framework to give counterexamples to $\Box\varphi \to \Box\psi$. However, we note that some aspects of epistemic clutter, as one may understand the notion, are preserved in our framework as, for instance, $\Box\varphi \to \Box(\varphi \vee \psi)$ is valid, for all φ and ψ (even if ψ is "totally unrelated" to φ).

We conclude this section by commenting on the relation of explicit and implicit belief in our framework. Proposition 2 says that, in a specific sense, implicit beliefs of an agent are the *classical closure* of the agent's explicit beliefs.

Lemma 5 (Implicit-explicit). *For every W-model $\mathfrak{M}$ and $\varphi \in \mathfrak{L}$: $\Box\varphi \models^c_{\mathfrak{M}} \Box_I\varphi$.*

Proof. This follows from Condition (8).

Lemma 6 (Classical-implicit). *For every W-model $\mathfrak{M}$ and $\varphi_1, \dots, \varphi_n, \psi \in \mathfrak{L}$: $\varphi_1, \dots, \varphi_n \models^c_{\mathfrak{M}} \psi \Rightarrow \Box_I\varphi_1, \dots, \Box_I\varphi_n \models^c_{\mathfrak{M}} \Box_I\psi$.*

Proof. This follows from Condition (7).

Proposition 2 (Classical closure). *For all $\varphi_1, \dots, \varphi_n, \psi \in \mathfrak{L}$ without occurrences of modal operators, the following are equivalent:*

1. *$\varphi_1 \wedge \dots \wedge \varphi_n \to \psi$ is a classical tautology;*
2. *$\Box\varphi_1 \wedge \dots \wedge \Box\varphi_n \to \Box_I\psi$ is valid in all W-models.*

Proof. 1 implies 2: If $\bigwedge_{i\leq n}\varphi_i \to \psi$ is a classical tautology, then $\bigwedge_{i\leq n}\varphi_i \models^c_{\mathfrak{M}} \psi$ for all W-models $\mathfrak{M}$ by Lemma 3. Then, $\bigwedge_{i\leq n}\Box_I\varphi_i \models^c_{\mathfrak{M}} \Box_I\psi$ by Lemma 6, which entails $\bigwedge_{i\leq n}\Box\varphi_i \models^c_{\mathfrak{M}} \Box_I\psi$ by Lemma 5. Consequently, $\bigwedge_{i\leq n}\Box\varphi_i \to \Box_I\psi$ is valid in all W-models $\mathfrak{M}$ by Lemma 4 (item 2).

2 implies 1: If $\bigwedge_{i\leq n}\varphi_i \to \psi$ is a propositional formula that is not a classical tautology, then there is a classical valuation v such that $v(\varphi_i) = 1$ for all φ_i and $v(\psi) = 0$. We may turn this valuation into a W-model $\mathfrak{M}$ with the set of states $S = \{0, v, 1\}$ and V such that $v \in V(p)$ iff $v(p) = 1$ for all $p \in At$. Moreover, we assume that $W = \{v\}$, $Q_{(I)}vv$, and the rest is added so that this structure is indeed a W-model[4]. It is obvious that $\Box\varphi_1 \wedge \dots \wedge \Box\varphi_n \to \Box_I\psi$ is not valid in $\mathfrak{M}$.

From a semantic point of view, implicit belief is stronger than the classical closure of explicit belief, as Conditions (7–8) ensure only that $Q_I(w) \subseteq Q(w) \cap W$ for all $w \in W$ and not the stronger condition $Q_I(w) = Q(w) \cap W$. However, the above proposition tells us that this does not matter in general. The present weaker semantics is more amenable to the canonical model technique.

[4] We can define $\mathfrak{M}$ similarly as in the +-construction used in the proof of Lemma 7, with the proviso that we do not add a new possible world w since v itself is seen as the only possible world in the model.

3 Axiomatization

In this section we introduce a Hilbert-style axiomatisation for our logic of explicit and implicit belief and prove that it is sound and complete with respect to the class of W-models. In fact, we provide a modular soundness and completeness result for a family of several logics CI.L, where L ranges over a number of relevant logics, extending our basic system at the propositional and modal level. The methods employed here are the same as the ones used in [13]. In particular, we use a Henkin-style canonical model construction (see Definition 8) which combines the usual strategies for completeness in classical propositional logic (defining worlds as maximally consistent CI.L-theories) and relevant modal logics (defining information states as prime L-theories). We note that a crucial step in the proof is a model construction allowing to transform every L-model into a suitable W-model, so that $\vdash_{\mathsf{L}} \varphi \Rightarrow \vdash_{\mathsf{CI.L}} \Box_L \varphi$ is an admissible meta-rule (see Lemma 7 Item (1)). A similar result for the framework without implicit belief was proven in [13].

We begin by recalling Fuhrmann's axiomatization of the basic conjunctively regular relevant modal logic BM.C [6]. In our formulation, the logic contains three modal operators, not one.

Definition 5 (Axiom system BM.C). *Let* BM.C *be a conjunctively regular multi-modal axiom system comprising the following axioms and rules:*

– *The following axioms and rules of the propositional relevant logic* BM *[12]:*

(BM1)	$\varphi \to \varphi$	(BM8)	$\neg(\varphi \wedge \psi) \to (\neg\varphi \vee \neg\psi)$
(BM2)	$(\varphi \wedge \psi) \to \varphi$	(BM9)	$(\neg\varphi \wedge \neg\psi) \to \neg(\varphi \vee \psi)$
(BM3)	$(\varphi \wedge \psi) \to \psi$	(BM10)	$((\varphi \to \psi) \wedge (\varphi \to \chi)) \to (\varphi \to (\psi \wedge \chi))$
(BM4)	$\varphi \to (\varphi \vee \psi)$	(BM11)	$((\varphi \to \chi) \wedge (\psi \to \chi)) \to ((\varphi \vee \psi) \to \chi)$
(BM5)	$\psi \to (\varphi \vee \psi)$	(BM12)	$(\varphi \wedge (\psi \vee \chi)) \to ((\varphi \wedge \psi) \vee (\varphi \wedge \chi))$
(BM6)	$\dfrac{\varphi \qquad \varphi \to \psi}{\psi}$	(BM13)	$\dfrac{\varphi \to \chi \qquad \psi \to \xi}{(\varphi \to \psi) \to (\chi \to \xi)}$
(BM7)	$\dfrac{\varphi \qquad \psi}{\varphi \wedge \psi}$	(BM14)	$\dfrac{\varphi \to \psi}{\neg\psi \to \neg\varphi}$

– *The following axioms and rules, for* $\Box_{(IC)} \in \{\Box, \Box_I, \Box_L\}$*:*

$$(\Box_{(IC)}.\mathsf{C}) \quad \Box_{(IC)}\varphi \wedge \Box_{(IC)}\psi \to \Box_{(IC)}(\varphi \wedge \psi)$$

$$(\Box_{(IC)}.\mathsf{M}) \quad \frac{\varphi \to \psi}{\Box_{(IC)}\varphi \to \Box_{(IC)}\psi}$$

Figure 1 lists further axioms and rules one may add to BM.C in order to obtain well-known relevant axiom systems (see [6] for a taxonomy).

Our goal is to set up a general framework that allows the user to use the relevant logic that is most suitable given their intuitions or the situation at hand. Obvious candidates include modal extensions of the strong relevant logics E or R, which received detailed discussion and motivation in [1] and [10], respectively.

In particular, the logic E.C (the conjunctively regular modal extension of E) is obtained by adding (L1–L8) and (L11) to BM.C, and R.C results from E.C by adding (L9).

In what follows, we use the variable L for an axiom system extending BM.C with axioms and rules of Fig. 1, and we stipulate that $Rstuv := \exists x(Rstx \ \& \ Rxuv)$, $Rs(tu)v := \exists x(Rsxv \ \& \ Rtux)$, $RQ_{(I)}stu := \exists x(Rstx \ \& \ Q_{(I)}xu)$ and $Q_{(I)}Rstu := \exists x(Q_{(I)}sx \ \& \ Rxtu)$. Moreover, we assume that in (L12)–(L17) the frame condition with the suitable accessibility relation $Q_{(I)} \in \{Q, Q_I\}$ corresponds to the $\Box_{(I)}$-variant of each. Let a L-*model* be a W-model satisfying the frame conditions corresponding to the axioms and rules of L from Fig. 1. Finally, let the $\Box_L$-*version* of a L-axiom (L-rule) be obtained by prefixing $\Box_L$ to the axiom (to each of the premises and the conclusion of the rule).

Axiom/rule		**Frame condition**
(L1)	$\varphi \leftrightarrow \neg\neg\varphi$	$s^{**} = s$
(L2)	$(\varphi \to \psi) \to (\neg\psi \to \neg\varphi)$	$Rstu \Rightarrow Rsu^*t^*$
(L3)	$((\varphi \to \psi) \wedge (\psi \to \chi)) \to (\varphi \to \chi)$	$Rstu \Rightarrow Rs(st)u$
(L4)	$\varphi \vee \neg\varphi$	$s \in L \Rightarrow s^* \leq s$
(L5)	$(\varphi \to \neg\varphi) \to \neg\varphi$	Rss^*s
(L6)	$(\varphi \to \psi) \to ((\chi \to \varphi) \to (\chi \to \psi))$	$Rstuv \Rightarrow Rs(tu)v$
(L7)	$(\varphi \to \psi) \to ((\psi \to \chi) \to (\varphi \to \chi))$	$Rstuv \Rightarrow Rt(su)v$
(L8)	$(\varphi \to (\varphi \to \psi)) \to (\varphi \to \psi)$	$Rstu \Rightarrow Rsttu$
(L9)	$(\varphi \to (\psi \to \chi)) \to (\psi \to (\varphi \to \chi))$	$Rstuv \Rightarrow Rsutv$
(L10)	$\varphi \to (\varphi \to \varphi)$	$Rstu \Rightarrow (s \leq u \vee t \leq u)$
(L11)	$\dfrac{\varphi}{(\varphi \to \psi) \to \psi}$	$\exists x(x \in L \ \& \ Rsxs)$
(L12)	$\dfrac{\varphi}{\Box\varphi}$	$(x \in L \ \& \ Qxs) \Rightarrow s \in L$
(L13)	$\Box_{(I)}(\varphi \to \psi) \to (\Box_{(I)}\varphi \to \Box_{(I)}\psi)$	$RQ_{(I)}stu \Rightarrow \exists x(Q_{(I)}tx \ \& \ Q_{(I)}Rsxu)$
(L14)	$\Box_{(I)}\varphi \to \varphi$	$Q_{(I)}ss$
(L15)	$\Box_{(I)}\neg\varphi \to \neg\Box_{(I)}\varphi$	$\exists x(Q_{(I)}sx^* \ \& \ Q_{(I)}s^*x)$
(L16)	$\Box_{(I)}\varphi \to \Box_{(I)}\Box_{(I)}\varphi$	$(Q_{(I)}st \ \& \ Q_{(I)}tu) \Rightarrow Q_{(I)}su$
(L17)	$\neg\Box_{(I)}\varphi \to \Box_{(I)}\neg\Box_{(I)}\varphi$	$(Q_{(I)}s^*u \ \& \ Q_{(I)}st) \Rightarrow Q_{(I)}t^*u$

Fig. 1. Frame conditions with the corresponding axioms and rules for L.

Definition 6 (Axiom system CI.L). *Let the logic* CI.L *consist of the following:*

- *an axiomatisation of classical propositional logic (*CPC*);*
- *the $\Box_L$-versions of axioms and rules of* L*;*
- *the following axioms and rules:*

$$(\Box\Box_I) \quad \Box\varphi \to \Box_I\varphi$$

$$(\Box_I.\mathsf{K}) \quad \Box_I(\varphi \to \psi) \to (\Box_I\varphi \to \Box_I\psi)$$

$$(\Box_I.\mathsf{N}) \quad \frac{\varphi}{\Box_I\varphi}$$

$$(\mathsf{BR}) \quad \frac{\Box_L(\varphi \to \psi)}{\varphi \to \psi}$$

Let provability of a formula φ in CI.L*, written $\vdash_{\mathsf{CI.L}} \varphi$, be defined as usual.*

Theorem 1 (CI.L soundness). *For every* L*-model* $\mathfrak{M}$*:* $\vdash_{\mathsf{CI.L}} \varphi \Rightarrow \mathfrak{M} \models \varphi$.

Proof. By induction on the length of CI.L-proofs. The axioms of CPC are valid thanks to Lemma 3. The fact that, for each L, all L-axioms are satisfied in all states $s \in L$ in all L-models is established as usual in relevant modal logic; see [13] for details. The cases corresponding to the $\Box_I$-variants of (L13)–(L17) are established similarly as their $\Box$-variant. Then, we infer that $\Box_L\varphi$ is valid in each L-model for each L-axiom φ using (9). The fact that the $\Box_L$-versions of L-rules preserve validity is established similarly. The cases corresponding to the remaining axioms and rules is established as follows. For ($\Box_I$.K), assume $w \models \Box_I(\varphi \to \psi)$ and $w \models \Box_I\varphi$ for all $w \in W$. Hence, for all s such that $Q_I ws$, $s \models \varphi \to \psi$ and $s \models \varphi$. By (7) we have $s \in W$, hence $s \models \psi$, by which we conclude that $w \models \Box_I\psi$. For ($\Box\Box_I$), assume $w \models \Box\varphi$ and $Q_I ws$ for some $w \in W$ and $s \in S$. By (8) we have Qws, hence by $w \models \Box\varphi$ we have $s \models \varphi$. Hence, we conclude that $w \models \Box_I\varphi$. For ($\Box_I$.N), assume $w \models \varphi$ for all $w \in W$ and $Q_I ws$ for some arbitrary $s \in S$. By (7) we have $s \in W$, hence $s \models \varphi$, by which we conclude that $\models \Box_I\varphi$. For (BR), assume $w \models \Box_L(\varphi \to \psi)$ for all $w \in W$. By Lemma 4 we have $[\![\varphi]\!] \subseteq [\![\psi]\!]$, hence by $W \subseteq S$ and Lemma 3 we conclude that $w \models \varphi \to \psi$. □

The following lemma clarifies the relationship between the logics L and CI.L. In particular, by item (1), the modal operator $\Box_L$ expresses L-provability within CI.L, and items (2) and (3) state some interesting consequences of item (1). We note that item (1) will be crucial in establishing that Condition (9) holds in the canonical model.

Lemma 7 (L-CI.L). *For every* $\varphi \in \mathfrak{L}$*:*

1. $\vdash_{\mathsf{L}} \varphi \Leftrightarrow \vdash_{\mathsf{CI.L}} \Box_L\varphi$*;*
2. $\vdash_{\mathsf{L}} \varphi \Rightarrow \vdash_{\mathsf{CI.L}} \varphi$*, for* L *not containing* (L12)*;*
3. $\vdash_{\mathsf{CI.L}} \Box_L\varphi \Rightarrow \vdash_{\mathsf{CI.L}} \varphi$*, for* L *not containing* (L12)*.*

Proof. For item (1), one direction is established by induction on the length of L-proofs. If φ is an L-axiom, by definition of CI.L $\Box_L\varphi$ is a CI.L-axiom. If φ is obtained by a L-inference rule with premises $\varphi_1, \ldots, \varphi_n$, by IH $\vdash_{\mathsf{CI.L}} \Box_L\varphi_1, \ldots, \vdash_{\mathsf{CI.L}} \Box_L\varphi_n$, hence by application of the $\Box_L$-version of the rule we conclude $\vdash_{\mathsf{CI.L}} \Box_L\varphi$. For the other direction we construct a W-model $\mathfrak{M}^+$ from a L-model $\mathfrak{M} = (S, \leq, L, R, *, Q, Q_I, V)$ such that if $\mathfrak{M} \not\models \varphi$, then $\mathfrak{M}^+ \not\models \Box_L\varphi$ (the result then follows by 1). Let $\mathfrak{M}^+ = (S^+, \leq^+, W^+, R^+, *^+, Q^+, Q^+_L, Q^+_I, V^+)$ be defined as follows:

$$S^+ = S \cup \{w, 0, 1\}$$
$$\leq^+ = \leq \cup \{(w, w)\} \cup \{(s, 1) \mid s \in S^+\} \cup \{(0, s) \mid s \in S^+\}$$
$$W = \{w\}$$
$$L^+ = L \cup \{w, 1\}$$
$$R^+ = R \cup \{(w, w, w)\} \cup \{(0, s, t), (s, 0, t), (s, t, 1) \mid s, t \in S^+\}$$

$$*^+ = * \cup \{(w,w)\} \cup \{(0,1),(1,0)\}$$
$$Q^+ = Q \cup \{(w,w)\} \cup \{(s,1) \mid s \in S^+\} \cup \{(0,s) \mid s \in S^+\}$$
$$Q_L^+ = Q_L \cup \{(w,w)\} \cup \{(w,s) \mid s \in L\} \cup \{(s,1) \mid s \in S^+\} \cup \{(0,s) \mid s \in S^+\}$$
$$Q_I^+ = Q_I \cup \{(w,w)\} \cup \{(0,s) \mid s \in S^+\}$$
$$V^+(p) = V(p) \cup \{1\} \text{ for all } p$$

It suffices to prove (i) that $\mathfrak{M}^+$ is a W-model; (ii) that for all $s \in S$, $\mathfrak{M}, s \models \varphi \Leftrightarrow \mathfrak{M}^+, s \models \varphi$; and (iii) that each frame condition of Fig. 1 holds in $\mathfrak{M}^+$ whenever it holds in $\mathfrak{M}$. Putting (i)–(iii) together, we conclude that if there is $l \in L$ such that $\mathfrak{M}, l \not\models \varphi$, then $(\mathfrak{M}^+, w) \not\models \Box_L\varphi$. (i) is established as in [13], with the new cases involving (9) and the monotonicity property of Q_I^+ holding by inspection of the definition of $\mathfrak{M}^+$. (ii) is established by induction on the structure of φ, as in [13], with the new case $\varphi = \Box_I\psi$ established as follows. If $\mathfrak{M}, s \not\models \Box_I\psi$, then $\mathfrak{M}^+, s \not\models \Box_I\psi$ by $Q_I \subseteq Q_I^+$ and IH. Conversely, if $\mathfrak{M}^+, s \not\models \Box_I\psi$, then there is t such that Q_I^+st and $\mathfrak{M}^+, t \not\models \psi$. By inspection of the definition of Q_I^+, $t \in S$ and so Q_Ist, which implies using IH that $\mathfrak{M}, s \not\models \Box_I\psi$. (iii) is established virtually as in [13] thanks to the observation that $s \in L^+$ iff Q_L^+ws (the case corresponding to the $\Box_I$-variants of (L12)–(L17) is almost identical as their $\Box$-variants).

Item (2) is established by induction on the length of L-proofs. All implicational axioms and rules of L are provable (preserve provability) in CI.L by item (1) and (BR), and (Adj) preserves provability thanks to $\vdash_{\mathsf{CPC}} \varphi \to (\psi \to (\varphi \wedge \psi))$ for all φ, ψ[5]. Item (3) follows from items (1) and (2). □

Definition 7 (Theories, Pairs). *A* L-theory *is a set of formulas T closed under provable implications and under conjunction, i.e. for all $\varphi, \psi \in \mathfrak{L}$ (i) $\varphi \in T$ and $\vdash_{\mathsf{L}} \varphi \to \psi$ implies $\psi \in T$ and (ii) $\varphi, \psi \in T$ implies $\varphi \wedge \psi \in T$. A* L*-theory is* regular *if it contains all theorems of* L*;* prime *if for all $\varphi, \psi \in \mathfrak{L}$ $\varphi \vee \psi \in T$ implies $\varphi \in T$ or $\psi \in T$;* proper *if it does not contain all formulas of $\mathcal{L}$.*

A pair of sets of formulas (Γ, Δ) is (C).L-independent *(for* (C.)L $\in$ {CI.L, L}*) iff there are no finite non-empty sets $\Gamma' \subseteq \Gamma$ and $\Delta' \subseteq \Delta$ such that $\vdash_{(\mathsf{C}).\mathsf{L}} \bigwedge \Gamma' \to \bigvee \Delta'$.*

Lemma 8 (Extension Lemma). *If (Γ, Δ) is* L*-independent (*CI.L*-independent and both Γ and Δ are non-empty), then there is a prime* L*-theory (non-empty proper prime* CI.L*-theory) Σ such that $\Gamma \subseteq \Sigma$ and $\Delta \cap \Sigma = \emptyset$.*

Proof. [11]. □

Definition 8 (Canonical model). *Let the* canonical CI.L-model *be the following tuple:*

$$\mathfrak{M}^c = (S^c, W^c, L^c, 0^c, 1^c \leq^c, R^c, *^c, Q^c, Q_L^c, V^c)$$

– *S^c is the set of prime* L*-theories;*

[5] Note that (L12) is problematic since φ in general is not an implication, so we cannot use item (1) and (BR).

- W^c *is the set of non-empty proper prime* CI.L*-theories;*
- L^c *is the set of regular prime* L*-theories;*
- $0^c = \emptyset$ *and* $1^c = \mathfrak{L}$*;*
- $\leq^c = \subseteq$*;*
- $\varphi \in s^{*^c}$ *iff* $\neg\varphi \notin s$*;*
- $R^c stu$ *iff* $\varphi \to \psi \in s \quad \& \quad \varphi \in t \Rightarrow \psi \in u$*;*
- $Q^c st$ *iff* $\Box\varphi \in s \Rightarrow \varphi \in t$*;*
- $Q^c_L st$ *iff* $\Box_L\varphi \in s \Rightarrow \varphi \in t$*;*
- $Q^c_I st \text{ iff } \begin{cases} \Box_I\varphi \in s \Rightarrow \varphi \in t & \text{if } s \notin W^c \\ (\Box_I\varphi \in s \Rightarrow \varphi \in t) \quad \& \quad t \in W^c & \text{if } s \in W^c \end{cases}$
- $s \in V^c(p)$ *iff* $p \in s$*.*

In what follows, we omit the superscript from the canonical CI.L-model $\mathfrak{M}^c$ whenever the context allows this.

Lemma 9 (Canonical model). $\mathfrak{M}^c$ *is a* L*-model.*

Proof. First, the canonical model is well-defined since $W \subseteq S$ ($\vdash_{\mathsf{L}} \varphi \to \psi$ implies $\vdash_{\mathsf{CI.L}} \Box_L(\varphi \to \psi)$ by the first item of Lemma 7, and $\vdash_{\mathsf{CI.L}} \Box_L(\varphi \to \psi)$ implies $\vdash_{\mathsf{CI.L}} \varphi \to \psi$ using (BR)). The monotonicity properties of $*, R, Q, Q_L$ hold by inspection of the definition of $\mathfrak{M}$. To show that Q_I is downward monotone in its first argument, assume $Q_I st$, $u \leq s$ and $\Box_I\varphi \in u$. If $s \in W$, then by $Q_I st$ we have $t \in W$ and by $u \leq s$ we have $\Box_I\varphi \in s$, by which we conclude that $\varphi \in t$. If $s \notin W$, then by $u \leq s$ we have $\Box_I\varphi \in s$, by which we conclude by $Q_I st$ that $\varphi \in t$. The proof for 1–6 and 10–15 is as in [13][6]. The remaining conditions are established as follows. (7) holds since, assuming $Q_I st$ and $s \in W$, by definition of Q_I we have that $t \in W$. (8) holds since, assuming $Q_I st$, $\Box\varphi \in s$ and $s \in W$, by ($\Box\Box_I$) we have $\Box_I\varphi \in s$, hence by definition of Q_I we conclude that $\varphi \in t$. (9) holds by the following argument. By contraposition, assume $s \notin L$. i.e. $\varphi \notin s$ for some φ such that $\vdash_{\mathsf{L}} \varphi$. By Lemma 7 $\vdash_{\mathsf{CI.L}} \Box_L\varphi$, hence $\Box_L\varphi \in w$ for all $w \in W$, which together with $\varphi \notin s$ implies that not $Q_L ws$ for all $w \in W$. Hence, $s \notin Q_L(W)$. Conversely, assume $s \in L$. We have to prove that there is $w \in W$ such that $Q_L ws$. If $s = 1$, then it is sufficient to show that W is non-empty. This follows from the fact that each CI.L considered here is consistent by Theorem 1. If $s \neq 1$, then we reason as follows. Consider the pair $(\{\psi \mid\vdash_{\mathsf{CI.L}} \psi\}, \{\Box_L\varphi \mid \varphi \notin s\})$ and note that both sets in the pair are non-empty. The pair is CI.L-independent, since otherwise

[6] The presence of the bounds $0, 1$ is necessary for the following reason. The bound-free versions of Conditions (5–6) are sufficient for Lemma 3, but these simpler versions do not hold in the canonical model. For instance, $\emptyset$ is a perfectly legitimate prime L-theory, and $Rw\emptyset t$ obviously holds for all $w \in W^c$ and $t \in S^c$. Hence, $Rwst \Rightarrow w \subseteq t$ fails. (The argument that $Rwst \Rightarrow s \subseteq w$ fails is similar, exploiting the possibility that $t = \mathfrak{L}$.) In this situation we can either add extensional truth constants to the language, and so rule out $\emptyset$ and $\mathfrak{L}$ as legitimate L-theories, or work with $\emptyset$ and $\mathfrak{L}$ as special kinds of states in the model while modifying the frame conditions (5–6) so that they refer to these special states. We chose the second option.

- $\vdash_{\mathsf{CI.L}} \bigvee_{i<n} \Box_L \varphi_i$ for some $n > 0$ only if (by $\vdash_{\mathsf{CI.L}} \Box_L \varphi \vee \Box_L \psi \to \Box_L(\varphi \vee \psi)$)
- $\vdash_{\mathsf{CI.L}} \Box_L \bigvee_{i<n} \varphi_i$ only if (by Lemma 7)
- $\vdash_{\mathsf{L}} \bigvee_{i<n} \varphi_i$ only if (by $s \in L$)
- $\bigvee_{i<n} \varphi_i \in s$ only if (since s is prime)
- $\varphi_i \in s$ for some $i < n$

which contradicts $\varphi_i \notin s$. It follows using Lemma 8 that there is a non-empty proper prime CI.L-theory w such that $Q_L ws$. Finally, the fact that the frame conditions corresponding to (L1)–(L17) are canonical is established as in [13], where the new cases of the conditions corresponding to the $\Box_I$-variants of (L12)–(L17) are virtually identical to their $\Box$-variants. □

Lemma 10 (Truth). *For every $\varphi \in \mathfrak{L}$: $\varphi \in s \Leftrightarrow \mathfrak{M}^c, s \models \varphi$.*

Proof. By induction on the structure of φ. The proof employs the standard arguments of relevant modal logic (using the fact that $\Box, \Box_L$ are conjunctively regular modalities, see e.g. [6]), except for the case $\varphi := \Box_I \psi$, which we show as an illustration. For one direction, assume $\Box_I \varphi \in s$ and $Q^c st$. Hence, $\varphi \in t$, by which we conclude that $t \models \varphi$ by IH. For the other direction, assume $\Box_I \psi \notin s$ and consider the pair $t_0 = (\{\chi \mid \Box_I \chi \in s\}, \{\psi\})$. In case $s \in W^c$, we have to show that t_0 is CI.L-independent. This holds, since otherwise

- $\vdash_{\mathsf{CI.L}} \chi_1 \wedge \cdots \wedge \chi_n \to \psi$ only if (by ($\Box_I$.C) and ($\Box_I$.M), which are derivable using ($\Box_I$.K) and ($\Box_I$.N) in the usual way)
- $\vdash_{\mathsf{CI.L}} \Box_I \chi_1 \wedge \cdots \wedge \Box_I \chi_n \to \Box_I \psi$ only if (by construction of t_0)
- $\Box_I \psi \in s$

contradicting $\Box_I \psi \notin s$. Hence, by Lemma 8 there is, $t \in W^c$ such that $Q_I st$ and $\psi \notin t$. If $s \notin W^c$, then the argument is similar – we just need to show that t_0 is L-independent. In both cases, $s \not\models \Box_I \psi$, using the induction hypothesis. □

Theorem 2 (Completeness). *For all $\varphi \in \mathfrak{L}$: If $\mathfrak{M} \models \varphi$ for every* L*-model $\mathfrak{M}$, then $\vdash_{\mathsf{CI.L}} \varphi$.*

Proof. The theorem follows from Lemmas 9 and 10. □

4 Conclusion

This paper extends our framework from [13] with a formalization of implicit belief. In the spirit of Levesque [9], we model explicit beliefs of an agent by a set of accessible situations that may contain counterexamples to classically valid entailments, and we model implicit beliefs by a subset of accessible situations that behave like classical possible worlds. In our setting, explicit belief is closed under the underlying relevant logic, while implicit belief is closed under classical logic and corresponds to the classical closure of explicit belief. The framework is best seen as formalizing agents that reason using a relevant logic because they prioritize relevance over classical consequence with the goal of not cluttering their belief bases by irrelevant consequences of their information. Our main technical result is a modular completeness theorem for a family of relevant epistemic logics based on the framework, extending the completeness result of [13].

We note that undecidability of some relevant logics L (such as E and R, for instance [15]) implies undecidability of CI.L in view of Lemma 7. We conjecture that, conversely, if L is decidable, then so is CI.L.

Natural topics for future research include a study of extensions of the present framework with a formalization of group-epistemic notions (common and distributed belief) and with a formalization of epistemic dynamics (public announcement, or action models in general). Another topic is a deeper investigation of possible applications in knowledge representation.

Acknowledgement. We thank the anonymous reviewers for comments. This work was supported by the Czech Science Foundation grant no. GA22-01137S.

References

1. Anderson, A.R., Belnap, N.D.: Entailment: The Logic of Relevance and Necessity, Volume I. Princeton University Press, Princeton (1975)
2. Barwise, J., Perry, J.: Situations and Attitudes. MIT Press, Cambridge (1983)
3. Fagin, R., Halpern, J.Y.: Belief, awareness, and limited reasoning. Artif. Intell. **34**(1), 39–76 (1988)
4. Fagin, R., Halpern, J.Y., Moses, Y., Vardi, M.Y.: Reasoning About Knowledge. MIT Press, Cambridge (1995)
5. Fagin, R., Halpern, J.Y., Vardi, M.: A nonstandard approach to the logical omniscience problem. Artif. Intell. **79**, 203–240 (1995)
6. Fuhrmann, A.: Models for relevant modal logics. Stud. Logica **49**(4), 501–514 (1990). https://doi.org/10.1007/BF00370161
7. Harman, G.: Change in View: Principles of Reasoning. MIT Press, Cambridge (1986)
8. Lakemeyer, G.: Tractable meta-reasoning in propositional logics of belief. In: IJCAI 1987, pp. 401–408. Morgan Kaufmann Publishers Inc. (1987)
9. Levesque, H.: A logic of implicit and explicit belief. In: Proceedings of AAAI 1984, pp. 198–202 (1984)
10. Mares, E.D.: Relevant logic and the theory of information. Synthese **109**(3), 345–360 (1996). https://doi.org/10.1007/BF00413865
11. Restall, G.: An Introduction to Substructural Logics. Routledge, London (2000)
12. Routley, R., Plumwood, V., Meyer, R.K., Brady, R.T.: Relevant Logics and Their Rivals, Volume 1. Ridgeview (1982)
13. Sedlár, I., Vigiani, P.: Relevant reasoners in a classical world. In: Duque, D.F., Palmigiano, A., Pichinat, S. (eds.) Advances in Modal Logic, London, vol. 14, pp. 697–718. College Publications (2022)
14. Seki, T.: General frames for relevant modal logics. Notre Dame J. Formal Log. **44**(2), 93–109 (2003)
15. Urquhart, A.: The undecidability of entailment and relevant implication. J. Symb. Log. **49**(4), 1059–1073 (1984)

Decidability of Modal Logics of Non-k-Colorable Graphs

Ilya Shapirovsky(✉)

New Mexico State University, Las Cruces, USA
ilshapir@nmsu.edu

Abstract. We consider the bimodal language, where the first modality is interpreted by a binary relation in the standard way, and the second is interpreted by the relation of inequality. It follows from Hughes (1990), that in this language, non-k-colorability of a graph is expressible for every finite k. We show that modal logics of classes of non-k-colorable graphs (directed or non-directed), and some of their extensions, are decidable.

Keywords: chromatic number · modal logic · difference modality · decidability · finite model property · filtration

1 Introduction

It is known that a non-k-colorability of a graph can be expressed by propositional modal formulas [Hug90]. In [GHV04], such formulas were used to construct a canonical logic which cannot be determined by a first-order definable class of relational structures; this gave a solution of a long-standing problem by Fine [Fin75].

In this paper, we are interested in decidability of modal logics given by axioms of non-k-colorability, and some of their extensions. We consider the bimodal language, where the first modality is interpreted by a binary relation in the standard way, and the second (difference modality) is interpreted by the relation of inequality.

The paper has the following structure. Section 2 provides preliminary syntactic and semantic facts. In Sect. 3, the finite model property and decidability are shown for logics of non-k-colorable graphs. In Sect. 4, these results are obtained for the connected non-directed case. Further results on the finite model property of logics of non-k-colorable graphs are obtained in Sect. 5. A discussion is given in Sect. 6.

2 Preliminaries

We assume that the reader is familiar with basic notions in modal logic (see, e.g., [CZ97, BdRV01] for the references). Below we briefly remind some of them.

The original version of the chapter has been revised. A correction to this chapter can be found at
https://doi.org/10.1007/978-3-031-39784-4_25

H. H. Hansen et al. (Eds.): WoLLIC 2023, LNCS 13923, pp. 351–361, 2023.
https://doi.org/10.1007/978-3-031-39784-4_22

Modal Syntax and Relational Semantics. The set of *n-modal formulas* is built from a countable set of *variables* $\mathrm{PV} = \{p_0, p_1, \ldots\}$ using Boolean connectives $\bot, \to$ and unary connectives $\Diamond_i$, $i < n$ (*modalities*). Other logical connectives are defined as abbreviations in the standard way, in particular $\Box_i\varphi$ denotes $\neg\Diamond_i\neg\varphi$.

An *n-frame* is a structure $F = (X, (R_i)_{i<n})$, where X is a non-empty set and $R_i \subseteq X \times X$ for $i < n$. A *valuation in a frame* F is a map $\mathrm{PV} \to \mathcal{P}(X)$, where $\mathcal{P}(X)$ is the set of all subsets of X. A *(Kripke) model on* F is a pair (F, θ), where θ is a valuation. The *truth* of formulas in models is defined in the usual way:

- $M, x \models p_i$ iff $x \in \theta(p_i)$;
- $M, x \not\models \bot$;
- $M, x \models \varphi \to \psi$ iff $M, x \models \varphi$ implies $M, x \models \psi$;
- $M, x \models \Diamond_i\varphi$ iff there exists y such that xR_iy and $M, y \models \varphi$.

A formula φ is *true in a model* M, in symbols $M \vDash \varphi$, if $M, x \vDash \varphi$ for all x in M. A formula φ is *valid in a frame* F, in symbols $F \vDash \varphi$, if φ is true in every model on F. For a class $\mathcal{C}$ of structures (frames or models) and a set of formulas Φ, we write $\mathcal{C} \vDash \Phi$, if $S \vDash \varphi$ for all $S \in \mathcal{C}$ and $\varphi \in \Phi$.

For the standard notions of *generated* and *point-generated subframe* and *submodel*, and *p-morphism*, we refer the reader to [CZ97, Section 3.3] or [BdRV01, Sections 2.1 and 3.3].

Modal Logics. A (*propositional normal n-modal*) *logic* is a set L of n-modal formulas that contains all classical tautologies, the axioms $\neg\Diamond_i\bot$ and $\Diamond_i(p_0 \vee p_1) \to \Diamond_i p_0 \vee \Diamond_i p_1$ for each $i < n$, and is closed under the rules of modus ponens, substitution and *monotonicity*; the latter means that for each $i < n$, $\varphi \to \psi \in L$ implies $\Diamond_i\varphi \to \Diamond_i\psi \in L$.[1] We write $L \vdash \varphi$ for $\varphi \in L$. For a set Φ of n-modal formulas, $L + \Phi$ is the smallest normal logic containing $L \cup \Phi$. For a formula φ, $L + \varphi$ abbreviates $L + \{\varphi\}$. K denotes the smallest unimodal logic.

An *L-frame* is a frame where L is valid.

For a class $\mathcal{C}$ of n-frames, the set of n-modal formulas φ such that $\mathcal{C} \vDash \varphi$ is called the *logic of* $\mathcal{C}$ and is denoted by $\mathrm{Log}\,\mathcal{C}$. It is straightforward that $\mathrm{Log}\,\mathcal{C}$ is a normal logic. Such logics are called *Kripke complete*. A logic has the *finite model property* (fmp), if it is the logic of a class of finite frames (by the cardinality of a frame or model we mean the cardinality of its domain). We say that L has the *exponential fmp*, if for every formula $\varphi \notin L$, φ is falsified in an L-frame of cardinality $\leq 2^{\ell(\varphi)}$, where $\ell(\varphi)$ is the number of subformulas of φ.

The *canonical model* $M_L = (X_L, (R_{i,L})_{i<n}, \theta_L)$ *of* L is built from maximal L-consistent sets X_L of n-modal formulas; the canonical relations and the valuation are defined in the standard way. Namely, for $\Gamma, \Delta \in X_L$, put $(\Gamma, \Delta) \in R_{i,L}$, if $\{\Diamond_i\varphi \mid \varphi \in \Delta\} \subseteq \Gamma$, and set $\theta_L(p) = \{\Gamma \in X_L \mid p \in \Gamma\}$ for $p \in \mathrm{PV}$. The following fact is well known, see e.g., [BdRV01, Chapter 4.2].

[1] For this version of the definition of normal modal logic, see, e.g., [BdRV01, Remark 4.7].

Proposition 1 *[Canonical model theorem].* $L \vdash \varphi$ *iff* $M_L \vDash \varphi$.

L is *canonical*, if L is valid in its *canonical frame* $F_L = (X_L, (R_{i,L})_{i<n})$. A formula φ is *canonical*, if $F_L \vDash \varphi$ whenever $\varphi \in L$.

Proposition 2. *Let L be a canonical n-modal logic. Then for any n-modal logic $L' \supseteq L$, we have $F_{L'} \vDash L$.*

This fact is well known and follows from a simple observation that $F_{L'}$ is a generated subframe of F_L.

Logics with the Difference Modality. It is known that adding the difference modality allows to increase the expressive power of propositional modal language (see, e.g., [dR92, GG93] in the relational context, or [KS14] for topological semantics).

Is this paper we will consider bimodal ($n = 2$) and unimodal ($n = 1$) languages. We write $\Diamond$ for $\Diamond_0$, and $\langle\neq\rangle$ for $\Diamond_1$; likewise for boxes. We also use abbreviations $\exists\varphi$ for $\langle\neq\rangle\varphi \vee \varphi$ and $\forall\varphi$ for $[\neq]\varphi \wedge \varphi$.

For a unimodal frame $F = (X, R)$, let $F_{\neq}$ be the bimodal frame $(X, R, \neq_X)$, where $\neq_X$ is the inequality relation on X, i.e., the set of pairs $(x, y) \in X \times X$ such that $x \neq y$. For a class $\mathcal{F}$ of frames, put $\mathcal{F}_{\neq} = \{F_{\neq} \mid F \in \mathcal{F}\}$

For a unimodal logic L, let $L_{\neq}$ be the smallest bimodal logic that contains L and the following formulas:

$$p \to [\neq]\langle\neq\rangle p, \quad \langle\neq\rangle\langle\neq\rangle p \to \exists p, \quad \Diamond p \to \exists p. \tag{1}$$

Recall that the validity of $p \to [\neq]\langle\neq\rangle p$ in a frame (X, R, D) expresses that D is symmetric, the formula $\langle\neq\rangle\langle\neq\rangle p \to \exists p$ means that the relation $D \cup Id_X$ is transitive (Id_X denotes the diagonal relation on X), and the formula $\Diamond p \to \exists p$ expresses that $R \subseteq D \cup Id_X$; see, e.g., [dR92] for details.

In particular, it follows that we have the following characterization of bimodal point-generated frames that validate $\mathrm{K}_{\neq}$:

Proposition 3. *$F = (X, R, D)$ is a point-generated $\mathrm{K}_{\neq}$-frame iff $\neq_X \subseteq D$.*

The formulas (1) are Sahlqvist formulas, and hence are canonical (see, e.g., [CZ97, Theorem 10.30]). In particular, it follows that $\mathrm{K}_{\neq}$ is Kripke complete. It is well-known that this logic has the finite model property: for every non-theorem φ of $\mathrm{K}_{\neq}$, consider a submodel M of the canonical model of $\mathrm{K}_{\neq}$ generated by a point x where φ is refuted, and take a filtration of M.

Proposition 4 ([dR92]). *$\mathrm{K}_{\neq}$ is the logic of the class of all (finite) frames of the form $(X, R, \neq_X)$.*

This proposition follows from Proposition 3 and the following standard move that "repairs" D-reflexive points. For a point-generated $\mathrm{K}_{\neq}$-frame $F = (X, R, D)$, let $F^{(\neq)}$ be the frame $(Y, S, \neq_Y)$, where

$$\begin{aligned} Y &= \{(x, 0) : x \in X\} \cup \{(x, 1) : x \in X \,\&\, xDx\}, \\ (x, i)S(y, j) &\text{ iff } xRy. \end{aligned}$$

Let $f : X \to Y$ be the map defined by $f(x, i) = x$. Readily, f is a p-morphism from $F^{(\neq)}$ onto F. Now Proposition 4 follows from the p-morphism lemma (see, e.g., [BdRV01, Theorem 3.14(i)]).

The frame $F^{(\neq)}$ will be used later; we will call it the *repairing of* F.

3 Logics of Non-*k*-Colorable Graphs

By a *graph* we mean a unimodal frame (X, R) in which R is symmetric. A *directed graph* is a unimodal frame. As usual, a *partition* $\mathcal{A}$ of a set X is a family of non-empty pairwise disjoint sets such that $X = \bigcup \mathcal{A}$.

Definition 1. Let X be a set, $R \subseteq X \times X$. A partition $\mathcal{A}$ of X is *proper*, if $\forall A \in \mathcal{A}\, \forall x \in A\, \forall y \in A\, \neg xRy$. Let

$$C(X, R) = \{|\mathcal{A}| : \mathcal{A} \text{ is a finite proper partition of } X\}.$$

Let $\chi(X, R)$ be the least k in $C(X, R)$, if $C(X, R) \neq \varnothing$, and ∞ otherwise.

In the case when R is symmetric, $\chi(X, R)$ is called the *chromatic number of the graph* (X, R).

Put

$$\chi_k^{>} = \forall \bigvee_{i<k} (p_i \wedge \bigwedge_{i \neq j < k} \neg p_j) \to \exists \bigvee_{i<k} (p_i \wedge \Diamond p_i).$$

Proposition 5 ([Hug90, GHV04]). *Let $F = (X, R, D)$ be a point-generated* $\mathrm{K}_{\neq}$*-frame. Then $\chi(X, R) > k$ iff $F \vDash \chi_k^{>}$.*

Remark 1. Formulas considered in [Hug90, GHV04] are formally different.

Proof. The premise of $\chi_k^{>}$ says that non-empty values of p_i's form a partition of X, the conclusion says that this partition is not proper. □

In particular, it follows that for every graph G,

$$\text{the chromatic number of } G > k \text{ iff } G_{\neq} \vDash \chi_k^{>}.$$

To show that logics of non-k-colorable graphs have the finite model property, we will use filtrations.

For a model $M = (X, (R_i)_{i<n}, \theta)$ and a set of n-modal formulas Γ, put

$$x \sim_\Gamma y \text{ iff } \forall \psi \in \Gamma (M, x \models \psi \text{ iff } M, y \models \psi)$$

.

For a formula φ, let $\mathrm{Sub}\,\varphi$ be the set of all subformulas of φ. A set Γ of formulas is Sub-*closed*, if $\mathrm{Sub}\,\varphi \subseteq \Gamma$ whenever $\varphi \in \Gamma$.

Definition 2. Let Γ be a Sub-closed set of formulas. A *Γ-filtration* of a model $M = (X, (R_i)_{i<n}, \theta)$ is a model $\widehat{M} = (\widehat{X}, (\widehat{R}_i)_{i<n}, \widehat{\theta})$ such that

1. $\widehat{X} = X/{\sim}$ for some equivalence relation $\sim$ such that $\sim \subseteq \sim_\Gamma$;
2. $\widehat{M}, [x] \models p$ iff $M, x \models p$ for all $p \in \Gamma$. Here $[x]$ is the $\sim$-class of x.
3. For all $i < n$, we have $(R_i)_\sim \subseteq \widehat{R}_i \subseteq (R_i)^\Gamma_\sim$, where

$$[x]\,(R_i)_\sim\,[y] \text{ iff } \exists x' \sim x\ \exists y' \sim y\ (x'\,R_i\,y'),$$
$$[x]\,(R_i)^\Gamma_\sim\,[y] \text{ iff } \forall \psi\ (\Diamond_i\psi \in \Gamma\ \&\ M, y \models \psi \Rightarrow M, x \models \Diamond_i\psi).$$

The relations $(R_i)_\sim$ are called the *minimal filtered relations*.

If $\sim = \sim_\Psi$ for some finite set of formulas $\Psi \supseteq \Gamma$, then $\widehat{M}$ is called a *definable Γ-filtration* of the model M.

The following fact is well known, see, e.g., [CZ97]:

Proposition 6 (Filtration lemma). *Suppose that Γ is a finite* Sub-*closed set of formulas and $\widehat{M}$ is a Γ-filtration of a model M. Then, for all points x in M and all formulas $\varphi \in \Gamma$, we have:*

$$M, x \models \varphi \textit{ iff } \widehat{M}, [x] \models \varphi.$$

For a bimodal formula φ, let $[\varphi]$ be the set of bimodal formulas that are substitution instances of φ (the axiom scheme).

Lemma 1. *Let $M = (X, R, D, \theta)$ be a bimodal model, $k < \omega$, $M \vDash [\chi^>_k]$, and let Γ be a finite* Sub-*closed set of bimodal formulas. Then for every finite $\Psi \supseteq \Gamma$, for every Γ-filtration $\widehat{M} = (X/{\sim_\Psi}, \widehat{R}, \widehat{D}, \widehat{\theta})$ of M, we have $\chi(X/{\sim_\Psi}, \widehat{R}) > k$.*

Remark 2. We do not make the assumption that (X, R, D) is a $\mathrm{K}_{\neq}$-frame or even that $M \vDash \mathrm{K}_{\neq}$. We also do not assume that $\chi(X, R) > k$: in general, $M \vDash [\chi^>_k]$ is a weaker condition.

Proof. Let $\widehat{X} = X/{\sim_\Psi}$. Since Ψ is finite, for every $A \in \widehat{X}$ there is a modal formula ψ_A such that

$$M, x \vDash \psi_A \text{ iff } x \in A. \tag{2}$$

Hence, for every $B \subseteq \widehat{X}$, for the formula $\varphi_B = \bigvee_{A \in B} \psi_A$ we have:

$$M, x \vDash \varphi_B \text{ iff } x \in \bigcup B. \tag{3}$$

We say that φ_B *defines* B.

Let $\mathcal{B}$ be a partition of $\widehat{X}$ and $|\mathcal{B}| = n \le k$. Then $\{\bigcup B : B \in \mathcal{B}\}$ is a partition of X. Let $\varphi_0, \dots \varphi_{n-1}$ be formulas that define elements of $\mathcal{B}$. For $n - 1 < i < k$, let $\varphi_i = \bot$. By (3), we have

$$M \vDash \forall \bigvee_{i<k} (\varphi_i \wedge \bigwedge_{i \neq j < k} \neg\varphi_j).$$

The result of substitution of φ_i's for p_i's in $\chi^>_k$ is true in M, so

$$M \vDash \exists \bigvee_{i<k} (\varphi_i \wedge \Diamond\varphi_i).$$

It follows from (3) that for some i, for some $x, y \in \bigcup B_i$ we have xRy. Let $[x]_\Psi$ denote the $\sim_\Psi$-class of x. We have $[x]_\Psi, [y]_\Psi \in B_i$. Since $\widehat{R}$ contains the minimal filtered relation, $[x]_\Psi \widehat{R} [y]_\Psi$. So $\mathcal{B}$ is not a proper partition of $(\widehat{X}, \widehat{R})$. □

Recall that the modal formula $p \to \Box\Diamond p$ expresses the symmetry of a binary relation. Let KB be the smallest unimodal logic containing this formula. It is well known that this logic is canonical.

Theorem 1. *For each $k < \omega$, the logics $\mathrm{K}_{\neq} + \chi_k^{>}$ and $\mathrm{KB}_{\neq} + \chi_k^{>}$ have the exponential finite model property and are decidable.*

Proof. Let $M_1 = (X_1, R_1, D_1, \theta_1)$ and $M_2 = (X_2, R_2, D_2, \theta_2)$ be the canonical models of the logics $\mathrm{K}_{\neq} + \chi_k^{>}$ and $\mathrm{KB}_{\neq} + \chi_k^{>}$, respectively. By Proposition 2, the canonical frames (X_1, R_1, D_1) and (X_2, R_2, D_2) validate the logic $\mathrm{K}_{\neq}$, and also R_2 is symmetric.

Let L be one of these logics, $\varphi \notin L$. Then φ is false at a point x in the canonical model of L. Let $M = (Y, R, D, \theta)$ be its submodel generated by x. By Proposition 3, for all $y, z \in Y$ we have:

$$\text{if } y \neq z, \text{ then } yDz. \tag{4}$$

Let $\Gamma = \mathrm{Sub}\,\varphi$, $\sim\, = \,\sim_\Gamma$. Put $\widehat{Y} = Y/{\sim}$, and consider the filtration $\widehat{M} = (\widehat{Y}, R_\sim, D_\sim, \widehat{\theta})$. Clearly, the size of $\widehat{Y}$ is bounded by $2^{\ell(\varphi)}$

By Filtration lemma (Proposition 6), φ is falsified in $\widehat{M}$. Let us show that the frame $(\widehat{Y}, R_\sim, D_\sim)$ validates L.

From (4), it follows that $(\widehat{Y}, R_\sim, D_\sim)$ validates the logic $K_{\neq}$. In the case of symmetric R, the minimal filtered relation $R_\sim$ is also symmetric. Finally, by Lemma 1, $\chi(\widehat{Y}, R_\sim) > k$. By Proposition 5, $(\widehat{Y}, R_\sim, D_\sim)$ validates L.

Hence L is complete with respect to its finite frames. □

Theorem 2. *Let $\mathcal{G}^{>k}$ be the class of graphs G such that $\chi(G) > k$, and let $\mathcal{D}^{>k}$ be the class of directed graphs G such that $\chi(G) > k$. Then $\mathrm{Log}\,\mathcal{G}_{\neq}^{>k} = \mathrm{KB}_{\neq} + \chi_k^{>}$, and $\mathrm{Log}\,\mathcal{D}_{\neq}^{>k} = \mathrm{K}_{\neq} + \chi_k^{>}$.*

Proof. By Theorem 1, the logics $\mathrm{K}_{\neq} + \chi_k^{>}$ and $\mathrm{KB}_{\neq} + \chi_k^{>}$ are complete with respect to their finite point-generated frames.

Consider a point-generated $\mathrm{K}_{\neq}$-frame $F = (X, R, D)$ and its repairing $F^{(\neq)} = (Y, S, \neq_Y)$. Recall that F is a p-morphic image of $F^{(\neq)}$. Let $\mathcal{A}$ be a partition of Y, $|\mathcal{A}| \leq k$. Consider the following partition $\mathcal{B}$ of X: $B \in \mathcal{B}$ iff there is $A \in \mathcal{A}$ such that $B = \{x : (x, 0) \in A\}$ and $B \neq \varnothing$.

Assume that $\chi(X, R) > k$. It follows that for some $B \in \mathcal{B}$ and some $x, y \in B$ we have xRy. Then for some $A \in \mathcal{A}$ we have $(x, 0), (y, 0) \in A$ and $(x, 0)S(y, 0)$. Thus, $\mathcal{A}$ is not a proper partition of (Y, S). Hence, $\chi(Y, S) > k$. This completes the proof in the directed case: $\mathrm{Log}\,\mathcal{D}_{\neq}^{>k} = \mathrm{K}_{\neq} + \chi_k^{>}$.

Clearly, if R is symmetric, then S is symmetric is well. This observation completes the proof in the non-directed case. □

Remark 3. These theorems can be extended for the case of graphs where the relation is irreflexive, if instead of the formula $\Diamond p \to \exists p$ in the definition of $L_{\neq}$ we use the formula $\Diamond p \to \langle\neq\rangle p$. Then in any frame (X, R, D) validating this version of $L_{\neq}$, the second relation contains R, and so if a point is R-reflexive, it is also D-reflexive. In this case, the repairing $F^{(\neq)}$ should be modified in the following way:

$$\begin{aligned} Y &= \{(x,0) : x \in X\} \cup \{(x,i) : x \in X \,\&\, xDx \,\&\, 0 < i \leq k\}, \\ (x,i)S(y,j) &\text{ iff } xRy \,\&\, ((x,i) \neq (y,j)). \end{aligned}$$

Then S is irreflexive, the map $(x,i) \mapsto x$ remains a p-morphism, and R-reflexive points in F become cliques of size $> k$. Also, it follows that $\chi(Y,S) > k$ whenever $\chi(X,R) > k$.

Remark 4. A related result was obtained very recently in [DLW23]: it was shown that in neighborhood semantics of modal language, the non-k-colorability of hypergraphs is expressible, and the resulting modal systems are decidable as well.[2]

4 Logics of Connected Graphs

A frame $F = (X, R)$ is *connected*, if for any points x, y in X, there are points $x_0 = x, x_1, \ldots, x_n = y$ such that for each $i < n$, $x_i R x_{i+1}$ or $x_{i+1} R x_i$.

Let Con be the following formula:

$$\exists p \wedge \exists \neg p \to \exists (p \wedge \Diamond \neg p). \tag{5}$$

Proposition 7. *Let $F = (X, R, D)$ be a point-generated* $\mathrm{KB}_{\neq}$*-frame. Then (X, R) is connected iff $F \vDash$* Con.

Proof. Assume that (X, R) is connected and M is a model on F such that $\exists p \wedge \exists \neg p$ is true (at some point) in M. Hence there are points x, y in M such that $M, x \vDash p$ and $M, y \vDash \neg p$. Then there are $x_0 = x, x_1, \ldots, x_n = y$ such that $x_i R x_{i+1}$ for each $i < n$. Let $k = \max\{i : M, x_i \vDash p\}$. Then $M, x_k \vDash p \wedge \Diamond \neg p$. Hence Con is valid in F.

Assume that (X, R) is not connected. Then there are x, y in X such that $(x, y) \notin R^*$, where R^* is the reflexive transitive closure of R. Put $\theta(p) = \{z : (x, z) \in R^*\}$s. In the model $M = (F, \theta)$, we have $M \vDash \exists p \wedge \exists \neg p$. On the other hand, at every point z in M we have $M, z \vDash p \to \Box p$, so the conclusion of Con is not true in M. So Con is not valid in F. □

In particular, it follows that for every graph G,

$$G \text{ is connected iff } G_{\neq} \vDash \text{Con}.$$

[2] I am grateful to Gillman Payette for sharing with me this reference after my talk at WoLLIC.

Remark 5. There are different ways to express connectedness in propositional modal languages [She90]. In particular, in the directed case, the connectedness can be expressed by the following modification of (5):

$$\exists p \wedge \exists \neg p \to \exists (p \wedge \Diamond \neg p) \vee \exists (\neg p \wedge \Diamond p);$$

Following the line of [She90], one can modally express the property of a graph to have at most n connected components for each finite n.

It is known that in many cases, adding axioms of connectedness preserves the finite model property [She90, GH18]. The following lemma shows that this is the case in our setting as well.

Lemma 2. *Assume that* (X, R, D) *is a point-generated* $\mathrm{KB}_{\neq}$*-frame. Let* $M = (X, R, D, \theta)$ *be a model such that* $M \vDash [\textsc{Con}]$*, and let* Γ *be a finite* Sub*-closed set of bimodal formulas. Then for every finite* $\Psi \supseteq \Gamma$*, for every* Γ*-filtration* $\widehat{M} = (X/{\sim_\Psi}, \widehat{R}, \widehat{D}, \widehat{\theta})$ *of* M*,* $(X/{\sim_\Psi}, \widehat{R})$ *is connected.*

Remark 6. Similarly to Lemma 1, connectedness of (X, R) does not follow from $M \vDash [\textsc{Con}]$.

Proof. Let $\widehat{X} = X/{\sim_\Psi}$, c the number of elements in $\widehat{X}$. We recursively define c distinct elements $A_0, \ldots, A_{c-1}$ of $\widehat{X}$, and auxiliary sets $\widehat{Y}_n = \{A_0, \ldots, A_n\}$, $\widehat{R}_n = \widehat{R} \cap (\widehat{Y}_n \times \widehat{Y}_n)$ for $n < c$ such that

$$\text{the restriction } (\widehat{Y}_n, \widehat{R}_n) \text{ of } (\widehat{X}, \widehat{R}) \text{ to } \widehat{Y}_n \text{ is connected.} \tag{6}$$

Let A_0 be any element of $\widehat{X}$. The frame $(\widehat{Y}_0, \widehat{R}_0)$ is connected, since it is a singleton.

Assume $0 < n < c$ and define A_n. By the same reasoning as in Lemma 1, there is a formula φ_n such that

$$M, x \vDash \varphi_n \text{ iff } x \in A_i \text{ for some } i < n. \tag{7}$$

The formula

$$\exists \varphi_n \wedge \exists \neg \varphi_n \to \exists (\varphi_n \wedge \Diamond \neg \varphi_n). \tag{8}$$

is a substitution instance of $\textsc{Con}$, so it is true in M. Let $V = \bigcup \widehat{Y}_{n-1}$. The set $\widehat{Y}_{n-1}$ has $n < c$ elements, so there are points x, y in X such that $x \in V$, and $y \notin V$. So $M, x \vDash \varphi_n$ and $M, y \vDash \neg \varphi_n$. By Proposition 3, the premise of (8) is true in M. Hence we have $M, z \vDash \varphi_n \wedge \Diamond \neg \varphi_n$ for some z in M. Then $z \in V$ and there exists u in $X \setminus V$ with zRu. Since $\widehat{R}$ contains the minimal filtered relation, $[z]_\Psi \widehat{R} [u]_\Psi$. We put $A_n = [u]_\Psi$. By the hypothesis (6), $(\widehat{Y}_{n-1}, \widehat{R}_{n-1})$ is connected, and so $(\widehat{Y}_n, \widehat{R}_n)$ is connected as well.

Finally, observe that $(\widehat{Y}_{c-1}, \widehat{R}_{c-1})$ is the frame $(\widehat{X}, \widehat{R})$. □

Theorem 3. *For each* $k < \omega$*, the logics* $\mathrm{KB}_{\neq} + \{\textsc{Con}, \Diamond \top\}$ *and* $\mathrm{KB}_{\neq} + \{\chi_k^{>}, \textsc{Con}, \Diamond \top\}$ *have the exponential finite model property and are decidable.*

Proof. Similar to the proof of Theorem 1. Let φ be a non-theorem of one these logics, M a point-generated submodel of the canonical model of the logic where φ is falsified. Consider the frame F of the minimal filtration of M via the subformulas of φ. We only need to check that F validates $\Diamond\top$ and Con (validity of other axioms was checked in the proof of Theorem 1). That $\Diamond\top$ is valid is trivial. The validity of Con follows from Lemma 2 and Proposition 7. □

Theorem 4. *Let $\mathcal{C}$ be the class of connected non-singleton graphs, $\mathcal{C}^{>k}$ the class of non-k-colorable graphs in $\mathcal{C}$. Then* $\mathrm{Log}\,\mathcal{C}_{\neq} = \mathrm{KB}_{\neq} + \{\mathrm{Con}, \Diamond\top\}$*, and* $\mathrm{Log}\,\mathcal{C}_{\neq}^{>k} = \mathrm{KB}_{\neq} + \{\chi_k^{>}, \mathrm{Con}, \Diamond\top\}$.

Proof. Similar to the proof of Theorem 2. Completeness of $\mathrm{KB}_{\neq} + \{\mathrm{Con}, \Diamond\top\}$ and $\mathrm{KB}_{\neq} + \{\chi_k^{>}, \mathrm{Con}, \Diamond\top\}$ with respect to their finite point-generated frames follows from Theorem 3.

Assume that $F = (X, R, D)$ is a point-generated $\mathrm{KB}_{\neq}$-frame, and (X, R) is connected and validates $\Diamond\top$. Consider the repairing $F^{(\neq)} = (Y, S, \neq_Y)$ of F. Clearly, $\Diamond\top$ is valid in $F^{(\neq)}$. Let (x, i) and (y, j) be in Y. First, assume that $x \neq y$. Since (X, R) is connected, there is a path between x and y in (X, R), which induces a path between (x, i) and (y, j) in (Y, S) by the definition of S. Now consider two distinct points (x, i) and (x, j) in Y. Since $\Diamond\top$ is valid in F, we have xRy for some y in F. Then we have $(x, i)S(y, 0)$ and $(x, j)S(y, 0)$. It follows that (Y, S) is connected and so $F^{(\neq)}$ validates Con by Proposition 7.

That other axioms hold in $(Y, S, \neq_Y)$ was shown in Theorem 2. Now the theorem follows from the fact that F is a p-morphic image of $F^{(\neq)}$. □

5 Corollaries

Lemmas 1 and 2 were stated in a more general way than it was required for the proofs of Theorems 1 and 3. The aim of using these, more technical, statements is the following.

Definition 3. A logic L *admits (rooted) definable filtration*, if for any (point-generated) model M with $M \vDash L$, and for any finite Sub-closed set of formulas Γ, there exists a finite model $\widehat{M}$ with $\widehat{M} \vDash L$ that is a definable Γ-filtration of M.

In [KSZ14,KSZ20], it was shown that if a modal logic L admits definable filtration, then its enrichments with modalities for the transitive closure and converse relations also admit definable filtration.

Notice that if $L = \mathrm{K}_2 + \varphi$, where K_2 is the smallest bimodal logic and φ is a bimodal formula, then $M \vDash L$ iff $M \vDash [\varphi]$. In particular, the logics $\mathrm{K}_2 + \chi_k^{>}$ admit definable filtration by Lemma 1. This fact immediately extends to any bimodal logic $L + \chi_k^{>}$, whenever L admits definable filtration.

Corollary 1. *If a bimodal logic L admits definable filtration, then all $L + \chi_k^{>}$ admit definable filtration, and consequently have the finite model property.*

Applying Lemmas 1 and 2 to the case of point-generated models, we obtain the following version of Theorems 1 and 3.

Corollary 2. *Assume that a bimodal logic L admits rooted definable filtration, $k < \omega$. Then $L + \chi_k^{>}$ has the finite model property. If also L extends* $\mathrm{KB}_{\neq}$*, then $L + \{\chi_k^{>}, \textsc{Con}\}$ has the finite model property.*

6 Discussion

We have shown that modal logics of different classes of non-k-colorable graphs are decidable. It is of definite interest to consider logics of certain graphs, for which the chromatic number is unknown.

Let $F = (\mathbb{R}^2, R_{=1})$ be the unit distance graph of the real plane. It is a long-standing open problem what is $\chi(F)$ (Hadwiger-Nelson problem). It is known that $5 \leq \chi(F) \leq 7$ [DG18, EI20].

Let $L_{=1}$ be the bimodal logic of the frame $(\mathbb{R}^2, R_{=1}, \neq_{\mathbb{R}^2})$. In modal terms, the problem asks whether $\chi_5^{>}, \chi_6^{>}$ belong to $L_{=1}$. We know that $L_{=1}$ extends $L = \mathrm{KB}_{\neq} + \{\chi_4^{>}, \textsc{Con}, \Diamond\top, \Diamond p \to \langle\neq\rangle p\}$ (it is an easy corollary of the above results that L is decidable). However, $L_{=1}$ contains extra formulas. For example, consider the formulas

$$\mathrm{P}(k,m,n) = \bigwedge_{i<k} \Diamond^m \Box^n p_i \to \bigvee_{i \neq j < k} \Diamond^m (p_i \wedge p_j).$$

For various k, m, n, $\mathrm{P}(k,m,n)$ is in $L_{=1}$ (and not in L); this can be obtained from known solutions for problems of packing equal circles in a circle.

Problem 1. Is $L_{=1}$ decidable? Finitely axiomatizable? Recursively enumerable? Does it have the finite model property?

Notice that instead of considering the difference auxiliary modality, one can consider the logic with the universal modality: this logic is a fragment of $L_{=1}$, but still can express formulas $\chi_k^{>}$.

Let $V_r \subseteq \mathbb{R}^2$ be a disk of radius r. It follows from de Bruijn-Erdős theorem, that if $\chi(F) > k$, then $\chi(V_r, R_{=1}) > k$ for some r.

Let $L_{=1,r}$ be the unimodal logic of the frame $(V_r, R_{=1})$. If $r > 1$, then the universal modality is expressible, and so are the formulas $\chi_k^{>}$. Hence, it is of interest to consider axiomatization problems and algorithmic problems for these logics.

Problem 2. To analyze the unimodal logics $L_{=1,r}$.

Acknowledgement. The author would like to thank the reviewers for their helpful comments on an earlier version of the paper.

References

[BdRV01] Blackburn, P., de Rijke, M., Venema, Y.: Modal Logic, Volume 53 of Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, Cambridge (2001)

[CZ97] Chagrov, A., Zakharyaschev, M.: Modal Logic, Volume 35 of Oxford Logic Guides. Oxford University Press, Oxford (1997)

[DG18] De Grey, A.D.N.J.: The chromatic number of the plane is at least 5. arXiv preprint arXiv:1804.02385 (2018)

[DLW23] Ding, Y., Liu, J., Wang, Y.: Someone knows that local reasoning on hypergraphs is a weakly aggregative modal logic. Synthese **201**(2), 46 (2023)

[dR92] de Rijke, M.: The modal logic of inequality. J. Symb. Log. **57**(2), 566–584 (1992)

[EI20] Exoo, G., Ismailescu, D.: The chromatic number of the plane is at least 5: a new proof. Discrete Comput. Geom. **64**(1), 216–226 (2020). https://doi.org/10.1007/s00454-019-00058-1

[Fin75] Fine, K.: Some connections between elementary and modal logic. In: Kanger, S. (ed.) Proceedings of the Third Scandinavian Logic Symposium, Volume 82 of Studies in Logic and the Foundations of Mathematics, pp. 15–31. Elsevier (1975)

[GG93] Gargov, G., Goranko, V.: Modal logic with names. J. Philos. Log. **22**(6), 607–636 (1993). https://doi.org/10.1007/BF01054038

[GH18] Goldblatt, R., Hodkinson, I.: The finite model property for logics with the tangle modality. Stud. Logica **106**(1), 131–166 (2018). https://doi.org/10.1007/s11225-017-9732-1

[GHV04] Goldblatt, R., Hodkinson, I., Venema, Y.: Erdös graphs resolve Fine's canonicity problem. Bull. Symb. Log. **10**(2), 186–208 (2004)

[Hug90] Hughes, G.E.: Every world can see a reflexive world. Stud. Logica Int. J. Symb. Log. **49**(2), 175–181 (1990). https://doi.org/10.1007/BF00935597

[KS14] Kudinov, A., Shehtman, V.: Derivational modal logics with the difference modality. In: Bezhanishvili, G. (ed.) Leo Esakia on Duality in Modal and Intuitionistic Logics. OCL, vol. 4, pp. 291–334. Springer, Dordrecht (2014). https://doi.org/10.1007/978-94-017-8860-1_11

[KSZ14] Kikot, S., Shapirovsky, I., Zolin, E.: Filtration safe operations on frames. In: Goré, R., Kooi, B.P., Kurucz, A. (eds.) Advances in Modal Logic, no. 10, pp. 333–352. College Publications (2014)

[KSZ20] Kikot, S., Shapirovsky, I., Zolin, E.: Modal logics with transitive closure: completeness, decidability, filtration. In: Olivetti, N., Verbrugge, R., Negri, S., Sandu, G. (eds.) 13th Conference on Advances in Modal Logic, AiML 2020, Helsinki, Finland, 24–28 August 2020, pp. 369–388. College Publications (2020)

[She90] Shehtman, V.: Derived sets in Euclidean spaces and modal logic. Technical report, ITLI Prepublication Series, University of Amsterdam, X-1990-05 (1990)

Subsumption-Linear Q-Resolution for QBF Theorem Proving

Allen Van Gelder(✉)

Computer Science Department, SOE–3, University of California, Santa Cruz, CA 95064, USA
avg@cs.ucsc.edu

Abstract. Subsumption-Linear Q-Resolution (SLQR) is introduced for proving theorems from Quantified Boolean Formulas. It is an adaptation of SL-Resolution, which applies to propositional and first-order logic. In turn SL-Resolution is closely related to model elimination and tableau methods. A major difference from QDPLL (DPLL adapted for QBF) is that QDPLL guesses variable assignments, while SLQR guesses clauses.

In prenex QBF (PCNF, all quantifier operations are outermost) a propositional formula D is called a nontrivial consequence of a QBF Ψ if Ψ is true (has at least one model) and D is true in every model of Ψ. Due to quantifiers, one cannot simply negate D and look for a refutation, as in propositional and first-order logic. Most previous work has addressed only the case that D is the empty clause, which can never be a nontrivial consequence.

This paper shows that SLQR with the operations of resolution on both existential and universal variables as well as universal reduction is inferentially complete for closed PCNF that are free of asymmetric tautologies; i.e., if D is logically implied by Ψ, there is a SLQR derivation of D from Ψ. A weaker form called SLQR–ures omits resolution on universal variables. It is shown that SLQR–ures is not inferentially complete, but is refutationally complete for closed PCNF.

1 Introduction

Theorem proving, i.e., showing that a given formula $\mathcal{F}$ logically implies another formula $\mathcal{G}$, is a fundamental task in any logic. We assume the reader is familiar with standard terminology of logic, as found in several texts [4,8]. Recent work on high-performance solvers for propositional formulas and quantified boolean formulas (QBFs) has focused on determining a given formula's *satisfiability*, or *truth value*. For propositional formulas this emphasis is partly justified by the fact that $\mathcal{F}$ logically implies $\mathcal{G}$ if and only if $(\mathcal{F} \wedge \neg\mathcal{G})$ is *unsatisfiable*. The QBF analogy of this simple relationship does not hold. That is, $\overrightarrow{Q} \cdot (\mathcal{F} \wedge \neg\mathcal{G})$ may be false but $\overrightarrow{Q} \cdot \mathcal{F}$ does not logically imply $\overrightarrow{Q} \cdot \mathcal{G}$, where QBF logical implication is defined in Definition 1.1.

Definition 1.1. Let $\Phi = \overrightarrow{Q}.\mathcal{F}$ be a closed QBF; that is, $\overrightarrow{Q}$ is the quantifier prenex, $\mathcal{F}$ is a propositional formula, and every variable in $\mathcal{F}$ appears in the

H. H. Hansen et al. (Eds.): WoLLIC 2023, LNCS 13923, pp. 362–376, 2023.
https://doi.org/10.1007/978-3-031-39784-4_23

prenex. We say that a propositional formula D on the same set of variables as $\mathcal{F}$ is a ***logical consequence*** of $\mathcal{F}$, written $\mathcal{F} \models D$, if D is true in every model of $\mathcal{F}$. We say that a propositional formula D on the same set of variables as $\mathcal{F}$ is a ***nontrivial consequence*** of Φ if Φ is true (has at least one model tree) and $\mathcal{F} \models D$. □

Due to quantifiers, one cannot simply negate D and look for a refutation, as in propositional and first-order logic. Most previous work has addressed only the case that D is an empty clause [3,17], which can never be a *nontrivial* consequence. We say a proof system is *inferentially complete* if any Ψ that is logically implied by Φ can be proven from Φ in the proof system; in other words all logical consequences of Φ are provable in the proof system.

All propositional formulas have a logically equivalent formula in *conjunctive normal form* (CNF), i.e., as a set of conjunctively joined *clauses* that are themselves disjunctively joined sets of literals. Propositional resolution is essentially inferentially complete for propositional CNF; technically, *clausal subsumption* is also needed in case a clause derived from $\mathcal{F}$ by resolution properly subsumes a clause that is a logical consequence of $\mathcal{F}$.

Alls QBFs have a logically equivalent formula in *prenex conjunctive normal form* (PCNF), i.e., all quantifiers are outermost operations and the remaining propositional formula, commonly called the *matrix*, is expressed in CNF. A QBF is said to be *closed* if every variable is quantified.

A further technical condition is important for inferential completeness: A QBF is said to be *AT-free* if it contains no *asymmetric tautologies*, as defined and studied by Heule *et al.* [6]. QBFs translated from applications are normally AT-free, but certain preprocessing operations might introduce asymmetric tautologies.

Although the precise definition is quite technical, a simple example of asymmetric tautology is a set of clauses in which a certain variable, say x is accompanied by some other variable, say y with the same polarity as x in each clause containing x. The variable y may occur in some other clauses, as well. All resolutions with x as the clashing variable are tautologous. Please see the cited paper for further details.

Although it is known that QU-resolution is inferentially complete for closed AT-free PCNF [20], we are not aware of any *implemented* QBF proof system with this property.

For propositional CNF formulas a *model* is a partial assignment that satisfies every clause. For closed PCNF formulas with k universal variables a *model* is a *set* of 2^k prefix-ordered total assignments that comprises a *strategy for the E-player*, such that each assignment in the set has a different assignment to the universal variables and satisfies every clause in the matrix [16,20,21]. The term *model tree* is often used to emphasize the structural constraints. (See Definition 2.4 for structural details). For both logics Φ logically implies Ψ if and only if *every* model of Φ is also a model of Ψ. The additional complexity of model trees compared to a single assignment explains why many theorem-proving ideas do not transfer easily from CNF to PCNF.

This paper introduces Subsumption-Linear Q-Resolution (SLQR) for ***proving theorems*** from Quantified Boolean Formulas in PCNF. SLQR is an adaptation of SL-Resolution, which applies to propositional and first-order logic.

A major difference between SLQR and QDPLL (DPLL adapted for QBF) is that QDPLL guesses and backtracks on *variable assignments*, while SLQR guesses and backtracks on *clauses*. The inferential power of SLQR is compared with other Q-Resolution and tableau strategies.

A primary motivation for the SLQR discipline is to reduce the search space compared to ad-hoc heuristics for choosing the next resolution operation. Several optimizations reduce the choices while preserving completeness:

1. One operand of every resolution operation is the immediately preceding derived clause (linearity).
2. When a clashing literal needs to be chosen in the first operand, there are restrictions on which literals need to be considered, and once a literal that meets those restrictions has been chosen, alternative choices of clashing literal need not be considered.
3. When backtrackable choices are made for the second clause operand, logical analysis is used to rule out many unnecessary choices.

SL-Resolution is closely related to model elimination [2,12,13,15,18] and tableau methods [11]. Discussions and thorough bibliographies may be found in several texts [8,14]. Terminology varies among these sources.

Letz adapted a tableau-oriented point of view for QBF solving [10]. However, his solver `Semprop` branches on variables, similarly to QDPLL solvers such as `depQBF`, `QuBE`, and others.

After introducing and analyzing needed technical machinery for *prefix-ordered QU-resolution* in Sect. 2 this paper introduces *Subsumption-Linear Q-Resolution* (SLQR) in Sect. 4, including the special operation *ancestor resolution*, and proves that SLQR is inferentially complete.

2 Preliminaries

In their most general form, *quantified boolean formulas* (QBFs) generalize propositional formulas by adding universal and existential quantification of boolean variables (often abbreviated to "variables"). A *quantified variable* is denoted by $\forall\, u$ (variable u is universal) or $\exists\, e$ (variable e is existential). A *literal* is a variable or a negated variable. See [8] for a thorough introduction and a review of any unfamiliar terminology.

Definition 2.1. The ***truth value*** of a *closed* QBF is either 0 (*false*) or 1 (*true*), as defined by induction on its principal operator.

1. $(\exists\, e\, \Psi(e)) = 1$ iff $(\Psi(0) = 1$ or $\Psi(1) = 1)$.
2. $(\forall\, u\, \Psi(u)) = 0$ iff $(\Psi(0) = 0$ or $\Psi(1) = 0)$.
3. Other operators have the same semantics as in propositional logic.

This definition emphasizes the connection of QBF to two-person games, in which player E (Existential) tries to set existential variables to make the QBF evaluate to 1, and player A (Universal) tries to set universal variables to make the QBF evaluate to 0 (see [9] for more details). □

Definition 2.2. For this paper QBFs are in ***prenex conjunctive normal form*** **(PCNF)**, and are *closed*; i.e., $\Psi = \overrightarrow{Q}\,.\,\mathcal{F}$ consists of a quantifier prefix $\overrightarrow{Q}$ and a set of quantifier-free clauses $\mathcal{F}$ (often called the *matrix*) such that every variable in $\mathcal{F}$ occurs in $\overrightarrow{Q}$. The number of clauses in $\mathcal{F}$ is denoted by $|\mathcal{F}|$. In the context of a matrix, clauses are understood to be combined conjunctively.

A ***clause*** is a disjunctively connected set of literals. A clause is called *tautologous* if it contains some literal and its complement; otherwise it is called *non-tautologous*. Clauses are frequently written enclosed in square brackets (e.g., $[p, q, \overline{r}]$) and $[]$ denotes the empty clause.

We follow certain notational conventions for boolean variables and literals (signed variables) to make reading easier: Lowercase letters near the ***beginning*** of the alphabet (e.g., b, c, d, e) denote existential literals, while lowercase letters near the ***end*** of the alphabet (e.g., u, v, w, x) denote universal literals, while ***middle*** letters (e.g., p, q, r) are of unspecified quantifier type. Quantifier types are implied frequently throughout the paper without restating this convention.

In contexts where a literal is expected, p might denote a positive or negative literal, while $\overline{p}$ denotes the negation of p. To emphasize that p stands for a ***variable***, rather than a *literal*, the notation $|p|$ is used. Clauses may be written as $[p, q, \overline{r}]$); $[]$ denotes the empty clause.

For set-combining operations on clauses, besides $\cup$ for union and $\cap$ for intersection, we use $+$ for ***disjoint union***, $-$ for ***set difference***, and write p instead of $[p]$ when it is an operand for one of these operations. Thus $C + p$ adds p to a clause that does not already contain p, while $C - p$ removes p from a clause that might or might not contain p.

The symbols α, β, and γ denote (possibly empty) sequences of literals or sets of literals, depending on context; $vars(\alpha)$ denotes the set of variables underlying the literals of α. (Because a clause is a set, a notation like $[p, \alpha]$ implicitly specifies that p is not in α.) The symbol $\perp$ is sometimes used as a literal denoting *false* and is treated as being outer to all other literals. □

Definition 2.3. The ***quantifier prefix*** (often shortened to ***prenex***) is a sequence of quantified variables. A variable closer to the beginning (end) of the sequence is said to be ***outer*** (***inner***) to another variable. The prenex is partitioned into ***quantifier blocks*** (abbreviated to ***qblocks***). Each quantifier block is a maximal consecutive subsequence of the prenex with variables with the same quantifier type, and has a unique ***quantifier depth***, denoted as *qdepth*, with the outermost qblock having $qdepth = 1$. The notation $p \prec q$ means that p is in a qblock outer to the qblock of q. The notation $p \preceq q$ means that p is the same qblock as q or $p \prec q$. There is no special notation for p and q being in the same qblock. The notation is extended to sets of variables or literals in the obvious ways; e.g., $P \prec q$ means that each $p \in P$ satisfies $p \prec q$. In situations where variables within a quantifier block are considered to have a fixed order, $p \prec\!\prec q$ means: p precedes q in the same quantifier block or $p \prec q$.

A few special operations on sets of literals are defined. A prenex $\overrightarrow{Q}$ is assumed to be known by the context. For a set S of literals:

$$\mathsf{exist}(S) = \{\text{the existential literals in } S\} \tag{1}$$

$$\mathsf{univ}(S) = \{\text{the universal literals in } S\} \tag{2}$$

$$(S \prec q) = \{\text{the literals in } S \text{ outer to } q\} \tag{3}$$

$$(q \prec S) = \{\text{the literals in } S \text{ inner to } q\} \tag{4}$$

Depending on context, the set of literals might be a clause, a prenex, a partial assignment, or other logical expression. □

Definition 2.4. Let a closed PCNF $\Psi = \overrightarrow{Q}.\mathcal{F}$ be given. Let $\mathbf{V}$ denote the variables of Ψ. A ***QBF strategy*** for Ψ is a set of boolean functions $\{p_j(\beta_j)\}$, where p_j ranges over the variables of one quantifier type and β_j consists of all variables q of the *opposite quantifier type* such that $q \prec p_j$. The function $p_j(\beta_j)$ is called a *Skolem function* if p_j is existential and is called an *Herbrand function* if p_j is universal. For Skolem functions $\beta_j = \mathsf{univ}(\mathbf{V}) \prec p_j$; for Herbrand functions $\beta_j = \mathsf{exist}(\mathbf{V}) \prec p_j$.

A ***winning strategy*** for player E is a QBF strategy in which p_j ranges over the *existential* variables such that $\mathcal{F}$ always evaluates to 1 if player E always chooses $p_j = p_j(\beta_j)$ when p_j is the outermost unassigned variable in the two-person game mentioned in Definition 2.1. A ***winning strategy*** for player A is a QBF strategy in which p_j ranges over the *universal* variables such that $\mathcal{F}$ always evaluates to 0 if player A always chooses $p_j = p_j(\beta_j)$ when p_j is the outermost unassigned variable in the same game. Exactly one of the players has a winning strategy. Winning strategies can be generalized to closed QBFs that are not in prenex conjunctive normal form, whose variables may have only a *partial* order [9].

A clause D is said to be ***logically implied*** by Ψ if $\overrightarrow{Q}.(\mathcal{F} \cup \{D\})$ has the same set of winning strategies for player E as does Ψ. The term ***logical consequence*** is also used. In this case, D is said to be a ***strategy-sound inference*** from Ψ, following [21]. As a less stringent requirement, a clause D is said to be a ***safe inference*** from Ψ if $\overrightarrow{Q}.(\mathcal{F} \cup \{D\})$ has the same truth value as Ψ (i.e., adding D does not change the set of winning strategies for player E from nonempty to empty).

Dually, deletion of a clause D from Ψ is said to be a ***strategy-sound operation*** if $\overrightarrow{Q}.(\mathcal{F} - \{D\})$ has the same set of winning strategies for player E as does Ψ. A clause deletion is said to be a ***safe operation*** if $\overrightarrow{Q}.(\mathcal{F} - \{D\})$ has the same truth value as Ψ (i.e., deleting D does not change the set of winning strategies for player E from empty to nonempty). □

Definition 2.5. The proof system known as *Q-resolution* consists of two operations, *resolution* and *universal reduction*. Resolution is defined as usual, except that the clashing literal is always existential; resolvents must be non-tautologous for Q-resolution. Universal reduction is special to QBF.

$$\mathsf{res}_e(C_1, C_2) = \alpha \cup \beta \qquad \text{where } C_1 = [\overline{e}, \alpha]\,,\; C_2 = [e,\, \beta] \tag{5}$$

$$\mathsf{unrd}_u(C_3) = \gamma \qquad \text{where } C_3 = [\gamma,\, u]\,. \tag{6}$$

$\mathsf{unrd}_u(C_3)$ is defined only if u is ***tailing*** for γ, which means that the *qdepth* of u is greater than that of any existential literal in γ, i.e., $(u \prec \mathsf{exist}(\gamma)) = \emptyset$.

A clause is *fully reduced* if no universal reductions on it are possible. The fully reduced form of C is denoted as $\mathsf{unrd}_*(C)$. For this paper all clauses in given PCNFs are assumed to be fully reduced and non-tautological, unless stated otherwise. □

Q-resolution is of central importance for PCNFs because it is a strategy-sound and refutationally complete proof system [7,8], as restated in Theorem 2.6 below. Recall that a clause-based proof system is *refutationally complete* if the empty clause can be derived from every formula whose truth value is 0.

Theorem 2.6 [7]**.** Let the closed PCNF $\Psi = \overrightarrow{Q}.\mathcal{F}$ be given. Ψ evaluates to *false* if and only if [] can be derived from Ψ by Q-resolution.

We say that a proof system is *inferentially complete* if whenever D is logically implied (see Definition 2.4), then some subset of D can be derived in the proof system. Note that Q-resolution is not inferentially complete. A simple example is

$$\forall u\, \exists e\, \exists f.\, \{[u,\, e]\,,\, [\overline{u},\, f]\}\,.$$

Nothing can be derived by Q-resolution, but the clause $[e,\, f]$ is logically implied, which can be seen by enumerating all the winning strategies $\{e(u),\, f(u)\}$ and observing that $[e,\, f]$ evaluates to 1 for all values of u in each strategy.

The proof system known as QU-resolution is Q-resolution with the added operation of resolution on universal variables. QU-resolution is inferentially complete for closed PCNF and is able to provide exponentially shorter refutations for certain QBF families [20]. However, the challenge for using QU-resolution in practice is knowing when universal resolution is likely to be productive.

Definition 2.7. A ***QU-derivation*** or ***Q-derivation***, often denoted as Π or Γ or Σ, is a rooted directed acyclic graph (DAG) in which each vertex is either an original clause (a DAG leaf), or a proof operation (an internal vertex). A *Q(U)-refutation* is a Q(U)-derivation of the empty clause. This paper follows the convention that DAG edges are directed *away from* the root. A Q(U)-derivation is *tree-like* if every internal vertex has only one incoming edge, except that the root has no incoming edge.

A *subderivation* of a Q(U)-derivation Π is any rooted sub-DAG of Π whose vertices consist of some root vertex V and all DAG vertices of Π reachable from V and whose edges are the induced edges for this vertex set.

In a proof DAG, each internal vertex is represented as a tuple with fields consisting of:

– a specified operation type (resolution or universal reduction or "copy"),
– a specified clashing literal or universal-reduction literal (null for "copy"),

– one or two directed edge(s) to its operand(s),
– a derived clause.

(See Fig. 1.) The same tuple may be used to represent a leaf, in which case the operation type is “leaf”, the clashing literal is null, there are no outgoing edges, and the clause is an original clause. When there is no confusion, a vertex may be referred to by its clause; however, the same clause may appear in more than one vertex.

The “copy” just transfers the same clause to another vertex, and is included for technical reasons. A DAG containing copy operations (and correctly derived clauses) is called a ***generalized derivation***. The copy operations can be “spliced out” in the obvious manner to produce a derivation: If V contains a copy operation, replace all incoming edges to V by edges to the child of V. See [19] for details on propositional derivations. The QBF variant is developed in [5].

In the normal case of a resolution operation, the first, or left, edge goes to a vertex whose clause contains the *negation* of the clashing literal, and the second, or right, edge goes to a vertex whose clause contains the clashing literal. In any case, the union of the two operand clauses may not contain any complementary pair of literals other than the clashing literals.

We say that a literal q *has a proof operation* at the (internal) DAG vertex V if q or $\overline{q}$ is the literal specified in V; we say that a literal q *has a proof operation* in Π if q has a proof operation at some DAG vertex in Π.

For a proof DAG Π, $\mathsf{root}(\Pi)$ is the clause at the root, $\mathsf{leaves}(\Pi)$ is the *set* of clauses in the leaves, and

$$\mathsf{support}(\Pi) = \overrightarrow{Q}'.\,\mathsf{leaves}(\Pi), \tag{7}$$

where $\overrightarrow{Q}'$ is the subsequence of $\overrightarrow{Q}$ that contains only the variables that appear in $\mathsf{leaves}(\Pi)$. □

Definition 2.8. An ***assignment*** is a partial function from variables to truth values, and is usually represented as the set of literals that it maps to *true*. Assignments are denoted by ρ, σ, τ, etc. A *total assignment* assigns a truth value to every variable.

Application of an assignment σ to a logical expression, followed by truth-value simplifications,[1] is called a ***restriction***. Restrictions are denoted by $q\lceil_\sigma$, $C\lceil_\sigma$, $\mathcal{F}\lceil_\sigma$, etc. If σ assigns variables that are quantified in Ψ, those quantifiers are deleted in $\Psi\lceil_\sigma$, and their variables receive the assignment specified by σ. □

3 Prefix-Ordered QU-Resolution

This section examines the restriction on QU-resolution derivations to be prefix-ordered, as defined below. The main result of this section is Lemma 3.6, which

[1] I.e., simplifications where one operand is 0 or 1.

concludes that prefix-ordered QU-resolution is inferentially complete. This is a stepping stone to the main results of the paper about SLQ resolution in Sect. 4.

In analogy with *regular propositional resolution* as defined by Kleine Büning and Lettmann [8], who cite Tseitin's classical paper, we define regularity for QU-resolution derivations. Definition 3.1 is more precise than one that is often seen, which specifies that no variable has more than one proof operation on any path in Γ. The two definitions are equivalent for refutations, but not for derivations in general.

For example, the four propositional clauses $[b, \neg e]$ $[e, \neg c]$ $[c, \neg d]$ $[b, e]$ derive $[d, e]$, but the derivation should not be called regular because a proof operation on e is needed..

Definition 3.1. A QU-resolution derivation Γ is said to be ***regular in*** p if no derived clause D that contains $|p|$ has a proof operation on $|p|$ on some path in Γ from D to a leaf. A QU-resolution derivation Γ is said to be ***regular*** if it is regular in p for all variables $|p|$ that have proof operations in Γ. □

Definition 3.2. We define QU-resolution to be ***prefix-ordered*** if the literals that have proof operations appear in the quantifier-prefix order on every path in the proof DAG, with the outermost closest to the root. □

A prefix-ordered QU-refutation is necessarily regular, but other prefix-ordered QU-derivations are not necessarily regular. The ensuing material requires some technical terminology, defined next.

Definition 3.3. A clause C ***subsumes*** clause D if the literals of C comprise a subset of the literals of D or if D is tautologous. Subsumption is *proper* if the subset is proper. In this sense, any tautologous clause is treated as containing every possible literal and is properly subsumed by any non-tautologous clause.

Minimality of clauses and sets of clauses is important in the technical material. A set of clauses is ***minimal*** under stated conditions if no proper subset of its clauses satisfies all of the conditions. Minimality of the set does not require minimum cardinality.

A clause C is ***QU-minimal*** for a PCNF Ψ if it is derivable from Ψ by QU-resolution and no proper subset of $\mathsf{unrd}_*(C)$ is derivable from Ψ by QU-resolution. A clause C is ***Q-minimal*** for a PCNF Ψ if it is derivable from Ψ by Q-resolution and no proper subset of $\mathsf{unrd}_*(C)$ is derivable from Ψ by Q-resolution.

Q-minimality of C does not require minimum cardinality; that is, some other clause E such that $|E| < |C|$ may be derivable by Q-resolution, provided that E does not properly subsume $\mathsf{unrd}_*(C)$. The same holds for QU-minimality. □

Definition 3.4. A QU-derivation Π is ***QU-irreducible*** if:

1. The clause derived in $\mathsf{root}(\Pi)$, say D, is QU-minimal for $\mathsf{support}(\Pi)$,
2. $\mathsf{leaves}(\Pi)$ is minimal for the QU-derivation of D from $\mathsf{support}(\Pi)$,
3. Π contains no proof operations on variables in D,
4. all proper subderivations of Π are QU-irreducible.

Ψ	$\exists d$	$\exists e$	$\forall u$	$\exists f$
C_1		e	u	$\overline{f}$
C_2	d			f
C_3	$\overline{d}$	$\overline{e}$		f

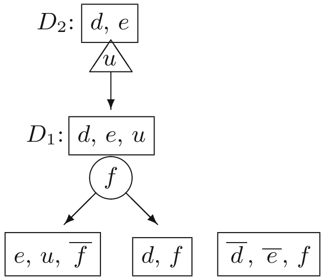


Fig. 1. PCNF Ψ in chart form (left) and proof DAG (right) for Example 3.5. Circles enclose the clashing literal for resolution; triangles denote universal reduction. C_3 is not part of the DAG rooted at D_2, but is its own trivial DAG.

Note that this definition does not require that the set of *DAG vertices* is minimal. In particular, every QU-irreducible derivation has a tree-like version.

Q-irreducible derivations are defined analogously. □

Example 3.5. To illustrate Q-minimality and QU-minimality, consider the formula Ψ, shown in Fig. 1 as a clause-literal incidence graph (chart form for short).

No universal resolutions are possible so Q and QU properties are the same. Let:

$$\begin{array}{l|l} D_1 = \mathsf{res}_f(C_1, C_2) = [d, e, u] & \Pi_1 = \text{the subderivation whose root is } D_1 \\ D_2 = \mathsf{unrd}_u(D_1) = [d, e] & \Pi_2 = \text{the derivation of } D_2 \\ & \Pi_3 = \text{the zero-step derivation of } C_3. \end{array}$$

Then D_1 is Q-minimal for Ψ even though D_2 is a proper subset, because the difference is only tailing universal literals. Also, $[d, f]$ is Q-derivable and narrower than D_1, but it is not a subset of D_1.

However, C_3 is not Q-minimal for Ψ even though it is an original clause, because $\mathsf{res}_d(C_3, C_2)$ is a proper subset of $\mathsf{unrd}_*(C_3)$. But the trivial subderivation Π_3 *is* Q-irreducible, because $\mathsf{leaves}(\Pi_3) = \{C_3\}$. □

To see that points 1 and 2 of Definition 3.4 are consistent, add a new "indicator" literal a_j to each clause $C_j \in \mathcal{F}$, the matrix. Replace the clause to be derived by $\left[D, \bigvee_j a_j\right]$. Then points 1 and 2 are both true if and only if $\left[D, \bigvee_j a_j\right]$ is Q-minimal for the modified clauses.

We need the following Lemma 3.6 for analyzing SLQR. QU-minimal clauses and minimal sets of clauses are important in the ensuing material. Recall the terminology in Definition 3.3 and Definition 3.4.

Lemma 3.6. Let the closed PCNF $\Psi = \overrightarrow{Q}.\mathcal{F}$ be given. By convention, every clause in $\mathcal{F}$ is non-tautological and fully reduced. Let clause D be QU-minimal for Ψ. Then D can be derived from Ψ by a QU-derivation Γ such that Γ is prefix-ordered, regular, tree-like and QU-irreducible.

Proof: Let $\mathcal{G} \subseteq \mathcal{F}$ be any subset such that D is not logically implied by any proper subset of $\mathcal{G}$. Let $\Phi = \overrightarrow{Q}.\mathcal{G}$. Then D is also QU-minimal for Φ. The proof of inferential completeness of QU-resolution in [20, Th. 5.4] constructs a QU-derivation of D from Φ with the required properties, and this is also a QU-derivation from Ψ. The cited theorem promises to derive $D^{(-)}$ but by minimality of D it must derive D exactly. ■

4 Subsumption-Linear Q-Resolution

This section defines subsumption-linear Q-resolution (SLQR) derivations and derives the main results of the paper.. We show that SLQR has the same inferential power as full QU-resolution; i.e., SLQR is inferentially complete for AT-free PCNF formulas. As mentioned in Sect. 1, a QBF is said to be *AT-free* if it contains no *asymmetric tautologies* [6]. QBFs translated from applications are normally AT-free, but certain preprocessing operations might introduce asymmetric tautologies.

We also define a weaker variant SLQR–ures that does not include resolution on clashing universal literals, and show that SLQR–ures has the same inferential power as full Q-resolution when all literals in the derived clause are outermost. Hence SLQR–ures is refutationally complete. Lemma 3.6 is an important stepping stone. We also show a PCNF and a Q-derivable clause for which there is no SLQR–ures derivation.

Definition 4.1. Given a QBF $\Phi = \overrightarrow{Q}.\mathcal{F}$ and a target clause T, a ***subsumption-linear Q-resolution* (SLQR))** derivation of T consists of a *top clause* $D_0 \in \mathcal{F}$ and a sequence of $m \geq 0$ derivation steps with $D_m = T$ of the form

$$D_i = \begin{cases} \mathsf{res}_{p(i)}(D_{i-1}, C_i) & \text{where } p(i) \text{ is any literal and } 1 \leq i \leq m \\ \mathsf{unrd}_{u(i)}(D_{i-1}) & \text{where } u(i) \text{ is universal and } 2 \leq i \leq m \end{cases} \quad (8)$$

such that each C_i is either a clause in $\mathcal{F}$ or is an earlier derived clause D_j that meets the precise criteria given below and is called an *ancestor clause.*

The D_i are called ***center clauses***. The C_i are called ***side clauses***. The literals of a side clause C_i are categorized as follows: $p(i)$ is the ***clashing*** literal; if C_i is derived, $p(i)$ is also called an ***ancestor*** literal; $q \in C_i$ is a ***target*** literal if $q \in T$; $q \in C_i$ is a ***merge*** literal if $q \in D_{i-1}$ and q is not a target literal; $q \in C_i$ is an ***extension*** literal if $q \in D_i$ and q is not in any of the preceding categories.

At the step where D_i is to be derived let D_j $(j \leq i-2)$ be an earlier derived clause and let $q \in D_j$ be the clashing literal for the derivation of D_{j+1}. Then D_j is defined to be an ***ancestor clause*** at this step in the proof if $D_j - \{q\}$ is a proper subset of each subsequently derived clause $D_{j+1}, \ldots, D_{i-1}$. If $q = \overline{p(i)}$ (the clashing literal in D_{i-1}), then the resolution of D_{i-1} and D_j is called ***ancestor resolution***, $\overline{p(i)}$ is called the ***ancestor literal***, and D_i consists of all literals in D_{i-1} except $\overline{p(i)}$. The word "subsumption" in the name "SLQR" is explained by the last relationship. If ancestor resolution is possible, other choices for side clause can be disregarded.

If D_j is an ancestor clause but $q \neq \overline{p(i)}$, q still plays a role as an ancestor literal: Some original clause must be chosen to resolve with D_{i-1}. If any *extension literal* of this resolution would be q, then this clause is inadmissible as a side clause at this step. A derivation that adheres to this policy (and also disallows derivation of tautologous clauses) is called ***tight*** [14]. □

Considering SLQR as a proof search system, the procedure to extend D_{i-1} to D_i consists of selecting a literal in D_{i-1} for the proof operation, and if the operation is resolution, selecting a side clause. It is known from antiquity [1] that propositional SL-resolution is complete for any literal-selection policy; i.e., it is not necessary to backtrack on the selected literal and try other selections. For simplicity and attention to implementation concerns, we consider only the LIFO policy for SLQR, defined next.

Definition 4.2. Given a QBF $\Phi = \overrightarrow{Q}.\mathcal{F}$ and a target clause T, the ***LIFO selection policy***, also called the *most recently introduced policy* is defined informally as follows. In a SLQR derivation, assume that each center clause D_{i-1} is represented by a last-in, first-out stack (LIFO) of its literals that are not in T, called the ***L-stack***, as well as a separate set of literals that are in T, which we call the ***T-subset***.

The L-stack is partitioned into contiguous sections such that all literals in a given section were introduced into a center clause D_j, $j \leq i-1$, as extension literals in the earlier resolution operation that derived D_j, and these literals are in quantifier order within the section with the innermost closest to the top of the L-stack. Further, this section has been intact for all center clauses between D_j and D_{i-1}. The L-stack as a whole may not be quantifier ordered. The *LIFO selection policy* selects the literal on top of the L-stack of the current center clause, say D_{i-1}. □

Whatever proof operation derives D_i, the selected literal will not be in D_i, so the L-stack of D_i may be formed by starting with that of D_{i-1}, popping the selected literal, and then possibly pushing a new section on top consisting of extension literals. A SLQR derivation develops by working on the section on top of the current L-stack until the current L-stack is empty. Readers familiar with Prolog will recognize the similarity to how the Prolog interpreter works.

4.1 Derivation Power of SLQR

This section investigates when a QU-derivable clause T also has a SLQR derivation. For propositional resolution, it is well known that the answer is essentially "always".[2] The situation for closed PCNF is not so simple.

[2] If the derived clause is not minimal, propositional resolution may derive a subsuming clause.

The proof of the next theorem employs the framework first published by Anderson and Bledsoe [1]. Minimal clauses and minimal sets of clauses are important in the ensuing material. Recall the terminology in Definition 3.3.

Theorem 4.3. Given a closed PCNF $\Psi = \overrightarrow{Q}\,.\,\mathcal{F}$, let T be a minimal clause such that there is a QU-resolution derivation of T from Ψ, call it Π, and no proper subset of $\mathcal{F}$ permits derivation of T. Then for every clause $C_0 \in \mathcal{F}$ and for the LIFO selection function (see Definition 4.2) there exists a SLQR derivation of T from Ψ whose top clause is C_0. Further, for each literal $q \in T$, q has no proof operation in the SLQR derivation.

Proof: Please see https://users.soe.ucsc.edu/~avg/Papers/slqr-long.pdf. ■

4.2 Derivation Power of SLQR–ures

SLQR–ures is the same as SLQR except that resolution on clashing universal literals is not permitted. This section investigates when a Q-derivable clause T also has a SLQR–ures derivation. For propositional resolution, it is well known that the answer is essentially "always," and this is just a special case of Theorem 4.4 below.[3] The situation for closed PCNF is not so simple.

The proof of the next theorem employs the framework first published by Anderson and Bledsoe [1]. Minimal clauses and minimal sets of clauses are important in the ensuing material. Recall the terminology in Definition 3.3.

Theorem 4.4. Given a closed PCNF $\Psi = \overrightarrow{Q}\,.\,\mathcal{F}$, let T be a minimal clause such that there is a Q-resolution derivation of T from Ψ, call it Π, and no proper subset of $\mathcal{F}$ permits derivation of T by Q-resolution. Further, let the literals of T be outermost among the literals of $\mathcal{F}$. Then for every clause $C_0 \in \mathcal{F}$ there exists a SLQR–ures derivation of T from Ψ whose top clause is C_0. Further, for each literal $q \in T$, q has no proof operation in the SLQR–ures derivation. In particular, SLQR–ures is refutationally complete for closed PCNF.

Proof: The proof is similar to that of Theorem 4.3 and is omitted. The hypothesis that T is outer to all literals with proof operations ensures that whenever a universal literal is the selected literal universal reduction is available, so universal resolution is not needed. Refutational completeness follows by letting $T = []$. ■

The preceding Theorem 4.4 shows that SLQR–ures has the full inferential power of Q-resolution for a very restricted set of derived clauses.

In fact, there are important clauses that can be derived by prefix-ordered tree-like Q-resolution, but not by SLQR–ures.

Theorem 4.5. There exists a closed PCNF such that the clause $[u, h]$ is derivable by prefix-ordered tree-like Q-resolution and not by SLQR–ures, u is universal and outermost, h is existential and innermost, $[u, h]$ is minimal, and the matrix is minimal.

Proof: Please see https://users.soe.ucsc.edu/~avg/Papers/slqr-long.pdf. ■

[3] If the derived clause is not minimal, propositional resolution may derive a subsuming clause.

4.3 Details for LIFO SLQR

Definition 4.6. The details of updating the stack are important, and some helpful terminology is now introduced. Proof operations are classified as follows:

1. *Reduction operation*: a universal reduction on a universal literal;
2. *Extension operation*: a resolution that introduces at least one literal not in the U-set or in the E-stack of the current center clause;
3. *Contraction operation*: a resolution that introduces no literals into the U-set or the E-stack, but possibly adds some literals to the T-subset.

For a resolution operation, the literals in the side clause are classified as follows:

1. *Clashing literal*: does not appear in the resolvent; pop its complement from the top of the E-stack;
2. *Target literal*: any literal in T; union this with the T-subset;
3. *Universal literal*: any universal literal not in T; union this with the U-set;
4. *Merge literal*: already in the E-stack; do not push this on the E-stack;
5. *Extension literal*: none of the above; all extension literals are pushed on the E-stack in outer to inner prefix order; the innermost extension literal is on top of the new E-stack.

Extension and merge literals are existential and the terminology stems from model elimination.

To get the center-clause data structure started, define the initial center clause to be $\top$, a tautologous clause that contains all literals. We use the sound extension that $\mathsf{res}_e(\top, C) = C$ for all non-tautologous C that do *not* contain the existential variable $|e|$. If the desired top clause is C_0, the literal selection rule simply chooses some literal whose variable is not among $vars(C_0)$. Then C_0 becomes the side clause for step 0. This artificial protocol makes all original clauses in the derivation appear as side clauses and simplifies later descriptions. The literals of the C_0 are processed as described in Definition 4.6.

The foregoing description can be formalized in mathematical terms of sets and sequences. We only note that the center clauses, disregarding the T-subset and U-set, can be regarded as existential literal sequences that can be partitioned into contiguous subsequences such that each subsequence is in prefix order and contains some subset of the extension literals of a single extension operation.

Definition 4.7. A *LIFO SLQR* is an SLQR that uses the LIFO selection function and also has an admissibility requirement for side clauses used for an extension operation.

At step i (to derive D_i) suppose the selected literal is $\overline{p}$. A side clause C (which necessarily contains p) is ***inadmissible*** if for some $j < i - 1$, D_j subsumes $\mathsf{res}_p(D_{i-1}, C)$ (including the case that the resolvent is tautologous). A derivation attempt fails if the current center clause was formed by resolution with an inadmissible side clause. In this case the LIFO-selected literal is $\bot$. $\Box$

Example 4.8. The motivation for inadmissible clauses is that it prevents looping [14]. Suppose the current center clause is $D_i = (\{\}, \{\beta\}, [\alpha, \overline{f}])$, where the T-subset is empty, the U-set is β, and the E-stack is $[\alpha, \overline{f}]$. Thus $\overline{f}$ is selected. Suppose there are clauses $C_1 = [f, \overline{g}]$ and $C_2 = [g, \overline{f}]$. Resolving (extending) D_i with C_1 gives $D_{i+1} = (\{\}, \{\beta\}, [\alpha, \overline{g}])$, then extending with C_2 would give $D_{i+2} = (\{\}, \{\beta\}, [\alpha, \overline{f}])$, creating a cycle. So C_2 is inadmissible to resolve with D_{i+1}. If no other side clause containing g is admissible, then the LIFO SLQR selected literal at step $i + 2$ is $\bot$, forcing the derivation attempt to fail. Thus a successfully completed LIFO SLQR never contains an inadmissible side clause. □

Corollary 4.9. Given a QBF $\Psi = \overrightarrow{Q}.\mathcal{F}$, let T be a minimal clause such that there is a Q-resolution derivation of T from Ψ, call it Π, and no proper subset of $\mathcal{F}$ permits derivation of T. Further, let the literals of T be outermost among the literals of $\mathcal{F}$. Then for every clause $C_0 \in \mathcal{F}$ there exists a LIFO SLQR derivation of T from Ψ whose top clause is C_0. Further, for each literal $q \in T$, q has no proof operation in the LIFO SLQR derivation.

Proof: Please see https://users.soe.ucsc.edu/~avg/Papers/slqr-long.pdf. ■

5 Conclusion

Subsumption-Linear Q-Resolution (SLQR) was introduced for proving theorems from Quantified Boolean Formulas. It is an adaptation of SL-Resolution, which in turn is closely related to model elimination and tableau methods. A major difference from QDPLL (DPLL adapted for QBF) is that QDPLL guesses variable assignments, while SLQR guesses clauses. Inferential completeness of SLQR for AT-free PCNFs is shown when it is allowed to use resolution with universal clashing variables; without that operation it is refutationally complete.

Future work should study heuristics for clause selection and lemma retention.

Acknowledgment. We thank the reviewers for their careful reading and suggestions for clarifying the paper.

References

1. Anderson, R., Bledsoe, W.W.: A linear format for resolution with merging and a new technique for establishing completeness. J. ACM **17**(3), 525–534 (1970)
2. Astrachan, O.L., Loveland, D.W.: The use of lemmas in the model elimination procedure. J. Autom. Reason. **19**, 117–141 (1997)
3. Beyersdorff, O., Chew, L., Janota, M.: New resolution-based QBF calculi and their proof complexity. ACM Trans. Comput. Theory **11**, 1–42 (2019)
4. Burris, S.: Logic for Mathematics and Computer Science. Prentice Hall, Upper Saddle River (1998)
5. Goultiaeva, A., Van Gelder, A., Bacchus, F.: A uniform approach for generating proofs and strategies for both true and false QBF formulas. In: Proceedings of IJCAI (2011)

6. Heule, M., Seidl, M., Biere, A.: Efficient extraction of skolem functions from QRAT proofs. In: Proceedings of FMCAD (2014)
7. Kleine Büning, H., Karpinski, M., Flögel, A.: Resolution for quantified Boolean formulas. Inf. Comput. **117**, 12–18 (1995)
8. Kleine Büning, H., Lettmann, T.: Propositional Logic: Deduction and Algorithms. Cambridge University Press, Cambridge (1999)
9. Klieber, W., Sapra, S., Gao, S., Clarke, E.: A non-prenex, non-clausal QBF solver with game-state learning. In: Strichman, O., Szeider, S. (eds.) SAT 2010. LNCS, vol. 6175, pp. 128–142. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14186-7_12
10. Letz, R.: Lemma and model caching in decision procedures for quantified Boolean formulas. In: Egly, U., Fermüller, C.G. (eds.) TABLEAUX 2002. LNCS (LNAI), vol. 2381, pp. 160–175. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45616-3_12
11. Letz, R., Mayr, K., Goller, C.: Controlled integration of the cut rule into connection tableau calculi. JAR **13**, 297–337 (1994)
12. Loveland, D.W.: Mechanical theorem-proving by model elimination. J. ACM **15**(2), 236–251 (1968)
13. Loveland, D.W.: A simplified format for the model elimination theorem-proving procedure. JACM **16**(3), 349–363 (1969)
14. Loveland, D.W.: Automated Theorem Proving: A Logical Basis. Elsevier, Amsterdam (1978)
15. Minker, J., Zanon, G.: An extension to linear resolution with selection function. Inf. Process. Lett. **14**(4), 191–194 (1982)
16. Samulowitz, H., Davies, J., Bacchus, F.: Preprocessing QBF. In: Benhamou, F. (ed.) CP 2006. LNCS, vol. 4204, pp. 514–529. Springer, Heidelberg (2006). https://doi.org/10.1007/11889205_37
17. Slivovsky, F.: Quantified CDCL with universal resolution. In: Proceedings of Sat 2022 (2022)
18. Van Gelder, A.: Autarky pruning in propositional model elimination reduces failure redundancy. J. Autom. Reason. **23**(2), 137–193 (1999)
19. Van Gelder, A.: Input distance and lower bounds for propositional resolution proof length. In: Theory and Applications of Satisfiability Testing (SAT) (2005)
20. Van Gelder, A.: Contributions to the theory of practical quantified Boolean formula solving. In: Proceedings of CP, pp. 647–673 (2012)
21. Van Gelder, A., Wood, S.B., Lonsing, F.: Extended failed literal detection for QBF. In: Proceedings of SAT, pp. 86–99 (2012)

Maximally Multi-focused Proofs for Skew Non-Commutative MILL

Niccolò Veltri(✉)

Tallinn University of Technology, Tallinn, Estonia
niccolo@cs.ioc.ee

Abstract. Multi-focusing is a generalization of Andreoli's focusing procedure which allows the parallel application of synchronous rules to multiple formulae under focus. By restricting to the class of maximally multi-focused proofs, one recovers permutative canonicity directly in the sequent calculus without the need to switch to other formalisms, e.g. proof nets, in order to represent proofs modulo permutative conversions. This characterization of canonical proofs is also amenable for the mechanization of the normalization procedure and the performance of further formal proof-theoretic investigations in interactive theorem provers.

In this work we present a sequent calculus of maximally multi-focused proofs for skew non-commutative multiplicative linear logic (SkNMILL), a logic recently introduced by Uustalu, Veltri and Wan which enjoys categorical semantics in the skew monoidal closed categories of Street. The peculiarity of the multi-focused system for SkNMILL is the presence of at most two foci in synchronous phase. This reduced complexity makes it a good starting point for the formal investigations of maximally multi-focused calculi for richer substructural logics.

Keywords: skew non-commutative MILL · maximal multi-focusing · skew monoidal closed categories · substructural logics · Agda

1 Introduction

Focusing is a technique introduced by Andreoli for reducing permutative non-determinism in proof search. It was originally applied to the cut-free sequent calculus of classical first-order linear logic [3] and subsequently ported to many other proof systems [9]. Andreoli's key idea was the organization of root-first proof search in the alternation of two distinct phases: the asynchronous phase, where invertible rules are eagerly applied, and the synchronous phase, where non-invertible rules are applied on a selected formula which is brought under focus.

We thank Alexis Saurin and Noam Zeilberger for valuable discussions. This work was supported by the Estonian Research Council grant PSG749, the ESF funded Estonian IT Academy research measure (project 2014-2020.4.05.19-0001) and COST CA19135 - Connecting Education and Research Communities for an Innovative Resource Aware Society.

H. H. Hansen et al. (Eds.): WoLLIC 2023, LNCS 13923, pp. 377–393, 2023.
https://doi.org/10.1007/978-3-031-39784-4_24

Focusing still retains a large amount of non-determinism in proof search, since many different formulae can possibly be brought under focus. Specifically, the non-determinism introduced by inessential permutative conversions is not resolved. Typically, a linear logician would solve this issue by leaving the sequent calculus and moving to a graphical representation of proofs, such as Girard's proof nets [7]. Chaudhuri et al. [6] showed that it is not necessary to depart from the sequent calculus formalism to represent canonical derivations wrt. the equational theory generated by the permutative conversions. They introduce a *multi-focused sequent calculus* where multiple formulae can simultaneously be brought under focus and decomposed during the synchronous phase. They then present a rewriting system on multi-focused proofs whose normal forms are *maximally multi-focused*. These are derivations f which, at the beginning of each synchronous phase, always pick the largest number of formulae to bring under focus among the multi-focused derivations which are equivalent to f wrt. the equational theory of permutative conversions. In this sense, maximally multi-focused proofs exhibit the maximal amount of parallelism. Chaudhuri et al. showed that these are equivalent to proof nets for unit-free multiplicative classical linear logic. Multi-focusing and maximality have subsequently been applied to other deductive systems [4,5], in particular variants of intuitionistic logic [12,13].

This work serves as a starting point for a comprehensive study of maximal multi-focused deductive systems for a large class of substructural logics. It is well-known that many substructural logics enjoy normalization procedures targeting variants of proof nets, e.g. the Lambek calculus [8]. Nevertheless, an extensive study of maximally multi-focused proofs for these logics is missing. We believe this to be especially beneficial for the development of proof-theoretic investigations of logical systems in interactive theorem provers, such as Agda or Coq, where the graphical syntax of proof nets would be harder to implement than sequent calculi, whose inference rules are standard example of inductive type families.

We initiate this endeavor by considering *skew non-commutative multiplicative linear logic* (SkNMILL), a weak substructural logic recently introduced by the author in collaboration with Uustalu and Wan [16]. This logic is a *semi-associative* and *semi-unital* variant of Lambek calculus (with only one residual): it validates structural rules of associativity $(A \otimes B) \otimes C \to A \otimes (B \otimes C)$ and unitality $\mathsf{I} \otimes A \to A$ and $A \to A \otimes \mathsf{I}$, but none of their inverses. Uustalu et al. introduce a cut-free sequent calculus for SkNMILL whose sequents are triples of the form $S \mid \Gamma \vdash A$, where S is an optional formula (called stoup), Γ is an ordered list of formulae and A is a single formula. A peculiarity of this calculus is that left logical rules act exclusively on the formula in the stoup position, never on formulae in context Γ. This makes this sequent calculus a good candidate for initiating the formal study of maximal multi-focusing of substructural logics: during the synchronous phase, at most *two* formulae can be brought under focus, the stoup formula and the succedent formula. From this perspective, the sequent calculus of SkNMILL is among the "simplest" deductive system which enjoys non-trivial multi-focusing.

The study of SkNMILL was initially motivated by its categorical semantics in the *skew monoidal closed categories* of Street [15]. These categories arise naturally in semantics of programming languages [1], while semi-associativity has found strong connections with combinatorial structures such as the Tamari lattice and Stasheff associahedra [10,20]. From a category-theoretic perspective, the maximal multi-focusing procedure described in this paper provides a solution to the *coherence* problem for skew monoidal closed categories.

The paper starts with a brief introduction of SkNMILL and its cut-free sequent calculus. It continues with a presentation of a sound and complete multi-focused sequent calculus. As expected, the latter does not resolve all the permutative non-determinism, but its introduction is pedagogically useful as it sets the stage for the more involved maximally multi-focused sequent calculus. The latter uses a system of *tags*, similarly employed by Uustalu et al. in their calculus of normal forms [16], which are inspired by Scherer and Rémy's saturation technique [14]. Tags are used to keep track of new formulae appearing in context from the application of invertible rules and to decide whether multi-focusing on both the stoup and succedent formulae is admissible or not.

An important contribution of this project is the formalization of the maximal multi-focusing calculus for SkNMILL and the proof of its correctness in the Agda proof assistant. The code, containing all the material presented in the paper, can be found at:

https://github.com/niccoloveltri/multifocus-sknmill

2 The Sequent Calculus of SkNMILL

We recall the definition of the sequent calculus for SkNMILL originally introduced in [16]. Formulae are generated by the grammar $A, B ::= X \mid \mathsf{I} \mid A \otimes B \mid A \multimap B$, where X comes from a fixed set At of atomic formulae, I is a multiplicative unit, $\otimes$ is a multiplicative tensor and $\multimap$ is a linear implication. Formulae I and $A \otimes B$ are *positive* while $A \multimap B$ is *negative.*

A sequent is a triple of the form $S \mid \Gamma \vdash A$, where the succedent A is a single formula (as in non-commutative multiplicative linear logic NMILL) and the antecedent is divided in two parts: an optional formula S, called *stoup*, and an ordered list of formulae Γ, called *context.* The metavariable S always denotes a stoup, i.e. S can be a single formula or empty, in which case we write $S = -$.

Derivations of a sequent $S \mid \Gamma \vdash A$ are inductively generated by the rules in Fig. 1. There are a few important differences with the sequent calculus of NMILL: 1) left rules can only act on the formula in stoup position, not on formulae in context; 2) the right $\otimes$-rule, when read bottom-up, forces the formula in the stoup (whenever it is present) to move to the stoup of the first premise, it cannot move to the antecedent of the second premise; 3) as in NMILL, there are no structural rules of exchange, weakening and contraction, but there is a new structural rule pass which moves the leftmost formula in context to the stoup, whenever the latter is empty.

$$\frac{}{X \mid\ \vdash X}\ \mathsf{ax} \qquad \frac{A \mid \Gamma \vdash C}{- \mid A, \Gamma \vdash C}\ \mathsf{pass}$$

$$\frac{- \mid \Gamma \vdash C}{\mathsf{I} \mid \Gamma \vdash C}\ \mathsf{IL} \qquad \frac{A \mid B, \Gamma \vdash C}{A \otimes B \mid \Gamma \vdash C}\ \otimes\mathsf{L} \qquad \frac{- \mid \Gamma \vdash A \quad B \mid \Delta \vdash C}{A \multimap B \mid \Gamma, \Delta \vdash C}\ \multimap\mathsf{L}$$

$$\frac{}{- \mid\ \vdash \mathsf{I}}\ \mathsf{IR} \qquad \frac{S \mid \Gamma \vdash A \quad - \mid \Delta \vdash B}{S \mid \Gamma, \Delta \vdash A \otimes B}\ \otimes\mathsf{R} \qquad \frac{S \mid \Gamma, A \vdash B}{S \mid \Gamma \vdash A \multimap B}\ \multimap\mathsf{R}$$

Fig. 1. Sequent calculus for `SkNMILL`.

$$\begin{array}{rl}
\otimes\mathsf{R}\ (\mathsf{IL}\ f, g) \doteq \mathsf{IL}\ (\otimes\mathsf{R}\ (f, g)) & (f : - \mid \Gamma \vdash A, g : - \mid \Delta \vdash B) \\
\otimes\mathsf{R}\ (\otimes\mathsf{L}\ f, g) \doteq \otimes\mathsf{L}\ (\otimes\mathsf{R}\ (f, g)) & (f : A' \mid B', \Gamma \vdash A, g : - \mid \Delta \vdash B) \\
\mathsf{pass}\ (\multimap\mathsf{R}\ f) \doteq \multimap\mathsf{R}\ (\mathsf{pass}\ f) & (f : A' \mid \Gamma, A \vdash B) \\
\mathsf{IL}\ (\multimap\mathsf{R}\ f) \doteq \multimap\mathsf{R}\ (\mathsf{IL}\ f) & (f : - \mid \Gamma, A \vdash B) \\
\otimes\mathsf{L}\ (\multimap\mathsf{R}\ f) \doteq \multimap\mathsf{R}\ (\otimes\mathsf{L}\ f) & (f : A \mid B, \Gamma, C \vdash D) \\
\multimap\mathsf{L}\ (f, \multimap\mathsf{R}\ g) \doteq \multimap\mathsf{R}\ (\multimap\mathsf{L}\ (f, g)) & (f : - \mid \Gamma \vdash A', g : B' \mid \Delta, A \vdash B) \\
\otimes\mathsf{R}\ (\mathsf{pass}\ f, g) \doteq \mathsf{pass}\ (\otimes\mathsf{R}\ (f, g)) & (f : A' \mid \Gamma \vdash A, g : - \mid \Delta \vdash B) \\
\otimes\mathsf{R}\ (\multimap\mathsf{L}\ (f, g), h) \doteq \multimap\mathsf{L}\ (f, \otimes\mathsf{R}\ (g, h)) & (f : - \mid \Gamma \vdash A, g : B \mid \Delta \vdash C, h : - \mid \Lambda \vdash D)
\end{array}$$

Fig. 2. Equivalence of derivations in the sequent calculus.

As in `NMILL` rules IL, $\otimes\mathsf{L}$ and $\multimap\mathsf{R}$ are invertible, while the other logical rules are not. The structural rule pass is also non-invertible. Two forms of *cut* are admissible, since the cut formula can either be located in the stoup or in the context of the second premise. A general axiom, or *identity*, rule is also admissible.

$$\frac{S \mid \Gamma \vdash A \quad A \mid \Delta \vdash C}{S \mid \Gamma, \Delta \vdash C}\ \mathsf{scut} \qquad \frac{- \mid \Gamma \vdash A \quad S \mid \Delta_0, A, \Delta_1 \vdash C}{S \mid \Delta_0, \Gamma, \Delta_1 \vdash C}\ \mathsf{ccut} \qquad \frac{}{A \mid\ \vdash A}\ \mathsf{ax}_A$$

A stoup S is called *irreducible* if it is either empty, an atom or a negative formula. This means that the stoup formula cannot be further reduced using left invertible rules IL and $\otimes\mathsf{L}$ in root-first proof search. Analogously, a succedent formula A is irreducible when it is atomic or positive, so it cannot be reduced by the right invertible rule $\multimap\mathsf{R}$.

We consider an equivalence relation $\doteq$ on sets of derivations. This is the congruence generated by the pairs of derivations in Fig. 2, which are permutative conversions. The congruence $\doteq$ has been chosen to serve as the proof-theoretic counterpart of the equational theory of skew monoidal closed categories [15]. In fact, there exists a *syntactic* skew monoidal closed category which has formulae of `SkNMILL` as objects, and morphisms between formulae A and B are given by the set of derivations of $A \mid\ \vdash B$ quotiented by the equivalence relation $\doteq$. This category is the *free* skew monoidal closed category generated by the set At. We refer to [16] for more details on categorical semantics.

ASYNCHRONOUS PHASE

$$\frac{S \mid \Gamma, A \Uparrow B}{S \mid \Gamma \Uparrow A \multimap B}\ {\multimap}\mathsf{R} \qquad \frac{A \mid B, \Gamma \Uparrow Q}{A \otimes B \mid \Gamma \Uparrow Q}\ \otimes\mathsf{L} \qquad \frac{- \mid \Gamma \Uparrow Q}{\mathsf{I} \mid \Gamma \Uparrow Q}\ \mathsf{IL} \qquad \frac{T \mid \Gamma \Downarrow Q}{T \mid \Gamma \Uparrow Q}\ \mathsf{foc}$$

SYNCHRONOUS PHASE

$$\frac{T \mid \Gamma \Downarrow_{\mathsf{lf}} Q \quad \boxed{Q} \mid \Delta \Downarrow \boxed{A}_b}{T \mid \Gamma, \Delta \Downarrow \boxed{A}_b}\ \mathsf{foc_L} \qquad \frac{\boxed{S}_b \mid \Gamma \Downarrow \boxed{T} \quad T \mid \Delta \Downarrow_{\mathsf{rf}} Q}{\boxed{S}_b \mid \Gamma, \Delta \Downarrow Q}\ \mathsf{foc_R}$$

$$\frac{}{\boxed{X} \mid \quad \Downarrow \boxed{X}}\ \mathsf{ax} \qquad \frac{S \mid \Gamma \Uparrow A \quad \mathsf{UT}(b, c, S, A)}{\boxed{S}_b \mid \Gamma \Downarrow \boxed{A}_c}\ \mathsf{unfoc}$$

LEFT-FOCUSING PHASE

$$\frac{A \mid \Gamma \Downarrow_{\mathsf{lf}} Q}{- \mid A, \Gamma \Downarrow_{\mathsf{lf}} Q}\ \mathsf{pass} \qquad \frac{- \mid \Gamma \Uparrow A \quad B \mid \Delta \Downarrow_{\mathsf{lf}} Q}{A \multimap B \mid \Gamma, \Delta \Downarrow_{\mathsf{lf}} Q}\ {\multimap}\mathsf{L} \qquad \frac{}{Q \mid \quad \Downarrow_{\mathsf{lf}} Q}\ \mathsf{blur_L}$$

RIGHT-FOCUSING PHASE

$$\frac{}{- \mid \quad \Downarrow_{\mathsf{rf}} \mathsf{I}}\ \mathsf{IR} \qquad \frac{T \mid \Gamma \Downarrow_{\mathsf{rf}} A \quad - \mid \Delta \Uparrow B}{T \mid \Gamma, \Delta \Downarrow_{\mathsf{rf}} A \otimes B}\ \otimes\mathsf{R} \qquad \frac{}{M \mid \quad \Downarrow_{\mathsf{rf}} M}\ \mathsf{blur_R}$$

Fig. 3. Multi-focused sequent calculus for SkNMILL.

We employ the following convention for naming formulae and stoups:

P	positive formula
N	negative formula
Q	positive or atomic formula
M	negative or atomic formula
T	irreducible stoup ($-$ or M)

3 A Multi-focused Sequent Calculus

We now present a multi-focused sequent calculus for SkNMILL, which draws inspiration from the one given by Chaudhuri et al. for multiplicative-additive classical linear logic [6]. Inference rules are given in Fig. 3. As in the original formulation by Andreoli [3], the (multi-)focused calculus describes, in a declarative fashion, a root-first proof search strategy in the original sequent calculus.

In this calculus, sequents can take four forms, corresponding to four distinct phases of proof search:

$S \mid \Gamma \Uparrow A$	asynchronous (or invertible)
$S \mid \Gamma \Downarrow A$	synchronous (or focusing)
$S \mid \Gamma \Downarrow_{\mathsf{lf}} Q$	left synchronous
$T \mid \Gamma \Downarrow_{\mathsf{rf}} A$	right synchronous

Proof search starts in asynchronous phase $S \mid \Gamma \Uparrow A$. In this phase, invertible rules are repeatedly applied until both the stoup formula (when present) and the succedent formula become irreducible. We have fixed an order on invertible rules and decided to apply $\multimap\mathsf{R}$ before $\mathsf{IL}/\otimes\mathsf{R}$, which is enforced by asking the succedent formula in the left invertible rules to be positive or atomic (so we use our notation Q).

Proof search then progresses to the synchronous phase via the rule foc. At this point we can choose to focus on the stoup or succedent position.

If we pick the first option, the irreducible stoup T is brought under focus with an application of rule $\mathsf{foc_L}$. The context is split in two parts Γ and Δ and the left focusing phase initiates in the first premise. A proof of $T \mid \Gamma \Downarrow_{\mathsf{lf}} Q$ consists of repeated application of left synchronous rules pass and $\multimap\mathsf{L}$ on stoup T and context Γ, until the stoup formula becomes the positive or atomic formula Q, at which point the left focus is "blurred" by the rule $\mathsf{blur_L}$. In synchronous phase, blurred formulae are surrounded by a *dashed* box $\boxed{A}$. We use notation $\boxed{A}_b$, with b a Boolean value, to denote a formula which is possibly blurred: $\boxed{A}_1 = \boxed{A}$ and $\boxed{A}_0 = A$. Blurred formulae are used to remember that a certain left or right synchronous phase has been performed.

If we pick the second option, proof search proceeds by bringing the succedent formula Q under focus with an application of rule $\mathsf{foc_R}$. The context is split in two parts Γ and Δ and the right focusing phase initiates in the second premise. The right focusing phase consists of repeated applications of the right synchronous rule $\otimes\mathsf{R}$. The optional formula T in sequent $T \mid \Delta \Downarrow_{\mathsf{rf}} Q$ indicates whether the right focusing phase terminates when the succedent formula becomes negative or atomic (in which case $T = M$) or it terminates with an application of IR (in which case $T = -$). In the first case, the succedent formula M is blurred by the rule $\mathsf{blur_R}$. The notation $\boxed{S}_b \mid \Gamma \Downarrow \boxed{T}$ is an abbreviation for: $\boxed{S}_b \mid \Gamma \Downarrow \boxed{M}$, when $T = M$, while its set of proofs is a singleton if $T = -$. In other words, $\mathsf{foc_R}$ does not have a first premise in case the proof of the second premise ends with IR.

A couple of observations on left- and right-focusing. A peculiarity of the sequent calculus in Fig. 3, when compared with other (multi-)focused calculi appearing in the literature, e.g. the one in [6], is that, during the application of non-invertible rules in the focusing phase, one of the premises always releases the focus. In rule $\otimes\mathsf{R}$, the right premise releases the focus on the succedent formula, and similarly for the first premise in rule $\multimap\mathsf{L}$. Without the loss of focus in these premises, the multi-focused sequent calculus would not be complete wrt. the calculus in Fig. 1, e.g. the sequent $X \mid Y \otimes Z \Uparrow X \otimes (Y \otimes Z)$ would not admit a derivation. This behaviour was already present in the focused sequent calculi for the $\otimes$- and $(\mathsf{I}, \otimes)$-fragments of the sequent calculus, originally studied by Zeilberger et al. [17,20].

The design of rule $\mathsf{foc_L}$, with a whole left-focusing phase compressed in a proof of $T \mid \Gamma \Downarrow_{\mathsf{lf}} Q$, is chosen specifically for the purpose of maximal multi-focusing, where we will be interested in whether a certain left-focusing phase has happened rather than the specific left synchronous rules that have been applied. Notice also that in the first premise $T \mid \Gamma \Downarrow_{\mathsf{lf}} Q$ of $\mathsf{foc_L}$ there is no need to keep track of the succedent formula A since it is not affected by left synchronous rules, and similarly for the context Δ of the second premise. Analogous observations apply to the second premise of $\mathsf{foc_R}$.

When the (left-) right-focusing phase terminates, one can subsequently choose to focus on the (succedent) stoup formula. If the execution of both left- and right-focusing lead to a valid derivation, they can be performed in any order, first left then right, or vice versa. When no formula is under focus anymore, we *unfocus* and continue proof search in asynchronous phase. In order to unfocus, formulae that were previously under focus, which are now blurred, must have switched their polarity, which is reflected in the side condition $\mathsf{UT}(b, c, S, A)$ of rule unfoc (UT stands for "unfocusing table"):

b	c	$\mathsf{UT}(b, c, S, A)$
0	0	0
0	1	$A = N$
1	0	$S = P$
1	1	$S = P \vee (S = X \wedge A = N)$

The stoup formula must be positive if it was under focus ($b = 1$) but the succedent was not ($c = 0$). Dually, the succedent formula must be negative is it was under focus ($c = 1$) and the stoup formula was not ($b = 0$). If both formulae were under focus ($b = 1 \wedge c = 1$), one of them must have changed its polarity: either the stoup formula has become positive or, if it had become (or stayed) atomic, the succedent formula has become negative. Unfocusing also requires that at least one formula was previously under focus, hence the condition $b \vee c$ must be true.

For a sequent with atomic stoup and positive succedent $X \mid \Gamma \Uparrow P$ (or, dually, negative stoup and atomic succedent), one can choose whether to focus on the stoup formula or not, and both choices may lead to a valid proof. For an example, consider the valid sequent $X \mid \ \Uparrow (Y \multimap (X \otimes Y)) \otimes I$. This situation was also present in the multi-focused calculus for classical linear logic [6], where in similar circumstances one was given the choice of focusing on negated atoms or not.

Invertible rules are easily proved to be admissible in the $\Uparrow$ phase (with a general formula as succedent), and similarly IR and ax.

Proposition 1. *The following rules are admissible:*

$$\frac{A \mid B, \Gamma \Uparrow C}{A \otimes B \mid \Gamma \Uparrow C}\ \otimes\mathsf{L}_{\Uparrow} \qquad \frac{- \mid \Gamma \Uparrow C}{\mathsf{I} \mid \Gamma \Uparrow C}\ \mathsf{IL}_{\Uparrow} \qquad \frac{}{- \mid \ \Uparrow \mathsf{I}}\ \mathsf{IR}_{\Uparrow} \qquad \frac{}{X \mid \ \Uparrow X}\ \mathsf{ax}_{\Uparrow}$$

Rule $\otimes\mathsf{R}$ of Fig. 1, with $\vdash$ replaced everywhere by $\Uparrow$, is also admissible, but showing this requires more work. We prove the admissibility of a *macro* inference rule corresponding to multiple application of $\otimes\mathsf{R}$. To this end, given a formula A and a list of formulae $\Gamma = B_1, \ldots, B_n$, define $A \otimes^* \Gamma = (((A \otimes B_1) \otimes B_2) \otimes \ldots) \otimes B_n$, which is simply A when Γ is empty. If Γ is non-empty, we write $A \otimes^+ \Gamma$. Define also $\Gamma \multimap^* A = B_1 \multimap (B_2 \multimap (\ldots \multimap (B_n \multimap A)))$ and similarly $\Gamma \multimap^+ A$ when Γ is non-empty.

Proposition 2. *Let $\overrightarrow{\Delta} = \Delta_1, \dots, \Delta_n$ be a list of contexts and $\overrightarrow{B} = B_1, \dots, B_n$ a list of formulae, both non-empty. The following rule is admissible:*

$$\dfrac{S \mid \Gamma \Uparrow A \quad \{- \mid \Delta_i \Uparrow B_i\}_i}{S \mid \Gamma, \overrightarrow{\Delta} \Uparrow A \otimes^+ \overrightarrow{B}} \;\otimes\mathsf{R}^+_{\Uparrow}$$

where $\{- \mid \Delta_i \Uparrow B_i\}_i$ is a collection of premises $- \mid \Delta_i \Uparrow B_i$ for each $1 \le i \le n$.

Proof. The proof proceeds by inspecting the polarity of formula A and then by induction on the structure of the derivation $f : S \mid \Gamma \Uparrow A$. When A is negative, we need to strengthen the statement for the induction on f to succeed. So we prove the admissibility of the more general rule:

$$\dfrac{S \mid \Gamma, \Lambda \Uparrow A \quad \{- \mid \Delta_i \Uparrow B_i\}_i}{S \mid \Gamma, \overrightarrow{\Delta} \Uparrow (\Lambda \multimap^+ A) \otimes^+ \overrightarrow{B}} \;\otimes\mathsf{R}^+_{\Uparrow N}$$

The context Λ serves as an accumulator for dealing with the case $f = \multimap\mathsf{R}(f')$:

$$\dfrac{\dfrac{\overset{f'}{S \mid \Gamma, \Lambda, A' \Uparrow B'}}{S \mid \Gamma, \Lambda \Uparrow A' \multimap B'} \;{\multimap}R \quad \{- \mid \Delta_i \Uparrow B_i\}_i}{S \mid \Gamma, \overrightarrow{\Delta} \Uparrow (\Lambda \multimap^+ (A' \multimap B')) \otimes^+ \overrightarrow{B}} \;\otimes\mathsf{R}^+_{\Uparrow N} \quad = \quad \dfrac{\overset{f'}{S \mid \Gamma, \Lambda, A' \Uparrow B'} \quad \{- \mid \Delta_i \Uparrow B_i\}_i}{S \mid \Gamma, \overrightarrow{\Delta} \Uparrow ((\Lambda, A') \multimap^+ B') \otimes^+ \overrightarrow{B}} \;\otimes\mathsf{R}^+_{\Uparrow N}$$

which type checks since $\Lambda \multimap^+ (A' \multimap B') = (\Lambda, A') \multimap^+ B'$. Another representative case is $f = \mathsf{foc}(f')$, where right-focusing can immediately be executed:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overset{f'}{T \mid \Gamma, \Lambda \Downarrow Q}}{T \mid \Gamma, \Lambda \Uparrow Q} \;\mathsf{foc}}{T \mid \Gamma \Uparrow \Lambda \multimap^+ Q} \;{\multimap}\mathsf{R}^+}{T \mid \Gamma \Downarrow \Lambda \multimap^+ Q} \;\mathsf{unfoc} \quad \dfrac{\dfrac{}{\Lambda \multimap^+ Q \mid \;\Downarrow_{\mathsf{rf}} \Lambda \multimap^+ Q} \;\mathsf{blur_R} \quad \{- \mid \Delta_i \Uparrow B_i\}_i}{\Lambda \multimap^+ Q \mid \overrightarrow{\Delta} \Downarrow_{\mathsf{rf}} (\Lambda \multimap^+ Q) \otimes^+ \overrightarrow{B}} \;\otimes\mathsf{R}^+}{T \mid \Gamma, \overrightarrow{\Delta} \Downarrow (\Lambda \multimap^+ Q) \otimes^+ \overrightarrow{B}} \;\mathsf{foc_R}}{T \mid \Gamma, \overrightarrow{\Delta} \Uparrow (\Lambda \multimap^+ Q) \otimes^+ \overrightarrow{B}} \;\mathsf{foc} \qquad (1)$$

We now move to the admissibility of left-synchronous rules. To this end, we introduce an inductive ternary (proof-relevant) relation $A \rightsquigarrow_{\mathsf{li}} S \mid \Gamma$ which holds when the antecedent $S \mid \Gamma$ is obtained by repeated applications of left-invertible rules on the antecedent $A \mid$, where A is in the stoup and the context is empty:

$$\dfrac{A \rightsquigarrow_{\mathsf{li}} \mathsf{I} \mid \Gamma}{A \rightsquigarrow_{\mathsf{li}} - \mid \Gamma} \;\mathsf{IL}^{-1} \qquad \dfrac{A \rightsquigarrow_{\mathsf{li}} A' \otimes B' \mid \Gamma}{A \rightsquigarrow_{\mathsf{li}} A' \mid B', \Gamma} \;\otimes\mathsf{L}^{-1} \qquad \dfrac{}{A \rightsquigarrow_{\mathsf{li}} A \mid} \;\varepsilon$$

Given a proof $\ell : A \rightsquigarrow_{\mathsf{li}} S \mid \Gamma$, we can turn a derivation $f : S \mid \Gamma, \Delta \Uparrow C$ into a derivation $\mathsf{inv}_{\mathsf{li}}(f, \ell) : A \mid \Delta \Uparrow C$:

$$\mathsf{inv}_{\mathsf{li}}(f,\ell) = \begin{array}{c} f \\ S \mid \Gamma, \Delta \Uparrow C \\ \vdots \quad \text{(left rules obtained by inverting } \ell) \\ A \mid \Delta \Uparrow C \end{array} \tag{2}$$

Proposition 3. *The following rules are admissible:*

$$\frac{A \mid \Gamma \Uparrow C}{- \mid A, \Gamma \Uparrow C}\ \mathsf{pass}_{\Uparrow} \qquad \frac{\{- \mid \Gamma_i \Uparrow A_i\}_i \quad B \mid \Delta \Uparrow C}{\overrightarrow{A} \multimap^{+} B \mid \overrightarrow{\Gamma}, \Delta \Uparrow C}\ \multimap\mathsf{L}^{+}_{\Uparrow}$$

Proof. We only discuss $\multimap\mathsf{L}^{+}$. Proving its admissibility proceeds by inspecting the polarity of formula B and then by induction on the structure of the derivation $g : B \mid \Delta \Uparrow C$. When B is positive, we need to strengthen the statement for the induction to succeed. We prove the admissibility of the more general rule:

$$\frac{\{- \mid \Gamma_i \Uparrow A_i\}_i \quad B \rightsquigarrow_{\mathsf{li}} S \mid \Lambda \quad S \mid \Lambda, \Delta \Uparrow C}{\overrightarrow{A} \multimap^{+} B \mid \overrightarrow{\Gamma}, \Delta \Uparrow C}\ \multimap\mathsf{L}^{+}_{\Uparrow P}$$

The additional assumption $\ell : B \rightsquigarrow_{\mathsf{li}} S \mid \Lambda$ serves as an accumulator for dealing with the cases when g is a left-invertible rule and it allows to state that the proof of the third premise is a subderivation of sequent $B \mid \Delta \Uparrow C$ in the sense depicted in (2). A representative case is $g = \mathsf{foc}(g)$, where we can immediately execute left-focusing, obtaining a derivation dual to the one in (1).

The multi-focused sequent calculus in Fig. 3 is sound and complete wrt. the sequent calculus in Fig. 1. In the upcoming theorem and in the rest of the paper, we also write $S \mid \Gamma \vdash A$ and $S \mid \Gamma \Uparrow A$ for the sets of proofs of the corresponding sequents.

Theorem 1. *There exist functions* $\mathsf{focus} : S \mid \Gamma \vdash A \rightarrow S \mid \Gamma \Uparrow A$ *and* $\mathsf{emb} : S \mid \Gamma \Uparrow A \rightarrow S \mid \Gamma \vdash A$*, turning sequent calculus derivations into multi-focused derivations, and vice versa.*

Proof. Function emb is obtained by erasing all phase-shifting rules and dashed boxes around blurred formulae. Function focus is defined by induction on the structure of the input derivation, noticing that each rule in Fig. 1 has an admissible counterpart in the multi-focused sequent calculus, which follows from Propositions 1, 2 and 3.

Multi-focused proofs are not canonical wrt. to the equational theory in Fig. 2. When the stoup formula is negative and the succedent is positive, we have the choice of whether left-focusing and subsequently unfocus, right-focusing and subsequently unfocus, or performing both left- and right-focusing before unfocusing, and the latter can also be achieved in two distinct ways. For example, there exist four distinct proofs of $X \multimap \mathsf{I} \mid X, Y \Uparrow (Z \multimap Z) \otimes Y$ which correspond to four $\doteq$-related derivations in the unfocused sequent calculus. As discussed before, in general we also have the choice of whether focusing on atomic formulae or not, which further increases the amount of non-determinism.

It is possible to fully capture this remaining non-determinism in a congruence relation $\doteq_{\Uparrow}$ on derivations of sequents $S \mid \Gamma \Uparrow A$. This is inductively specified simultaneously with congruences $\doteq_{\Downarrow}$, $\doteq_{\mathsf{lf}}$ and $\doteq_{\mathsf{rf}}$. The generators of this collection of relations are exhibited in Fig. 4. Notice that all these generators belong to the relation $\doteq_{\Downarrow}$. This means that $\doteq_{\Uparrow}$ is the smallest equivalence relation which rules $\multimap\mathsf{R}$, IL and $\otimes\mathsf{L}$ respect (in the sense that they send $\doteq_{\Uparrow}$-related premises to $\doteq_{\Uparrow}$-related conclusions), and moreover $f \doteq_{\Downarrow} g$ implies $\mathsf{foc}(f) \doteq_{\Uparrow} \mathsf{foc}(g)$.

We can show that functions focus and emb respect congruences $\doteq$ and $\doteq_{\Uparrow}$, and moreover define an equivalence between sets of proofs in the different sequent calculi, strengthening the statement of Theorem 1.

Theorem 2. *Functions* focus *and* emb *underlie an isomorphism between the set of proofs of a sequent* $S \mid \Gamma \vdash A$ *quotiented by the equivalence relation* $\doteq$ *and the set of proofs of* $S \mid \Gamma \Uparrow A$ *quotiented by the equivalence relation* $\doteq_{\Uparrow}$.

Details about the proof can be found in our Agda formalization.

4 Maximal Multi-focusing Using Tags

In order to design a calculus of permutative-canonical derivations, we have to answer the following question: in which situation does a right-focusing phase *need* to be performed strictly before a left-focusing phase? And dually, when must left-focusing be done before right-focusing? Consider the valid sequent $X \multimap Y \mid Z \Downarrow (X \multimap Y) \otimes Z$. Attempting to focus on the stoup formula would fail, because no splitting of the context, consisting of the singleton formula Z, leads to a valid derivation. We would be able to appropriately split the context only after performing right-focusing, specifically after an application of $\otimes\mathsf{R}$, and a subsequent application of $\multimap\mathsf{R}$. This is because the formula X, that we would like to send to the first premise during left-focusing, is not initially in context, it becomes available only after right-focusing.

Dually, consider the valid sequent $X \multimap (Y \otimes Z) \mid X \Downarrow Y \otimes Z$. It is not hard to see that any attempt to focus on the succedent formula would fail. But after left-focusing and an application of $\otimes\mathsf{L}$, right-focusing becomes possible and leads to a valid proof. This is because the formula Z, which should be sent to the second premise by $\mathsf{foc}_{\mathsf{R}}$, appears in context only after executing the left-focusing phase. Another simple example is given by the valid sequent $- \mid X \otimes Y \Downarrow X \otimes Y$. Again left-focusing, specifically pass, must happen before right-focusing, since the formula Y is not in context and cannot otherwise be sent to the second premise during right-focusing.

We need a mechanism for keeping track of *new* formulae appearing in context from applications of invertible rules $\otimes\mathsf{L}$ and $\multimap\mathsf{R}$. In proof search, when we choose to perform left-focusing but we decide to postpone right-focusing, after

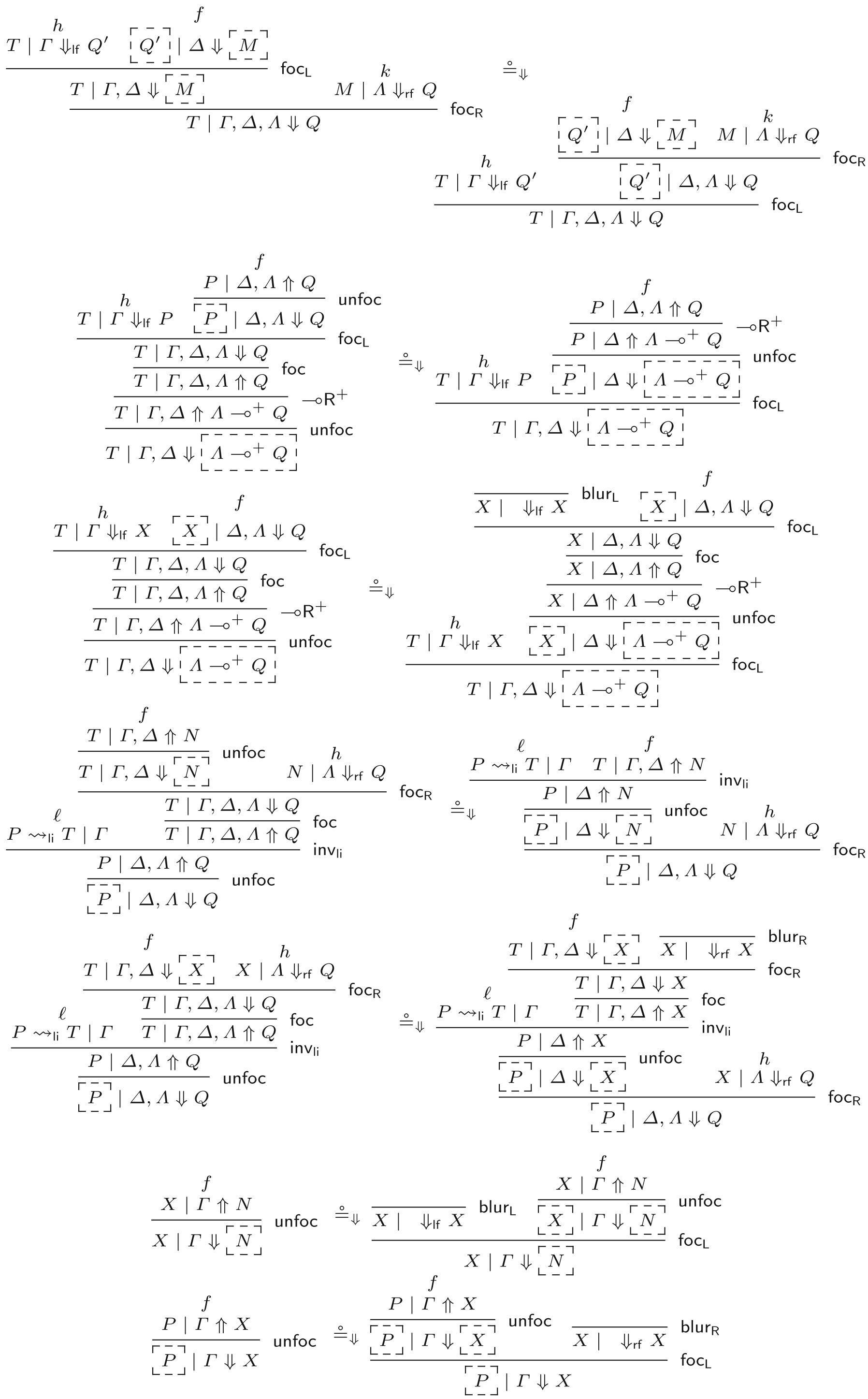


Fig. 4. Equivalence of derivations in the multi-focused sequent calculus.

releasing the focus we have to justify this decision by showing that the subsequent application of $\mathsf{foc_R}$ splits the context in-between new formulae that appeared in context only after the termination of the left-focusing phase. And dually if right-focusing strictly precedes left-focusing.

We employ a mechanism from the recent work of Uustalu et al. [16] which was inspired by Scherer and Rémy's saturation for intuitionistic logic [14]. Formulae appearing in a sequent will now be decorated with a superscript Boolean value, which we call a *tag*: A^0 or A^1. Stoups are also tagged: S^0 or S^1. Tagged contexts consist of tagged formulae. Sequents in the maximally multi-focused sequent calculus also take four forms:

$S \mid \Gamma \Uparrow_{\mathsf{m}} A$	asynchronous
$S \mid \Gamma \Downarrow_{\mathsf{m}} A$	synchronous
$S \mid \Gamma \Downarrow_{\mathsf{lfm}} Q$	left synchronous
$T \mid \Gamma \Downarrow_{\mathsf{rfm}} A$	right synchronous

The above are all triples consisting of a tagged stoup S (or an irreducible tagged stoup T in the last case), a tagged context Γ and a tagged formula A (or an irreducible tagged formula Q in the third case). If in a sequent we do not want to specify the tag of a tagged formula, we simply write it without superscript. Given a tagged formula A, we also write A^0 when we want to replace the tag of A by 0 and A^1 when the tag is replaced by 1. These conventions also apply to tagged stoups and contexts in a sequent.

Tags serve two purposes:

1. They are used to remember which (if any) among left- or right-focusing was *not* performed during the preceding focusing phase. If the stoup is S^1, only right-focusing was previously executed. Dually, if the succedent formula is A^1, only left-focusing took place.
2. In case one (and only one) among the stoup and the succedent has tag 1, new formulae moved to context via the application of invertible rules are also assigned tag 1. So tags are used to remember which formulae in context are new.

Inference rules for the maximally multi-focused sequent calculus are displayed in Fig. 5. In the premise of rule $\multimap\mathsf{R}$, the stoup S and the formula A must have the same tag t: if the stoup is S^1 in the conclusion, so left-focusing did not happen in the previous synchronous phase, we track the new formula A moving to the right-most end of the context by assigning it tag 1. Similarly for tagged formulae A^t and Q^t in the premise of rule $\otimes\mathsf{L}$.

Proof search starts again in asynchronous phase, where initially the sequent is $S^0 \mid \Gamma^0 \Uparrow_{\mathsf{m}} A^0$. At this point of the search, this phase is analogous to the one in the multi-focused calculus of Fig. 3. Tag 1 may start to appear with an application of unfoc. If left-focusing was not performed, so $b = 0$, then the stoup is given tag 1, which in the rule is denoted $S^{\neg b}$. If right-focusing was not executed, so $c = 0$, then the succedent has given tag 1, so it becomes $A^{\neg c}$. If either the stoup or the succedent has tag 1, new formulae moved to the context via applications of $\multimap\mathsf{R}$ and $\otimes\mathsf{L}$ are also assigned tag 1.

Asynchronous Phase

$$\frac{S^t \mid \Gamma, A^t \Uparrow_{\mathsf{m}} B}{S^t \mid \Gamma \Uparrow_{\mathsf{m}} A \multimap B}\ {\multimap}\mathsf{R} \qquad \frac{A \mid B^t, \Gamma \Uparrow_{\mathsf{m}} Q^t}{A \otimes B \mid \Gamma \Uparrow_{\mathsf{m}} Q^t}\ {\otimes}\mathsf{L} \qquad \frac{- \mid \Gamma \Uparrow_{\mathsf{m}} Q}{\mathsf{I} \mid \Gamma \Uparrow_{\mathsf{m}} Q}\ \mathsf{IL} \qquad \frac{T \mid \Gamma \Downarrow_{\mathsf{m}} Q}{T \mid \Gamma \Uparrow_{\mathsf{m}} Q}\ \mathsf{foc}$$

Synchronous Phase

$$\frac{T^0 \mid \Gamma^0 \Downarrow_{\mathsf{lfm}} Q^0 \qquad \boxed{Q^0} \mid \Delta \Downarrow_{\mathsf{m}} A \qquad 1 \in \Gamma}{T^1 \mid \Gamma, \Delta \Downarrow_{\mathsf{m}} A}\ \mathsf{foc}^1_{\mathsf{L}}$$

$$\frac{\boxed{S}_b \mid \Gamma \Downarrow_{\mathsf{m}} \boxed{T^0} \qquad T^0 \mid \Delta^0 \Downarrow_{\mathsf{rfm}} Q^0 \qquad T = M \supset 1 \in \Delta}{\boxed{S}_b \mid \Gamma, \Delta \Downarrow_{\mathsf{m}} Q^1}\ \mathsf{foc}^1_{\mathsf{R}}$$

$$\frac{T^0 \mid \Gamma^0 \Downarrow_{\mathsf{lfm}} Q^0 \qquad \boxed{Q^0} \mid \Delta \Downarrow_{\mathsf{m}} A}{T^0 \mid \Gamma, \Delta \Downarrow_{\mathsf{m}} A}\ \mathsf{foc}_{\mathsf{L}} \qquad \frac{\boxed{S}_b \mid \Gamma \Downarrow_{\mathsf{m}} \boxed{T^0} \qquad T^0 \mid \Delta^0 \Downarrow_{\mathsf{rfm}} Q^0}{\boxed{S}_b \mid \Gamma, \Delta \Downarrow_{\mathsf{m}} Q^0}\ \mathsf{foc}_{\mathsf{R}}$$

$$\frac{}{\boxed{X^0} \mid \ \Downarrow_{\mathsf{m}} \boxed{X^0}}\ \mathsf{ax} \qquad \frac{S^{\neg b} \mid \Gamma^0 \Uparrow_{\mathsf{m}} A^{\neg c} \qquad \mathsf{UT}(b, c, S, A)}{\boxed{S^0}_b \mid \Gamma \Downarrow_{\mathsf{m}} \boxed{A^0}_c}\ \mathsf{unfoc}$$

Fig. 5. Maximally multi-focused sequent calculus for `SkNMILL`.

If we want to left-focus, we first inspect the tag of the stoup formula. If it is T^1, we need to justify *why* left-focusing was not performed together with right-focusing in the preceding synchronous phase. This can be done by requiring a formula tagged with 1 to appear in Γ, which is the meaning of the side condition $1 \in \Gamma$ in the premise of $\mathsf{foc}^1_{\mathsf{L}}$. Proof search continues with a stoup formula Q^0. Dually, if we want to right-focus and the succedent is Q^1, and moreover T is non-empty, we require a formula tagged with 1 to appear in Δ when applying $\mathsf{foc}^1_{\mathsf{R}}$. When T is empty, so the right-focusing phase terminates with IR, there is no need to check whether Δ contains formulae tagged with 1, since right-focusing could not have happened together with the preceding left-focusing phase. Phases $\Downarrow_{\mathsf{lfm}}$ and $\Downarrow_{\mathsf{rfm}}$ are omitted in Fig. 5, since they are the same as $\Downarrow_{\mathsf{lf}}$ and $\Downarrow_{\mathsf{rf}}$ in Fig. 3 but with all formulae in sequents having tag 0, and $\Uparrow$ replaced by $\Uparrow_{\mathsf{m}}$ in the premises of ${\multimap}\mathsf{L}$ and ${\otimes}\mathsf{R}$.

When releasing the focus via unfoc, stoup and succedent must have tag 0, meaning that all the reasons for "not maximally focus" in a preceding focusing phase must have been successfully justified. Apart from tags, there are a couple of differences with the multi-focused system in Fig. 3.

1. In synchronous phase, we have the choice of first applying $\mathsf{foc}_{\mathsf{L}}$ and then applying $\mathsf{foc}_{\mathsf{R}}$, i.e. we remove non-determinism in the choice of left- or right-focusing when both are executable. In Fig. 5 this can be observed in $\mathsf{foc}_{\mathsf{L}}$, where succedents cannot be blurred.

2. Another difference lays in the treatment of atomic formulae. The axiom rule ax requires the atomic formula to have tag 0 and to be blurred in both positions. More generally, each derivation of $X \mid \Gamma \Downarrow_{\mathsf{m}} A$ necessarily focuses on the stoup and each derivation of $S \mid \Gamma \Downarrow_{\mathsf{m}} X$ necessarily focuses on the succedent.

All tags from a maximally multi-focused derivation can be removed to obtain a proof in the non-maximally multi-focused sequent calculus. More interestingly, each multi-focused derivation can be normalized to a maximally multi-focused one.

Theorem 3. *There exist functions*

$$\begin{aligned}\mathsf{max}_{\odot} &: S \mid \Gamma \odot A \;\rightarrow\; S^0 \mid \Gamma^0 \odot_{\mathsf{m}} A^0\\ \mathsf{untag}_{\odot} &: S^0 \mid \Gamma^0 \odot_{\mathsf{m}} A^0 \;\rightarrow\; S \mid \Gamma \odot A\end{aligned}$$

for all $\odot \in \{\Uparrow, \Downarrow, \Downarrow_{\mathsf{lf}}, \Downarrow_{\mathsf{rf}}\}$*, turning multi-focused proofs into maximally multi-focused ones, and vice versa.*

Proof. We only sketch the construction of $\mathsf{max}_{\Downarrow}$, which is the most challenging function to define. We refer the interested reader to the associated Agda formalization for the complete proof. The input derivation can either be: (i) an application of $\mathsf{foc_R}$ followed by ax or unfoc; (ii) an application of $\mathsf{foc_L}$ followed by ax or unfoc; (iii) an application of both $\mathsf{foc_L}$ and $\mathsf{foc_R}$. In case (iii), we can safely apply both $\mathsf{foc_L}$ and $\mathsf{foc_R}$ in the maximally multi-focused calculus. The most interesting cases are (i) and (ii) when the focus is subsequently released. We only look at case (i) when the input derivation is of the form $f = \mathsf{foc_R}(\mathsf{unfoc}(f'), r)$ for some $f' : S \mid \Gamma \Uparrow M$ and $r : M \mid \Delta \Downarrow_{\mathsf{rf}} Q$. To deal with this case, we prove the following rule admissible:

$$\frac{S^0 \mid \Gamma^0 \Uparrow_{\mathsf{m}} M^0 \quad M^0 \mid \Delta^0 \Downarrow_{\mathsf{rfm}} Q^0}{S^0 \mid \Gamma^0, \Delta^0 \Uparrow_{\mathsf{m}} Q^0}\ \mathsf{foc}_{\mathsf{R}\Uparrow_{\mathsf{m}}}$$

The proof proceeds by checking whether M is atomic or negative. In the latter case we further need to generalize the statement and prove the admissibility of

$$\frac{T^0 \mid \Gamma^0, \Lambda^0 \Uparrow_{\mathsf{m}} A^0 \quad \Lambda^0 \multimap^{+} A^0 \mid \Delta^0 \Downarrow_{\mathsf{rfm}} Q^0}{T^0 \mid \Gamma^0, \Delta^0 \Uparrow_{\mathsf{m}} Q^0}\ \mathsf{foc}_{\mathsf{R}\Uparrow_{\mathsf{m}}N}$$

We proceed by induction on the structure of the proof of the first premise $g : T^0 \mid \Gamma^0, \Lambda^0 \Uparrow_{\mathsf{m}} A^0$. We look at the case $g = \mathsf{foc}(\mathsf{foc_L}(l, \mathsf{unfoc}(h)))$ for some $l : T^0 \mid \Omega^0 \Downarrow_{\mathsf{lfm}} P^0$ and $h : P^0 \mid \Xi^0 \Uparrow_{\mathsf{m}} A^1$. In this case, we have the equality of contexts $\Omega, \Xi = \Gamma, \Lambda$ and we check whether Λ is split between Ω and Ξ, or it is fully contained in Ξ.

1. If $\Lambda = \Phi, \Xi$ and $\Omega = \Gamma, \Phi$ for some non-empty Φ, then multi-focusing on both stoup and succedent is not possible. We return:

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overset{l}{T^0 \mid \Omega^0 \Downarrow_{\mathsf{lfm}} P^0}}{T^0 \mid \Gamma^0, \Phi^0 \Downarrow_{\mathsf{lfm}} P^0}= \quad \dfrac{\overset{h}{P^0 \mid \Xi^0 \Uparrow_{\mathsf{m}} A^1}}{\boxed{P^0} \mid \Xi^1 \Downarrow_{\mathsf{m}} A^0}\,\mathsf{unfoc}}{T^1 \mid \Gamma^0, \Phi^1, \Xi^1 \Downarrow_{\mathsf{m}} A^0}\,\mathsf{foc}^1_{\mathsf{L}}}{T^1 \mid \Gamma^0, \Phi^1, \Xi^1 \Uparrow_{\mathsf{m}} A^0}\,\mathsf{foc}}{T^1 \mid \Gamma^0, \Lambda^1 \Uparrow_{\mathsf{m}} A^0}=}{T^1 \mid \Gamma^0 \Uparrow_{\mathsf{m}} \Lambda^0 {\multimap^+} A^0}\,{\multimap}\mathsf{R}^+}{T^0 \mid \Gamma^0 \Downarrow_{\mathsf{m}} \boxed{\Lambda^0 {\multimap^+} A^0}}\,\mathsf{unfoc} \quad \Lambda^0 {\multimap^+} A^0 \mid \Delta^0 \Downarrow_{\mathsf{rfm}} Q^0}{T^0 \mid \Gamma^0, \Delta^0 \Downarrow_{\mathsf{m}} Q^0}\,\mathsf{foc}_{\mathsf{R}}}{T^0 \mid \Gamma^0, \Delta^0 \Uparrow_{\mathsf{m}} Q^0}\,\mathsf{foc}
$$

The double-line rule is the equality rule (we simply rewrite the contexts).

2. If $\Gamma = \Omega, \Phi$ and $\Xi = \Phi, \Lambda$, then multi-focusing on both stoup and succedent is possible. We return:

$$
\dfrac{\dfrac{\dfrac{\overset{l}{T^0 \mid \Omega^0 \Downarrow_{\mathsf{lfm}} P^0} \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overset{h'}{P^0 \mid \Xi^0 \Uparrow_{\mathsf{m}} A^0}}{P^0 \mid \Phi^0, \Lambda^0 \Uparrow_{\mathsf{m}} A^0}=}{P^0 \mid \Phi^0 \Uparrow_{\mathsf{m}} \Lambda {\multimap^+} A^0}\,{\multimap}\mathsf{R}^+}{\boxed{P^0} \mid \Phi^0 \Downarrow_{\mathsf{m}} \boxed{\Lambda {\multimap^+} A^0}}\,\mathsf{unfoc} \quad \Lambda^0 {\multimap^+} A^0 \mid \Delta^0 \Downarrow_{\mathsf{rfm}} Q^0}{\boxed{P^0} \mid \Phi^0, \Delta^0 \Downarrow_{\mathsf{m}} Q^0}\,\mathsf{foc}_{\mathsf{R}}}{T^0 \mid \Omega^0, \Phi^0, \Delta^0 \Downarrow_{\mathsf{m}} Q^0}\,\mathsf{foc}_{\mathsf{L}}}{T^0 \mid \Gamma^0, \Delta^0 \Downarrow_{\mathsf{m}} Q^0}=}{T^0 \mid \Gamma^0, \Delta^0 \Uparrow_{\mathsf{m}} Q^0}\,\mathsf{foc}
$$

where h' is obtained from h by turning all applications of rules $\mathsf{foc}^1_{\mathsf{L}}$ and $\mathsf{foc}^1_{\mathsf{R}}$ in h to $\mathsf{foc}_{\mathsf{L}}$ and $\mathsf{foc}_{\mathsf{R}}$.

It is possible to show that proofs in the maximally multi-focused calculus are canonical wrt. the equational theory in Fig. 4 on multi-focused derivations. Therefore, by Theorem 2, they are also canonical wrt. the equational theory in Fig. 2 on unfocused derivations.

Theorem 4. *Functions* $\mathsf{max}_{\Uparrow}$ *and* $\mathsf{untag}_{\Uparrow}$ *underlie an isomorphism between the set of proofs of a sequent* $S \mid \Gamma \Uparrow A$ *quotiented by the equivalence relation* $\doteq_{\Uparrow}$ *and the set of proofs of* $S^0 \mid \Gamma^0 \Uparrow_{\mathsf{m}} A^0$.

We refer the reader to our Agda formalization for details about the proofs.

Corollary 1. *Functions* $\mathsf{max}_{\Uparrow} \circ \mathsf{focus}$ *and* $\mathsf{emb} \circ \mathsf{untag}_{\Uparrow}$ *underlie an isomorphism between the set of proofs of a sequent* $S \mid \Gamma \vdash A$ *quotiented by the equivalence relation* $\doteq$ *and the set of proofs of* $S^0 \mid \Gamma^0 \Uparrow_{\mathsf{m}} A^0$.

Proof. By Theorems 2 and 4.

5 Conclusions

SkNMILL is a relatively weak logic, low in the substructural hierarchy and with a restricted selection of logical connectives. Nevertheless, its simplicity allows to properly investigate complex proof-theoretic procedure such as maximal multi-focusing, which can be potentially extended to sequent calculi for richer logics. Porting the technique to extensions of SkNMILL with other structural laws, such as full associativity/unitality (recovering the Lambek calculus without left residual) or exchange (as in the sequent calculus of symmetric skew monoidal categories [18]), should be relatively straightforward. Extensions with additive connectives will make things more complicated. To this end, it would be interesting to study semantic approaches to maximal multi-focusing, akin to normalization-by-evaluation [2,19] or (proof-relevant) semantic cut elimination [11].

Uustalu et al. [16] define a normalization procedure for SkNMILL using tags, which is also inspired by focusing. Their canonical derivations arise as normal forms of the confluent and strongly normalizing rewriting system obtained by orienting the equations in Fig. 2 from left to right. This means that, during root-first proof search, invertible rules are again applied first, but the application of non-invertible rules pass and $\multimap$L is prioritized over $\otimes$R. Moreover, focus is released after each application of a non-invertible rule and the asynchronous phase is immediately resumed. Maximal multi-focusing, on the other hand, is unbiased with respect to the application of non-invertible rules. We plan to further investigate the relationship between the normal forms of the two normalization strategies, for SkNMILL and other substructural logics.

References

1. Altenkirch, T., Chapman, J., Uustalu, T.: Monads need not be endofunctors. Log. Methods Comput. Sci. **11**(1), 3 (2015). https://doi.org/10.2168/lmcs-11(1:3)2015
2. Altenkirch, T., Dybjer, P., Hofmann, M., Scott, P.J.: Normalization by evaluation for typed lambda calculus with coproducts. In: Proceedings of 16th Annual IEEE Symposium on Logic in Computer Science, LICS 2001, pp. 303–310. IEEE Computer Society (2001). https://doi.org/10.1109/LICS.2001.932506
3. Andreoli, J.M.: Logic programming with focusing proofs in linear logic. J. Log. Comput. **2**(3), 297–347 (1992). https://doi.org/10.1093/logcom/2.3.297
4. Brock-Nannestad, T., Guenot, N.: Multi-focused cut elimination. Math. Struct. Comput. Sci. **28**(5), 614–650 (2018). https://doi.org/10.1017/S0960129516000451
5. Chaudhuri, K., Hetzl, S., Miller, D.: A multi-focused proof system isomorphic to expansion proofs. J. Log. Comput. **26**(2), 577–603 (2016). https://doi.org/10.1093/logcom/exu030
6. Chaudhuri, K., Miller, D., Saurin, A.: Canonical sequent proofs via multi-focusing. In: Ausiello, G., Karhumäki, J., Mauri, G., Ong, L. (eds.) TCS 2008. IIFIP, vol. 273, pp. 383–396. Springer, Boston, MA (2008). https://doi.org/10.1007/978-0-387-09680-3_26
7. Girard, J.: Linear logic. Theor. Comput. Sci. **50**, 1–102 (1987). https://doi.org/10.1016/0304-3975(87)90045-4

8. Lamarche, F., Retoré, C.: Proof nets for the Lambek calculus - an overview. In: Proceedings of 3rd Roma Workshop on Proofs and Linguistic Categories 1996, pp. 241–262 (1996). https://hal.archives-ouvertes.fr/inria-00098442
9. Liang, C.C., Miller, D.: Focusing and polarization in linear, intuitionistic, and classical logics. Theor. Comput. Sci. **410**(46), 4747–4768 (2009). https://doi.org/10.1016/j.tcs.2009.07.041
10. Moortgat, M.: The Tamari order for d^3 and derivability in semi-associative Lambek-Grishin Calculus. In: Talk at 16th Workshop on Computational Logic and Applications, CLA 2020 (2020). https://cla.tcs.uj.edu.pl/history/2020/pdfs/CLA_slides_Moortgat.pdf
11. Okada, M.: Phase semantic cut-elimination and normalization proofs of first- and higher-order linear logic. Theor. Comput. Sci. **227**(1–2), 333–396 (1999). https://doi.org/10.1016/S0304-3975(99)00058-4
12. Pimentel, E., Nigam, V., Neto, J.: Multi-focused proofs with different polarity assignments. In: Benevides, M.R.F., Thiemann, R. (eds.) Proceedings of 10th Workshop on Logical and Semantic Frameworks, with Applications, LSFA 2015. Electron. Notes Theoretical Computer Science, vol. 323, pp. 163–179. Elsevier (2015). https://doi.org/10.1016/j.entcs.2016.06.011
13. Scherer, G.: Multi-focusing on extensional rewriting with sums. In: Altenkirch, T. (ed.) Proceedings of 13th International Conference on Typed Lambda Calculi and Applications, TLCA 2015. LIPIcs, vol. 38, pp. 317–331. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2015). https://doi.org/10.4230/LIPIcs.TLCA.2015.317
14. Scherer, G., Rémy, D.: Which simple types have a unique inhabitant? In: Proceedings of 20th ACM SIGPLAN International Conference on Functional Programming, ICFP 2015, pp. 243–255. ACM (2015). https://doi.org/10.1145/2784731.2784757
15. Street, R.: Skew-closed categories. J. Pure Appl. Alg. **217**(6), 973–988 (2013). https://doi.org/10.1016/j.jpaa.2012.09.020
16. Uustalu, T., Veltri, N., Wan, C.S.: Proof theory of skew non-commutative MILL. In: Indrzejczak, A., Zawidzki, M. (eds.) Proceedings of 10th International Conference on Non-classical Logics: Theory and Applications, NCL 2022. Electronics Proceedings in Theoretical Computer Science, vol. 358, pp. 118–135. Open Publishing Association (2022). https://doi.org/10.4204/eptcs.358.9
17. Uustalu, T., Veltri, N., Zeilberger, N.: The sequent calculus of skew monoidal categories. In: Joachim Lambek: The Interplay of Mathematics, Logic, and Linguistics. OCL, vol. 20, pp. 377–406. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-66545-6_11
18. Veltri, N.: Coherence via focusing for symmetric skew monoidal categories. In: Silva, A., Wassermann, R., de Queiroz, R. (eds.) WoLLIC 2021. LNCS, vol. 13038, pp. 184–200. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-88853-4_12
19. Veltri, N.: Normalization by evaluation for the lambek calculus. In: Indrzejczak, A., Zawidzki, M. (eds.) Proceedings of 10th International Conference on Non-Classical Logics: Theory and Applications, NCL 2022. Electronics Proceedings in Theoretical Computer Science, vol. 358, pp. 102–117 (2022). https://doi.org/10.4204/EPTCS.358.8
20. Zeilberger, N.: A sequent calculus for a semi-associative law. Log. Methods Comput. Sci. **15**(1), 9 (2019). https://doi.org/10.23638/lmcs-15(1:9)2019

Correction to: Decidability of Modal Logics of Non-k-Colorable Graphs

Ilya Shapirovsky

Correction to:
Chapter 22 in: H. H. Hansen et al. (Eds.): *Logic, Language, Information, and Computation*, LNCS 13923, https://doi.org/10.1007/978-3-031-39784-4_22

The original version of this paper some information in Remark 4, the bibitem [DLW23] and Footnote 2 on page 7 was missing. This has been corrected.

The updated version of this chapter can be found at
https://doi.org/10.1007/978-3-031-39784-4_22

H. H. Hansen et al. (Eds.): WoLLIC 2023, LNCS 13923, p. C1, 2023.
https://doi.org/10.1007/978-3-031-39784-4_25

Author Index

H. H. Hansen et al. (Eds.): WoLLIC 2023, LNCS 13923, p. 395, 2023.
https://doi.org/10.1007/978-3-031-39784-4