Sergei Artemov
Anil Nerode (Eds.)

LNCS 10703

Logical Foundations of Computer Science

International Symposium, LFCS 2018
Deerfield Beach, FL, USA, January 8–11, 2018
Proceedings

LF
CS
2018

Springer

Lecture Notes in Computer Science 10703

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

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

Sergei Artemov · Anil Nerode (Eds.)

Logical Foundations of Computer Science

International Symposium, LFCS 2018
Deerfield Beach, FL, USA, January 8–11, 2018
Proceedings

Springer

Editors

Sergei Artemov
City University of New York
New York, NY
USA

Anil Nerode
Cornell University
Ithaca, NY
USA

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-319-72055-5 ISBN 978-3-319-72056-2 (eBook)
https://doi.org/10.1007/978-3-319-72056-2

Library of Congress Control Number: 2017960856

LNCS Sublibrary: SL1 – Theoretical Computer Science and General Issues

Printed on acid-free paper

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

Preface

The Symposium on Logical Foundations of Computer Science provides a forum for the fast-growing body of work on the logical foundations of computer science, e.g., those areas of fundamental theoretical logic related to computer science. The LFCS series began with "Logic at Botik," Pereslavl-Zalessky, 1989, which was co-organized by Albert R. Meyer (MIT) and Michael Taitslin (Tver). After that, organization passed to Anil Nerode.

Currently LFCS is governed by a Steering Committee consisting of Anil Nerode (General Chair), Stephen Cook, Dirk van Dalen, Yuri Matiyasevich, Gerald Sacks, Andre Scedrov, and Dana Scott.

The 2018 Symposium on Logical Foundations of Computer Science (LFCS 2018) took place at the Wyndham Deerfield Beach Resort, Deerfield Beach, Florida, USA, during January 8–11, 2018. This volume contains the extended abstracts of talks selected by the Program Committee for presentation at LFCS 2018.

The scope of the symposium is broad and includes constructive mathematics and type theory, homotopy type theory, logic, automata and automatic structures, computability and randomness, logical foundations of programming, logical aspects of computational complexity, parameterized complexity, logic programming and constraints, automated deduction and interactive theorem proving, logical methods in protocol and program verification, logical methods in program specification and extraction, domain theory logics, logical foundations of database theory, equational logic and term rewriting, lambda and combinatory calculi, categorical logic and topological semantics, linear logic, epistemic and temporal logics, intelligent and multiple-agent system logics, logics of proof and justification, non-monotonic reasoning, logic in game theory and social software, logic of hybrid systems, distributed system logics, mathematical fuzzy logic, system design logics, and other logics in computer science.

We thank the authors and reviewers for their contributions. We acknowledge the support of the U.S. National Science Foundation, The Association for Symbolic Logic, Cornell University, the Graduate Center of the City University of New York, and Florida Atlantic University.

October 2017

Anil Nerode
Sergei Artemov

Organization

Steering Committee

Stephen Cook	University of Toronto, Canada
Yuri Matiyasevich	Steklov Mathematical Institute, St. Petersburg, Russia
Anil Nerode (General Chair)	Cornell University, USA
Gerald Sacks	Harvard University, USA
Andre Scedrov	University of Pennsylvania, USA
Dana Scott	Carnegie-Mellon University, USA
Dirk van Dalen	Utrecht University, The Netherlands

Program Committee

Sergei Artemov (Chair)	The City University of New York, USA
Eugene Asarin	Université Paris Diderot - Paris 7, France
Steve Awodey	Carnegie Mellon University, USA
Matthias Baaz	The Vienna University of Technology, Austria
Lev Beklemishev	Steklov Mathematical Institute, Moscow, Russia
Andreas Blass	University of Michigan, Ann Arbor, USA
Samuel Buss	University of California, San Diego, USA
Robert Constable	Cornell University, USA
Thierry Coquand	University of Gothenburg, Sweden
Nachum Dershowitz	Tel Aviv University, Israel
Michael Fellows	University of Bergen, Norway
Melvin Fitting	The City University of New York, USA
Sergey Goncharov	Sobolev Institute of Mathematics, Novosibirsk, Russia
Denis Hirschfeldt	University of Chicago, USA
Martin Hyland	University of Cambridge, UK
Rosalie Iemhoff	Utrecht University, The Netherlands
Hajime Ishihara	Japan Advanced Institute of Science and Technology, Kanazawa, Japan
Bakhadyr Khoussainov	The University of Auckland, New Zealand
Roman Kuznets	The Vienna University of Technology, Austria
Daniel Leivant	Indiana University Bloomington, USA
Robert Lubarsky	Florida Atlantic University, USA
Victor Marek	University of Kentucky, Lexington, USA
Lawrence Moss	Indiana University Bloomington, USA
Anil Nerode	Cornell University, USA
Hiroakira Ono	Japan Advanced Institute of Science and Technology, Kanazawa, Japan
Alessandra Palmigiano	Delft University of Technology, The Netherlands
Ruy de Queiroz	The Federal University of Pernambuco, Recife, Brazil

Ramaswamy Ramanujam	The Institute of Mathematical Sciences, Chennai, India
Michael Rathjen	University of Leeds, UK
Jeffrey Remmel	University of California, San Diego, USA
Andre Scedrov	University of Pennsylvania, USA
Helmut Schwichtenberg	University of Munich, Germany
Philip Scott	University of Ottawa, Canada
Alex Simpson	University of Ljubljana, Slovenia
Sonja Smets	University of Amsterdam, The Netherlands
Sebastiaan Terwijn	Radboud University Nijmegen, The Netherlands
Alasdair Urquhart	University of Toronto, Canada

Additional Reviewers

Josef Berger
S. P. Suresh
Catalin Dima
Giuseppe Greco
Yanjing Wang
Heinrich Wansing
Toshiyasu Arai
Lutz Straßburger
Rohit Parikh

Contents

The Completeness Problem for Modal Logic

Antonis Achilleos(✉)

School of Computer Science, Reykjavik University, Reykjavik, Iceland
antonios@ru.is

Abstract. We introduce the completeness problem for Modal Logic and examine its complexity. For a definition of completeness for formulas, given a formula of a modal logic, the completeness problem asks whether the formula is complete for that logic. We discover that completeness and validity have the same complexity — with certain exceptions for which there are, in general, no complete formulas. To prove upper bounds, we present a non-deterministic polynomial-time procedure with an oracle from PSPACE that combines tableaux and a test for bisimulation, and determines whether a formula is complete.

Keywords: Modal logic · Completeness · Computational complexity Bisimulation

1 Introduction

For a modal logic l, we call a modal formula φ *complete* when for every modal formula ψ on the same propositional variables as φ, we can derive from φ in l either the formula ψ or its negation. For different modal logics l, we examine the following problem: given a modal formula φ, is it complete for l? We call this the completeness problem for l and we examine its complexity. Our main results show that the completeness problem has the same complexity as provability, at least for the logics we consider.

Given Modal Logic's wide area of applications and the importance of logical completeness in general, we find it surprising that, to the best of our knowledge, the completeness problem for Modal Logic has not been studied as a computational problem so far. On the other hand, the complexity of satisfiability (and thus validity) for Modal Logic has been studied extensively — for example, see [1–3]. We examine the completeness problem for several well-known modal logics, namely the extensions of **K** by the axioms Factivity, Consistency, Positive Introspection, and Negative Introspection (also known as T, D, 4, and 5, respectively) — i.e. the ones between **K** and **S5**. We discover that the complexity of provability and completeness tend to be the same: the completeness problem

This research was partly supported by the project "TheoFoMon: Theoretical Foundations for Monitorability" (grant number: 163406-051) of the Icelandic Research Fund.

S. Artemov and A. Nerode (Eds.): LFCS 2018, LNCS 10703, pp. 1–21, 2018.
https://doi.org/10.1007/978-3-319-72056-2_1

is PSPACE-complete if the logic does not have Negative Introspection and it is coNP-complete otherwise. There are exceptions: for certain logics (**D** and **T**), the completeness problem as we define it is trivial, as these logics have no finite complete theories.

Our motivation partly comes from [4] (see also [5]), where Artemov raises the following issue. It is the usual practice in Game Theory (and Epistemic Game Theory) to reason about a game based on a model of the game description. On the other hand, it is often the case in an epistemic setting that the game specification is not complete, thus any conclusions reached by examining any single model are precarious. He thus argues for the need to verify the completeness of game descriptions, and proposes a syntactic, proof-centered approach, which is more robust and general, and which is based on a syntactic formal description of the game. Artemov's approach is more sound, in that it allows one to draw only conclusions that can be safely derived from the game specification; on the other hand, the model-based approach has been largely successful in Game Theory for a long time. He explain that if we can determine that the syntactic specification of a game is complete, then the syntactic and semantic approaches are equivalent and we can describe the game efficiently, using one model. Furthermore, he presents a complete and an incomplete formulation of the Muddy Children puzzle.

For a formula–specification φ (for example, a syntactic description of a game), if we are interested in the formulas we can derive from φ (the conclusions we can draw from the game description), knowing that φ is complete can give a significant computational advantage. If φ is complete and consistent, for a model $\mathcal{M}$ for φ, ψ can be derived from φ exactly when ψ is satisfied in $\mathcal{M}$ at the same state as φ. Thus, knowing that φ is complete allows us to reduce a derivability problem to a model checking problem, which is easier to solve (see, for example, [3]). This approach may be useful when we need to examine multiple conclusions, especially if the model for φ happens to be small. On the other hand, if we discover that φ is incomplete, then, as a specification it may need to be refined.

Notions similar to complete formulas have been studied before. Characteristic formulas allow one to characterize a state's equivalence class for a certain equivalence relation. In our case, the equivalence relation is bisimulation on states of (finite) Kripke models and the notions of characteristic and complete formulas collapse, by the Hennessy-Milner Theorem [6], in that a formula is complete for one of the logics we consider if and only if it is characteristic for a state in a model for that logic. A construction of characteristic formulas for variants of CCS processes [7] was introduced in [8]. This construction allows one to verify that two CCS processes are equivalent by reducing this problem to model checking. Similar constructions were studied later in [9–11] for instance.

Normal forms for Modal Logic were introduced by Fine [12] and they can be used to prove soundness, completeness, and the finite frame property for several modal logics with respect to their classes of frames. Normal forms are modal formulas that completely describe the behavior of a Kripke model up to a certain distance from a state, with respect to a certain number of propositional variables. Therefore, every complete formula is equivalent to a normal form, but not all normal forms are complete, as they may be agnostic with respect to states located further away. We may define that a formula is complete up to depth d for logic l when it is equivalent to a normal form of modal depth (the nesting depth of a formula's modalities) at most d. We briefly discuss these topics in Sect. 6.

We focus on a definition of completeness that emphasizes on the formula's ability to either affirm or reject every possible conclusion. We can also consider a version of the problem that asks to determine if a formula is complete up to its modal depth — that is, whether it is equivalent to a normal form. If we are interested in completely describing a setting, the definition we use for completeness is more appropriate. However, it is not hard to imagine situations where this variation of completeness is the notion that fits better, either as an approximation on the epistemic depth agents reason with, or, perhaps, as a description of process behavior for a limited amount of time. We briefly examine this variation in Sect. 6.

Overview. Section 2 provides background on Modal Logic, bisimulation, and relevant complexity results. In Sect. 3, we draw our first conclusions about the completeness problem in relation to bisimulation and give our first complexity result for logics with Negative Introspection. In Sect. 4, we examine different logics and in which cases for each of these logics the completeness problem is non-trivial. In Sect. 5, we examine the complexity of the completeness problem. We first present a general lower bound. For logics with Negative Introspection we prove coNP-completeness. For the remaining logics — the ones without Negative Introspection for which the problem is not trivial — we present a non-deterministic polynomial-time procedure with an oracle from PSPACE that accepts incomplete formulas, as the section's main theorem, Theorem 6 demonstrates. This proves that the completeness problem for these cases is PSPACE-complete. These complexity results are summarized in Table 1. In Sect. 6, we consider variations of the problem and draw further conclusions. Full proofs for our results can be found in the extended version, [13].

2 Background

We present needed background on Modal Logic, its complexity, and bisimulation, and we introduce the completeness problem. For an overview of Modal Logic and its complexity, we refer the reader to [3,14,15].

2.1 Modal Logic

We assume a countably infinite set of propositional variables $p_1, p_2, \ldots$. Literals are all p and $\neg p$, where p is a propositional variable. Modal formulas are constructed from literals, the constants $\bot, \top$, the usual operators for conjunction and disjunction $\land, \lor$, and the dual modal operators, $\Box$ and $\Diamond$:

$$\varphi ::= \bot \mid \top \mid p \mid \neg p \mid \varphi \land \varphi \mid \varphi \lor \varphi \mid \Box\varphi \mid \Diamond\varphi.$$

The negation $\neg\varphi$ of a modal formula, implication $\varphi \rightarrow \psi$, and $\varphi \leftrightarrow \psi$ are constructed as usual. The language described by the grammar above is called L.

For a finite set of propositional variables P, $L(P) \subseteq L$ is the set of formulas that use only variables from P. For a formula φ, $P(\varphi)$ is the set of propositional variables that appear in φ, so $\varphi \in L(P(\varphi))$. If $\varphi \in L$, then $sub(\varphi)$ is the set of subformulas of φ and $\overline{sub}(\varphi) = sub(\varphi) \cup \{\neg\psi \mid \psi \in sub(\varphi)\}$. For Φ a nonempty finite subset of L, $\bigwedge \Phi$ is a conjunction of all elements of Φ and $\bigwedge \emptyset = \top$; we define $\bigvee \Phi$ similarly. The modal depth $md(\varphi)$ of φ is the largest nesting depth of its modal operators; the size of φ is $|\varphi| = |sub(\varphi)|$. For every $d \geq 0$, $\overline{sub}_d(\varphi) = \{\psi \in \overline{sub}_d(\varphi) \mid md(\psi) \leq d\}$.

Normal modal logics use all propositional tautologies and axiom K, Modus Ponens, and the Necessitation Rule:

$$K : \Box\varphi \wedge \Box(\varphi \rightarrow \psi) \rightarrow \Box\psi; \qquad \frac{\varphi \quad \varphi \rightarrow \psi}{\psi}; \qquad \frac{\varphi}{\Box\varphi}.$$

The logic that has *exactly* these axioms and rules is the smallest normal modal logic, **K**. We can extend **K** with more axioms:

$$D : \Diamond\top; \qquad T : \Box\varphi \rightarrow \varphi; \qquad 4 : \Box\varphi \rightarrow \Box\Box\varphi; \qquad 5 : \Diamond\varphi \rightarrow \Box\Diamond\varphi.$$

We consider modal logics that are formed from a combination of these axioms. Of course, not all combinations make sense: axiom D (also called the Consistency axiom) is a special case of T (the Factivity axiom). Axiom 4 is called Positive Introspection and 5 is called Negative Introspection. Given a logic l and axiom a, $l+a$ is the logic that has as axioms all the axioms of l and a. Logic **D** is **K**$+D$, **T** is **K**$+T$, **K4** $=$ **K**$+4$, **D4** $=$ **K**$+D+4 =$ **D**$+4$, **S4** $=$ **K**$+T+4 =$ **T**$+4 =$ **K4**$+T$, **KD45** $=$ **D4** $+ 5$, and **S5** $=$ **S4** $+ 5$. From now on, unless we explicitly say otherwise, by a logic or a modal logic, we mean one of the logics we defined above. We use $\vdash_l \varphi$ to mean that φ can be derived from the axioms and rules of l; when l is clear from the context, we may drop the subscript and just write $\vdash$.

A Kripke model is a triple $\mathcal{M} = (W, R, V)$, where W is a nonempty set of states (or worlds), $R \subseteq W \times W$ is an accessibility relation and V is a function that assigns to each state in W a set of propositional variables. If P is a set of propositional variables, then for every $a \in W$, $V_P(a) = V(a) \cap P$. To ease notation, when $(s,t) \in R$ we usually write sRt.

Truth in a Kripke model is defined through relation $\models$ in the following way: $\mathcal{M}, a \models p$ iff $p \in V(a)$, and

$\mathcal{M}, a \not\models \bot$ and $\mathcal{M}, a \models \top$;
$\mathcal{M}, a \models p$ iff $p \in V(a)$ and $\mathcal{M}, a \models \neg p$ iff $p \notin V(a)$;
$\mathcal{M}, a \models \varphi \wedge \psi$ iff both $\mathcal{M}, a \models \varphi$ and $\mathcal{M}, a \models \psi$;
$\mathcal{M}, a \models \varphi \vee \psi$ iff $\mathcal{M}, a \models \varphi$ or $\mathcal{M}, a \models \psi$;
$\mathcal{M}, a \models \Diamond\varphi$ iff there is some $b \in W$ such that aRb and $\mathcal{M}, b \models \varphi$; and
$\mathcal{M}, a \models \Box\varphi$ iff for all $b \in W$ such that aRb it is the case that $\mathcal{M}, b \models \varphi$.

If $\mathcal{M}, a \models \varphi$, we say that φ is true/satisfied in a of $\mathcal{M}$. (W, R) is called a *frame*. We call a Kripke model (W, R, V) (resp. frame (W, R)) finite if W is finite.[1] If $\mathcal{M}$ is a model (for logic l) and a is a state of $\mathcal{M}$, then $(\mathcal{M}, a)$ is a pointed model (resp. for l).

Each modal logic l is associated with a class of frames $F(l)$, that includes all frames (W, R) for which R meets certain conditions, depending on the logic's axioms. If l has axiom:

D**,** then R must be serial (for every state $a \in W$ there must be some $b \in W$ such that aRb);
T**,** then R must be reflexive (for all $a \in W$, aRa);
4**,** then R must be transitive (if $aRbRc$, then aRc);
5**,** then R must be euclidean (if aRb and aRc, then bRc).

A model (W, R, V) is a model for a logic l if and only if $(W, R) \in F(l)$. We call a formula satisfiable for logic l, if it is satisfied in a state of a model for l. We call a formula valid for logic l, if it is satisfied in all states of all models for l.

Theorem 1 (Completeness, Finite Frame Property). *A formula φ is valid for l if and only if it is provable in l; φ is satisfiable for l if and only if it is satisfied in a finite model for l.*

For the remainder of this paper we only consider finite Kripke models and frames. For a finite model $\mathcal{M} = (W, R, V)$, we define $|\mathcal{M}| = |W| + |R|$.

Definition 1. *A formula φ is called* complete *for logic l when for every $\psi \in L(P(\varphi))$, $\vdash_l \varphi \rightarrow \psi$ or $\vdash_l \varphi \rightarrow \neg\psi$; otherwise, it is* incomplete *for l.*

By Theorem 1, φ is complete for l exactly when for every $\psi \in L(P(\varphi))$, either ψ or its negation is true at every (finite) pointed model for l that satisfies φ.

2.2 Bisimulation

An important notion in Modal Logic (and other areas) is that of bisimulation. Let P be a (finite) set of propositional variables. For Kripke models $\mathcal{M} = (W, R, V)$ and $\mathcal{M}' = (W', R', V')$, a non-empty relation $\mathcal{R} \subseteq W \times W'$ is a *bisimulation* (respectively, bisimulation modulo P) from $\mathcal{M}$ to $\mathcal{M}'$ when the following conditions are satisfied for all $(s, s') \in \mathcal{R}$:

- $V(s) = V'(s')$ (resp. $V_P(s) = V'_P(s')$).
- For all $t \in W$ such that sRt, there exists $t' \in W'$ s.t. $(t, t') \in \mathcal{R}$ and $s'R't'$.
- For all $t' \in W'$ such that $s'R't'$, there exists $t \in W$ s.t. $(t, t') \in \mathcal{R}$ and sRt.

[1] According to our definition, for a finite model $\mathcal{M} = (W, R, V)$ and $a \in W$, $V(a)$ can be infinite. However, we are mainly interested in (W, R, V_P) for finite sets of propositions P, which justifies calling $\mathcal{M}$ finite.

We call pointed models $(\mathcal{M}, a), (\mathcal{M}', a')$ *bisimilar* (resp. bisimilar modulo P) and write $(\mathcal{M}, a) \sim (\mathcal{M}', a')$ (resp. $(\mathcal{M}, a) \sim_P (\mathcal{M}', a')$) if there is a bisimulation (resp. bisimulation modulo P) $\mathcal{R}$ from $\mathcal{M}$ to $\mathcal{M}'$, such that $a\mathcal{R}a'$. If $(\mathcal{M}, a)$ is a pointed model, and P a set of propositional variables, then $Th_P(\mathcal{M}, a) = \{\varphi \in L(P) \mid \mathcal{M}, a \models \varphi\}$. We say that two pointed models are equivalent and write $(\mathcal{M}, a) \equiv_P (\mathcal{M}', a')$ when $Th_P(\mathcal{M}, a) = Th_P(\mathcal{M}', a')$. The following simplification of the Hennessy-Milner Theorem [6] gives a useful characterization of pointed model equivalence; Proposition 1 is its direct consequence.

Theorem 2 (Hennessy-Milner Theorem). *If $(\mathcal{M}, a)$, $(\mathcal{M}', a')$ are finite pointed models, then*

$$(\mathcal{M}, a) \equiv_P (\mathcal{M}', a') \text{ if and only if } (\mathcal{M}, a) \sim_P (\mathcal{M}', a').$$

Proposition 1. *A formula φ is complete for a logic l if and only if for every two pointed models $(\mathcal{M}, a)$ and $(\mathcal{M}', a')$ for l, if $\mathcal{M}, a \models \varphi$ and $\mathcal{M}', a' \models \varphi$, then $(\mathcal{M}, a) \sim_P (\mathcal{M}', a')$.*

Paige and Tarjan in [16] give an efficient algorithm for checking whether two pointed models are bisimilar. Theorem 3 is a variation on their result to account for receiving the set P of propositional variables as part of the algorithm's input.

Theorem 3. *There is an algorithm which, given two pointed models $(\mathcal{M}, a)$ and $(\mathcal{M}', a')$ and a finite set of propositional variables P, determines whether $(\mathcal{M}, a) \sim_P (\mathcal{M}', a')$ in time $O(|P| \cdot (|\mathcal{M}| + |\mathcal{M}'|) \cdot \log(|\mathcal{M}| + |\mathcal{M}'|))$.*

2.3 The Complexity of Satisfiability

For logic l, the satisfiability problem for l, or l-satisfiability asks, given a formula φ, if φ is satisfiable. The provability problem for l asks if $\vdash_l \varphi$.

The classical complexity results for Modal Logic are due to Ladner [1], who established PSPACE-completeness for the satisfiability of **K**, **T**, **D**, **K4**, **D4**, and **S4** and NP-completeness for the satisfiability of **S5**. Halpern and Rêgo later characterized the NP–PSPACE gap by the presence or absence of Negative Introspection [2], resulting in Theorem 4.

Theorem 4. *If $l \in \{$**K**, **T**, **D**, **K4**, **D4**, **S4**$\}$, then l-provability is* PSPACE-*complete and $l + 5$-provability is* coNP-*complete.*

3 The Completeness Problem and Axiom 5

The completeness problem for l asks, given a formula φ, if φ is complete for l. In this section, we explain how to adjust Halpern and Rêgo's techniques from [2] to prove similar complexity bounds for the completeness problem for logics with Negative Introspection. In the course of proving the coNP upper bound for logics with Negative Introspection, Halpern and Rêgo give in [2] a construction that provides a small model for a satisfiable formula. We can adjust parts of their construction and conclude with Corollary 2 and from that, Lemma 1 and Corollary 1. The remaining results in this section are consequence of these.

For a logic $l+5$, we call a pointed model $(\mathcal{M}, s)$ for $l+5$ *flat* when

- $\mathcal{M} = (\{s\} \cup W, R, V)$;
- $R = R_1 \cup R_2$, where $R_1 \subseteq \{s\} \times W$ and R_2 is an equivalence relation on W; and
- if $l \in \{\mathbf{T}, \mathbf{S4}\}$, then $s \in W$.

Lemma 1 informs us that flat models are a normal form for models of logics with axiom 5. Negative Introspection. and it is part of the construction from [2].

Lemma 1. *Every pointed $l+5$-model $(\mathcal{M}, s)$ is bisimilar to a flat pointed $l+5$-model.*

Proof. Let W' be the set of states of $\mathcal{M}$ reachable from s and R the restriction of the accessibility relation of $\mathcal{M}$ on W'. It is easy to see that the identity relation is a bisimulation from $\mathcal{M}$ to $\mathcal{M}'$, so $(\mathcal{M}, s) \sim (\mathcal{M}', s)$; let $W = \{w \in W' \mid \exists w' R w\}$. Therefore $W' = W \cup \{s\}$ and if $l \in \{\mathbf{T}, \mathbf{S4}\}$, then $s \in W$. Since $\mathcal{M}$ is an $l+5$-model, R is euclidean. Therefore, the restriction of R on W is reflexive. This in turn means that R is symmetric in W: if $a, b \in W$ and aRb, since aRa, we also have bRa. Finally, R is transitive in W: if $aRbRc$ and $a, b, c \in W$, then bRa, so aRc. Therefore R is an equivalence relation when restricted on W. □

The construction from [1,2] continues to filter the states of the flat model, resulting in a small model for a formula φ. Using this construction, Halpern and Rêgo prove Corollary 1 [2]; the NP upper bound for $l+5$-satisfiability of Theorem 4 is a direct consequence.

Corollary 1. *Formula φ is $l+5$-satisfiable if and only if it is satisfied in a flat $l+5$-model of $O(|\varphi|)$ states.*

Since we are asking whether a formula is complete, instead of whether it is satisfiable, we want to be able to find two small non-bisimilar models for φ when φ is incomplete. For this, we need a characterization of bisimilarity between flat models.

Lemma 2. *Flat pointed models $(\mathcal{M}, a) = (\{a\} \cup W, R, V)$ and $(\mathcal{M}', a') = (\{a'\} \cup W', R', V')$ are bisimilar modulo P if and only if $V_P(a) = V_P(a')$ and:*

- *for every $b \in W$, there is some $b' \in W'$ such that $V_P(b) = V'_P(b')$;*
- *for every $b' \in W'$, there is some $b \in W'$ such that $V_P(b) = V'_P(b')$;*
- *for every $b \in W$, if aRb, then there is a $b' \in W'$ such that $a'Rb'$ and $V_P(b) = V'_P(b')$; and*
- *for every $b' \in W'$, if $a'Rb'$, then there is a $b \in W'$ such that aRb and $V_P(b) = V'_P(b')$.*

Proof. If these conditions are met, we can define bisimulation $\mathcal{R}$ such that $a\mathcal{R}a'$ and for $b \in W$ and $b' \in W'$, $b\mathcal{R}b'$ iff $V_P(b) = V'_P(b')$; on the other hand, if there is a bisimulation, then it is not hard to see by the definition of bisimulation that these conditions hold — for both claims, notice that the conditions above, given the form of the models, correspond exactly to the conditions from the definition of bisimulation. □

This gives us Corollary 2, which is a useful characterization of incomplete formulas.

Corollary 2. *Formula φ is incomplete for $l+5$ if and only if it has two non-bisimilar flat pointed models for $l+5$ of at most $O(|\varphi|)$ states.*

Proof. If φ has two non-bisimilar pointed models for $l+5$, then by Theorem 2, it is incomplete. On the other hand, if φ is incomplete, again by Theorem 2 and Lemma 1, φ has two non-bisimilar flat pointed models, $(\mathcal{M}, a) = (\{a\} \cup W, R, V)$ and $(\mathcal{M}', a') = (\{a'\} \cup W', R', V')$. By Lemma 2 and without loss of generality, we can distinguish three cases:

- there is some $p \in V_P(a) \setminus V_P(a')$: in this case let $\psi = p$;
- there is some $b \in W$, such that for all $b' \in W'$, $V_P(b) \neq V'_P(b')$: in this case let $\psi = \Diamond\Diamond(\bigwedge V_P(b) \wedge \neg \bigvee(P \setminus V_P(b)))$;
- there is some $b \in W$, such that aRb and for all $b' \in W'$ such that $a'Rb'$, $V_P(b) \neq V'_P(b')$: in this case let $\psi = \Diamond(\bigwedge V_P(b) \wedge \neg \bigvee(P \setminus V_P(b)))$.

In all these cases, both $\varphi \wedge \psi$ and $\varphi \wedge \neg\psi$ are satisfiable and of size $O(|\varphi|)$, so by Corollary 1, each is satisfied in a non-bisimilar flat pointed model for $l+5$ of at most $O(|\varphi|)$ states. □

Our first complexity result is a consequence of Corollary 2 and Theorem 3:

Proposition 2. *The completeness problem for logic $l+5$ is in* coNP.

Proof. By Corollary 2 and Theorem 3. □

In the following, when P is evident, we will often omit any reference to it and instead of bisimulation modulo P, we will call the relation simply bisimulation.

4 The Completeness Problem and Triviality

The first question we must answer concerning the completeness problem for l is whether there are any satisfiable and complete formulas for l. If not, then the problem is trivial. We examine this question with parameters the logic l and whether P, the set of propositional variables we use, is empty or not. If for a logic l the problem is nontrivial, then we give a complete formula φ^l_P that uses exactly the propositional variables in P. We see that for $P = \emptyset$, completeness can be trivial for another reason: for some logics, when $P = \emptyset$, all formulas are complete. On the other hand, when $P \neq \emptyset$, $\bigwedge P$ is incomplete for every logic.

4.1 Completeness and K

Whether $P = \emptyset$ or not, completeness is nontrivial for **K** and **K4**: let $\varphi_P^{\mathbf{K}} = \varphi_P^{\mathbf{K4}} = \bigwedge P \wedge \Box\bot$ for every finite P. Formula $\top$ is incomplete for **K** and **K4**.

Lemma 3. *Formula $\bigwedge P \wedge \Box\bot$ is complete and satisfiable for **K** and for **K4**.*

Proof. A model that satisfies $\varphi_P^{\mathbf{K}}$ is $\mathcal{M} = (\{a\}, \emptyset, V)$, where $V(a) = P$. If there is another model $\mathcal{M}', a' \models \varphi_P^{\mathbf{K}}$, then $\mathcal{M}', a' \models \Box\bot$, so there are no accessible worlds from a' in $\mathcal{M}'$; therefore, $\mathcal{R} = \{(a, a')\}$ is a bisimulation. □

Notice that if φ is complete for l, then it is complete for every extension of l. Thus, $\varphi_P^{\mathbf{K}}$ is complete for all other logics. However, we are looking for *satisfiable and complete* formulas for each logic, so finding one complete formula for **K** is not enough. On the other hand, if l' is an extension of l (by a set of axioms) and a formula φ is complete for l and satisfiable for l', then we know that φ is satisfiable and complete for all logics between (and including) l and l'. Unfortunately, the following lemma demonstrates that we cannot use this convenient observation to reuse $\varphi_P^{\mathbf{K}}$ — except perhaps for **K5** and **K45**, but these can be handled just as easily together with the remaining logics with Negative Introspection.

4.2 Completeness and Consistency

When l has axiom T or D, but not 4 or 5, P determines if a formula is complete:

Lemma 4. *Let l be either **D** or **T**. A satisfiable formula $\varphi \in L$ is complete with respect to l if and only if $P(\varphi) = \emptyset$.*

Proof. When $P = \emptyset$, all models are bisimilar through the total bisimulation; therefore, all formulas φ, where $P(\varphi) = \emptyset$ are trivially complete. We now consider the case for $P \neq \emptyset$; notice that we can assume that $l = \mathbf{D}$, as **D** is contained in **T**. Let the modal depth of φ be d and let $\mathcal{M}, a \models \varphi$, where $\mathcal{M} = (W, R, V)$; let $x \notin W^*$, $a_0 = a$, and

$$\Pi_d = \{a_0 \cdots a_k \in W^* \mid k \leq d \text{ and for all } 0 \leq i < k,\ a_i R a_{i+1}\}.$$

Then, we define $\mathcal{M}_1' = (W', R', V_1')$ and $\mathcal{M}_2' = (W', R', V_2')$, where

$$\begin{aligned} W' &= \Pi_d \cup \{x\}; \\ R' &= \{(\alpha, \alpha b) \in W'^2 \mid b \in W\} \ \cup\ \{(a_0 a_1 \cdots a_d, x) \in W'^2\} \ \cup\ \{(x, x)\} \\ V_i'(\alpha b) &= V(b),\ \text{for } i = 1, 2,\ 0 \leq |\alpha| < d; \\ V_1'(x) &= \emptyset; \quad \text{and} \quad V_2'(x) \ = \ P. \end{aligned}$$

To prove that $\mathcal{M}_1', a \models \varphi$ and $\mathcal{M}_2', a \models \varphi$, we prove that for $\psi \in sub(\varphi)$, for every $i = 1, 2$ and $w = a_0 \cdots a_k \in \Pi_d$, where $k \leq d - md(\psi)$, $\mathcal{M}_i', w \models \psi$ if and only if $\mathcal{M}, a_k \models \psi$. We use induction on ψ. If ψ is a literal or a constant, the claim is immediate and so are the cases of the $\wedge, \vee$ connectives. If $\psi = \Box\psi'$, then $md(\psi') = md(\psi) - 1$; $\mathcal{M}_i', w \models \psi$ iff for every $wR'w'$, $\mathcal{M}_i', w' \models \psi'$ iff for every $a_k R' b$, $\mathcal{M}, b \models \psi'$ (by the Inductive Hypothesis) iff $\mathcal{M}, a_k \models \psi$; the case of $\psi = \Diamond\psi'$ is symmetric.

If $(\mathcal{M}'_1, a) \sim (\mathcal{M}'_2, a)$ through bisimulation $\mathcal{R}$ from $\mathcal{M}'_1$ to $\mathcal{M}'_2$, then notice that in both models any sufficiently long path from a will end up at x; therefore, by the conditions of bisimulation, $x\mathcal{R}x$, which is a contradiction, since $V'_1(x) \neq V'_2(x)$. So, φ is satisfied in two non-bisimilar models for **D**. □

4.3 Completeness, Consistency, and Positive Introspection

For every finite P, let $\varphi_P^{\mathbf{D4}} = \varphi_P^{\mathbf{S4}} = \bigwedge P \wedge \Box \bigwedge P$. As the following lemma demonstrates, $\varphi_P^{\mathbf{D4}}$ is a complete formula for **D4** and **S4**.

Lemma 5. *For every finite P, $\varphi_P^{\mathbf{D4}}$ is complete for **D4** and **S4**; all formulas in $L(\emptyset)$ are complete for **D4** and **S4**.*

Proof. Let $\mathcal{M}, a \models \varphi_P^{\mathbf{D4}}$ and $\mathcal{M}', a' \models \varphi_P^{\mathbf{D4}}$; let $\mathcal{R}$ be the relation that connects all states of $\mathcal{M}$ that are reachable from a (including a) to all states of $\mathcal{M}'$ that are reachable from a' (including a'); it is not hard to verify that $\mathcal{R}$ is a bisimulation. Notice that if $P = \emptyset$, then $\varphi_P^{\mathbf{D4}}$ is a tautology, thus all formulas are complete. □

It is straightforward to see that $\varphi_P^{\mathbf{D4}}$ is satisfiable for every logic l: consider a model based on any frame for l, where $\bigwedge P$ holds at every state. Therefore:

Corollary 3. *$\varphi^{\mathbf{D4}}$ is satisfiable and complete for every extension of* **D4**.[2]

4.4 Consistency and Negative Introspection

For logic $l = l' + 5$, let $\varphi_P^l = \bigwedge P \wedge \Diamond\Box \bigwedge P$.

Lemma 6. *For any logic $l = l' + 5$, φ_P^l is a satisfiable complete formula for l.*

Proof. By Lemma 1, φ_P^l is complete. It is satisfied in $(\{a\}, \{(a,a)\}, V)$, where $V(a) = P$. □

When $P = \emptyset$, we can distinguish two cases. If $l' \in \{\mathbf{D}, \mathbf{D4}, \mathbf{T}, \mathbf{S4}\}$, then $\varphi_\emptyset^l$ is a tautology, therefore all formulas in $L(P)$ are complete for l.[3] If $l' \in \{\mathbf{K}, \mathbf{K4}\}$, then there are exactly two non-bisimilar modulo $\emptyset$ models for l; Therefore, if $P = \emptyset$ the completeness problem for **K5** and **K45** is not trivial, but it is easy to solve: a formula with no propositional variables is complete for $l \in \{\mathbf{K5}, \mathbf{K45}\}$ if it is satisfied in at most one of these two models.

Corollary 4. *If $P = \emptyset$, the completeness problem for **K5** and **K45** is in* P.

[2] Although for the purposes of this paper we only consider a specific set of modal logics, it is interesting to note that the corollary can be extended to a much larger class of logics.

[3] This is also a corollary of Lemma 4, as these are extensions of **D** and **T**.

4.5 Completeness and Modal Logics

A logic l has a nontrivial completeness problem if for $P \neq \emptyset$, there are complete formulas for l. From the logics we examined, only **D** and **T** have trivial completeness problems. Table 1 summarizes the results of this section and of Sect. 5 regarding the completeness problem. As the table demonstrates, we can distinguish the following cases. For **K**, the completeness problem is non-trivial and PSPACE-complete; this does not change when we add axiom 4. Once we add axiom D to **K**, but not 4 or 5, the completeness problem becomes trivial; adding the stronger axiom T does not change the situation. Adding both 4 and D or T to **K** makes completeness PSPACE-complete again, except when $P = \emptyset$. Regardless of other axioms, if the logic has Negative Introspection, completeness is coNP-complete — unless $P = \emptyset$, when the situation depends on whether the logic has D (or the stronger T) or not.

Table 1. The complexity of the completeness problem for different modal logics. Trivial (all) indicates that all formulas in this case are complete for the logic; trivial (none) indicates that there is no satisfiable, complete formula for the logic.

Modal logic	$P = \emptyset$	$P \neq \emptyset$
K, **K4**	PSPACE-complete	PSPACE-complete
D, **T**	Trivial (all)	Trivial (none)
D4, **S4**	Trivial (all)	PSPACE-complete
K5, **K45**	In P	coNP-complete
$l + 5$, $l \neq \mathbf{K}, \mathbf{K4}$	Trivial (all)	coNP-complete

5 The Complexity of Completeness

Our main result is that for a modal logic l, the completeness problem has the same complexity as provability for l, as long as we allow for propositional variables in a formula and l-completeness is nontrivial (see Table 1). For the lower bounds, we consider hardness under polynomial-time reductions. As the hardness results are relative to complexity classes that include coNP, these reductions suffice.

5.1 A Lower Bound

We present a lower bound for the complexity of the completeness problem: that the completeness problem is at least as hard as provability for a logic, as long as it is nontrivial.

Theorem 5. *Let l be a logic that has a nontrivial completeness problem and let C be a complexity class. If l-provability is C-hard, then the completeness problem for l is C-hard.*

Proof. To prove the theorem we present a reduction from l-provability to the completeness problem for l. From a formula φ, the reduction constructs in polynomial time a formula φ_c, such that φ is provable if and only is φ_c is complete. For each logic l with nontrivial completeness and finite set of propositional variables P, in Sect. 4 we provided a complete formula φ^l_P. This formula is satisfied in a model of at most two states, which can be generated in time $O(|P|)$. Let $(\mathcal{M}_l, a_l)$ be such a pointed model for φ^l_P.

Any pointed model that satisfies φ^l_P is bisimilar to $(\mathcal{M}_l, a_l)$. Given a formula $\varphi \in L(P)$, we can determine in linear time if $\mathcal{M}_l, a_l \models \varphi$. There are two cases:

- $\mathcal{M}_l, a_l \not\models \varphi$, in which case φ is not provable and we set $\varphi_c = \bigwedge P$.
- $\mathcal{M}_l, a_l \models \varphi$, so $\neg\varphi \wedge \varphi^l_P$ is not satisfiable, in which case we set $\varphi_c = \varphi \rightarrow \varphi^l_P$. We demonstrate that φ is provable if and only if $\varphi \rightarrow \varphi^l_P$ is complete.
  - If φ is provable, then $\varphi \rightarrow \varphi^l_P$ is equivalent to φ^l_P, which is complete.
  - On the other hand, if $\varphi \rightarrow \varphi^l_P$ is complete and $(\mathcal{M}, a)$ is any pointed model, we show that $\mathcal{M}, a \models \varphi$, implying that if $\varphi \rightarrow \varphi^l_P$ is complete, then φ is provable. If $(M, a) \sim_P (M_l, a_l)$, then from our assumptions $\mathcal{M}, a \not\models \neg\varphi$, thus $\mathcal{M}, a \models \varphi$. On the other hand, if $(M, a) \not\sim_P (M_l, a_l)$, since $(M_l, a_l) \models \varphi \rightarrow \varphi^l_P$ and $\varphi \rightarrow \varphi^l_P$ is complete, $\mathcal{M}, a \not\models \varphi \rightarrow \varphi^l_P$, therefore $\mathcal{M}, a \models \varphi$. □

Theorem 5 applies to more than the modal logics that we have defined in Sect. 2. For Propositional Logic, completeness amounts to the problem of determining whether a formula does not have *two* distinct satisfying assignments, therefore it is coNP-complete. By similar reasoning, completeness for First-order Logic is undecidable, as satisfiability is undecidable.

5.2 Upper Bounds

The case of logics with axiom 5 is now straightforward; from Theorem 5 and Proposition 2:

Proposition 3. *The completeness problem for logic $l + 5$ is* coNP*-complete.*

For the logics without axiom 5, by Theorem 4, satisfiability and provability are both PSPACE-complete. So, completeness is PSPACE-hard, if it is nontrivial. It remains to show that it is also in PSPACE. To this end we present a procedure that decides completeness for a modal formula. We call it the CC Procedure. Parts of this procedure are similar to the tableaux by Fitting [17] and Massacci [18] for Modal Logic, in that the procedure explores local views of a tableau. For more on tableaux the reader can see [19]. The CC Procedure is a nondeterministic polynomial time algorithm that uses an oracle from PSPACE. It accepts exactly the incomplete formulas, thus establishing that the completeness problems for these logics is in PSPACE. We have treated the case for logics with axiom 5, and the completeness problem for **D** and **T** is trivial. Therefore, form now on, we fix a logic l that can either be **K**, or have axiom 4 and be one of **K4**, **D4**, and **S4**.

The CC Procedure for Modal Logic l on φ. Intuitively, the procedure tries to demonstrate that there are two models for φ that are not bisimilar. We first give a few definitions that we need to describe the procedure.

For our procedure, *states* are sets of formulas from $\overline{sub}(\varphi)$. The procedure generates structures that we call *views*. A view S is a pair $(p(S), C(S))$ of a (possibly empty) set $C(S)$ of states, that are called the *children-states* of S and a distinguished state $p(S)$ called the *parent-state* of S. Each view is allowed to have up to $|\varphi|$ children-states.

Definition 2. *We call a set s of formulas l-*closed *if the following conditions hold:*

- *if $\varphi_1 \wedge \varphi_2 \in s$, then $\varphi_1, \varphi_2 \in s$;*
- *if $\varphi_1 \vee \varphi_2 \in s$, then $\varphi_1 \in s$ or $\varphi_2 \in s$;*
- *if $\Box\psi \in s$ and l has axiom T, then $\psi \in s$;*
- *for every $p \in P$, either $p \in s$ or $\neg p \in s$.*

*We call a view S l-*complete *(or* complete *if l is fixed) if the following conditions hold:*

- *the parent-state and every child-state of that view are l-closed;*
- *for every $\Diamond\psi \in p(S)$, $\psi \in \bigcup C(S)$;*
- *for every $\Box\psi \in p(S)$, $\psi \in \bigcap C(S)$;*
- *if l has axiom 4, then for every $\Box\psi \in p(S)$, $\Box\psi \in \bigcap C(S)$;*
- *if l has axiom D, then $C(S) \neq \emptyset$.*

For state a, $th(a) = \bigwedge a$. A state $a \subseteq \overline{sub}(\varphi)$ is maximal *if it is a maximally consistent subset of $\overline{sub}(\varphi)$. A child-state c of a view S is $\boldsymbol{K}$-maximal when it is a maximally consistent subset of $\overline{sub}_d(\varphi)$, where $d = \max\{md(c') \mid c' \in C(S)\}$. A view S is* consistent *when every state of S is a consistent set of formulas. A view S'* completes *view S when: S' is l-complete; $p(S) \subseteq p(S')$; for every $a \in C(S)$ there is an $a' \in C(S')$ such that $a \subseteq a'$; and: if $l = \boldsymbol{K}$, then every $a' \in C(S')$ is $\boldsymbol{K}$-maximal; if l has axiom 4, then every $a' \in C(S')$ is maximal.*

A view gives a local view of a model, as long as it is consistent. The procedure generates views and ensures that they are complete — so that all relevant information is present in each view — and consistent — so that the view indeed represents parts of a model. If the parent-state can represent two non-bisimilar states of two models (say, s and t), then the procedure should be able to provide a child, representing a state accessible from s or t that is not bisimilar to any state accessible from s or t, respectively. Since the states are (**K**-)maximal, two states that are not identical can only be satisfied in non-bisimilar models. The procedure is given in Table 2.

This section's main theorem is Theorem 6 and informs us our procedure can determine the completeness of formula φ in at most $|\varphi| + 2$ steps. We conclude that the completeness problem for logics without axiom 5 is in PSPACE.

Theorem 6. *The CC Procedure accepts φ if and only if φ is incomplete.*

Table 2. The CC Procedure on φ for logic $l \in \{\mathbf{K}, \mathbf{K4}, \mathbf{D4}, \mathbf{S4}\}$.

Initial conditions:	Non-deterministically generate maximal states a and b that include φ; if there are none, then return "`reject`".
	If $a \neq b$, then return "`accept`."
	Initialize N to $\lvert\varphi\rvert + 2$.
Construction:	Non-deterministically generate a consistent view S that completes $(a, \emptyset)$, having up to $\lvert\varphi\rvert$ children-states.
Condition:	If $C(S) = \emptyset$, then return "`reject`."
	If there is a child-state $c \in C(S)$, such that $\not\vdash_l th(a) \rightarrow \Diamond th(c)$, then return "`accept`."
Next step:	Otherwise, non-deterministically pick a child $c \in C(S)$ and set $a := c$.
	If $N > 0$, then set $N := N - 1$ and continue from "Construction."
	If $N = 0$, then return "`reject`".

Proof (Part of Proof). We give the proof of the theorem, but we omit certain details. The interested reader can see [13] for a full proof. We prove that the CC Procedure has a way to accept φ if and only if φ is satisfied in two non-bisimilar models. By Theorem 2, the theorem follows.

We assume that there are two non-bisimilar pointed models (A, w) and (B, w'), such that $A, w \models \varphi$ and $B, w' \models \varphi$. We prove that the CC Process accepts φ in $\lvert\varphi\rvert + 2$ steps. We call these models the underlying models; the states of the underlying models are called model states to distinguish them from states that the process uses. Let $A = (W^A, R^A, V^A)$ and $B = (W^B, R^B, V^B)$; we can assume that $W^A \cap W^B = \emptyset$. Let $f : W^A \times W^B \rightarrow W^A \cup W^B$ be a partial function that maps every pair (s, t) of non-bisimilar pairs to a model state c accessible from s or t that is non-bisimilar to every state accessible from t or s, respectively. We call f a choice-function. We can see that the procedure can maintain that the maximal state it generates each time is satisfied in two non-bisimilar states s, t, one from A and the other from B, respectively: at the beginning these are w and w'. At every step, the procedure can pick a child c that is satisfied in $f(s, t)$. If $\not\vdash_l th(a) \rightarrow \Diamond th(c)$, then the procedure terminates and accepts the input. Otherwise, c is satisfied in $f(s, t)$ and in another state that is non-bisimilar to $f(s, t)$. Let that other state be called a counterpart of $f(s, t)$.

If $l = \mathbf{K}$, then at every step, the procedure can reduce the modal depth of a, and therefore, after at most $\lvert\varphi\rvert$ steps, the procedure can simply choose $P = P(\varphi)$ as a state. Since $\Diamond \bigwedge P$ is not derivable from any consistent set of modal depth 0, the procedure can terminate and accept the input. We now assume that $l \neq \mathbf{K}$.

We demonstrate that if φ is incomplete, then the CC Procedure will accept φ after a finite number of steps. As we have seen above, the procedure, given non-bisimilar pointed models (A, a) and (B, b) of φ, always has a child to play according to f. For convenience, we can assume that models A and B have no cycles, so the choice-function never repeats a choice during a process run. If for every choice of f, the process does not terminate, then we show that $(A, w) \sim (B, w')$, reaching a contradiction. Let $\mathcal{R} = \sim \cup Z$, where $\sim$ is the bisimilarity relation between the states of A and the states of B, and xZy when for some choice-function, there is an infinite execution of the procedure, in which y is a counterpart of x, or x a counterpart of y. If $x\mathcal{R}y$, either $(A, x) \sim (B, y)$, so $V^A_P(x) = V^B_P(y)$, or xZy, so, again, $V^A_P(x) = V^B_P(y)$, since x and y satisfy he same maximal state. If $x\mathcal{R}y$ and xR^Ax', then if $(A, x) \sim (B, y)$, immediately there is some yR^By' so that $(A, x') \sim (B, y')$; if x is a counterpart of y or y is a counterpart of x during a non-terminating run, then for every x' accessible from x (the case is symmetric for a y' accessible from y), either x' is bisimilar to some y' accessible from y, or we can alter the choice-function f that the procedure uses so that $x' = f(x, y)$. Since for that altered f, the procedure does not terminate, x' has a counterpart as well. Therefore, the bisimulation conditions are satisfied and $\mathcal{R}$ is a bisimulation. If for every choice-function, the procedure never terminates, then $(A, w) \sim (B, w')$, and we have reached a contradiction. Therefore, there is a choice-function f that ensures the procedure terminates after a finite number of steps. We call that number of steps the length of choice-function f. For every state a, let $D(a) = \{\Diamond\psi \in a\}$ and $B(a) = \{\Box\psi \in a\}$. Then, $0 \leq |D(a)| \leq k_1$ and $0 \leq |B(a)| \leq k_2$, where $0 \leq k_1 + k_2 \leq |\varphi| - 1$. Notice that according to the definition of f above, as the process runs, $D(a)$ decreases and $B(a)$ increases — though, not necessarily strictly.

Lemma 7. *Let $l \in \{$**K4**, **D4**, **S4**$\}$ and let a, b, c be maximal states. If $B(a) = B(b)$, $D(a) = D(b)$, $\vdash th(a) \to_l \Diamond th(c)$, and $\not\vdash_l th(b) \to \Diamond th(c)$, then $c = a \neq b$ and $l =$ **S4**.*

Proof. See [13]. □

We can safely assume that the procedure never repeats the same choice of child-state — otherwise, it could continue from the second repetition and shorten its run. If during an execution, the CC Procedure picks states a, and in a following step, a state b, so that $B(a) = B(b)$ and $D(a) = D(b)$, and immediately after b the procedure picks child-state c, we claim that either the procedure could pick c right after a without affecting its run, or a and b are consecutive picked states and after picking c, the procedure terminates. Since c can be a child-state for a view that has b as parent-state, it satisfies all necessary closure conditions for l-complete views, so it can appear as a child-state for a view that has a as parent-state. If $\not\vdash_l th(a) \to \Diamond th(c)$, then the procedure can pick c right after a and terminate immediately; if $\vdash_l th(a) \to \Diamond th(c)$, but $\not\vdash_l th(b) \to \Diamond th(c)$, then the procedure terminates at c and, by Lemma 7, $l =$ **S4** and $a = c$. If a and b are not consecutive states, then there is a maximal state a' picked after a and before b, so that $B(a') = B(b)$ and $D(a') = D(b)$. Similarly to the above, $a' = c$, and therefore, $a = a'$ — so, the procedure repeated the same child-state choice. Therefore, a minimal-length choice function can ensure that the CC Procedure terminates after $|\varphi| + 2$ steps.

On the other hand, we prove that if φ is complete, then the CC Procedure can never accept φ. For this, we use the following lemmata:

Lemma 8. *If a view S is consistent and complete and $C(S) \neq \emptyset$, then*

- *if l does not have axiom 4 ($l = \boldsymbol{K}$), then the following formula is consistent:*

$$th(p(S)) \wedge \bigwedge_{c \in C(S)} \Diamond th(c) \wedge \Box \bigvee_{c \in C(S)} th(c);$$

- *if l has axiom 4 ($l \in \{\boldsymbol{K4}, \boldsymbol{D4}, \boldsymbol{S4}\}$), then the following formula is consistent:*

$$th(p(S)) \wedge \bigwedge_{c \in C(S)} \Diamond th(c).$$

Proof. See [13]. □

Lemma 9. *Let s be a consistent, and complete state, and for $l \neq \boldsymbol{K}$, also a maximal state; d a maximal state; and ψ a formula. If*

- *$\vdash_l th(s) \rightarrow \Diamond th(d)$,*
- *$th(d)$ is not equivalent to $th(s)$, and*
- *$d \cup \{\Box\psi\}$ is consistent,*

then $th(s) \wedge \Box(\neg th(d) \vee \Box\psi)$ is consistent.

Proof. See [13]. □

Lemma 10. *For a consistent view S that completes itself, for every child $c \in C(S)$, if $th(p(S))$ is complete, then so is $th(c)$.*

Proof. See [13]. □

By Lemma 10, all parent-states that appear during a run are complete. If at some point, the process picks a child-state c and a is the parent-state, then by Lemma 8, $th(a) \wedge \Diamond th(c)$ is consistent; since a is complete, $\vdash_l th(a) \rightarrow \Diamond th(c)$. Therefore, there is no way for the procedure to accept if the input formula is complete. □

Corollary 5. *The completeness problem for $\boldsymbol{K}$, $\boldsymbol{K4}$, $\boldsymbol{D4}$, and $\boldsymbol{S4}$ is PSPACE-complete.*

Proof. PSPACE-hardness is a consequence of Theorem 5. The CC Procedure is a non-deterministic polynomial-time algorithm with an oracle from PSPACE. Each condition that it needs to check is either a closure condition or a condition for the consistency or provability of formulas of polynomial size with respect to $|\varphi|$; therefore, they can be verified either directly or with an oracle from PSPACE. Thus, the completeness problem for these logics is in $\mathsf{coNP}^{\mathsf{PSPACE}} = \mathsf{PSPACE}$. □

6 Variations and Other Considerations

There are several variations one may consider for the completeness problem. One may define the completeness of a formula in a different way, consider a different logic, depending on the intended application, or wonder whether we could attempt a solution to the completeness problem by using Fine's normal forms [12].

6.1 Satisfiable and Complete Formulas

It may be more appropriate, depending on the case, to check whether a formula is *satisfiable and complete*. In this case, if the modal logic does not have axiom 5, we can simply alter the CC Procedure so that it accepts right away if the formula is not satisfiable. Therefore, the problem remains in PSPACE; for PSPACE-completeness, notice that the reduction for Theorem 5 constructs satisfiable formulas. For logics with axiom 5 (and plain Propositional Logic), the language of satisfiable and complete formulas is US-complete, where a language U is in US when there is a nondeterministic Turing machine T, so that for every instance x of U, $x \in U$ if and only if T has exactly one accepting computation path for x[4] [20]: UniqueSAT is a complete problem for US and a special case of this variation of the completeness problem.

6.2 Completeness with Respect to a Model

A natural variation of the completeness problem would be to consider completeness of a formula over a satisfying model. That is, the problem would ask: given a formula φ and pointed model $(\mathcal{M}, s)$, such that $\mathcal{M}, s \models \varphi$, is formula φ complete? For this variation, we are given one of φ's pointed models, so it is a reasonable expectation that the problem became easier. Note that in many cases, this problem may be more natural than the original one, as we are now testing whether the formula completely describes the pointed model (that is, whether the formula is characteristic for the model). Unfortunately, this variation has the same complexity as the original completeness problem. We can easily reduce completeness with respect to a model to plain completeness by dropping the model from the input. On the other hand, the reduction from provability to completeness of Sect. 5 still works in this case, as it can easily be adjusted to additionally provide the satisfying model of the complete formula φ^l_P.

[4] We note that US is different from UP; for UP, if T has an accepting path for x, then it is *guaranteed* that it has a unique accepting path for x.

6.3 Completeness and Normal Forms for Modal Logic

In [12], Fine introduced normal forms for Modal Logic. The sets F_P^d are defined recursively on the depth d, which is a nonnegative integer, and depend on the set of propositional variables P (we use a variation on the presentation from [21]):

$$F_P^0 = \left\{ \bigwedge_{p \in S} p \wedge \bigwedge_{p \notin S} \neg p \mid S \subseteq P \right\}; \quad \text{and}$$

$$F_P^{d+1} = \left\{ \varphi_0 \wedge \bigwedge_{\varphi \in S} \Diamond \varphi \wedge \Box \bigvee_{\varphi \in S} \varphi \mid S \subseteq F_P^d,\ \varphi_0 \in F_P^0 \right\}.$$

For example, formula $\varphi_P^{\mathbf{K}}$ from Sect. 4 is a normal form in F_P^1.

Theorem 7 (from [12]). *For every modal formula φ of modal depth at most d, if φ is consistent for $\boldsymbol{K}$, then there is some $S \subseteq F_P^d$, so that $\vdash_{\boldsymbol{K}} \varphi \leftrightarrow \bigvee S$.*

Furthermore, as Fine [12] demonstrated, normal forms are mutually exclusive: no two distinct normal forms from F_P^d can be true at the same state of a model. Normal forms are not necessarily complete by our definition (for example, consider $p \wedge \Diamond p \wedge \Box p$ for $P = \{p\}$), but, at least for $\mathbf{K}$, it is not hard to distinguish the complete ones; by induction on d, $\varphi \in F_P^d$ is complete for $\mathbf{K}$ if and only if $md(\varphi) < d$. Therefore, for $\mathbf{K}$, the satisfiable and complete formulas are exactly the ones that are equivalent to such a complete normal form. However, we cannot use this observation to test formulas for completeness by guessing a complete normal form and verifying that it is equivalent to our input formula, as normal forms can be of very large size: $|F_P^0| = 2^{|P|}$; $|F_P^{d+1}| = |P| \cdot 2^{|F_P^d|}$; and if $\psi \in F_P^d$, $|\psi|$ can be up to $|P| + 2|F_P^{d-1}|$. We would be guaranteed a normal form of reasonable (that is, polynomial w.r.to $|\varphi|$) size to compare to φ only if φ uses a small (logarithmic with respect to $|\varphi|$) number of variables and its modal depth is very small compared to $|\varphi|$ (that is, $md(\varphi) = O(\log^*(|\varphi|))$).

6.4 Completeness up to Depth

Fine's normal forms [12] can inspire us to consider a relaxation of the definition of completeness. We call a formula φ *complete up to its depth* for a logic l exactly when for every formula $\psi \in L(P(\varphi))$ of modal depth at most $md(\varphi)$, either $\vdash_l \varphi \rightarrow \psi$ or $\vdash_l \varphi \rightarrow \neg\psi$. Immediately from Theorem 7:

Lemma 11. *All normal forms are complete up to their depths.*

Lemma 12. *Formula φ is satisfiable and complete up to its depth for logic l if and only if it is equivalent in l to a normal form from $F_P^{md(\varphi)}$.*

Proof. From Theorem 7, if φ is satisfiable, then it is equivalent to some $\bigvee S$, where $S \subseteq F_P^{md(\varphi)}$, but if it is also complete up to its depth, then it can derive a the normal form $\psi \in S$; so, $\vdash_l \varphi \to \psi$, but also $\vdash_l \psi \to \bigvee S$ and $\bigvee S$ is equivalent to φ. For the other direction, notice that every normal form in $F_P^{md(\varphi)}$ is either complete or has the same modal depth as φ, so by Lemma 11, if φ is equivalent to a normal form, in the first case it is complete and in the second case it is complete up to its depth.

Therefore, all modal logics have formulas that are complete up to their depth. In fact, for any finite set of propositional variables P and $d \geq 0$, we can define $\varphi_P^d = \bigwedge_{i=0}^{d} \Box^i \bigwedge P$, which is equivalent in **T** and **D** to a normal form (by induction on d). Then, we can use a reduction similar to the one from the proof of Theorem 5 to prove that for every modal logic, completeness up to depth is as hard as provability.

Proposition 4. *For any complexity class C and logic l, if l-provability is C-hard, then completeness up to depth is C-hard.*

Proof. The proof is similar to that of Theorem 5 and can be found in [13]. □

We demonstrate that this variation of the completeness problem is in PSPACE when the logic is **K**; it seems plausible that one can follow similar approaches that use normal forms for the remaining modal logics.

Proposition 5. *A formula φ is complete up to its depth for **K** if and only if $\varphi \wedge \Box^{md(\varphi)+1}\bot$ is complete for **K**.*

Proof. Let $\psi \in F_P^d$ be a normal form. Then, $\psi \wedge \Box^{d+1}\bot$ is equivalent in **K** to $\psi^{+1} \in F_P^{d+1}$, which is ψ after we replace all $\Diamond\psi'$ in ψ by $\Diamond(\psi' \wedge \Box\bot)$, where $\psi' \in F_P^0$. Notice that $\psi_1, \psi_2 \in F_P^d$ are distinct normal forms if and only if ψ_1^{+1}, ψ_2^{+1} are distinct normal forms in F_P^r for every $r > d$. So, φ is complete up to its depth for **K** if and only if $\varphi \wedge \Box^{md(\varphi)+1}\bot$ is complete for **K**. □

6.5 More Logics

There is more to Modal Logic— and more modal logics,— so, perhaps, there is also more to discover about the completeness problem. We based the decision procedure for the completeness problem for each logic on a decision procedure for satisfiability. We distinguished two cases, depending on the logic's satisfiability-testing procedures.

– If the logic has axiom 5, then to test satisfiability we guess a small model and we use model checking to verify that the model satisfies the formula. This procedure uses the small model property of these logics (Corollary 1). To test for completeness, we guess *two* small models; we verify that they satisfy the formula and that they are non-bisimilar. We could try to use a similar approach for another logic based on a decision procedure for satisfiability based on a small model property (for, perhaps, another meaning for "small"). To do so successfully, a small model property may not suffice. We need to first demonstrate that for this logic, a formula that is satisfiable and incomplete has *two* small non-bisimilar models.

– For the other logics, we can use a tableau to test for satisfiability. We were able to combine the tableaux for these logics with bisimulation games to provide an optimal — when the completeness problem is not trivial — procedure for testing for completeness. For logics where a tableau gives an optimal procedure for testing for satisfiability, this is, perhaps, a promising approach to also test for completeness.

Another direction of interest would be to consider axiom schemes as part of the input — as we have seen, axiom 5 together with $\varphi^{\mathbf{S5}}$ is complete for **T**, when no modal formula is.

Acknowledgments. The author is grateful to Luca Aceto for valuable comments that helped improve the quality of this paper.

References

1. Ladner, R.E.: The computational complexity of provability in systems of modal propositional logic. SIAM J. Comput. **6**(3), 467–480 (1977)
2. Halpern, J.Y., Rêgo, L.C.: Characterizing the NP-PSPACE gap in the satisfiability problem for modal logic. J. Logic Comput. **17**(4), 795–806 (2007)
3. Halpern, J.Y., Moses, Y.: A guide to completeness and complexity for modal logics of knowledge and belief. Artif. Intell. **54**(3), 319–379 (1992)
4. Artemov, S.: Syntactic epistemic logic. In: Book of Abstracts, 15th Congress of Logic, Methodology and Philosophy of Science CLMPS 2015, pp. 109–110 (2015)
5. Artemov, S.: Syntactic epistemic logic and games (2016)
6. Hennessy, M., Milner, R.: Algebraic laws for nondeterminism and concurrency. J. ACM (JACM) **32**(1), 137–161 (1985)
7. Milner, R.: Communication and Concurrency. Prentice-Hall Inc., Upper Saddle River (1989)
8. Graf, S., Sifakis, J.: A modal characterization of observational congruence on finite terms of CCS. Inf. Control **68**(1–3), 125–145 (1986)
9. Steffen, B., Ingólfsdóttir, A.: Characteristic formulas for processes with divergence. Inf. Comput. **110**(1), 149–163 (1994)
10. Mller-Olm, M.: Derivation of characteristic formulae. Electr. Notes Theor. Comput. Sci. **18**, 159–170 (1998)
11. Aceto, L., Della Monica, D., Fábregas, I., Ingólfsdóttir, A.: When are prime formulae characteristic? In: Italiano, G.F., Pighizzini, G., Sannella, D.T. (eds.) MFCS 2015. LNCS, vol. 9234, pp. 76–88. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48057-1_6
12. Fine, K.: Normal forms in modal logic. Notre Dame J. Formal Logic **16**(2), 229–237 (1975)
13. Achilleos, A.: The completeness problem for modal logic. CoRR abs/1605.01004 (2016)
14. Blackburn, P., de Rijke, M., Venema, Y.: Modal Logic. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, Cambridge (2001)
15. Chagrov, A., Zakharyaschev, M.: Modal Logic. Oxford University Press, Oxford (1997)
16. Paige, R., Tarjan, R.E.: Three partition refinement algorithms. SIAM J. Comput. **16**(6), 973–989 (1987)

17. Fitting, M.: Tableau methods of proof for modal logics. Notre Dame J. Formal Logic **13**(2), 237–247 (1972)
18. Massacci, F.: Single step tableaux for modal logics. J. Autom. Reasoning **24**(3), 319–364 (2000)
19. D'Agostino, M., Gabbay, D.M., Hähnle, R., Posegga, J.: Handbook of Tableau Methods. Springer, Dordrecht (1999). https://doi.org/10.1007/978-94-017-1754-0
20. Blass, A., Gurevich, Y.: On the unique satisfiability problem. Inf. Control **55**(1–3), 80–88 (1982)
21. Moss, L.S.: Finite models constructed from canonical formulas. J. Philos. Logic **36**(6), 605–640 (2007)

Justification Awareness Models

Sergei Artemov(✉)

The City University of New York, The Graduate Center,
365 Fifth Avenue, New York City, NY 10016, USA
sartemov@gc.cuny.edu

Abstract. Justification Awareness Models, *JAM*s, incorporate two principal ideas: (i) *justifications are prime objects of the model*: knowledge and belief are defined evidence-based concepts; (ii) *awareness restrictions are applied to justifications* rather than to propositions, which allows for the maintaining of desirable closure properties. *JAM*s naturally include major justification models, Kripke models and, in addition, represent situations with multiple possibly fallible justifications. As an example, we build a *JAM* for Russell's well-known Prime Minister scenario which, in full generality, was previously off the scope of rigorous epistemic modeling.

Keywords: Modal logic · Justification logic · Epistemology
Knowledge · Belief

1 Context and Motivations

Proof systems of justification logic and general purpose classes of models for these systems have been studied in [1–3,9,10,16,18,20] and many other sources. However, for formalizing epistemic scenarios, one needs specific *domain-dependent models* with additional features that are not necessary for standard soundness and completeness analysis of proof systems.

Awareness is an important concept in epistemic modeling, but, when applied to propositions directly, it may seriously diverge from the intuition due to lack of natural closure properties [7,8,17]. We suggest applying awareness to justifications

$$agent\ is\ aware/unaware\ of\ a\ justification\ t\ for\ a\ proposition\ F$$

rather then to propositions "agent is aware/unaware of a proposition F"; this approach allows for the maintaining of natural closure properties.

We introduce *justification awareness models*, *JAM*s, in which justifications are primary objects and a distinction is made between *accepted* and *knowledge-producing* justifications. In *JAM*s, belief and knowledge are derived notions which depend on the status of supporting justifications. We argue that *JAM*s can work in situations in which standard non-hyperintensional tools (Kripke, topological, algebraic) fail to fairly represent the corresponding epistemic structure.

S. Artemov and A. Nerode (Eds.): LFCS 2018, LNCS 10703, pp. 22–36, 2018.
https://doi.org/10.1007/978-3-319-72056-2_2

2 Preliminaries

Standard modal epistemic models have "propositional" precision, i.e., they do not distinguish sentences with the same truth values at each possible world. The expressive power of such models for analysis of justification, belief, and knowledge is rather limited, and so we have to "go hyperintensional."[1] Specifically, if, at all possible worlds, t is a justification for F

$$\Vdash t{:}F,$$

and G has the same truth value as F

$$\Vdash F \leftrightarrow G,$$

we still cannot conclude that t is a justification for G

$$\nVdash t{:}G.$$

A natural example from mathematics: both statements $0 = 0$ and *Fermat's Last Theorem*, FLT, are true (proven) mathematical facts and hence are true at all possible worlds. However, we cannot claim that a proof of $0 = 0$ is a proof of FLT as well.

A sample justification logic analysis of some standard epistemic situations (Gettier examples, Red Barn example) is presented in [2] using justification Fitting models [9] though, due to the relative simplicity of those examples, this analysis could be replicated in a bi-modal language (cf. [21]).

However, we cannot go much farther without adopting a justification framework: the situation changes when we have to represent several conflicting pieces of evidence for a stated fact, cf. the following Russell example of 1912 ([19]):

> *If a man believes that the late Prime Minister's last name began with a 'B,' he believes what is true, since the late Prime Minister was Sir Henry Campbell Bannerman*[2]*. But if he believes that Mr. Balfour was the late Prime Minister, he will still believe that the late Prime Minister's last name began with a 'B,' yet this belief, though true, would not be thought to constitute knowledge.*

To keep it simple, we consider proposition Q

the late Prime Minister's last name began with a 'B,'

with two justifications for Q: the right one r and the wrong one w; the agent chooses w as a reason to believe that Q holds.

To avoid a misleading reduction of failures of justifications to "false premises," consider another Russell example from [19].

[1] From [6]: "Hyperintensional contexts are simply contexts which do not respect logical equivalence".

[2] Which was true in 1912.

> *If I know that all Greeks are men and that Socrates was a man, and I infer that Socrates was a Greek, I cannot be said to-know-that Socrates was a Greek, because, although my premisses and my conclusion are true, the conclusion does not follow from the premisses.*

This Russell's example illustrates that "false premises" in the Prime Minister story is an instance of a more general phenomenon: an erroneous justification which, in principle, can fail for many different reasons: unreliable premises, hidden assumptions, deduction errors, an erroneous identification of the goal sentence, etc.[3]

There is a mathematical version of the story with a true proposition and its two justifications; one is correct, the other is not.

> *Consider the picture*[4]*:*
>
> $$\frac{1\cancel{6}}{\cancel{6}4} = \frac{1}{4}. \tag{1}$$
>
> *The true proposition is "16/64 = 1/4," the right justification is dividing both the numerator and the denominator by 16, and the wrong (but shorter and more attractive) justification is simplifying as in* (1).

Given these considerations, we prefer speaking about *erroneous justifications* in a general setting without reducing them to propositional entities such as "false premises." To be specific, we'll continue with Russell's Prime Minister example.

To formalize Russell's scenario in modal logic (cf. [21]), we introduce two modalities: **K** for knowledge and **J** for justified belief. In the real world,

- Q holds;
- $\mathbf{J}Q$ holds, since the agent has a justification w for Q;
- $\mathbf{K}Q$ does not hold;

thus yielding the set of assumptions

$$\Gamma = \{Q,\ \mathbf{J}Q\ \neg\mathbf{K}Q\}.$$

However, Γ doesn't do justice to Russell's scenario: the right justification r is not represented and Γ rather corresponds to the same scenario but lacking r. The epistemic structure of the example is not respected.

Within the *JAM* framework, we provide a model for Russell's Prime Minister example which, we wish to think, fairly represents its intrinsic epistemic structure.

[3] Moreover, one can easily imagine knowledge-producing reasoning from a source with false beliefs (both an atheist and a religious scientist can produce reliable knowledge products though one of them has false beliefs), so "false premises" are neither necessary nor sufficient for a justification to fail.

[4] Which the author saw on the door of the Mathematics Support Center at Cornell in 2017.

3 Generic Logical Semantics of Justifications

What kinds of logical objects are justifications? When asked in a mathematical context "what is a predicate?" we have a ready answer: a subset of a Cartesian product of the domain set. Within an exact mathematical theory, there should be a similar kind of answer to the question "what is a justification?"

We consider this question in its full generality which, surprisingly, yields a clean and meaningful answer. We assume the language of justification logic consists of two disjoint sets of syntactic objects:

1. a set of **justification terms** Tm;
2. a set of **formulas** Fm, built inductively from propositional atoms using Boolean connectives and the justification formula formation rule: if F is a formula, $F \in Fm$, and t a justification term, $t \in Tm$, then $t{:}F$ is again a formula, $t{:}F \in Fm$.

The meaning assigned to formulas is a classical truth value, 0 for *false* and 1 for *true*, and we retain classical logic behavior for propositional connectives. The key item is to give meaning to justification terms, and this will be a *set of formulas* interpreted as *the set of formulas for which it is a justification.* A formal definition follows.

Definition 1 (Basic Model). *A basic model, simply called* $*$*, consists of an interpretation of the members of* Fm*, and an interpretation of the members of* Tm*.*

The interpretation of a formula in a basic model is a truth value. That is,

$$* : Fm \mapsto \{0, 1\}.$$

We assume the Boolean truth tables: $(X \to Y)^* = 1$ *if and only if* $X^* = 0$ *or* $Y^* = 1$*, etc. Let also* $\models_* X$ *stand for* $X^* = 1$*.*

We interpret justification terms as sets of formulas. That is,

$$* : Tm \mapsto 2^{Fm}.$$

Our final requirement connects the two mapping roles that $*$ *plays in a basic model. For any* $X \in Fm$ *and any* $t \in Tm$*,*

$$\models_* t{:}X \textit{ if and only if } X \in t^*.$$

It is easy to check that any mapping $*$ from propositional letters to truth values, and from justification terms to sets of formulas, determines a unique basic model.

So far, a basic model is merely a classical propositional model in which justification assertions $t{:}F$ are treated as *independent propositional atoms.*

Note that while propositions are interpreted semantically as truth values, justifications are interpreted syntactically as sets of formulas. This is a principal *hyperintensional* feature: a basic model may treat distinct formulas F and G as equal, i.e. $F^* = G^*$, but still be able to distinguish justification assertions $t{:}F$ and $t{:}G$, e.g., when $F \in t^*$, but $G \notin t^*$ yielding $\models_* t{:}F$ but $\not\models_* t{:}G$.

Definition 2. *Let S a set of formulas, $S \subseteq Fm$, and X be a formula, $X \in Fm$. We write $S \vdash X$ if X is derivable from S in classical logic that treats justification assertions $t{:}F$ as propositional atoms (with Modus Ponens as the only rule of inference). We say that S is consistent if $S \nvdash \bot$.*

A basic model of S is merely a possible world containing S in the canonical model, i.e., a maximal consistent set Γ of formulas, with the convenience agreement reading $t{:}F \in \Gamma$ as $F \in \{X \mid t{:}X \in \Gamma\}$. In this respect, basic models and the canonical model are slightly different but obviously equivalent ways of presenting the same object. When we move to more sophisticated models (Fitting models, modular models), the advantage of dealing with sets and operations (e.g. basic models) over logical conditions (e.g. the canonical model) becomes clear.

Definition 3. *For $S \subseteq Fm$, $BM(S)$ is the class of all basic models of S.*

Theorem 1. *Each set of formulas S is sound and complete with respect to its class of basic models $BM(S)$. In other words, $S \vdash F$ iff F is true in each basic model of S.*

Proof. This theorem is merely a reformulation of the soundness and completeness of classical propositional logic with hypotheses. Indeed, if $S \vdash F$ and $\models_* S$, then $\models_* F$ since propositional derivations respect validity.

If $S \nvdash F$, then there is a Boolean evaluation $*$ which makes all formulas from S true, $S^* = 1$, and F false, $F^* = 0$. In this case, there are two types of atomic propositions: propositional letters P and justification assertions $t{:}X$. Define

$$t^* = \{X \mid (t{:}X)^* = 1\}$$

and note that $(t{:}X)^* = 1$ iff $X \in t^*$. Therefore, $*$ is a propositional evaluation and $*$ is a basic model yielding the same truth values of atomic formulas P and $t{:}X$. Since $S^* = 1$ and $F^* = 0$, we have $\models_* S$ and $\not\models_* F$ for basic model $*$.

An easy corollary: $\vdash F$ iff F is a tautology (with $t{:}X$es as distinct propositional atoms).

Example 1. In Definition 2, take $S = \emptyset$.

1. For any justification term t,
$$\nvdash t{:}F.$$
Straightforward, since $t{:}F$ is not a propositional tautology. For a specific countermodel, define $t^* = \emptyset$ for each term $t \in Tm$, which makes $\not\models_* t{:}F$.
2. For any propositional letter P, and term t,
$$\nvdash t{:}P \rightarrow P.$$
Likewise, this holds because $t{:}P \rightarrow P$ is not a propositional tautology. Specifically, put $t^* = Fm$ and $P^* = 0$, with other assignments being arbitrary. In this model, all justification assertions are true, but $t{:}P \rightarrow P$ is false.

3. For any propositional letter P, and term t,

$$\not\vdash P \to t{:}P.$$

   Again, this holds since $P \to t{:}P$ is not a propositional tautology. For example, put $t^* = \emptyset$ and $P^* = 1$. In this model, t is not a justification for P (i.e., $\not\models_* t{:}P$) and $P \to t{:}P$ is false.
4. A somewhat less trivial example illustrating hyperintensionality: for a justification variable x and formula F

$$\not\vdash x{:}F \to x{:}(F \wedge F).$$

   A high-level argument is the same: formulas $x{:}F$ and $x{:}(F \wedge F)$, evaluated from a Boolean point of view, can be regarded as distinct propositional variables. Hence $x{:}F \to x{:}(F \wedge F)$ is not a tautology. For a countermodel, take $x^* = \{F\}$. Then $\models_* x{:}F$, but $\not\models_* x{:}(F \wedge F)$. This demonstrates hyperintensionality of a justification logic base, since F and $F \wedge F$ are provably equivalent, but not $x{:}F$ and $x{:}(F \wedge F)$.

4 Basic Justification Logic J^-

Within the Justification Logic framework, there are two sorts of logical objects: justification terms *Tm* and formulas *Fm*. Let us become more specific about both.

- For *Tm*, reserve a set of justification constants $a, b, c, \ldots$ with indices, and variables $x, y, z, \ldots$ with indices. Justification terms are built from constants and variables by a binary operation $\cdot$ (application).
- Formulas are built from propositional letters $p, q, r, \ldots$ (with indices) and Boolean constant $\bot$ (falsum) by the standard Boolean connectives $\wedge, \vee, \to, \neg$ with a new formation rule: *whenever t is a justification term and F is a formula, $t : F$ is a formula (with the informal reading "t is a justification for F").* For better readability, we will interchangeably use brackets $0, 0$ and parentheses $(,)$. Our preferred notation is $[s \cdot t]{:}(F \to G)$ which is the same as $(s \cdot t){:}(F \to G)$.

The *logical system* J^- consists of two groups of postulates.

- **Background logic**: axioms of classical propositional logic, rule *Modus Ponens.*
- **Application**: $s{:}(F \to G) \to (t{:}F \to [s{\cdot}t]{:}G)$.

Basic models corresponding to J^- are those in which the application axiom holds. They can be specified by a natural combinatorial condition.

Definition 4. *For sets of formulas S and T, we define*

$$S \rhd T = \{F \mid G \to F \in S \textit{ and } G \in T \textit{ for some } G\}.$$

Informally, $S \rhd T$ is the result of applying Modus Ponens once to all members of S and of T (in a given order).

Theorem 2. *$BM(\mathsf{J}^-)$ is the class of basic models with the following closure condition*

$$s^* \rhd t^* \subseteq [s\cdot t]^*. \tag{2}$$

Proof. Let us assume the closure condition (2) and check the validity of the application axiom. Indeed, $\models_* s{:}(F \rightarrow G)$ and $\models_* t{:}F$ yield $(F \rightarrow G) \in s^*$ and $F \in t^*$. By the closure condition, $G \in [s\cdot t]^*$, i.e., $\models_* [s\cdot t]{:}G$.

Now assume the application axiom and derive the closure condition (2). Let $(F \rightarrow G) \in s^*$ and $F \in t^*$. By definition, this yields $\models_* s{:}(F \rightarrow G)$ and $\models_* t{:}F$. By the application axiom, $\models_* [s\cdot t]{:}G$, hence $G \in [s\cdot t]^*$.

Example 2. None of the formulas from Example 1: $t:F$, $t:P \rightarrow P$, $P \rightarrow t:P$, $x:F \rightarrow x:(F \wedge F)$ is derivable in J^-. Indeed, every specific evaluation from Example 1.1–3 satisfies the closure condition (2), hence their countermodels are J^--models. Consider the latter formula 4. Put $x^* = \{F\}$ and $t^* = Fm$ for all other terms t. The closure condition (2) holds vacuously, hence $*$ is a J^--model. Obviously, $\models_* x{:}F$ and $\not\models_* x{:}(F \wedge F)$.

Constants in justification logic are used to denote justifications of assumptions, in particular, axioms. Indeed, as we have already seen in Example 2, no formula $t{:}F$ is derivable in J^-. In particular, no logical axiom is assumed justified in J^- which is not realistic.

Definition 5. *A set X of formulas is reflexive if for each $s{:}t{:}F \in X$, $t{:}F$ is also in X. By constant specification CS we understand a reflexive set of formulas of the type*

$$c_n{:}c_{n-1}{:}c_{n-2}{:}\ \ldots\ c_1{:}A$$

where A is a J^--axiom and c_i are justification constants. The major classes of constant specifications are empty, total— (each constant is a justification for each axiom), axiomatically appropriate (each axiom has a justification at any depth).

Let CS be a constant specification. Then by $\mathsf{J}^-(CS)$, we understand J^- with additional axioms CS. A *CS-model* is a model in which all formulas from CS hold.

Corollary 1. *Basic models for $\mathsf{J}^-(CS)$ are the basic CS-models for J^-. $\mathsf{J}^-(CS)$ is sound and complete with respect to the class of its basic models.*

4.1 Other Justification Logics

There is a whole family of justification logics and they all extend J^-; the reader is referred to [2,11] for details. Here we list just the main systems of justification logic for purposes of general orientation.

Logic J is obtained from J^- by adding a new operation on justifications '+' and the principle

$$s{:}F \vee t{:}F \rightarrow [s+t]{:}F.$$

Logics JD, JT, J4, J5, etc., are obtained by adding the corresponding combination of principles

$$D = \neg t{:}\bot,$$

$$T = t{:}F \rightarrow F,$$

$$4 = t{:}F \rightarrow\ !t{:}t{:}F,$$

$$5 = \neg t{:}F \rightarrow\ ?t{:}\neg t{:}F.$$

The family of justification logics has now grown to be infinite, cf. [11].

4.2 Sharp Models

In closure condition (2) from Theorem 2, one cannot, generally speaking, replace the inclusion "$\subseteq$" by the equality "$=$" without violating completeness Theorem 1.

Indeed, fix a justification constant 0 and consider logic

$$\mathcal{L} = \mathsf{J}^- + \{\neg 0{:}F \mid F \in Fm\}.$$

Informally, justification 0 receives empty evaluation in any basic model, $0^* = \emptyset$. We claim that formula $G = \neg[0\cdot 0]{:}P$ is not derivable in $\mathcal{L}$, but is true in any basic model of $\mathcal{L}$ with the closure condition $s^* \triangleright t^* = [s\cdot t]^*$. To show that $\mathcal{L} \not\vdash G$, it suffices to find a basic model for $\mathcal{L}$ in which G is false. Consider a basic model $\sharp$ such that $0^\sharp = \emptyset$ and $t^\sharp = Fm$ for any other justification term t. Obviously, the closure condition from Theorem 2, together with $0^\sharp = \emptyset$, is met. Therefore, $\sharp$ is a basic model of $\mathcal{L}$. It is immediate that G is false in $\sharp$, since $[0\cdot 0]^\sharp = Fm$. On the other hand, G holds in any basic model of $\mathcal{L}$ with the closure condition $[0\cdot 0]^* = 0^* \triangleright 0^*$. Indeed, in such a model, $[0\cdot 0]^* = \emptyset$ since $0^* = \emptyset$ and $\emptyset \triangleright \emptyset = \emptyset$.

Definition 6. Sharp *basic models are those in which the application closure condition has the form*

$$[s\cdot t]^* = s^* \triangleright t^*. \tag{3}$$

Note that a sharp model is completely defined by evaluations of atomic propositions and atomic justifications.

5 Justification Awareness

We need more expressive power to capture epistemic differences between justifications and their use by the knower. Some justifications are knowledge-producing, some are not. The agent makes choices on which justifications to base an agent's beliefs/knowledge and which justifications to ignore in this respect. These actions are present in epistemic scenarios, from which we will primarily focus on Russell's Prime Minister example, which has them all:

- there are justifications w (Balfour was the late prime minister) and r (Bannerman was the late prime minister) for Q;
- r is knowledge-producing whereas w is not;
- the agent opts to base his belief on w and ignores r;
- the resulting belief is evidence-based, but is not knowledge.

5.1 Justification Awareness Models

Fix $\mathsf{J}^-(CS)$ for some axiomatically appropriate constant specification CS.

Definition 7. *A set X of justification terms is properly closed if X contains all constants and is closed under applications. If X is a set of justification terms, then by $\overline{X}$ we mean the proper closure of X, i.e., the minimal properly closed superset of X.*

Definition 8. *A (basic) Justification Awareness Model is $(*, \mathcal{A}, \mathcal{E})$ where*

- *$*$ is a basic $\mathsf{J}^-(CS)$-model;*
- *$\mathcal{A} \subseteq Tm$ is a properly closed set $\mathcal{A}$ of accepted justifications;*
- *$\mathcal{E} \subseteq Tm$ is a properly closed set $\mathcal{E}$ of knowledge-producing justifications.*

Unless stated otherwise, we also assume consistency of accepted justifications: $\models_* \neg t\!:\!\bot$ for any $t \in \mathcal{A}$, and factivity of knowledge-producing justifications, $\models_* t{:}F \rightarrow F$ for each F and each $t \in \mathcal{E}$. In models concerning beliefs rather then knowledge, the component $\mathcal{E}$ can be dropped.

Both sets $\mathcal{A}$ and $\mathcal{E}$ contain all constants. This definition presumes that constants in a model are knowledge-producing and accepted.

Definition 9. *In a JAM $(*, \mathcal{A}, \mathcal{E})$, a sentence F is believed if there is $t \in \mathcal{A}$ such that $\models_* t{:}F$. Sentence F is known if there is $t \in \mathcal{A} \cap \mathcal{E}$ such that $\models_* t{:}F$.*

By *ground term* we understand a term containing no (justification) variables. In other words, a term is ground iff it is built from justification constants only.

Sets of accepted and knowledge-producing justifications overlap on ground terms but otherwise can be in a general position[5]. There may be accepted, but not knowledge-producing, justifications and vice versa. So, *JAM*s do not analyze **why** certain justifications are knowledge-producing or accepted, but rather provide a formal framework that accommodates these notions.

5.2 Single-Conclusion Justifications

The notions of *accepted* and *knowledge-producing* justifications should be utilized with some caution. Imagine a justification t for F (i.e., $t{:}F$ holds) and for G ($t{:}G$) such that, intuitively, t is a knowledge-producing justification for F but not for G. Is such a t knowledge-producing, trustworthy, acceptable for a reasonable agent? The answers to these questions seem to depend on F and G, and if we prefer to handle justifications as objects rather than as justification assertions, it is technically convenient to assume that justifications are *single-conclusion* (or, equivalently, *pointed*):

$$\textit{there is at most one formula } F \textit{ such that } t{:}F \textit{ holds.}$$

[5] In principle, one could consider smaller sets $\mathcal{A}$, which would correspond to the high level of skepticism of an agent who does not necessarily accept logical truths (axioms) as justified. We leave this possibility for further studies.

Conceptually, by turning to pointed justifications, one does not lose generality: if p is a proof of F and of something else, then the same p with a designated statement F, symbolically, a pair (p, F), can be regarded as a single-conclusion (or pointed) proof of F.

In model $\mathcal{R}$ for the Russell Prime Minister example, Sect. 6, all justifications are pointed.

Note that J^- is not complete with respect to the class of basic models which are both sharp and pointed (as model $\mathcal{R}$ for the Russell Example). Indeed, consider formula F,

$$F = \neg(x{:}(P \rightarrow Q) \wedge y{:}P \wedge [x\cdot y]{:}R)$$

where P, Q, R are distinct propositional letters and x, y justification variables. Obviously, F holds in any basic model $*$ which is sharp and pointed. Imagine a sharp pointed $*$ in which $x{:}(P \rightarrow Q)$ and $y{:}P$ hold. In such $*$, $[x{\cdot}y]^* = \{Q\}$, hence both $\neg[x\cdot y]{:}R$, and F hold. On the other hand, F is not derivable in J^-, e.g., F fails in the basic model $*$ with $x^* = \{P \rightarrow Q\}$, $y^* = \{P\}$, and $t^* = Fm$ for any other t (check closure condition (2)!). So, "sharp and pointed" justification tautologies constitute a proper extension SP of J^-. The problem of finding complete axiomatization of SP was first stated in [5]. This question was answered in [15] along the lines of studying single-conclusion logic of proofs [13,14].

6 Russell Scenario as a *JAM*

Consider the version of J^- in a language with two justification variables w and r, one propositional letter Q, and pointed constant specification CS:

$$c_n{:}A \in CS \ \text{ iff } \ A \textit{ is an axiom and } n \textit{ is the Gödel number of } A.$$

Define a model $*$ such that

- $Q^* = 1$, i.e., $\models_* Q$;
- $c_n^* = \{A\}$ if A is an axiom and n is the Gödel number $|A|$ of A, and $c_n^* = \emptyset$ otherwise;
- $w^* = r^* = \{Q\}$, e.g., $\models_* r{:}Q$ and $\not\models_* r{:}F$ for any F other than Q (the same for w);
- application is sharp: $[s{\cdot}t]^* = s^* \rhd t^*$.

A *JAM* $\mathcal{R}$ (for Russell's scenario) is $(*, \mathcal{A}, \mathcal{E})$ with

- $\mathcal{A} = \overline{\{w\}}$, i.e., the set of accepted justifications is $\{w\}$, properly closed;
- $\mathcal{E} = \overline{\{r\}}$, i.e., the set of knowledge-producing justifications is $\{r\}$, properly closed.

Though the idea behind $\mathcal{R}$ is quite intuitive, we need to fill in some technical details: extending truth evaluations to all terms and formulas and checking closure conditions.

6.1 Technicalities of the Model

Define $c^*_{|A|} = \{A\}$ for each axiom A of $\mathsf{J}^-(CS)$. Technically, this is an inductive definition with induction on n in c_n.

Base: $n = 0$. Here $c^*_0 = \emptyset$, given 0 is not a Gödel number of any formula.

Inductive step: suppose n is the Gödel number $|F|$ of some formula F. If F is an axiom of J^-, put $c^*_n = \{F\}$. If $F = c_k{:}G$ for some c_k and G, then, by monotonicity of Gödel numbering, $k < n$, hence c^*_k is defined. If $c^*_k = \{G\}$, then $c_k{:}G$ is an axiom of $\mathsf{J}^-(CS)$ and we can define $c^*_n = \{F\}$. In all other cases, $c^*_n = \emptyset$.

Since application is sharp, the evaluation of each term is, at most, a singleton. Together with Boolean truth tables, this determines the truth value of any formula.

Lemma 1. *Each $t \in Tm$ is factive, $\models_* t{:}F \rightarrow F$.*

Proof. Induction on t. Assume $\models_* t{:}F$; that means $t^* = \{F\}$. If t is w or r, then F is Q, which is true in the model $*$. If t is a constant, then F is an axiom and hence true in $*$. The induction step corresponds to application, which preserves the truth of justified formulas.

It follows from Lemma 1, that accepted justifications are consistent and knowledge-producing justifications are factive. Therefore, $\mathcal{R} = (*, \mathcal{A}, \mathcal{E})$ is indeed a *JAM*.

Theorem 3. *In model $\mathcal{R}$, sentence Q is true, justified and believed, but not known.*

Proof. In model $\mathcal{R}$, sentence Q is

- true, since $\models_* Q$;
- justified, since $\models_* w{:}Q$;
- believed, since $w \in \mathcal{A}$.

We have to show that Q is not known, i.e., for any justification $g \in \mathcal{A} \cap \mathcal{E}$, $\not\models_* g{:}Q$.

Consider an auxiliary basic model $\bullet$ which is the same as $*$ but with $Q^\bullet = 0$, i.e., the truth value of Q is flipped from 'true' to 'false.' In particular, application in $\bullet$ is sharp.

Lemma 2. *For each justification term t,*

$$t^* = t^\bullet.$$

Proof. The inductive process (based on sharp application) of evaluating all justifications, given evaluations of atomic justifications, operates only with formulas of type $t{:}F$ and starts with the same initial set of such formulas in $*$ and $\bullet$. Hence the results of these processes in $*$ and $\bullet$ coincide.

In particular, for all $g \in \mathcal{A} \cap \mathcal{E}$, $g^* = g^\bullet$, and if $\models_* g{:}Q$, then $\models_\bullet g{:}Q$ as well.

Lemma 3. *Each $g \in \mathcal{A} \cap \mathcal{E}$ is factive in $\bullet$, i.e., $\models_\bullet g{:}F \rightarrow F$.*

Proof. All $g \in \mathcal{A} \cap \mathcal{E}$ are obtained from constants by application. By construction, if $\models_\bullet c{:}X$, then $X \in c^*$ and X is an axiom, hence true. Application obviously preserves factivity.

To complete the proof of Theorem 3, suppose $\mathcal{R} \models g{:}Q$, i.e., $\models_* g{:}Q$, for some $g \in \mathcal{A} \cap \mathcal{E}$. By Lemma 2, $\models_\bullet g{:}Q$, and, by Lemma 1, $\models_\bullet Q$, which is not the case.

6.2 Can Russell's Scenario Be Made Modal?

One could try to express Russell's scenario in a modal language by introducing the justified belief modality

$$\mathbf{J}F \quad \Leftrightarrow \quad \textit{there is } t \in \mathcal{A} \textit{ such that } \models t{:}F,$$

and the knowledge-producing modality

$$\mathbf{E}F \quad \Leftrightarrow \quad \textit{there is } t \in \mathcal{E} \textit{ such that } \models t{:}F,$$

and by stipulating that F is known iff F is both accepted and supported by a knowledge-producing justification:

$$\mathbf{K}F \quad \Leftrightarrow \quad \mathbf{J}F \wedge \mathbf{E}F.$$

This, however, fails, since both $\mathbf{J}Q$ and $\mathbf{E}Q$ hold in $\mathcal{R}$, but $\mathbf{K}Q$ does not. We are facing a Gettier-style phenomenon (cf. [12]), when a proposition is supported by a knowledge-producing justification (hence true), and believed, but not known (since knowledge-producing and accepted justifications for Q are different). This once again illustrates the limitations of modal language in tracking and sorting justifications.

7 Kripke Models and Master Justification

From the Justification Logic point of view, Kripke models may be regarded as a special case of multi-world *JAM*s[6]; the Kripkean accessibility relation between worlds, uRv, can be recovered by the usual rule *what is believed at u, holds at v.* Moreover, such representation of Kripke models as justification models reveals and formalizes the observation made in [4] that epistemic reading of Kripke models relies on a hidden assumption of (common) knowledge of the model.

The informal argument is as follows. We have to find a justification $m{:}F$ for each knowledge/belief assertion $\Box F$ in a model $\mathcal{K}$. We claim that the model $\mathcal{K}$ itself is such a justification. Indeed, let $u \Vdash \Box F$ in $\mathcal{K}$. Then a complete description

[6] In which we suppress the knowledge-producing component $\mathcal{E}$ to capture beliefs.

of $\mathcal{K}$ yields that at state u, the agent knows/believes F **because the agent knows the model $\mathcal{K}$ and knows that F holds at all possible worlds**. So, the knowledge/belief-producing evidence for F is delivered by $\mathcal{K}$ itself, assuming the agent is aware of $\mathcal{K}$.

Syntactically, we consider a very basic justification language in which the set of justification terms consists of just one term m, called *master justification*. Think of m as representing a complete description of model $\mathcal{K} = (W, R, \Vdash)$. Specifically, we extend the truth evaluation in $\mathcal{K}$ to justification assertions by stipulating at each $u \in W$

$$\mathcal{K}, u \Vdash m{:}X \qquad \textit{iff} \qquad \mathcal{K}, v \Vdash X \ \textit{for any } v \in R(u) \qquad \textit{iff} \qquad \mathcal{K}, u \Vdash \Box X.$$

This reading provides a meaningful justification semantics of epistemic assertions in $\mathcal{K}$ via the master justification m representing the whole $\mathcal{K}$. Since a Kripkean agent is logically omniscient, along with $\mathcal{K}$, the agent knows all its logical consequences. Technically, we can assume that the description $\mathcal{K}$ is closed under logical consequence and hence m is idempotent w.r.t. application, $m \cdot m = m$. This condition manifests itself in a special form of the application principle

$$m{:}(A \rightarrow B) \rightarrow (m{:}A \rightarrow m{:}B).$$

On the technical side, a switch from $\Box X$ to $m{:}X$ is a mere transliteration which does not change the epistemic structure of a model. Finally, for each $u \in W$, we define a basic model – maximal consistent set Γ_u in the propositional language with $Tm = \{m\}$:

$$\Gamma_u = \{X \mid u \Vdash X\}.$$

So, from a justification perspective, a Kripke model is a collection of basic models with master justification that represents (common) knowledge of the model.

8 Discussion

Comparisons of justification awareness models with other justification epistemic structures such as Fitting, Mkrtychev, and modular models, can be found in [5]. Technically, basic models and Mkrtychev models may be regarded as special cases of Fitting models. On the other hand, Fitting models can be identified as modular models with additional assumptions, cf. [3]. This provides a natural hierarchy of the aforementioned classes of models:

$$\textit{basic and Mkrtychev models} \ \subset \ \textit{Fitting models} \ \subset \ \textit{modular models} \ \subset \ \textit{JAMs}.$$

Even the smallest class, basic models, is already sufficient for mathematical completeness of justification logics. So, the main idea of progressing to Fitting models, modular models, or *JAM*s is not a pursuit of completeness but rather a desire to offer natural models for a variety of epistemic situations involving evidence, belief, and knowledge.

*JAM*s do not offer a complete self-contained analysis of knowledge but rather reduce knowledge to knowledge-producing justifications accepted by the agent. This, however, constitutes a meaningful progress; it decomposes knowledge in a way that moves justification objects to the forefront of epistemic modeling. Note that Gettier and Russell examples, clearly indicate which justifications are knowledge-producing or accepted. So *JAM*s fairly model situations in which the corresponding properties of justifications (knowledge-producing, accepted) are given.

There are many natural open questions that indicate possible research directions. Are justification assertions checkable, decidable for an agent? Is the property of a justification to be knowledge-producing checkable by the agent? In multi-agent cases, how much do agents know about each other and about the model? Do agents know each other's accepted and knowledge-producing justifications? What is the complexity of these new justification logics and what are their feasible fragments which make sense for epistemic modeling?

Acknowledgements. The author is grateful to Melvin Fitting, Vladimir Krupski, Elena Nogina, and Tudor Protopopescu for helpful suggestions. Special thanks to Karen Kletter for editing and proofreading this text.

References

1. Artemov, S.: Explicit provability and constructive semantics. Bull. Symbolic Logic **7**(1), 1–36 (2001)
2. Artemov, S.: The logic of justification. Rev. Symbolic Logic **1**(4), 477–513 (2008)
3. Artemov, S.: The ontology of justifications in the logical setting. Stud. Logica. **100**(1–2), 17–30 (2012)
4. Artemov, S.: Knowing the model. Published online at: arXiv:1610.04955 [math.LO] (2016)
5. Artemov, S.: Epistemic modeling with justifications. Published online at: arXiv:1703.07028 [math.LO] (2017)
6. Cresswell, M.J.: Hyperintensional logic. Stud. Logica. **34**(1), 25–38 (1975)
7. Fagin, R., Halpern, J.: Belief, awareness, and limited reasoning. Artif. Intell. **34**(1), 39–76 (1988)
8. Fagin, R., Halpern, J., Moses, Y., Vardi, M.: Reasoning About Knowledge. MIT Press, Cambridge (1995)
9. Fitting, M.: The logic of proofs, semantically. Ann. Pure Appl. Logic **132**(1), 1–25 (2005)
10. Fitting, M.: Possible world semantics for first-order logic of proofs. Ann. Pure Appl. Logic **165**(1), 225–240 (2014)
11. Fitting, M.: Modal logics, justification logics, and realization. Ann. Pure Appl. Logic **167**(8), 615–648 (2016)
12. Gettier, E.: Is justified true belief knowledge? Analysis **23**, 121–123 (1963)
13. Krupski, V.N.: Operational logic of proofs with functionality condition on proof predicate. In: Adian, S., Nerode, A. (eds.) LFCS 1997. LNCS, vol. 1234, pp. 167–177. Springer, Heidelberg (1997). https://doi.org/10.1007/3-540-63045-7_18
14. Krupski, V.: The single-conclusion proof logic and inference rules specification. Ann. Pure Appl. Logic **113**(1), 181–206 (2002)

15. Krupski, V.: On the sharpness and the single-conclusion property of basic justification models. In: Artemov, S., Nerode, A. (eds.) LFCS 2018. LNCS, vol. 10703, pp. 211–220. Springer, Cham (2018)
16. Kuznets, R., Struder, 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, London (2012)
17. Meyer, J.-J.C., van der Hoek, W.: Epistemic Logic for AI and Computer Science. CUP, Cambridge (1995)
18. Mkrtychev, A.: Models for the logic of proofs. In: Adian, S., Nerode, A. (eds.) LFCS 1997. LNCS, vol. 1234, pp. 266–275. Springer, Heidelberg (1997). https://doi.org/10.1007/3-540-63045-7_27
19. Russell, B.: The Problems of Philosophy. Williams and Norgate, London (1912)
20. Sedlár, I.: Justifications, awareness and epistemic dynamics. In: Artemov, S., Nerode, A. (eds.) LFCS 2013. LNCS, vol. 7734, pp. 307–318. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-35722-0_22
21. Williamson, T.: A note on Gettier cases in epistemic logic. Philos. Stud. **172**(1), 129–140 (2015)

A Minimal Computational Theory of a Minimal Computational Universe

Arnon Avron[1] and Liron Cohen[2](✉)

[1] Tel Aviv University, Tel-Aviv, Israel
aa@post.tau.ac.il
[2] Cornell University, Ithaca, NY, USA
lironcohen@cornell.edu

Abstract. In [3] a general logical framework for formalizing set theories of different strength was suggested. We here employ that framework, focusing on the exploration of *computational* theories. That is, theories whose set of closed terms suffices for denoting every concrete set (including infinite ones) that might be needed in applications, as well as for computations with sets. We demonstrate that already the minimal computational level of the framework, in which only a minimal computational theory and a minimal computational universe are employed, suffices for developing large portions of scientifically applicable mathematics.

Keywords: Formalized mathematics · Computational theories
Computational universes · Rudimentary set theory

1 Introduction

Formalized mathematics and mathematical knowledge management (MKM) are extremely fruitful and quickly expanding fields of research at the intersection of mathematics and computer science (see, e.g., [2,8,23]). The declared goal of these fields is to develop computerized systems that effectively represent all important mathematical knowledge and techniques, while conforming to the highest standards of mathematical rigor. At present there is no general agreement what should be the best framework for this task. However, since most mathematicians view *set theory* as the basic foundation of mathematics, formalized set theories seem to us as the most natural choice.[1,2]

[1] Already in [9] it was argued that "a main asset gained from Set theory is the ability to base reasoning on just a handful of axiom schemes which, in addition to being conceptually simple (even though surprisingly expressive), lend themselves to good automated support". More recently, H. Friedman wrote (in a message on FOM on Sep 14, 2015): "I envision a large system and various important weaker subsystems. Since so much math can be done in systems much weaker than ZFC, this should be reflected in the choice of Gold Standards. There should be a few major Gold Standards ranging from Finite Set Theory to full blown ZFC".

[2] Notable set-based automated provers are Mizar [29], Metamath [25] and SETL [30].

S. Artemov and A. Nerode (Eds.): LFCS 2018, LNCS 10703, pp. 37–54, 2018.
https://doi.org/10.1007/978-3-319-72056-2_3

In [3,4] a logical framework for developing and mechanizing set theories was introduced. Its key properties are that it is based on the usual (type-free) set theoretic language and makes extensive use of statically defined abstract set terms. Furthermore, it enables the use of different logics and set theories of different strength. This modularity of the system has been exploited in [5], where a hierarchy of set theories for formalizing different levels of mathematics within this framework was presented.

The current paper concentrates on one very basic theory, RST_{HF}^{FOL}, from the above-mentioned hierarchy, and on its minimal model. The latter is shown to be the universe J_2 in Jensen's hierarchy [22]. Both RST_{HF}^{FOL} and J_2 are *computational* (in a precise sense defined below). With the help of the formal framework of [3–5] they can therefore be used to make explicit the potential computational content of set theories (first suggested and partially demonstrated in [9]). On the other hand, they also suffice (as we show) for developing large portions of scientifically applicable mathematics [17], especially analysis.[3] In [15–17] it was forcefully argued by Feferman that scientifically applicable mathematics, i.e. the mathematics that is actually indispensable to present-day natural science, can be developed using only predicatively acceptable mathematics. We here support this claim, using a much simpler framework than the systems employed by Feferman.

The restriction to a minimal, concrete framework has of course its price. Not all standard mathematical structures are elements of J_2. (The real line is a case in point.) Hence we have to treat such objects in a different manner: as proper classes. Accordingly, in this paper we introduce for the first time classes into the formal framework of [3–5], and develop efficient ways for handling them.

The paper is organized as follows: In Sect. 2 we present the formal framework, define the notions of computational theory and universe, and describe the computational theories which are minimal within the framework. Section 3 is dedicated to the introduction of standard extensions by definitions of the framework, done in a static way. We define the notions of sets and classes in our framework, and describe the way standard set theoretical notions are dealt with in the system. In Sect. 4 we turn to real analysis, and demonstrate how it can be developed in our minimal computational framework, although the reals are a proper class in it. This includes the introduction of the real line and real functions, as well as formulating and proving classical results concerning these notions.[4] Section 5 concludes with directions for future continuation of the work.[5]

[3] The thesis that J_2 is sufficient for core mathematics was already put forward in [33].

[4] A few of the claims in Sect. 4 have counterparts in [5]. The main difference is that in this paper the claims and their proofs have to be modified to handle classes.

[5] Due to page constraints, all proofs in the paper were omitted, and will appear in an extended version of the current paper.

2 Preliminaries

2.1 The Framework

Notation. To avoid confusion, the parentheses $\{\!\!\{\,\}\!\!\}$ are used in our formal languages, while in the meta-language we use $\{\,\}$. We use the letters $X, Y, Z, \ldots$ for collections; Φ, Θ for finite sets of variables; and $x, y, z, \ldots$ for variables in the formal language. $Fv(exp)$ denotes the set of free variables of exp, and $\varphi\left[{}^{t_1}/_{x_1}, \ldots, {}^{t_n}/_{x_n}\right]$ denotes the result of simultaneously substituting t_i for x_i in φ.

Definition 1. *Let C be a finite set of constants. The language $\mathcal{L}^C_{RST}$ and the associated safety relation $\succ$ are simultaneously defined as follows:*

- *Terms:*
  - *Every variable is a term.*
  - *Every $c \in C$ is a term (taken to be a constant).*
  - *If x is a variable and φ is a formula such that $\varphi \succ \{x\}$, then $\{\!\!\{x \mid \varphi\}\!\!\}$ is a term ($Fv\left(\{\!\!\{x \mid \varphi\}\!\!\}\right) = Fv\left(\varphi\right) - \{x\}$).*
- *Formulas:*
  - *If s, t are terms, then $t = s$, $t \in s$ are atomic formulas.*
  - *If φ, ψ are formulas and x is a variable, then $\neg\varphi, \left(\varphi \wedge \psi\right), \left(\varphi \vee \psi\right)$, $\exists x\varphi$ are formulas.*[6]
- *The safety relation $\succ$:*
  - *If φ is an atomic formula, then $\varphi \succ \emptyset$.*
  - *If t is a term such that $x \notin Fv\left(t\right)$, and $\varphi \in \{x \in x, x \in t, x = t, t = x\}$, then $\varphi \succ \{x\}$.*
  - *If $\varphi \succ \emptyset$, then $\neg\varphi \succ \emptyset$.*
  - *If $\varphi \succ \Theta$ and $\psi \succ \Theta$, then $\varphi \vee \psi \succ \Theta$.*
  - *If $\varphi \succ \Theta$, $\psi \succ \Phi$ and $\Phi \cap Fv\left(\varphi\right) = \emptyset$ or $\Theta \cap Fv\left(\psi\right) = \emptyset$, then $\varphi \wedge \psi \succ \Theta \cup \Phi$.*
  - *If $\varphi \succ \Theta$ and $y \in \Theta$, then $\exists y\varphi \succ \Theta - \{y\}$.*

Notation. We take the usual definition of $\subseteq$ in terms of $\in$, according to which $t \subseteq s \succ \emptyset$. $\{\!\!\{t\}\!\!\}$ denotes the term $\{\!\!\{x \mid x = t\}\!\!\}$, and $s \cup t$ the term $\{\!\!\{x \mid x \in s \vee x \in t\}\!\!\}$.

Definition 2. *The system RST^{FOL}_C is the classical first-order system with variable binding term operator (vbto; see, e.g., [13]) in $\mathcal{L}^C_{RST}$ which is based on the following set of axioms:*[7]

- *Extensionality:* $\forall z\left(z \in x \leftrightarrow z \in y\right) \rightarrow x = y$
- *Comprehension Schema:* $\forall x\left(x \in \{\!\!\{x \mid \varphi\}\!\!\} \leftrightarrow \varphi\right)$
- *Restricted $\in$-induction Schema:*

$$\left(\forall x\left(\forall y\left(y \in x \rightarrow \varphi\left[{}^{y}/_{x}\right]\right) \rightarrow \varphi\right)\right) \rightarrow \forall x\varphi \text{ , for } \varphi \succ \emptyset$$

[6] Though the official language does not include $\forall$ and $\rightarrow$, since we assume classical logic we take $\forall x_1 \ldots \forall x_n\left(\varphi \rightarrow \psi\right)$ as an abbreviation for $\neg\exists x_1 \ldots \exists x_n\left(\varphi \wedge \neg\psi\right)$.

[7] RST^{FOL} can be shown to be equivalent to the system obtained from Gandy's basic set theory [20] by adding to it the Restricted $\in$-induction schema.

In case $HF \in C$, the following axioms are added:

- $\emptyset \in HF$ *(where* $\emptyset = \{\!\{x \in HF \mid x \neq x\}\!\}$*)*
- $\forall x \forall y\, (x \in HF \wedge y \in HF \rightarrow x \cup \{\!\{y\}\!\} \in HF)$
- $\forall y\, (\emptyset \in y \wedge \forall v, w \in y.v \cup \{\!\{w\}\!\} \in y \rightarrow HF \subseteq y)$

Notation. In what follows, in case $C = \emptyset$ we elide C from our notations (e.g., we write RST^{FOL} for $RST^{FOL}_{\emptyset}$). Also, if $C = \{HF\}$ we simply write RST^{FOL}_{HF}.

An important feature of RST^{FOL}_{C} is that its first two axioms directly lead (and are equivalent) to the *set-theoretical* β and η reduction rules (see [3]).

In [3] it was suggested that the computationally meaningful instances of the Comprehension Axiom are those which determine the collections they define in an absolute way, independently of any "surrounding universe". In the context of set theory, a formula φ is "computable" w.r.t. x if the collection $\{x \mid \varphi\,(x, y_1, ..., y_n)\}$ is completely and uniquely determined by the identity of the parameters $y_1, ..., y_n$, and the identity of other objects referred to in the formula (all of which are well-determined beforehand). Note that φ is computable for $\emptyset$ iff it is absolute in the usual sense of set theory. In order to translate this idea into an exact, *syntactic* definition, the safety relation is used. Thus, only those formulas which are safe with respect to $\{x\}$ are allowed in the Comprehension Scheme.

Concerning $\in$-induction, even the full one does not seem to be in any conflict with the notion of a computational theory since it only imposes further restrictions on the collection of acceptable sets. Nevertheless, to be on the safe side, we adopt here only a very restricted variation of it. Moreover, we try to avoid (when possible) the use of this axiom, and shall point out the places where it is used.

It is easy to verify that the system RST^{FOL}_{C} is a proper subsystem of ZF. On the other hand, in [3] it was shown that the full power of ZF can be achieved by simply adding certain syntactic clauses to the definition of the safety relation.

While the formal language allows the use of set terms, it also provides a mechanizable static check of their validity due to the syntactic safety relation. To obtain decidable syntax logically equivalent formulas are not taken to be safe w.r.t. the same set of variables. However, if $\varphi \leftrightarrow \psi$ is provable in RST^{FOL}_{C}, then so is $x \in \{\!\{x \mid \varphi\}\!\} \leftrightarrow \psi$. Thus, we freely write $\{\!\{x \mid \psi\}\!\}$ for of $\{\!\{x \mid \varphi\}\!\}$ for such φ, ψ.

Definition 3. *Let C be a set of constants.*

1. *A function is called C-rudimentary if it rudimentary relative to the interpretations of the constants in C.*[8]
2. *A C-universe is a transitive collection of sets closed under C-rudimentary functions.*

For simplicity, in what follows we do not distinguish between a C-universe W and a structure for $\mathcal{L}^{C}_{RST}$ with domain W and an interpretation function I that

[8] Rudimentary functions are obtained by omitting the recursion schema from the usual list of schemata for primitive recursive set functions (see, e.g., [14]).

assigns the obvious interpretations to the symbols $\in$, $=$, the set of hereditary finite sets to HF (if $HF \in C$), and an element in W to every $c \in C$.

Definition 4. *Let v be an assignment in a C-universe W. For a term t and formula φ of $\mathcal{L}^{C}_{RST}$, a collection $\|t\|^{W}_{v}$ and a truth value $\|\varphi\|^{W}_{v} \in \{\mathbf{t}, \mathbf{f}\}$ are standardly defined, with the additional clause: $\|\{\!\!\{x \mid \varphi\}\!\!\}\|^{W}_{v} = \left\{a \in W \mid \|\varphi\|^{W}_{v[x:=a]} = \mathbf{t}\right\}$.*[9]

From Corollary 6 below it follows that $\|t\|^{W}_{v}$ is an element of W, and $\|\varphi\|^{W}_{v}$ denotes the truth value of the formula φ under W and v.

Notation. In case exp is a closed expression, we denote by $\|exp\|^{W}$ the value of exp in W, and at times we omit the superscript W and simply write $\|exp\|$.

The following theorem is a slight generalization of a theorem in [4].

Theorem 5. *Let C be a set of constants.*

1. *If F is an n-ary C-rudimentary function, then there exists a formula φ_F of $\mathcal{L}^{C}_{RST}$ s.t. $Fv(\varphi_F) \subseteq \{y, x_1, ..., x_n\}$, $\varphi_F \succ \{y\}$ and $F(x_1, ..., x_n) = \{y \mid \varphi_F\}$.*
2. *If φ is a formula of $\mathcal{L}^{C}_{RST}$ s.t. $Fv(\varphi) \subseteq \{y_1, ..., y_k, x_1, ..., x_n\}$ and $\varphi \succ \{y_1, ..., y_k\}$, then there exists a C-rudimentary function F_φ s.t. $F_\varphi(x_1, ..., x_n) = \{\langle y_1, ..., y_k\rangle \mid \varphi\}$.*
3. *If t is a term of $\mathcal{L}^{C}_{RST}$ s.t. $Fv(t) \subseteq \{x_1, ..., x_n\}$, then there exists a C-rudimentary function F_t s.t. $F_t(x_1, ..., x_n) = t$ for every $x_1, ..., x_n$.*

Corollary 6. *Let v be an assignment in a C-universe W.*

1. *For a term t of $\mathcal{L}^{C}_{RST}$, $\|t\|^{W}_{v} \in W$.*
2. *For a formula φ of $\mathcal{L}^{C}_{RST}$ s.t. $\{y_1, ..., y_n\} \subseteq Fv(\varphi)$:*
   (a) *If $\varphi \succ \{y_1, ..., y_n\}$ $(n > 0)$, $\left\{\langle a_1, ..., a_n\rangle \in W^n \mid \|\varphi\|^{W}_{v[\boldsymbol{y}:=\overrightarrow{a}]} = \mathbf{t}\right\} \in W$.*
   (b) *If $\varphi \succ \emptyset$ and $X \in W$, then $\left\{\langle a_1, ..., a_n\rangle \in X^n \mid \|\varphi\|^{W}_{v[\boldsymbol{y}:=\overrightarrow{a}]} = \mathbf{t}\right\} \in W$.*

If t is a closed term s.t. $\|t\|^{W} = X$, we say that t defines X (X is definable by t).

Corollary 7. *Any C-universe is a model of RST^{FOL}_{C}.*

Lemma 8. *[5] The following notations are available in RST^{FOL} (i.e. they can be introduced as abbreviations in $\mathcal{L}_{RST}$ and their basic properties are provable in RST^{FOL}): $\emptyset$, $\langle t_1, ..., t_n\rangle$, $\{\!\!\{t_1, ..., t_n\}\!\!\}$, $\{\!\!\{x \in t \mid \varphi\}\!\!\}$ (provided $\varphi \succ \emptyset$ and $x \notin Fv(t)$), $\{\!\!\{t \mid x \in s\}\!\!\}$ (provided $x \notin Fv(s)$), $s \times t$, $s \cup t$, $s \cap t$, $s - t$, $\cup t$, $\cap t$, $\pi_1(t)$, $\pi_2(t)$, $Dom(t)$, $Im(t)$, $\iota x.\varphi$ (provided $\varphi \succ \{x\}$), $\lambda x \in s.t$ (provided $x \notin Fv(s)$).*

[9] $v[x := a]$ denotes the x-variant of v which assigns a to x.

2.2 Computational Theories and Universes

Computations within a set of objects require concrete representations of these objects. Accordingly, we call a theory *computational* if its set of closed terms induces in a natural way a minimal model of the theory, and it enables the key properties of these elements to be provable within it. Next we provide a more formal definition for the case of set theories which are defined within our general framework. Note that from a Platonist point of view, the set of closed terms of such a theory $\mathcal{T}$ induces some subset $\mathcal{S}_{\mathcal{T}}$ of the cumulative universe of sets V, as well as some subset $\mathcal{M}_{\mathcal{T}}$ of any transitive model $\mathcal{M}$ of $\mathcal{T}$.

Definition 9. *1. A theory $\mathcal{T}$ in the above framework is called* computational *if the set $\mathcal{S}_{\mathcal{T}}$ it induces is a transitive model of $\mathcal{T}$, and the identity of $\mathcal{S}_{\mathcal{T}}$ is absolute in the sense that $\mathcal{M}_{\mathcal{T}} = \mathcal{S}_{\mathcal{T}}$ for any transitive model $\mathcal{M}$ of $\mathcal{T}$ (implying that $\mathcal{S}_{\mathcal{T}}$ is actually a* minimal *transitive model of $\mathcal{T}$).*
2. A set is called computational if it is $\mathcal{S}_{\mathcal{T}}$ for some computational theory $\mathcal{T}$.

The most basic computational theories are the two minimal theories in the hierarchy of systems developed in [5]. This fact, as well as the corresponding computational universes, are described in the following three results from [5].

Proposition 10. *Let J_1, J_2 be the first two universes in Jensen's hierarchy [22].*

1. *J_1 is a model of RST^{FOL}.*
2. *J_2 with the interpretation of HF as J_1 is a model of RST^{FOL}_{HF}.*

Theorem 11

- *$X \in J_1$ iff there is a closed term t of $\mathcal{L}_{RST}$ s.t. $\|t\|^{J_1} = X$.*
- *$X \in J_2$ iff there is a closed term t of $\mathcal{L}^{HF}_{RST}$ such that $\|t\|^{J_2} = X$.*

Corollary 12. *RST^{FOL} and RST^{FOL}_{HF} are computational, and J_1 and J_2 are their computational universes.*

Now J_1, the minimal computational universe, is the set of hereditary finite sets. This universe captures the standard data structures used in computer science, like strings and lists. However, in order to be able to capture computational structures with infinite objects, we have to move to RST^{FOL}_{HF}, whose computational universe, J_2, seems to be the minimal universe that suffices for this purpose. RST^{FOL}_{HF} still allows for a very concrete, computationally-oriented interpretation, and it is appropriate for mechanical manipulations and interactive theorem proving. Moreover, as noted in the introduction, its corresponding universe J_2 is rich enough for a systematic development of applicable mathematics.

3 Static Extensions by Definitions

When working in a minimal computational universe such as J_2 (as done in the next section), many of the standard mathematical objects (such as the real line and real functions) are only available in our framework as proper classes. Thus, in order to be able to formalize standard theorems regarding such objects we must enrich our language to include them. Introducing classes into our framework, however, is a part of the more general method of extensions by definitions which is an essential part of every mathematical research and its presentation. Now, there are two principles that govern this process in our framework. First, the static nature of our framework demands that conservatively expanding the language of a given theory should be reduced to the use of *abbreviations*. Second, since the introduction of new predicates and function symbols creates new atomic formulas and terms, one should be careful that the basic conditions concerning the underlying safety relation $\succ$ are preserved. Thus only formulas φ s.t. $\varphi \succ \emptyset$ can be used for defining new predicate symbols.

We start with the problem of introducing new unary predicate symbols to the base langauge.[10] In standard practice such extensions are carried out by introducing a new unary predicate symbol P and either treating $P(t)$ as an abbreviation for $\varphi(t)$ for some formula φ, or (what is more practical) adding $\forall x\,(P(x) \leftrightarrow \varphi)$ as an axiom to the (current version of the base) theory, obtaining by this a conservative theory in the extended language. However, in the set theoretical framework it is possible and frequently more convenient to uniformly use class terms, rather than introduce a new predicate symbol each time. Thus, instead of writing "$P(t)$" one uses an appropriate class term S and writes "$t \in S$". Whatever approach is chosen – in order to respect the definition of a safety relation, class terms should be restricted so that "$t \in S$" is safe w.r.t. $\emptyset$. Accordingly, we extend our language by incorporating class terms which are objects of the form $\{\!\!\{x \,\hat{|}\, \varphi\}\!\!\}$, where $\varphi \succ \emptyset$. The use of these terms is done in the standard way. In particular, $t \in \{\!\!\{x \,\hat{|}\, \varphi\}\!\!\}$ (where t is free for x in φ) is equivalent to (and may be taken as an abbreviation for) $\varphi\left[{}^{t}/_{x}\right]$. It should be emphasized that a class term is not a valid term in the language, only a definable predicate. The addition of the new notation does not enhance the expressive power of $\mathcal{L}_{RST}^{C}$, but only increases the ease of using it.

A further conservative extension of the language that we shall use incorporates free class variables, $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}$, and free function variables, $\boldsymbol{F}, \boldsymbol{G}$, into $\mathcal{L}_{RST}^{C}$ (as in free-variable second-order logic [31]). These variables stand for arbitrary class or function terms (the latter is defined in Definition 20), and they may only appear as *free* variables, *never to be quantified*. We allow occurrences of such variables inside a formula in a class term or a function term. One may think of a formula with such variables as a schema, where the variables play the role of "place holders", and whose substitution instances abbreviate official formulas of the language (see Example 2). In effect, a formula $\psi(\boldsymbol{X})$ with free class variable $\boldsymbol{X}$ can be intuitively interpreted as "for any *given* class X, $\psi(X)$ holds". Thus, a free-variable formulation has the flavor of a universal formula. Therefore, this addition allows us to make statements about *all* potential classes as well as *all* potential functions.

[10] The use of n-ary predicates can standardly be reduced, of course, to unary predicates.

We define $\left\| \{\!\!\{ x \mid \varphi \}\!\!\} \right\|_v^W = \left\{ a \in W \mid \|\varphi\|_{v[x:=a]}^W = \mathbf{t} \right\}$. We say that the class term defines the latter collection (which might not be an element of W).

Definition 13. *Let X be a collection of elements in W.*

- *X is a $\succ$-*set *if there is a closed term that defines it. If X is a $\succ$-set, $\widetilde{X}$ denotes some closed term that defines it.*
- *X is a $\succ$-class if there is a closed class term that defines it. If X is a $\succ$-class, $\bar{X}$ denotes some closed class term that defines it.*

Note that, by Corollary 6, if X is a $\succ$-*set* then $X \in W$.

Proposition 14. *The following holds:*

1. *Every $\succ$-set is a $\succ$-class.*
2. *The intersection of a $\succ$-class with a $\succ$-set is a $\succ$-set.*
3. *Every $\succ$-class that is contained in a $\succ$-set is a $\succ$-set.*

Remark 15. A semantic counterpart of our notion of a $\succ$-class was used in [33], and is there called an ι-class. It is defined as a definable subset of J_2 whose intersection with any element of J_2 is in J_2. The second condition in this definition seems somewhat ad hoc. More importantly, it is unclear how it can be checked in general, and what kind of set theory is needed to establish that certain collections are ι-classes. The definition of a $\succ$-class used here is, in contrast, motivated by and based on purely syntactical considerations. It is also a simplification of the notion of ι-class as by Proposition 14(2) every $\succ$-class is an ι-class.[11]

Proposition 16. *The following holds:*

- *Let Y be a $\succ$-set. If $\varphi \succ \emptyset$ and $Fv(\varphi) \subseteq \{x\}$, then $\{x \in Y \mid \varphi\}$ is a $\succ$-set.*
- *If $\varphi \succ \{x_1, ..., x_n\}$, then $\{\langle x_1, ..., x_n \rangle \mid \varphi\}$ is a $\succ$-set.*

Proposition 17. *For every n-ary C-rudimentary function f there is a term t with $Fv(t) \subseteq \{x_1, ..., x_n\}$ s.t. for any $\langle A_1, ..., A_n \rangle \in W^n$, f returns the $\succ$-set $\|t\|_{[x_1:=A_1,...,x_n:=A_n]}^W$.*

Proposition 18. *If X, Y are $\succ$-classes, so are $X \cup Y$, $X \cap Y$, $X \times Y$, $J_2 - X$, and $P_{J_2}(X) = \{z \in J_2 \mid z \subseteq X\}$.*

For a class term s we denote by 2^s the class term $\{\!\!\{ z \mid z \subseteq s \}\!\!\}$. Note that for any assignment v in W and class term s, $\|2^s\|_v^W$ is equal to $P_W\left(\|s\|_v^W\right)$, i.e., the intersection of the power set of $\|s\|_v^W$ and W. This demonstrates the main difference between set terms and class terms. The interpretation of set terms is absolute, whereas the interpretation of class terms might not be (though membership in the interpretation of a class term is absolute).

[11] Two other ideas that appear in the sequel were adopted from [33]: treating the collection of reals as a proper class, and the use of codes for handling certain classes. It should nevertheless be emphasized that the framework in [33] is exclusively based on semantical considerations, and it is unclear how it can be turned into a formal theory like ZF or PA (and it is certainly not suitable for mechanization as is).

Definition 19. *A* $\succ$-relation from *a* $\succ$-class X to *a* $\succ$-class Y *is a* $\succ$*-class* A *s.t.* $A \subseteq X \times Y$. *A* $\succ$*-relation is called* small *if it is a* $\succ$*-set.*

Next we extend our framework by the introduction of new function symbols. This poses a new difficulty. While new relation symbols are commonly introduced in a static way, new function symbols are usually introduced *dynamically*: a new function symbol is made available after appropriate existence and uniqueness theorems had been proven.

However, one of the main guiding principles of our framework is that its languages should be treated exclusively in a *static* way. Thus function symbols, too, are introduced only as abbreviations for definable operations on sets.[12]

Definition 20. – *For a closed class term* T *and a term* t *of* $\mathcal{L}_{RST}^{C}$, $\lambda x \in \mathsf{T}.t$ *is a* function term *which is an abbreviation for* $\lbrace\!\!\vert z \,\hat{\vert}\, \exists x \exists y \left(z \check{\doteq} \langle x, y\rangle \wedge x \in \mathsf{T} \wedge y = t\right) \vert\!\!\rbrace$.[13]

- *A* $\succ$*-class* F *is called a* $\succ$-function on a $\succ$*-class* X *if there is a function term* $\lambda x \in \mathsf{T}.t$ *such that* $X = \|\mathsf{T}\|$, $Fv\left(t\right) \subseteq \{x\}$ *and* $F = \|\lambda x \in \mathsf{T}.t\|$. t *is called a term which* represents F.
- *A* $\succ$*-class is called a* $\succ$*-function if it is a* $\succ$*-function on some* $\succ$*-class.*
- *A* $\succ$*-function is called* small *if it is a* $\succ$*-set.*

Note that the standard functionality condition is always satisfied in a $\succ$*-function.*

Terminology. In what follows, claiming that an object is *available in* RST_C^{FOL} *as a* $\succ$*-function (*$\succ$*-relation)* means that it is definable as a $\succ$-function *(*$\succ$*-relation)* in $\mathcal{L}_{RST}^{C}$, and that its basic properties are provable in RST_C^{FOL}.[14]

Proposition 21. *Let* X, Y *be* $\succ$*-classes and* R *a* $\succ$*-relation from* X *to* Y.

1. R *is small iff* $Dom\left(R\right)$ *and* $Im\left(R\right)$ *are* $\succ$*-sets.*
2. $R^{-1} = \{\langle y, x\rangle \mid \langle x, y\rangle \in R\}$ *is available in* RST_C^{FOL} *as a* $\succ$*-relation from* Y *to* X. *If* R *is small, then so is* R^{-1}.
3. *If* $Z \subseteq X$ *and* $U \subseteq Y$ *are* $\succ$*-classes, then* $R \cap (Z \times U)$ *is available in* RST_C^{FOL} *as a* $\succ$*-relation from* Z *to* U.

Proposition 22. *A* $\succ$*-set is a function according to the standard mathematical definition (a single-valued relation) iff it is a small* $\succ$*-function.*

Notation. Let $F = \left\|\lambda x \in \bar{X}.t\right\|$ be a $\succ$-function. We employ standard β-reduction for λ terms. Thus, we write $F\left(s\right)$ for $t\left[{}^{s}/{}_{x}\right]$ if s is free for x in t. Hence $F\left(s\right) = y$ stands for $t\left[{}^{s}/{}_{x}\right] = y$, and so if $y \notin Fv\left[t\right] \cup Fv\left[s\right] \setminus \{x\}$, then $F\left(s\right) = y \succ \{y\}$.

[12] In this paper, as in standard mathematical textbooks, the term "function" is used both for collections of ordered pairs and for set-theoretical operations (such as $\cup$).

[13] We abbreviate by $z \check{\doteq} \langle x, y\rangle$ and $\langle x, y\rangle \check{\in} z$ the two formulas that are provably equivalent to $z = \langle x, y\rangle$ and $\langle x, y\rangle \in z$ and are safe w.r.t. $\{x, y\}$ which were introduced in [5].

[14] The "basic properties" of a certain object is of course a fuzzy notion. However, it is not difficult to identify its meaning in each particular case, as will be demonstrated in several examples below.

Proposition 23 (Replacement axiom in class form). *Let F be a $\succ$-function on a $\succ$-class X. Then for every $\succ$-set $A \subseteq X$, $F[A] = \{F(a) \mid a \in A\}$ is a $\succ$-set.*

Below is a natural generalization of Definition 20 to functions of several variables.

Lemma 24. *If* $X_1, ..., X_n$ are *$\succ$-classes and t is a term s.t. $Fv(t) \subseteq \{x_1, ..., x_n\}$, then $F = \left\|\lambda x_1 \in \bar{X}_1, ..., x_n \in \bar{X}_n.t\right\|$ is available in RST_C^{FOL} as a $\succ$-function on $X_1 \times ... \times X_n$. (where $\lambda x_1 \in \bar{X}_1, ..., x_n \in \bar{X}_n.t$ is an abbreviation for $\{\!\{ \langle\langle x_1, ..., x_n\rangle, t\rangle \;\hat{|}\; \langle x_1, ..., x_n\rangle \in \bar{X}_1 \times ... \times \bar{X}_n \}\!\}$).*

Corollary 25. *Every C-rudimentary function is available in RST_C^{FOL} as a $\succ$-function.*

Proposition 26. *Let F be a $\succ$-function on a $\succ$-class X.*

1. *F is small iff X is a $\succ$-set.*
2. *If Y_0 is a $\succ$-class, then $F^{-1}[Y_0] = \{a \in X \mid F(a) \in Y_0\}$ is a $\succ$-class. If F is small, then $F^{-1}[Y_0]$ is a $\succ$-set.*
3. *If $X_0 \subseteq X$ is a $\succ$-class, then $F \upharpoonright_{X_0}$ is available in RST_C^{FOL} as a $\succ$-function.*
4. *$G \circ F$ is available in RST_C^{FOL} as a $\succ$-function on X, in case G is a $\succ$-function on a $\succ$-class Y and $Im(F) \subseteq Y$.*
5. *If G is a $\succ$-function on a $\succ$-class Y and F and G agree on $X \cap Y$, then $G \cup F$ is available in RST_C^{FOL} as a $\succ$-function on $X \cup Y$.*
6. *If Z is a $\succ$-class then the identity on Z and any constant function on Z are available in RST_C^{FOL} as $\succ$-functions.*

4 Real Analysis in J_2

It is not difficult to formalize the definitions, claims, and proofs of this section in our formal framework. These translations are straightforward, but rather tedious. Hence we shall omit them, with the exception of a few outlined examples.

4.1 The Natural Numbers

We follow the standard construction of the natural numbers: $0 := \emptyset$; $\quad n+1 := S(n)$, where $S(n) = n \cup \{n\}$. Each $n \in \mathbb{N}$ is a $\succ$-set, and $\mathbb{N}$ (the set of natural numbers) is contained in the interpretation of HF.

In mainstream mathematics, as well as in standard computerized theorem provers, the collection of natural numbers is taken as a basic object. This is because it constitutes a well-understood, computational concept. Now, the computational universe associated with RST^{FOL} is J_1, in which $\mathbb{N}$ is available only as a proper $\succ$-class. To solve this, in RST_{HF}^{FOL} a special constant HF was added, whose axioms ensure (as far as possible on the first-order level) that it is to be interpreted as the set of hereditary finite sets. These axioms in fact replace the usual infinity axiom of ZF. This increases the computational power of the theory and captures the natural numbers as a $\succ$-set. Thus, in what follows we restrict our attention to the computational theory RST_{HF}^{FOL} and its computational universe J_2. Therefore, for readability, we simply write $\|exp\|_v$ instead of $\|exp\|_v^{J_2}$.

The induction rule is available in RST_{HF}^{FOL}, but only for $\varphi \succ \emptyset$.

Proposition 27. $\vdash_{RST_{HF}^{FOL}} \ (\varphi(0) \wedge \forall x (\varphi \rightarrow \varphi(S(x)))) \ \rightarrow \ \forall x \ \in \ \widetilde{\mathbb{N}}.\varphi$, *for* $\varphi \succ \emptyset$.

Basic properties of the natural numbers which can be formulated in the language of first-order Peano arithmetics are provable in RST_{HF}^{FOL} using the restricted induction principle given in Proposition 27. This is because in their translation to $\mathcal{L}_{RST}^{HF}$, all the quantifications are bounded in $\mathbb{N}$, and thus they are safe w.r.t. $\emptyset$.[15]

4.2 The Real Line

The standard construction of $\mathbb{Z}$, the set of integers, as the set of ordered pairs $(\mathbb{N} \times \{0\}) \cup (\{0\} \times \mathbb{N})$ can be easily carried out in RST_{HF}^{FOL}, as can the usual construction of $\mathbb{Q}$, the set of rationals, in terms of ordered pairs of relatively prime integers. There is also no difficulty in defining the standard orderings on $\mathbb{Z}$ and $\mathbb{Q}$ as small $\succ$-relations, as well as the standard functions of addition and multiplication as small $\succ$-functions. The main properties of addition and multiplication are provable in RST_{HF}^{FOL}, as the standard proofs by induction can be carried out within it. Furthermore, all the basic properties of $\mathbb{Z}$ and $\mathbb{Q}$ (such as $\mathbb{Q}$ being a dense unbounded field) are straightforwardly proven in RST_{HF}^{FOL}.

Now we turn to the standard construction of the real line using Dedekind cuts. Since it is well known that the real line and its open segments are not absolute, they cannot be $\succ$-sets, only proper $\succ$-classes. Thus the collection of real numbers in RST_{HF}^{FOL} will not be a term but merely a *definable predicate.*[16]

Let $\psi(u) = \forall x, y \in \widetilde{\mathbb{Q}}.x \in u \wedge y < x \rightarrow y \in u$, $\varphi(u) = \neg\exists x \in u \forall y \in u.y \leq x$.

Definition 28 (The Reals). $\mathbb{R}$ *is* $\left\| \lbrace\!\lbrace u \in \overline{P_{J_2}(\mathbb{Q}) \setminus \{\emptyset, \mathbb{Q}\}} \hat{\restriction} \psi(u) \wedge \varphi(u) \rbrace\!\rbrace \right\|$.

The above term is a valid class term as $P_{J_2}(\mathbb{Q}) \setminus \{\emptyset, \mathbb{Q}\}$ is a $\succ$-class, and $\varphi, \psi \succ \emptyset$.

Note that the $\succ$-class $\mathbb{R}$ is not the "real" real-line (if such a thing really exists). However, it does contain all *computable* real numbers, such as $\sqrt{2}$ and π (see [5]).

Notation. We employ the following notations: $\mathbb{Q}^+ = \{q \in \mathbb{Q} \mid 0 < q\}$, $\mathbb{R}^+ = \{r \in \mathbb{R} \mid 0 < r\}$, $(a, b) = \{r \in \mathbb{R} \mid a < r < b\}$ and $[a, b] = \{r \in \mathbb{R} \mid a \leq r \leq b\}$, for a, b real numbers.[17]

[15] It can be shown that the power of full induction over $\mathbb{N}$ (i.e. for *any* formula φ) can be achieved by adding to RST_{HF}^{FOL} the full $\in$-induction scheme.

[16] As noted in Footnote 4, some of claims in the sequel have counterparts in [5]. However, the minimality restriction on the universe employed in this paper, which in turn requires the use of classes, makes a crucial difference.

[17] Notice that $\mathbb{Q}^+$ is a $\succ$-set and $\mathbb{R}^+$ is a $\succ$-class.

Proposition 29. *The following holds:*

1. *The standard ordering $<$ on $\mathbb{R}$ is available in RST_{HF}^{FOL} as a $\succ$-relation.*
2. *The standard addition and multiplication of reals are available in RST_{HF}^{FOL} as $\succ$-functions.*

We next show that the least upper bound principle is provable in RST_{HF}^{FOL} for $\succ$-subsets of $\mathbb{R}$.

Theorem 30. *It is provable in RST_{HF}^{FOL} that every nonempty $\succ$-subset of $\mathbb{R}$ that is bounded above has a least upper bound in $\mathbb{R}$. Furthermore, the induced mapping (l.u.b) is available in RST_{HF}^{FOL} as a $\succ$-function.*

Theorem 30 only states that $\succ$-*subsets* of $\mathbb{R}$ have the least upper bound property. Thus, it is insufficient for the development of most of standard mathematics in RST_{HF}^{FOL}. The reason is that even the most basic substructures of $\mathbb{R}$, like the intervals, are not $\succ$-sets, but proper $\succ$-classes in RST_{HF}^{FOL}. Hence, a stronger version of the theorem, which ensures that the least upper bound property holds for standard $\succ$-subclasses of $\mathbb{R}$, is needed. Theorem 40 below provides such an extension, but it requires some additional definitions and propositions.[18]

First we consider $\succ$-classes $U \subseteq \mathbb{R}$ which are open. These $\succ$-classes are generally not $\succ$-sets (unless empty), since they contain an interval of positive length, which is a proper $\succ$-class and thus cannot be contained in a $\succ$-set (see Proposition 14(3)). Clearly, there is no such thing as a $\succ$-set of $\succ$-classes, as a proper $\succ$-class can never be an element of another $\succ$-set or $\succ$-class. However, the use of coding (following [32,33][19]) allows us, for example, to replace the meaningless statement "the union of a $\succ$-set of $\succ$-classes is a $\succ$-class" with "given a $\succ$-set of codes for $\succ$-classes, the union of the corresponding $\succ$-classes is a $\succ$-class".

The coding technique we use is based on the standard mathematical notation for a "family of sets", $(A_i)_{i \in I}$, where I is a set of indices and A_i is a set for each $i \in I$. In RST_{HF}^{FOL} we cannot construct the collection of all such A_i's if A_i is a $\succ$-class for some $i \in I$. Thus, we treat the $\succ$-set I as a code for the "family of classes" $(A_i)_{i \in I}$. In fact, we mainly use the union of such families, i.e., $\bigcup\limits_{i \in I} A_i$.

Definition 31. *For any $p \in \mathbb{R}$ and $q \in \mathbb{R}^+$, the open ball $B_q(p)$ is the $\succ$-class $\{r \in \mathbb{R} \mid |r - p| < q\}$.*

Definition 32. *Let $U \subseteq \mathbb{R}$ be a $\succ$-class. If there exists a $\succ$-set $u \subseteq \mathbb{Q} \times \mathbb{Q}^+$ s.t. $U = \bigcup\limits_{\langle p,q \rangle \in u} B_q(p) = \{r \in \mathbb{R} \mid \exists p, q\,(\langle p, q\rangle \in u \wedge |r - p| < q)\}$, then U is called* open *and u is a* code *for U.*

[18] It should be noted that the least upper bound principle is not derivable for all subsets also in Weyl's approach [34]. We next use similar coding techniques to the ones employed by Weyl to obtain the principle for standard mathematical objects.

[19] In [33] such codings are called "proxies".

In what follows, the formalizations in RST_{HF}^{FOL} are carried out as follows:

- To quantify over open $\succ$-classes: $Qu \subseteq \widetilde{\mathbb{Q} \times \mathbb{Q}^{+}}$ $(Q \in \{\forall, \exists\})$.
- To decode the open $\succ$-class whose code is u:

$$dec\,(u) := \{\!\!\{ r \in \bar{\mathbb{R}} \mathrel{\hat{|}} \exists p, q\,(\langle p, q\rangle \mathrel{\check{\in}} u \wedge |r - p| < q)\,\}\!\!\}$$

- To state that a class variable $\boldsymbol{U}$ is an open $\succ$-class:

$$Open\,(\boldsymbol{U}) := \exists u \subseteq \widetilde{\mathbb{Q} \times \mathbb{Q}^{+}}.\boldsymbol{U} = dec\,(u)$$

Proposition 33. *The following are provable in* RST_{HF}^{FOL}*:*

1. *For any $\succ$-set $u \subseteq \mathbb{R} \times \mathbb{R}^{+}$, $\{r \in \mathbb{R} \mid \exists p, q\,(\langle p, q\rangle \in u \wedge |r - p| < q)\}$ is an open $\succ$-class.*
2. *The open ball $B_q\,(p)$ is an open $\succ$-class for any $p \in \mathbb{R}$ and $q \in \mathbb{R}^{+}$.*

Proposition 34. *The following are provable in* RST_{HF}^{FOL}*:*

1. *The union of a $\succ$-set of open $\succ$-classes is an open $\succ$-class. i.e., given a $\succ$-set of codes of open $\succ$-classes, the union of the corresponding open $\succ$-classes is an open $\succ$-class.*
2. *The intersection of finitely many open $\succ$-classes is an open $\succ$-class.*

Example 1. As an example of the use of the coding technique, we demonstrate the formalization of Proposition 34(1):

$$\forall z.(\forall x \in z.x \subseteq \widetilde{\mathbb{Q} \times \mathbb{Q}^{+}}) \to \exists w \subseteq \widetilde{\mathbb{Q} \times \mathbb{Q}^{+}}.dec\,(w) = \{\!\!\{ r \mathrel{\hat{|}} \exists x \in z.r \in dec\,(x)\,\}\!\!\}$$

Definition 35. *A $\succ$-class $X \subseteq \mathbb{R}$ is* closed *if $\mathbb{R} - X$ is open.*

Lemma 36. *Let $X \subseteq \mathbb{R}$ be a $\succ$-class and $A \subseteq X$ be a $\succ$-set. The following are equivalent in* RST_{HF}^{FOL}*:*

1. *Every open ball about a point in X intersects A.*
2. *Every open $\succ$-class that intersects X also intersects A.*

Example 2. As an example of a full formalization which uses class variables, the formalization of the Lemma above is:

$$\begin{gathered}\phi := \boldsymbol{X} \subseteq \bar{\mathbb{R}} \to \forall a \subseteq \boldsymbol{X}\,\big(\forall x \in \boldsymbol{X} \forall \varepsilon \in \bar{\mathbb{R}}^{+}\,(B_{\varepsilon}\,(x) \cap a \neq \emptyset) \leftrightarrow \\ \forall u \subseteq \widetilde{\mathbb{Q} \times \mathbb{Q}^{+}}\,(dec\,(u) \cap \boldsymbol{X} \neq \emptyset \to dec\,(u) \cap a \neq \emptyset)\big)\end{gathered}$$

We now demonstrate how to obtain a formula in the basic $\mathcal{L}_{RST}^{HF}$ by replacing each appearance of a class term or variable with the formula it stands for. First, we explain the translation of $x \in \bar{\mathbb{R}}$ to $\mathcal{L}_{RST}^{HF}$. One iteration of the translation entails $x \in \overline{P_{J_2}\,(\mathbb{Q}) \setminus \{\emptyset, \mathbb{Q}\}} \wedge \varphi\,(x) \wedge \psi\,(x)$ for φ, ψ as in Definition 28. A second iteration yields $R\,(x) := x \subseteq \tilde{\mathbb{Q}} \wedge x \neq \tilde{\mathbb{Q}} \wedge x \neq \emptyset \wedge \varphi\,(x) \wedge \psi\,(x)$ which is in $\mathcal{L}_{RST}^{HF}$. For the translation of ϕ, first substitute $\{\!\!\{ x \mathrel{\hat{|}} \theta \}\!\!\}$ for $\boldsymbol{X}$, where $\theta \succ \emptyset$. Proceeding with the translation steps results in the following formula (scheme) of $\mathcal{L}_{RST}^{HF}$, for $\theta \succ \emptyset$:

$$\forall b\,(\theta\,(b) \rightarrow R\,(b)) \rightarrow \forall a\,((\forall z.z \in a \rightarrow \theta\,(z)) \rightarrow \forall x\,(\theta\,(x) \rightarrow \forall \varepsilon\,((R\,(\varepsilon) \wedge 0 < \varepsilon) \rightarrow$$
$$\exists w.\,|w - x| < \varepsilon \wedge w \in a \leftrightarrow \forall u \subseteq \widetilde{\mathbb{Q} \times \mathbb{Q}^{+}}\,(\exists w.R\,(w) \wedge \exists p, q\,(\langle p, q\rangle\,\check{\in} u \wedge |w - p| < q) \wedge$$
$$\theta\,(w)) \rightarrow \exists w.R\,(w) \wedge \exists p, q\,(\langle p, q\rangle\,\check{\in} u \wedge |w - p| < q) \wedge w \in a)$$

Remark 37. When we say that a theorem about a $\succ$-class or a $\succ$-function is provable in RST_{HF}^{FOL} (as in Lemma 36), we mean that it can be formalized and proved as a scheme. That is, that its proof can be carried out in RST_{HF}^{FOL} using a uniform scheme. The one exception is theorems about open $\succ$-classes, which due to the coding machinery can be fully formalized and proved in RST_{HF}^{FOL}.

Definition 38. *Let $X \subseteq \mathbb{R}$ be a $\succ$-class, and $A \subseteq X$ a $\succ$-set. A is called* dense *in X if one of the conditions of Lemma 36 holds. X is called* separable *if it contains a dense $\succ$-subset.*

Proposition 39. *It is provable in RST_{HF}^{FOL} that an open $\succ$-subclass of a separable $\succ$-class is separable.*

Now we can finally turn to prove a more encompassing least upper bound theorem.

Theorem 40. *It is provable in RST_{HF}^{FOL} that every nonempty separable $\succ$-subclass of $\mathbb{R}$ that is bounded above has a least upper bound in $\mathbb{R}$.*

Definition 41. *A $\succ$-class $X \subseteq \mathbb{R}$ is called an* interval *if for any $a, b \in X$ s.t. $a < b$: if $c \in \mathbb{R} \wedge a < c < b$ then $c \in X$.*

Proposition 42. *It is provable in RST_{HF}^{FOL} that a non-degenerate interval is separable. If it is also bounded above then it has a least upper bound.*

Proposition 43. *Let $X \subseteq \mathbb{R}$ be a $\succ$-class. It is provable in RST_{HF}^{FOL} that X is connected (i.e. cannot be disconnected by two open $\succ$-classes) iff it is an interval.*

4.3 Real Functions

Definition 44. *Let X be a $\succ$-class. A* $\succ$-sequence *in X is a $\succ$-function on $\mathbb{N}$ whose image is contained in X.*

Lemma 45. *It is provable in RST_{HF}^{FOL} that Cauchy $\succ$-sequences in $\mathbb{R}$ converge to limits in $\mathbb{R}$. The induced map (lim) is available in RST_{HF}^{FOL} as a $\succ$-function.*

Proposition 46. *It is provable in RST_{HF}^{FOL} that if $X \subseteq \mathbb{R}$ is closed, then every Cauchy $\succ$-sequence in X converges to a limit in X.*

Next we want to study sequences of functions, but Definition 44 cannot be applied as is, since $\succ$-functions which are proper $\succ$-classes cannot be values of a $\succ$-function (in particular, of a $\succ$-sequence). Instead, we use the standard Uncurrying procedure.

Definition 47. *For X, Y $\succ$-classes, a* $\succ$-sequence of $\succ$-functions *on X whose image is contained in Y is a $\succ$-function on $\mathbb{N} \times X$ with image contained in Y.*

Proposition 48. *Any point-wise limit of a $\succ$-sequence of $\succ$-functions on a $\succ$-class $X \subseteq \mathbb{R}$ whose image is contained in $\mathbb{R}$ is available in RST_{HF}^{FOL} as a $\succ$-function.*

Next we turn to continuous real $\succ$-functions. One possibility of doing so, adopted e.g., in [32,34], is to introduce codes for continuous real $\succ$-functions (similar to the use of codes for open $\succ$-classes). This is of course possible as such $\succ$-functions are determined by their values on the $\succ$-set $\mathbb{Q}$. However, we prefer to present here another approach, which allows for almost direct translations of proofs in standard analysis textbook into our system. This is done using free function variables. Accordingly, the theorems which follow are schemes. Implicitly, the previous sections of this paper can also be read and understood as done in this manner. Therefore, in what follows we freely use results from them.

Definition 49. *Let $X \subseteq \mathbb{R}$ be a $\succ$-class and let F be a $\succ$-function on X whose image is contained in $\mathbb{R}$. F is called a* continuous real *$\succ$-function if:*

$$\forall a \in X \forall \varepsilon \in \mathbb{R}^{+} \exists \delta \in \mathbb{R}^{+} \forall x \in X. \left|x-a\right| < \delta \rightarrow \left|F\left(x\right)-F\left(a\right)\right| < \varepsilon$$

Proposition 50. *Let $X \subseteq \mathbb{R}$ be a $\succ$-class and F be a $\succ$-function on X whose image is contained in $\mathbb{R}$. It is provable in RST_{HF}^{FOL} that if for every open $\succ$-class $B \subseteq \mathbb{R}$, there is an open $\succ$-class A s.t. $F^{-1}\left[B\right] = A \cap X$, then F is continuous.*

Lemma 51. *The following are provable in RST_{HF}^{FOL}:*

1. *The composition, sum and product of two continuous real $\succ$-functions is a continuous real $\succ$-function.*
2. *The uniform limit of a $\succ$-sequence of continuous real $\succ$-functions is a continuous real $\succ$-function.*

Theorem 52 (Intermediate Value Theorem). *Let F be a continuous real $\succ$-function on an interval $[a, b]$ with $F\left(a\right) < F\left(b\right)$. It is provable in RST_{HF}^{FOL} that for any $d \in \mathbb{R}$ s.t. $F\left(a\right) < d < F\left(b\right)$, there is $c \in [a, b]$ s.t. $F\left(c\right) = d$.*

Theorem 53 (Extreme Value Theorem). *Let F be a continuous real $\succ$-function on a non-degenerate interval $[a, b]$. It is provable in RST_{HF}^{FOL} that F attains its maximum and minimum.*

The next step is to introduce in RST_{HF}^{FOL} the concepts of differentiation, integration, power series, etc., and develop their theories. It should now be clear that there is no difficulty in doing so. Since a thorough exposition obviously could not fit in one paper we omit it here, but use some relevant facts in what follows.

We now show that all elementary functions that are relevant to J_2 are available in RST_{HF}^{FOL}. Even though for every real number y in J_2, $\lambda x \in \mathbb{R}.y$ is available in RST_{HF}^{FOL} as a $\succ$-function, not all constant functions on the "real" real line are available in J_2. The reason is that $\lambda x \in \mathbb{R}.y$ does not exists in J_2 for *every* "real" number y (simply since not every "real" real number is available in RST_{HF}^{FOL}).

Definition 54. *The collection of J_2-elementary functions is defined like the standard elementary functions (see, e.g., [28]), replacing the constant functions by J_2-constant functions, which are $\lambda x \in \mathbb{R}.c$ where c is a real in J_2.*

Proposition 55. *Let F be a continuous, strictly monotone real $\succ$-function on a real interval. Then it is provable in RST_{HF}^{FOL} that the inverse function F^{-1} is available in RST_{HF}^{FOL} as a $\succ$-function, and its continuity is provable in RST_{HF}^{FOL}.*

Proposition 56. *All J_2-elementary functions are available in RST_{HF}^{FOL}. Also, any piece-wise defined function with finitely many pieces such that its restriction to any of the pieces is a J_2-elementary function, is available in RST_{HF}^{FOL}.*

5 Conclusion and Further Research

In this paper we showed that a minimal computational framework is sufficient for the development of applicable mathematics. Of course, a major future research task is to implement and test the framework. A critical component of such implementation will be to scale the cost of checking the safety relation. We then plan to use the implemented framework to formalize even larger portions of mathematics, including first of all more analysis, but also topology and algebra.

Another important task is to fully exploit the computational power of our computational theories. This includes finding a good notion of canonical terms, and investigating various reduction properties such as strong normalization. We intend to try also to profit from this computational power in other ways, e.g., by using it for proofs by reflection as supported by well-known proof assistant like Coq [10], Nuprl [12] and Isabelle/HOL [27].

An intuitionistic variant of the system RST_C^{FOL}, RST_C^{iFOL}, can be also considered. It is based on intuitionistic first-order logic (which underlies constructive counterparts of ZF, like CZF [1] and IZF [7]), and is obtained by adding to RST_C^{FOL} the axiom of Restricted Excluded Middle: $\varphi \vee \neg\varphi$, where $\varphi \succ \emptyset$. This axiom is computationally acceptable since it simply asserts the definiteness of absolute formulas. The computational theory RST_{HF}^{iFOL} should allow for a similar formalization of constructive analysis (e.g., [26]).

Further exploration of the connection between our framework and other related works is also required. This includes works on: computational set theory [1,7,9,19,26], operational set theory [18,21], and rudimentary set theory [6,24].

Another direction for further research is to consider larger computational structures. This includes J_ω or even J_{ω^ω} (which is the minimal model of the minimal computational theory based on ancestral logic [4,11]). On the one hand, in such universes standard mathematical structures can be treated as sets. On the other hand, they are more comprehensive and less concrete, thus include more objects which may make computations harder.

Acknowledgements. The second author is supported by: Fulbright Post-doctoral Scholar program; Weizmann Institute of Science – National Postdoctoral Award program for Advancing Women in Science; Eric and Wendy Schmidt Postdoctoral Award program for Women in Mathematical and Computing Sciences.

References

1. Aczel, P., Rathjen, M.: Notes on constructive set theory. Technical report 40, Mittag-Leffler (2001)
2. Avigad, J., Harrison, J.: Formally verified mathematics. Commun. ACM **57**(4), 66–75 (2014)
3. Avron, A.: A framework for formalizing set theories based on the use of static set terms. In: Avron, A., Dershowitz, N., Rabinovich, A. (eds.) Pillars of Computer Science. LNCS, vol. 4800, pp. 87–106. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78127-1_6
4. Avron, A.: A new approach to predicative set theory. In: Schindler, R. (ed.) Ways of Proof Theory, Onto Series in Mathematical Logic, pp. 31–63. Verlag (2010)
5. Avron, A., Cohen, L.: Formalizing scientifically applicable mathematics in a definitional framework. J. Formalized Reasoning **9**(1), 53–70 (2016)
6. Beckmann, A., Buss, S.R., Friedman, S.D.: Safe recursive set functions. J. Symbolic Logic **80**(3), 730–762 (2015)
7. Beeson, M.J.: Foundations of Constructive Mathematics: Metamathematical Studies, vol. 6. Springer Science & Business Media, Boston (2012)
8. Autexier, S., Campbell, J., Rubio, J., Sorge, V., Suzuki, M., Wiedijk, F. (eds.): CICM 2008. LNCS, vol. 5144. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85110-3
9. Cantone, D., Omodeo, E., Policriti, A.: Set Theory for Computing: From Decision Procedures to Declarative Programming with Sets. Springer, New York (2001)
10. Chlipala, A.: Certified Programming with Dependent Types. MIT Press, Cambridge (2013)
11. Cohen, L., Avron, A.: The middle ground-ancestral logic. Synthese, 1–23 (2015)
12. Constable, R.L., Allen, S.F., Bromley, M., Cleaveland, R., et al.: Implementing Mathematics with the Nuprl Proof Development System. Prentice Hall, Englewood Cliffs (1986)
13. Corcoran, J., Hatcher, W., Herring, J.: Variable binding term operators. Math. Logic Q. **18**(12), 177–182 (1972)
14. Devlin, K.: Constructibility. Perspectives in Mathematical Logic. Springer, Heidelberg (1984)
15. Feferman, S.: Systems of predicative analysis. J. Symbolic Logic **29**(01), 1–30 (1964)
16. Feferman, S.: Systems of predicative analysis, II: representations of ordinals. J. Symbolic Logic **33**(02), 193–220 (1968)
17. Feferman, S.: Why a little bit goes a long way: logical foundations of scientifically applicable mathematics. In: PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association, pp. 442–455. JSTOR (1992)

18. Feferman, S.: Operational set theory and small large cardinals. Inf. Comput. **207**(10), 971–979 (2009)
19. Friedman, H.: Set theoretic foundations for constructive analysis. Ann. Math. **105**(1), 1–28 (1977)
20. Gandy, R.O.: Set-theoretic functions for elementary syntax. In: Proceedings of Symposia in Pure Mathematics, vol. 13, pp. 103–126 (1974)
21. Jäger, G., Zumbrunnen, R.: Explicit mathematics and operational set theory: some ontological comparisons. Bull. Symbolic Logic **20**(3), 275–292 (2014)
22. Jensen, R.B.: The fine structure of the constructible hierarchy. Ann. Math. Logic **4**(3), 229–308 (1972)
23. Kamareddine, F.D.: Thirty Five Years of Automating Mathematics, vol. 28. Springer, Netherlands (2003)
24. Mathias, A.R.D., Bowler, N.J., et al.: Rudimentary recursion, gentle functions and provident sets. Notre Dame J. Formal Logic **56**(1), 3–60 (2015)
25. Megill, N.: Metamath: A Computer Language for Pure Mathematics. Elsevier Science, Amsterdam (1997)
26. Myhill, J.: Constructive set theory. J. Symbolic Logic **40**(03), 347–382 (1975)
27. Nipkow, T., Wenzel, M., Paulson, L.C. (eds.): Isabelle/HOL: A Proof Assistant for Higher-Order Logic. LNCS, vol. 2283. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45949-9
28. Risch, R.H.: Algebraic properties of the elementary functions of analysis. Am. J. Math. **101**(4), 743–759 (1979)
29. Rudnicki, P.: An overview of the MIZAR project. In: Proceedings of the 1992 Workshop on Types for Proofs and Programs, pp. 311–330 (1992)
30. Schwartz, J.T., Dewar, R.B., Schonberg, E., Dubinsky, E.: Programming with Sets: An Introduction to SETL. Springer-Verlag New York Inc., New York (1986). https://doi.org/10.1007/978-1-4613-9575-1
31. Shapiro, S.: Foundations Without Foundationalism: A Case for Second-Order Logic. Oxford University Press, Oxford (1991)
32. Simpson, S.G.: Subsystems of Second Order Arithmetic, vol. 1. Cambridge University Press, Cambridge (2009)
33. Weaver, N.: Analysis in J_2. arXiv preprint arXiv:math/0509245 (2005)
34. Weyl, H.: Das Kontinuum: Kritische Untersuchungen über die Grundlagen der Analysis. W. de Gruyter (1932)

A Sequent-Calculus Based Formulation of the Extended First Epsilon Theorem

Matthias Baaz[1], Alexander Leitsch[2], and Anela Lolic[1](✉)

[1] Institute of Discrete Mathematics and Geometry 104, TU Wien, Vienna, Austria
{baaz,anela}@logic.at
[2] Institute of Computer Languages (E185), TU Wien, Vienna, Austria
leitsch@logic.at

Abstract. The optimal calculation of Herbrand disjunctions from unformalized or formalized mathematical proofs is one of the most prominent problems of computational proof theory. The up-to-date most direct approach to calculate Herbrand disjunctions is based on Hilbert's epsilon formalism (which is in fact also the oldest framework for proof theory). The algorithm to calculate Herbrand disjunctions is an integral part of the proof of the extended first epsilon theorem. This paper connects epsilon proofs and sequent calculus derivations with cuts. This leads to an improved notation for the epsilon formalism and a computationally improved version of the extended first epsilon theorem, which allows a nonelementary speed-up of the computation of Herbrand disjunctions.

Keywords: Extended first epsilon theorem · Herbrand disjunctions Epsilon calculus

1 Introduction

The optimal calculation of Herbrand disjunctions from unformalized or formalized mathematical proofs is one of the most prominent problems in proof theory of first-order logic. So far, the most direct approach to calculate Herbrand disjunctions is based on Hilbert's ε-formalism, cf. which is also the oldest framework for proof theory [7].

The ε-calculus uses ε-terms to represent $\exists x A(x)$ by $A(\varepsilon_x A(x))$ ($\forall x A(x)$ is consequently represented by $A(\varepsilon_x \neg A(x))$). As the ε-calculus is only based on the representation of substitutions (critical formulas $A(t) \rightarrow A(\varepsilon_x A(x))$ for $A(t) \rightarrow \exists x A(x)$) and propositional axioms and rules, the unrestricted deduction theorem of propositional calculus transfers to this formalization of first order logic: The ε-proof itself is a tautology (($\bigwedge_{i=1}^{n} A_i(t_i) \rightarrow A_i(\varepsilon_x A_i(x))) \rightarrow E$, where E is the original result translated into ε-calculus). Note that strong quantifier inferences are replaced by substitutions of $\varepsilon_x \neg A(x)$ for $\forall x A(x)$ positive and $\varepsilon_x A(x)$ for $\exists x A(x)$ negative. Valid propositional formulas do not influence an ε-proof. The extended first ε-theorem [7,9] eliminates algorithmically the critical formulas

S. Artemov and A. Nerode (Eds.): LFCS 2018, LNCS 10703, pp. 55–71, 2018.
https://doi.org/10.1007/978-3-319-72056-2_4

obtaining a Herbrand disjunction $\bigvee_{i=1}^{m} E(\overline{t_i})$, where E is the ε-translation of $\exists \overline{x} E'(\overline{x})$, E' being quantifier-free.

There are many advantages of the extended first ε-theorem w.r.t. the calculation of Herbrand disjunctions, which are not shared by more widespread approaches to obtain Herbrand disjunctions. First of all, the ε-calculus is not sensitive w.r.t. addition of (arbitrary) tautologies. This implies that bounds depend on first-order features only. More precisely, the complexity of the Herbrand disjunction does not depend on the complexity of the derivation, but only on the number of critical formulas and on the complexity of the ε-terms occurring in it. Note that the complexity of ε-terms measures just the quantifier complexity of the formulas involved, not the propositional complexity. Furthermore, the extended first ε-theorem develops the Herbrand disjunctions as contraposition of all case distinctions in the proof: the Herbrand disjunction may be understood as the only remaining case distinction. This explains why Herbrand disjunctions may be short even for high level mathematical proofs (cf. the complexity of the Herbrand disjunction in [8]). Another advantage is that a proof might be formalized disregarding propositional features: represent all substitutions in the proof; if the obtained ε-proof is not a tautology, a substitution has been overlooked.

Despite the advantages, the ε-calculus has never become popular in computational proof theory of first order logic. The main reasons are the untractability of almost all nonclassical logics by any adaptation of the ε-formalism and the clumsiness of the ε-formalism itself: consider the ε-translation of $\exists x \exists y \exists z$ $A(x,y,z)$: $A(\varepsilon_x A(x,\varepsilon_y A(x,y,\varepsilon_z A(x,y,z)), \varepsilon_z A(x,\varepsilon_y A(x,y,\varepsilon_z A(x,y,z)),z))$, $\varepsilon_y A(\varepsilon_x A(x,\varepsilon_y A(x,y,\varepsilon_z A(x,y,z)),\varepsilon_z A(x,\varepsilon_y A(x,y,\varepsilon_z A(x,y,z)),z))$, $y, \varepsilon_z A(\varepsilon_x A(x,\varepsilon_y A(x,y,\varepsilon_z A(x,y,z)),\varepsilon_z A(x,\varepsilon_y A(x,y,\varepsilon_z A(x,y,z)),z)),y,z))$, $\varepsilon_z A(\varepsilon_x A(x,\varepsilon_y A(x,y,\varepsilon_z A(x,y,z)),\varepsilon_z A(x,\varepsilon_y A(x,y,\varepsilon_z A(x,y,z)),z))$, $\varepsilon_y A(\varepsilon_x A(x,\varepsilon_y A(x,y,\varepsilon_z A(x,y,z)),\varepsilon_z A(x,\varepsilon_y A(x,y,\varepsilon_z A(x,y,z)),z))$, $y, \varepsilon_z A(\varepsilon_x A(x,\varepsilon_y A(x, y, \varepsilon_z A(x, y, z)),\varepsilon_z A(x,\varepsilon_y A(x,y,\varepsilon_z A(x,y,z)),z)),y,z)),z))$.

This paper addresses the second problem by defining a translation of LK-proofs with cuts to ε/τ-proofs[1]. This translation leads to a simplified notation and a new elimination order that eliminates the critical formulas based on the critical formulas that are present and not on internal features of the ε-terms. This leads to a non-elementary speed-up of the length of the obtained Herbrand sequents w.r.t. the length of the Herbrand sequents obtained by the original elimination order.

2 Hilbert's Extended First ε-Theorem in a Modern Setting

Hilbert's ε-calculus is formulated by representing $\exists x A(x)$ by $A(\varepsilon_x A(x))$ and $\forall x A(x)$ by $A(\varepsilon_x \neg A(x))$ (or by $A(\tau_x A(x))$ if τ-terms are used). This translation easily extends to the whole language. Critical formulas are of the form

[1] To represent the symmetry of LK we will work with critical τ-formulas, i.e. $A(\tau_x A(x)) \rightarrow A(t)$ represents $\forall x A(x) \rightarrow A(t)$. This is only a notational convenience.

$A(t) \rightarrow A(\varepsilon_x A(x))$ and $A(\tau_x A(x)) \rightarrow A(t)$, respectively. An ε/τ-matrix of an ε-term is obtained by replacing all different proper subterms by different variables. An ε/τ-proof in a modern setting is a valid quantifier-free sequent $C_1, \ldots, C_r, \Pi \vdash \Delta$, where $C_1, \ldots, C_r$ are critical formulas. ε/τ-proofs of translations of LK derivations can be easily obtained by replacing

$$\frac{\Pi \vdash \Delta, A(t)}{\Pi \vdash \Delta, \exists x A(x)}\ \exists_r \qquad \text{by} \qquad \frac{\Pi \vdash \Delta, A(t)}{\overline{A(t) \rightarrow A(\varepsilon_x A(x)), \Pi \vdash \Delta}}$$

analogously for $\forall_l$ and

$$\frac{\Pi \vdash \Delta, A(\alpha)}{\Pi \vdash \Delta, \forall x A(x)}\ \forall_r \qquad \text{by substituting } \alpha \text{ by } \tau_x A(x),$$

analogously for $\exists_l$. Otherwise the proof remains unchanged. The end-sequent is the ε/τ-proof.

The first ε-theorem states roughly speaking that the critical formulas can be eliminated from the sequent if Π and Γ do not contain ε/τ-terms. We are interested in the extended first ε-theorem, which constructs Herbrand sequents $\Pi\sigma_1, \ldots, \Pi\sigma_n \vdash \Delta\sigma_1, \ldots, \Delta\sigma_n$ from ε/τ-proofs $C_1, \ldots, C_r, \Pi \vdash \Delta$.

Lemma 1 (Hilbert's Ansatz). *If $A(t_1) \rightarrow A(\varepsilon_x A(x))$, …, $A(t_n) \rightarrow A(\varepsilon_x A(x))$, $\Pi \vdash \Delta$ is valid then $\Pi\{\varepsilon_x A(x) \rightarrow t_1\}$, …, $\Pi\{\varepsilon_x A(x) \rightarrow t_n\}$, $\Pi \vdash \Delta\{\varepsilon_x A(x) \rightarrow t_1\}$, …, $\Delta\{\varepsilon_x\ A(x) \rightarrow t_n\}$, Δ is valid ($\varepsilon_x A(x)$ can then be substituted by a fixed constant).*

Proof. (See [7].) Note that $A(t_i), \Pi\{\varepsilon_x A(x) \rightarrow t_i\} \vdash \Delta\{\varepsilon_x A(x) \rightarrow t_i\}$ and $\neg A(t_1), \ldots, \neg A(t_n), \Pi \vdash \Delta$ are valid.

For the ε-theorems the following definition is central.

Definition 1. *An ε/τ-term e is* nested *in an ε/τ-term e' if e is a proper subterm of e'. An ε/τ-term e is* subordinate *to an ε/τ-term $e' = \varepsilon_x A(x)$ (or $e' = \tau_x A(x)$) if e occurs in e' and x is free in e.*

The *rank* counts the subordination levels and the *degree* the length of the maximal inclusion chain.

Theorem 1 (Extended first ε-theorem). *Given a proof $C_1, \ldots, C_r, \Pi \vdash \Delta$ we obtain a valid sequent $\Pi\sigma_1, \ldots, \Pi\sigma_n \vdash \Delta\sigma_1, \ldots, \Delta\sigma_n$ containing no ε/τ-terms, where the σ_i are substituting ε/τ-terms by closed terms.*

Proof (Sketch). c.f. [7]. Hilbert's Ansatz is repeatedly applied to ε/τ-terms of maximal rank and maximal degree and to the remaining critical formulas to obtain an expansion both of the other critical formulas and of the rest of the sequent. The condition of maximal rank is necessary to guarantee that critical formulas are transformed into critical formulas by these substitutions. The maximal degree is necessary for termination.

The proof in [7] implicitly contains the following algorithm:

Definition 2

```
begin % algorithm AlgOld
1. let $F = C_1, \ldots, C_n, \Pi \vdash \Delta$ be an $\varepsilon/\tau$-proof and define $S_0 = C_1, \ldots, C_n$,
   $F'_0 = \Pi, F''_0 = \Delta$;
2. check whether $F'_k \vdash F''_k$ is a tautology, if yes substitute all remainung
   $\varepsilon/\tau$-terms by the fixed constant $c$ and terminate AlgOld;
3. delete $A \to A$ from $S_k$ and let $S'_k$ be the result;
4. choose an $\varepsilon/\tau$-term $e$ of maximal rank and maximal degree;
5. delete the critical formulas $S(e)$ belonging to $e$. Let $\sigma_1 \ldots \sigma_i$ be the
   corresponding substitutions. $S_{k+1} = (S'_k - S(e))\sigma_1\lambda \ldots (S'_k - S(e))\sigma_i\lambda$,
   $F'_{k+1} = F'_k\sigma_1\lambda \ldots F'_k\sigma_i\lambda, F'_k\lambda,\ F''_{k+1} = F''_k\sigma_1\lambda \ldots F''_k\sigma_i\lambda, F''_k\lambda$,
   where $\lambda$ is $e \leftarrow c$ for the fixed constant $c$. Repeat this operation until there
   are no critical formulas belonging to $e$. Proceed with 2.;
end.
```

3 A Function Variable Variant of ε/τ-Proofs

We define a new abstract format for ε/τ-proofs, where ε/τ-terms are replaced by critical terms fulfilling a minimal set of conditions. This format will make it easier to make use of structural properties of given LK derivations. Definition 7 will provide an inductive translation of LK proofs into function variable proofs. This translation is based on the identification of critical function symbols at some nodes of the proof: therefore our abstract notion of ε/τ-proofs is based on function variables.

Definition 3 (critical formulas). *Let $s_1, \ldots, s_n$ be arbitrary terms and f_Q for $Q \in \{\forall, \exists\}$ be n-ary function variables. Then formulas of the form $A(t) \to A(f_Q(s_1, \ldots, s_n))$ and $A(f_Q(s_1, \ldots, s_n)) \to A(t)$ are called* critical formulas. *f_Q is a* critical function variable *and $f_Q(s_1, \ldots, s_n)$ is a* critical function term. *$A(f_Q(s_1, \ldots, s_n))$ is the* head *of the critical formula.*

Definition 4 (function variable proof). *A* function variable proof *F is a quantifier-free sequent of the form $C_1, \ldots, C_n, \Gamma \vdash \Delta$ s.t.*

1. *$C_1, \ldots, C_n, \Gamma \vdash \Delta$ is valid,*
2. *$C_1, \ldots, C_n$ are critical formulas,*
3. *the set $\mathbf{\Delta}(F)$ of all critical function variables can be partially ordered by an irreflexive, transitive relation $<$ s.t. whenever a critical function term of the form $g_Q(\ldots x \ldots)\{x \leftarrow f_{Q'}(\ldots)\}$ occurs in F then $g_Q < f_{Q'}$,*
4. *let σ be a substitution of critical function terms by terms s.t. $f_Q(u)\sigma = f_Q(v)\sigma$, where $Q \in \{\forall, \exists\}$, for critical function terms $f_Q(u), f_Q(v)$ associated with $A(x), B(x)$, then $A(f_Q(u))\sigma = B(f_Q(v))\sigma$.*

Note that condition 4 of Definition 4 replaces the fact that traditional ε/τ-terms contain all subterms of the heads of critical formulas.

Example 1. 1. $F = A(t) \rightarrow A(f_Q(t)), (A(t) \rightarrow A(f_Q(t))) \rightarrow (A(e_{Q'}) \rightarrow A(f_Q(\, e_{Q'}))) \vdash A(e_{Q'}) \rightarrow A(f_Q(e_{Q'}))$ with critical terms $f_Q(t)$, $e_{Q'}$, where $\mathbf{\Delta}(F) = \{f_Q, e_{Q'}\}$, with $f_Q < e_{Q'}$, is a function variable proof (in fact, a proof of the *drinker's principle*).

2. $E = (A(s, f_Q(s)) \rightarrow A(s, f_Q(g_{Q'}(s)))), (B(u, g_{Q'}(u)) \rightarrow B(u, g_{Q'}(f_Q(u)))) \vdash (A(s, f_Q(s)) \rightarrow A(s, f_Q(g_{Q'}(s)))) \wedge (B(u, g_{Q'}(u)) \rightarrow B(u, g_{Q'}(f_Q(u))))$ with critical terms $f_Q(s)$, $g_{Q'}(s)$, $g_{Q'}(u)$, where $\mathbf{\Delta}(E) = \{f_Q, g_{Q'}\}$, fulfils $1, 2, 4$ of Definition 4 but cannot fulfil 3.
3. $G = A(c) \rightarrow A(e_Q), \neg A(d) \rightarrow \neg A(e_Q) \vdash A(c) \rightarrow A(d)$ with critical term e_Q, where $\mathbf{\Delta}(G) = \{e_Q\}$ fulfills conditions $1, 2, 3$ but not condition 4.

Proposition 1. *It can be effectively checked if sequents fulfilling condition* 1 *and* 2 *of Definition 4 are function variable proofs.*

Proof. First determine the critical terms and from the critical terms the critical function variables (their outermost function symbol). Check whether the transitive closure of the relation in the sense of condition 3 of Definition 4 determines a partial order, i.e. is loop-free. To check condition 4 of Definition 4 consider all pairs of critical function terms with identical outermost function variable. Replace all critical function terms properly within the selected pair with different variables and unify. If the unification is successful apply the most general unifier in the same way to the corresponding heads of critical formulas, they have to be identical.

Example 2. Let $E = A(c_{Q'}, s) \rightarrow A(c_{Q'}, f_Q(c_{Q'})), A(t, s) \rightarrow A(t, f_Q(t)), B(u) \rightarrow B(c_{Q'}) \vdash C(c_{Q'}, f_Q(c_{Q'}))$ with critical terms $f_Q(c_{Q'})$, $f_Q(t)$, $c_{Q'}$, where $\mathbf{\Delta}(E) = \{c_{Q'}, f_Q\}$, with $f_Q < c_{Q'}$. Condition 4 of Definition 4 on E can be checked in the following way: We check the unifiability of $f_Q(x)$ and $f_Q(t)$, the most general unifier is $\sigma = \{x \leftarrow t\}$. Then $A(x, f_Q(x))\sigma = A(t, f_Q(t))\sigma$.

Proposition 2. *ε/τ-proofs can be considered as instances of function variable proofs.*

Proof. Replace the variables defined by the matrices of occurring ε-terms by function variables. Condition 3 of Definition 4 is fulfilled as the order of function variables reflects the overbinding of the matrices. Condition 4 is fulfilled as critical ε-terms and heads are in a $1 - 1$ relation.

The soundness of function variable proofs will be shown as corollary to the extended ε-theorem for the function variable format.

4 A Sequent-Based Translation into the Function Variable Proof Format

As noted in Sect. 1 the translation into ε-calculus is in general problematic and leads to very long and complicated formulas. To obtain a better readability of ε-expressions and making use of given LK-proofs, we propose a new nonextensional translation from formulas and proofs into a function variable format.

Definition 5 (function variable translation). *We define the* function variable translation $\mathrm{FT}(A)$*, the corresponding set of function variables* $\mathbf{\Delta}(A)$ *and the set of relations* $\mathbf{\Gamma}(A)$ *of a formula* A *inductively as follows:*

1. A *is an atom, then* $\mathrm{FT}(A) = A$ *and* $\mathbf{\Delta}(A) = \emptyset$, $\mathbf{\Gamma}(A) = \emptyset$,
2. $A = \neg B$, *then* $\mathrm{FT}(A) = \neg\mathrm{FT}(B)$ *and* $\mathbf{\Delta}(A) = \mathbf{\Delta}(B)$, $\mathbf{\Gamma}(A) = \mathbf{\Gamma}(B)$,
3. $A = B \circ C$, *where* $\circ \in \{\wedge, \vee, \rightarrow\}$, *then* $\mathrm{FT}(A) = \mathrm{FT}(B) \circ \mathrm{FT}(C)$ *and* $\mathbf{\Delta}(A) = \mathbf{\Delta}(B) \cup \mathbf{\Delta}(C)$, $\mathbf{\Gamma}(A) = \mathbf{\Gamma}(B) \cup \mathbf{\Gamma}(C)$,
4. $A = QxF(x)$ *where* $Q \in \{\exists, \forall\}$, *then* $\mathrm{FT}(A) = \mathrm{FT}(F(x))\{x \leftarrow f_Q(y_1, \ldots, y_n)\}$, *where* f_Q *is a new function variable,* $y_1, \ldots, y_n$ *are bound variables s.t. their corresponding quantifiers are in the scope of* Qx *and* $\mathbf{\Delta}(A) = \mathbf{\Delta}(F(x)) \cup \{f_Q\}$. *For all* $g \in \mathbf{\Delta}(F(x))$, *whenever* x *occurs in the form* $g(\ldots x \ldots)$ *in* $\mathrm{FT}(F(x))$ *we define* $\mathbf{\Gamma}(A)$ *to be the transitive closure of* $\mathbf{\Gamma}(F(x)) \cup \{g < f_Q\}$, *where* $<$ *is a partial order on function symbols.*

Remark 1. Note that $\mathrm{FT}(B)$ and $\mathbf{\Delta}(B)$ will differ for two different occurrences of a subformula B of A and the difference between functions in Δ and Skolem functions: functions in Δ depend on all quantifiers, Skolem functions only on strong quantifiers.

Note that this definition can be easily extended to sequents. Indeed, any sequent $S = A_1, \ldots, A_m \vdash B_1, \ldots, B_m$ is equivalent to $S' = \vdash (A_1 \wedge \ldots \wedge A_m) \rightarrow (B_1 \vee \ldots \vee B_n)$, containing only one formula. $(A_1 \wedge \ldots \wedge A_m) \rightarrow (B_1 \vee \ldots \vee B_n)$ can then be transformed into its function variable translation as usual.

Example 3. Let $E = \forall x(P(x) \rightarrow \exists y \forall z Q(x, y, z))$. Then $\mathrm{FT}(E) = P(c) \rightarrow Q(c, g(c), f(c, g(c)))$, $\mathbf{\Delta}(E) = \{c, g, f\}$ and $\mathbf{\Gamma}(E) = \{f < g, f < c, g < c\}$.

In the cases contraction and cut in Definition 7 we will deal with unification of terms containing function variables, where only function variables are substituted for function variables of the same arity. We call such a unification *restricted function unification.*

Definition 6 (restricted function unification). *Let* s *and* t *be terms containing function variables. A* restricted function unifier *is defined by a substitution* σ *s.t.* $s\sigma = t\sigma$, *where exclusively function variables are substituted for function variables of the same arity.* σ *is a most general restricted function unifier* $mf(s,t)$ *iff any other restricted function unifier* σ' *of* s *and* t *is an extension, i.e. there is a substitution* σ'' *s.t.* $\sigma' = mf(s,t)\sigma''$.

Proposition 3. *It is decidable whether there exists a restricted function unifier and in case there is one there is a most general one.*

Proof. Parse s and t from left to the right. At the first place of difference check whether there are two function variables of the same arity and substitute one by the other, otherwise there is no restricted function unifier. Repeat the procedure until the terms are identical or the output is non-unifiable.

Proofs in *sequent calculus* can also be transformed into the function variable format. The translation follows the height of the proof. We define the translation for proofs with atomic axioms, arbitrary cuts and end-sequents containing weak quantifiers only (note that strong quantifiers in the end-sequents can be replaced by Skolem functions without increasing the complexity of the proof [6] and that from a cut-free proof Skolem functions can be eliminated at exponential expense [4]). Technically we define an extension LK* of LK with the additional rules

$$\frac{\Gamma \vdash \Delta, A(t)}{\Gamma \vdash \Delta, A(s)}\, fv_r \qquad\qquad \frac{A(t), \Gamma \vdash \Delta}{A(t), \Gamma \vdash \Delta}\, fv_l$$

where s is a function term and add the formulas $A(t) \to A(s)$ for fv_r and $A(s) \to A(t)$ for fv_l. In the end of the transformation these formulas will be the critical formulas corresponding to a proof in LK and guarantee the soundness of LK*.

Every sequent occurrence in a proof φ is labelled by a label (node) ν and $\varphi.\nu$ is the subproof of φ ending in ν.

To make use of contraction more explicitly, we use a multiplicative version of LK in the following definition:

Definition 7 (minimal translation of an LK proof). *We define a proof transformation T transforming every LK proof φ into an* LK* *proof $T(\varphi)$ and simultaneously generating the set of critical formulas. We construct $T(\varphi)$ containing the critical formulas ($\mathcal{S}_\exists$ and $\mathcal{S}_\forall$), $\mathbf{\Delta}(\varphi)$ (the set of function variables) and $\mathbf{\Gamma}(\varphi)$ (a partial order on $\mathbf{\Delta}(\varphi)$) inductively over the nodes ν in φ.*

$\varphi.\nu$ is an axiom $A \vdash A$. Here we define $T(\varphi.\nu) = A \vdash A$ and $\mathcal{S}_\exists(\nu) = \mathcal{S}_\forall(\nu) = \mathbf{\Delta}(\nu) = \mathbf{\Gamma}(\nu) = \emptyset$.

Let ξ be a unary rule different from quantifier rules, contraction and weakening, μ the node of the premise and let $\varphi.\nu =$

$$\frac{\begin{matrix}(\psi)\\ \mu\colon \Gamma \vdash \Delta\end{matrix}}{\nu\colon \Gamma_1 \vdash \Delta_1}\ \xi$$

Let $T(\varphi.\mu)$ be a proof ψ^* of the sequent $\Gamma^* \vdash \Delta^*$. Then we define $T(\varphi.\nu)$ as

$$\frac{\begin{matrix}(\psi^*)\\ \Gamma^* \vdash \Delta^*\end{matrix}}{\Gamma_1^* \vdash \Delta_1^*}\ \xi$$

and $\mathcal{S}_\exists(\nu) = \mathcal{S}_\exists(\mu), \mathcal{S}_\forall(\nu) = \mathcal{S}_\forall(\mu)$, $\mathbf{\Delta}(\nu) = \mathbf{\Delta}(\mu)$, $\mathbf{\Gamma}(\nu) = \mathbf{\Gamma}(\mu)$.

Note that the auxiliary formulas in $\Gamma^* \vdash \Delta^*$ and principal formula in $\Gamma_1^* \vdash \Delta_1^*$ correspond to those in $\Gamma \vdash \Delta$ and $\Gamma_1 \vdash \Delta_1$, respectively.

Let $\varphi.\nu =$

$$\frac{\begin{matrix}(\psi_1)\\ \mu_1 : \Gamma \vdash \Delta\end{matrix} \quad \begin{matrix}(\psi_2)\\ \mu_2 : \Pi \vdash \Lambda\end{matrix}}{\nu\colon \Gamma_1, \Pi_1 \vdash \Delta_1, \Lambda_1}\ \xi$$

where ξ is a binary rule different from cut. Let us assume that $T(\varphi.\mu_1) = \psi_1^*$ with end-sequent $\Gamma^* \vdash \Delta^*$, and $T(\varphi.\mu_2) = \psi_2^*$ with end sequent $\Pi^* \vdash \Lambda^*$. Then we define $T(\varphi.\nu)$ as

$$\frac{\begin{matrix}(\psi_1^*) & (\psi_2^*)\\ \Gamma^* \vdash \Delta^* & \Pi^* \vdash \Lambda^*\end{matrix}}{\Gamma_1^*, \Pi_1^* \vdash \Delta_1^*, \Lambda_1^*}\ \xi$$

and $\mathcal{S}_\exists(\nu) = \mathcal{S}_\exists(\mu_1) \cup \mathcal{S}_\exists(\mu_2)$, $\mathcal{S}_\forall(\nu) = \mathcal{S}_\forall(\mu_1) \cup \mathcal{S}_\forall(\mu_2)$ and $\mathbf{\Delta}(\nu) = \mathbf{\Delta}(\mu_1) \cup \mathbf{\Delta}(\mu_2)$, $\mathbf{\Gamma}(\nu)$ is the transitive closure of $\mathbf{\Gamma}(\mu_1) \cup \mathbf{\Gamma}(\mu_2)$.
Let $\varphi.\nu$ be

$$\frac{\begin{matrix}(\psi)\\ \mu\colon \Gamma \vdash \Delta, A(t)\end{matrix}}{\nu\colon \Gamma \vdash \Delta, \exists x.A(x)}\ \exists_r$$

Assume that $T(\varphi.u) = \psi^*$ where ψ^* is a proof with end-sequent $\Gamma^* \vdash \Delta^*, A^*(t^*)$. Let $t_1, \ldots, t_n$ be the terms eliminated by quantifiers overbinding $\exists x$ in φ (in a derivation going to the end-sequent or to a cut). Let $f_\exists$ be a new function variable of arity n (not occurring in $\mathbf{\Delta}(\mu)$) and

$$\theta = \{x \leftarrow f_\exists(y_1, \ldots, y_n)\}\{y_1 \leftarrow t_1, \ldots, y_n \leftarrow t_n\}.$$

Then we define $T(\varphi.\nu)$ as

$$\frac{\begin{matrix}(\psi^*)\\ \Gamma^* \vdash \Delta^*, A^*(t^*)\end{matrix}}{\Gamma^* \vdash \Delta^*, A^*(x)\theta}\ fv_r$$

$\mathcal{S}_\exists(\nu) = \mathcal{S}_\exists(\mu) \cup \{A^*(t^*) \to A^*(x)\theta\}, \mathcal{S}_\forall(\nu) = \mathcal{S}_\forall(\mu)$, $\mathbf{\Delta}(\nu) = \mathbf{\Delta}(\mu) \cup \{f_\exists\}$ and if there are $g_1, \ldots, g_k$ in $\Delta(\mu)$ s.t. $A^*(t^*)$ is of the form $A^{**}(g_i(\ldots t^* \ldots))$ for $1 \leq i \leq k$, then $\mathbf{\Gamma}(\nu)$ is the transitive closure of $\mathbf{\Gamma}(\mu) \cup \{g_1 < f_\exists\} \cup \ldots \cup \{g_k < f_\exists\}$.

The last inference in $\varphi.\nu$ is $\forall_l$ inferring $\forall x.A(x)$: analogous to the case $\exists_r$ above. The last rule in $T(\varphi.\nu)$ is fv_l, the new variable is $f_\forall$,

$$\theta = \{x \leftarrow f_\forall(y_1, \ldots, y_n)\}\{y_1 \leftarrow t_1, \ldots y_n \leftarrow t_n\},$$

and $\mathcal{S}_\forall(\nu) = \mathcal{S}_\forall(\mu) \cup \{A^*(x)\theta \to A^*(t^*)\}$. $\mathbf{\Delta}(\nu) = \mathbf{\Delta}(\mu) \cup \{f_\forall\}$ and if there are $g_1, \ldots, g_k$ in $\Delta(\mu)$ s.t. $A^*(t^*)$ is of the form $A^{**}(g_i(\ldots t^* \ldots))$ for $1 \leq i \leq k$, then $\mathbf{\Gamma}(\nu)$ is the transitive closure of $\mathbf{\Gamma}(\mu) \cup \{g_1 < f_\exists\} \cup \ldots \cup \{g_k < f_\exists\}$.
Let $\varphi.\nu =$

$$\frac{\begin{matrix}(\psi)\\ \mu\colon \Gamma \vdash \Delta, A(\alpha)\end{matrix}}{\nu\colon \Gamma \vdash \Delta, \forall x.A(x)}\ \forall_r$$

Let $T(\varphi.\mu) = \psi^*$ where ψ^* is a proof of $\Gamma^* \vdash \Delta^*, A^*(\alpha)$. Let $t_1, \ldots, t_n$ be the terms eliminated by quantifiers overbinding $\forall x$ in φ (in a derivation going to the end-sequent or to a cut). Let $f_\forall$ be a new function variable of arity n (not occurring in $\mathbf{\Delta}(\mu)$) and

$$\theta = \{\alpha \leftarrow f_\forall(y_1, \ldots, y_n)\}\{y_1 \leftarrow t_1, \ldots, y_n \leftarrow t_n\}.$$

Then we define $T(\varphi.\nu) = \psi^*\theta$, $\mathcal{S}_\exists(\nu) = \mathcal{S}_\exists(\mu)\theta, \mathcal{S}_\forall(\nu) = \mathcal{S}_\forall(\mu)\theta$. $\mathbf{\Delta}(\nu) = \mathbf{\Delta}(\mu) \cup \{f_\forall\}$ *and if there are* $g_1, \ldots, g_k$ *in* $\Delta(\mu)$ *s.t.* $A^*(\alpha)$ *is of the form* $A^{**}(g_i(\ldots\alpha\ldots))$ *for* $1 \leq i \leq k$, *then* $\mathbf{\Gamma}(\nu)$ *is the transitive closure of* $\mathbf{\Gamma}(\mu) \cup \{g_1 < f_\exists\} \cup \ldots \cup \{g_k < f_\exists\}$.

The case of $\exists_l$ *is analogous to* $\forall_r$.

Contraction: let $\varphi.\nu$ *be of the form*

$$\cfrac{\begin{matrix}(\psi)\\ \mu\colon \Gamma \vdash \Delta, A, A\end{matrix}}{\nu\colon \Gamma \vdash \Delta, A}\ c_r$$

Assume that $T(\varphi.\mu) = \psi^*$ *where* ψ^* *is a proof of* $\Gamma^* \vdash \Delta^*, A_1^*, A_2^*$. *Note that* A_1^* *and* A_2^* *differ only in the occurrence of function variables, i.e. there exists a formula* A_0 *and function variables* $f_1, \ldots, f_n, g_1, \ldots, g_n$ *s.t.*

$$A_1^* = A_0(f_1, \ldots, f_n),\ A_2^* = A_0(g_1, \ldots, g_n).$$

We use a most general restricted function unifier of A_1^* *and* A_2^*: $\Theta = mf(A_1^*, A_2^*)$. *We define* $T(\varphi.\nu)$ *as*

$$\cfrac{\begin{matrix}(\psi^*\Theta)\\ \Gamma^*\Theta \vdash \Delta^*\Theta, A_1^*\Theta, A_2^*\Theta\end{matrix}}{\Gamma^*\Theta \vdash \Delta^*\Theta, A_1^*\Theta}\ c_r$$

and $\mathcal{S}_\exists(\nu) = \mathcal{S}_\exists(\mu)\Theta$, $\mathcal{S}_\forall(\nu) = \mathcal{S}_\forall(\mu)\Theta$. $\mathbf{\Delta}(\nu) = \mathbf{\Delta}(\mu)\Theta$ *and* $\mathbf{\Gamma}(\nu)$ *is the transitive closure of* $\mathbf{\Gamma}(\mu)\Theta$.

The case of c_l *is analogous.*

The case of cut: Let $\varphi.\nu =$

$$\cfrac{\begin{matrix}(\psi_1)\\ \mu_1\colon \Gamma \vdash \Delta, A\end{matrix} \quad \begin{matrix}(\psi_2)\\ \mu_2\colon A, \Pi \vdash \Lambda\end{matrix}}{\nu\colon \Gamma, \Pi \vdash \Delta, \Lambda}\ cut$$

assume that $T(\varphi.\mu_1) = \psi_1^*$ *where* ψ_1^* *is a proof of* $\Gamma^* \vdash \Delta^*, A_1^*$ *and* $T(\varphi.\mu_2) = \psi_2^*$ *where* ψ_2^* *is a proof of* $A_2^*, \Pi^* \vdash \Lambda^*$. *Like in the case of contraction there exists a formula* A_0 *and function variables* $f_1, \ldots, f_n, g_1, \ldots, g_n$ *s.t.*

$$A_1^* = A_0(f_1, \ldots, f_n),\ A_2^* = A_0(g_1, \ldots, g_n).$$

We define $\Theta = mf(A_1^*, A_2^*)$. *Then* $T(\varphi.\nu) =$

$$\cfrac{\begin{matrix}(\psi_1^*\Theta)\\ \Gamma^*\Theta \vdash \Delta^*\Theta, A_1^*\Theta\end{matrix} \quad \begin{matrix}(\psi_2^*\Theta)\\ A_2^*\Theta, \Pi^*\Theta \vdash \Lambda^*\Theta\end{matrix}}{\Gamma^*\Theta, \Pi^*\Theta \vdash \Delta^*\Theta, \Lambda^*\Theta}\ cut$$

We define $\mathcal{S}_\exists(\nu) = \mathcal{S}_\exists(\mu_1) \cup \mathcal{S}_\exists(\mu_2)\Theta$, $\mathcal{S}_\forall(\nu) = \mathcal{S}_\forall(\mu_1) \cup \mathcal{S}_\forall(\mu_2)\Theta$ *and* $\mathbf{\Delta}(\nu) = (\mathbf{\Delta}(\mu_1) \cup \mathbf{\Delta}(\mu_2))\Theta$, $\mathbf{\Gamma}(\nu)$ *is the transitive closure of* $(\mathbf{\Gamma}(\mu_1) \cup \mathbf{\Gamma}(\mu_2))\Theta$.

The case of weakening: let $\varphi.\nu$ be of the form

$$\dfrac{\begin{matrix}(\psi)\\ \mu\colon \Gamma \vdash \Delta\end{matrix}}{\nu\colon \Gamma \vdash \Delta, A}\ w_r$$

Let $T(\varphi.\mu)$ be a proof ψ^* of the sequent $\Gamma^* \vdash \Delta^*$ and let A^* be the function variable translation of A. Then we define $T(\varphi.\nu)$ as

$$\dfrac{\begin{matrix}(\psi^*)\\ \Gamma^* \vdash \Delta^*\end{matrix}}{\Gamma^* \vdash \Delta^*, A^*}\ w_r$$

and $\mathcal{S}_\exists(\nu) = \mathcal{S}_\exists(\mu), \mathcal{S}_\forall(\nu) = \mathcal{S}_\forall(\mu)$, $\mathbf{\Delta}(\nu) = \mathbf{\Delta}(\mu) \cup \mathbf{\Delta}(A)$, $\mathbf{\Gamma}(\nu)$ is the transitive closure of $\mathbf{\Gamma}(\mu) \cup \mathbf{\Gamma}(A)$.

The case of w_l is analogous.

Let $T(\varphi.\nu_0) = \Pi \vdash \Gamma$, where ν_0 is the root node of φ. The function variable proof corresponding to φ is the sequent $F(\varphi) = \mathcal{S}_\exists(\nu_0), \mathcal{S}_\forall(\nu_0), \Pi \vdash \Gamma$. $\mathbf{\Delta}(F(\varphi)) = \mathbf{\Delta}(\nu_0)$ and $\mathbf{\Gamma}(F(\varphi)) = \mathbf{\Gamma}(\nu_0)$.

Note that in all cases $\mathbf{\Gamma}(F(\varphi))$ is acyclic, because of the order of the quantifiers.

Proposition 4. *Let $F(\varphi) = \mathcal{S}_\exists(\nu_0), \mathcal{S}_\forall(\nu_0), \Pi \vdash \Gamma$ be the sequent obtained from an LK-proof φ as in Definition 7 and let $\mathbf{\Delta}(F(\varphi))$ and $\mathbf{\Gamma}(F(\varphi))$ be the corresponding sets of function variables and relations. Then $F(\varphi)$ is a function variable proof.*

Proof. Note that by construction $F(\varphi)$ is a valid sequent and $\mathcal{S}_\exists(\nu_0)$, $\mathcal{S}_\forall(\nu_0)$ are critical formulas, hence conditions 1, 2 of Definition 4 are fulfilled. The set $\mathbf{\Delta}(F(\varphi))$ contains all critical function variables and can be partially ordered by the set of relations defined in $\mathbf{\Gamma}(F(\varphi))$. Note that indeed, the transitive closure of the set of relations in $\mathbf{\Gamma}(F(\varphi))$ is irreflexive and whenever a critical function term of the form $g_Q(\ldots x \ldots)\{x \leftarrow f_{Q'}(\ldots)\}$ occurs in $F(\varphi)$ then $g_Q < f_{Q'}$ as $f_{Q'}$ occurs below g_Q in the proof-tree. Therefore, condition 3 of Definition 4 is fulfilled as well. Condition 4 holds because the critical function terms constructed in this translation contain all other critical terms in the head as subterms.

Example 4. Let π be the proof of "There are irrational numbers x and y s.t. x^y is rational". Let $u^v = exp(u, v)$ and $\sqrt{2}$ be a constant. Then $\pi =$

$$\dfrac{\begin{matrix}(\pi_1)\\ F_1, F_2 \vdash F_3, R(\sqrt{2}^{\sqrt{2}})\end{matrix} \qquad \begin{matrix}(\pi_2)\\ R(\sqrt{2}^{\sqrt{2}}), F_1, F_2 \vdash F_3\end{matrix}}{\neg R(\sqrt{2}), R(\sqrt{2}^{\sqrt{2}^{\sqrt{2}}}) \vdash \exists x \exists y \neg R(x) \wedge \neg R(y) \wedge R(x^y)}\ cut + \text{contractions}$$

where $F_1 = \neg R(\sqrt{2})$, $F_2 = R(\sqrt{2}^{\sqrt{2}^{\sqrt{2}}})$, $F_3 = \exists x \exists y(\neg R(x) \wedge \neg R(y) \wedge R(x^y))$ and $\pi_1 =$

$$\dfrac{\dfrac{\dfrac{\dfrac{R(\sqrt{2}^{\sqrt{2}}) \vdash R(\sqrt{2}^{\sqrt{2}})}{\vdash \neg R(\sqrt{2}^{\sqrt{2}}), R(\sqrt{2}^{\sqrt{2}})}\,\neg_r \qquad \begin{matrix}(\pi_1')\\ \neg R(\sqrt{2}), R(\sqrt{2}^{\sqrt{2}^{\sqrt{2}}}) \vdash \neg R(\sqrt{2}) \wedge R(\sqrt{2}^{\sqrt{2}^{\sqrt{2}}})\end{matrix}}{\neg R(\sqrt{2}), R(\sqrt{2}^{\sqrt{2}^{\sqrt{2}}}) \vdash \neg R(\sqrt{2}^{\sqrt{2}}) \wedge \neg R(\sqrt{2}) \wedge R(\sqrt{2}^{\sqrt{2}^{\sqrt{2}}}), R(\sqrt{2}^{\sqrt{2}})}\,\wedge_r}{\neg R(\sqrt{2}), R(\sqrt{2}^{\sqrt{2}^{\sqrt{2}}}) \vdash \exists y(\neg R(\sqrt{2}^{\sqrt{2}}) \wedge \neg R(y) \wedge R(\sqrt{2}^{\sqrt{2}^{y}})), R(\sqrt{2}^{\sqrt{2}})}\,\exists_r}{\neg R(\sqrt{2}), R(\sqrt{2}^{\sqrt{2}^{\sqrt{2}}}) \vdash \exists x \exists y(\neg R(x) \wedge \neg R(y) \wedge R(x^{y})), R(\sqrt{2}^{\sqrt{2}})}\,\exists_r$$

π_1' is

$$\dfrac{\neg R(\sqrt{2}) \vdash \neg R(\sqrt{2}) \qquad R(\sqrt{2}^{\sqrt{2}^{\sqrt{2}}}) \vdash R(\sqrt{2}^{\sqrt{2}^{\sqrt{2}}})}{\neg R(\sqrt{2}), R(\sqrt{2}^{\sqrt{2}^{\sqrt{2}}}) \vdash \neg R(\sqrt{2}) \wedge R(\sqrt{2}^{\sqrt{2}^{\sqrt{2}}})}\,\wedge_r$$

and $\pi_2 =$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\neg R(\sqrt{2}) \vdash \neg R(\sqrt{2}) \qquad \dfrac{\neg R(\sqrt{2}) \vdash \neg R(\sqrt{2}) \qquad R(\sqrt{2}^{\sqrt{2}}) \vdash R(\sqrt{2}^{\sqrt{2}})}{\neg R(\sqrt{2}), R(\sqrt{2}^{\sqrt{2}}) \vdash \neg R(\sqrt{2}) \wedge R(\sqrt{2}^{\sqrt{2}})}\,\wedge_r}{\neg R(\sqrt{2}), \neg R(\sqrt{2}), R(\sqrt{2}^{\sqrt{2}}) \vdash \neg R(\sqrt{2}) \wedge \neg R(\sqrt{2}) \wedge R(\sqrt{2}^{\sqrt{2}})}\,\wedge_r}{\neg R(\sqrt{2}), R(\sqrt{2}^{\sqrt{2}}) \vdash \neg R(\sqrt{2}) \wedge \neg R(\sqrt{2}) \wedge R(\sqrt{2}^{\sqrt{2}})}\,c_l}{R(\sqrt{2}^{\sqrt{2}}), \neg R(\sqrt{2}), R(\sqrt{2}^{\sqrt{2}^{\sqrt{2}}}) \vdash \neg R(\sqrt{2}) \wedge \neg R(\sqrt{2}) \wedge R(\sqrt{2}^{\sqrt{2}})}\,w_l}{R(\sqrt{2}^{\sqrt{2}}), \neg R(\sqrt{2}), R(\sqrt{2}^{\sqrt{2}^{\sqrt{2}}} \vdash \exists y(\neg R(\sqrt{2}) \wedge \neg R(y) \wedge R(\sqrt{2}^{y}))}\,\exists_r}{R(\sqrt{2}^{\sqrt{2}}), \neg R(\sqrt{2}), R(\sqrt{2}^{\sqrt{2}^{\sqrt{2}}}) \vdash \exists x \exists y(\neg R(x) \wedge \neg R(y) \wedge R(x^{y}))}\,\exists_r$$

Consider the minimal translation of π_2 in order to obtain the critical formulas: in a first step, π_2 is translated into π_2':

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\neg R(\sqrt{2}) \vdash \neg R(\sqrt{2}) \qquad \dfrac{\neg R(\sqrt{2}) \vdash \neg R(\sqrt{2}) \qquad R(\sqrt{2}^{\sqrt{2}}) \vdash R(\sqrt{2}^{\sqrt{2}})}{\neg R(\sqrt{2}), R(\sqrt{2}^{\sqrt{2}}) \vdash \neg R(\sqrt{2}) \wedge R(\sqrt{2}^{\sqrt{2}})}\,\wedge_r}{\neg R(\sqrt{2}), \neg R(\sqrt{2}), R(\sqrt{2}^{\sqrt{2}}) \vdash \neg R(\sqrt{2}) \wedge \neg R(\sqrt{2}) \wedge R(\sqrt{2}^{\sqrt{2}})}\,\wedge_r}{\neg R(\sqrt{2}), R(\sqrt{2}^{\sqrt{2}}) \vdash \neg R(\sqrt{2}) \wedge \neg R(\sqrt{2}) \wedge R(\sqrt{2}^{\sqrt{2}})}\,c_l}{R(\sqrt{2}^{\sqrt{2}}), \neg R(\sqrt{2}), R(\sqrt{2}^{\sqrt{2}^{\sqrt{2}}}) \vdash \neg R(\sqrt{2}) \wedge \neg R(\sqrt{2}) \wedge R(\sqrt{2}^{\sqrt{2}})}\,w_l}{\nu : R(\sqrt{2}^{\sqrt{2}}), \neg R(\sqrt{2}), R(\sqrt{2}^{\sqrt{2}^{\sqrt{2}}} \vdash \neg R(\sqrt{2}) \wedge \neg R(f(\sqrt{2})) \wedge R(\sqrt{2}^{f(\sqrt{(2)})})}\,fv_r}{R(\sqrt{2}^{\sqrt{2}}), \neg R(\sqrt{2}), R(\sqrt{2}^{\sqrt{2}^{\sqrt{2}}}) \vdash \exists x(\neg R(x) \wedge \neg R(f(x)) \wedge R(x^{f(x)}))}\,\exists_r$$

where $\mathcal{S}_{\exists}(\nu) = \{(\neg R(\sqrt{2}) \wedge \neg R(\sqrt{2}) \wedge R(\sqrt{2}^{\sqrt{2}})) \to (\neg R(\sqrt{2}) \wedge \neg R(f(\sqrt{2})) \wedge R(\sqrt{2}^{f(\sqrt{(2)})}))\}$, $\mathbf{\Delta}(\nu) = \{f\}$, $\mathbf{\Gamma}(\nu) = \emptyset$. To eliminate the second $\exists$ inference we construct $\pi_2'' =$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\neg R(\sqrt{2}) \vdash \neg R(\sqrt{2}) \qquad \dfrac{\neg R(\sqrt{2}) \vdash \neg R(\sqrt{2}) \qquad R(\sqrt{2}^{\sqrt{2}}) \vdash R(\sqrt{2}^{\sqrt{2}})}{\neg R(\sqrt{2}), R(\sqrt{2}^{\sqrt{2}}) \vdash \neg R(\sqrt{2}) \wedge R(\sqrt{2}^{\sqrt{2}})}\wedge_r}{\neg R(\sqrt{2}), \neg R(\sqrt{2}), R(\sqrt{2}^{\sqrt{2}}) \vdash \neg R(\sqrt{2}) \wedge \neg R(\sqrt{2}) \wedge R(\sqrt{2}^{\sqrt{2}})}\wedge_r}{\neg R(\sqrt{2}), R(\sqrt{2}^{\sqrt{2}}) \vdash \neg R(\sqrt{2}) \wedge \neg R(\sqrt{2}) \wedge R(\sqrt{2}^{\sqrt{2}})}c_l}{R(\sqrt{2}^{\sqrt{2}}), \neg R(\sqrt{2}), R(\sqrt{2}^{\sqrt{2}^{\sqrt{2}}}) \vdash \neg R(\sqrt{2}) \wedge \neg R(\sqrt{2}) \wedge R(\sqrt{2}^{\sqrt{2}})}w_l}{\nu : R(\sqrt{2}^{\sqrt{2}}), \neg R(\sqrt{2}), R(\sqrt{2}^{\sqrt{2}^{\sqrt{2}}} \vdash \neg R(\sqrt{2}) \wedge \neg R(f(\sqrt{2})) \wedge R(\sqrt{2}^{f(\sqrt{(2)})})}fv_r}{\mu : R(\sqrt{2}^{\sqrt{2}}), \neg R(\sqrt{2}), R(\sqrt{2}^{\sqrt{2}^{\sqrt{2}}}) \vdash \neg R(d) \wedge \neg R(f(d)) \wedge R(d^{f(d)})}fv_r$$

and $\mathcal{S}_\exists(\mu) = \mathcal{S}_\exists(\nu) \cup \{(\neg R(\sqrt{2}) \wedge \neg R(f(\sqrt{2})) \wedge R(\sqrt{2}^{f(\sqrt{(2)})})) \rightarrow (\neg R(d) \wedge \neg R(f(d)) \wedge R(d^{f(d)}))\}$, $\mathbf{\Delta}(\nu) = \{f, d\}$ and $\mathbf{\Gamma}(\nu) = \{f < d\}$. The same transformation is performed on π_1, hence the set of critical formulas is $\{C_1, C_2, C_3, C_4\}$, where

$$\begin{aligned}
C_1 = {} & (\neg R(\sqrt{2}) \wedge \neg R(\sqrt{2}) \wedge R(\sqrt{2}^{\sqrt{2}})) \rightarrow \\
& (\neg R(\sqrt{2}) \wedge \neg R(f(\sqrt{2})) \wedge R(\sqrt{2}^{f(\sqrt{2})})) \\
C_2 = {} & (\neg R(\sqrt{2}^{\sqrt{2}}) \wedge \neg R(\sqrt{2}) \wedge R(\sqrt{2}^{\sqrt{2}^{\sqrt{2}}})) \rightarrow \\
& (\neg R(\sqrt{2}^{\sqrt{2}}) \wedge \neg R(f(\sqrt{2}^{\sqrt{2}})) \wedge R(\sqrt{2}^{\sqrt{2}^{f(\sqrt{2}^{\sqrt{2}})}})) \\
C_3 = {} & (\neg R(\sqrt{2}) \wedge \neg R(f(\sqrt{2})) \wedge R(\sqrt{2}^{f(\sqrt{2})})) \rightarrow \\
& (\neg R(d) \wedge \neg R(f(d)) \wedge R(d^{f(d)})) \\
C_4 = {} & (\neg R(\sqrt{2}^{\sqrt{2}}) \wedge \neg R(f(\sqrt{2}^{\sqrt{2}})) \wedge R(\sqrt{2}^{\sqrt{2}^{f(\sqrt{2}^{\sqrt{2}})}})) \rightarrow \\
& (\neg R(d) \wedge \neg R(f(d)) \wedge R(d^{f(d)}))
\end{aligned}$$

The function variable proof is

$$C_1, C_2, C_3, C_4, \neg R(\sqrt{2}), R(\sqrt{2}^{\sqrt{2}^{\sqrt{2}}}) \vdash \neg R(d) \wedge \neg R(f(d)) \wedge R(d^{f(d)}).$$

5 A New Version of the Extended First ε-Theorem

In this section we make use of the partial order on terms and thus define a new elimination order, leading to a nonelementary speed-up of the computation of Herbrand disjunctions.

Lemma 2 (Hilbert's Ansatz in the new format). *Let*

$$A(t_1) \rightarrow A(e), \ldots, A(t_r) \rightarrow A(e), \mathit{\Gamma} \vdash \mathit{\Delta}$$

be a valid sequent, where $A(t_1) \rightarrow A(e), \ldots, A(t_r) \rightarrow A(e)$ are critical formulas and e is a term. Then the critical formulas and e can be eliminated and we obtain a valid sequent $(\mathit{\Gamma}\sigma_1, \ldots, \mathit{\Gamma}\sigma_r, \mathit{\Gamma} \vdash \mathit{\Delta}\sigma_1, \ldots, \mathit{\Delta}\sigma_r, \mathit{\Delta})\{e \leftarrow c\}$ where $\sigma_i = \{e \leftarrow t_i\}$ and c is a fixed constant. (Analogously for critical formulas of the form $A(e) \rightarrow A(t_1), \ldots, A(e) \rightarrow A(t_r)$).

Proof. Analogous to the proof of Lemma 1.

Theorem 2 (function variable elimination theorem). *Let $F = C_1, \ldots, C_r, A_1, \ldots, A_m \vdash A_{m+1}, \ldots, A_n$ be a function variable proof with $\mathbf{\Delta}(F)$ and $\mathbf{\Gamma}(F)$. Then F can be transformed into a valid sequent without function variables $\ldots, A_i\sigma_i^1, \ldots, A_i\sigma_i^{r_1}, \ldots \vdash \ldots, A_j\sigma_j^1, \ldots, A_j\sigma_j^{r_j}, \ldots$ where σ_v^u replace terms with outermost function variables by terms without function variables.*

Proof. Let $F = C_1, \ldots, C_r, A_1, \ldots, A_m \vdash A_{m+1}, \ldots, A_n$ be a function variables proof. By Hilbert's Ansatz we may eliminate any specific critical function term and all critical formulas belonging to it. The problem is that the substitution might render other critical formulas into formulas which are not critical formulas any more. Consider e.g. a critical formula $A(g_Q(t(s))) \rightarrow A(g_Q(t(f_\exists(u_1, \ldots, u_m))))$ where both $g_Q(t(f_\exists(u_1, \ldots, u_m)))$ and $f_\exists(u_1, \ldots, u_m)$ are critical terms. If we eliminate $g_Q(t(f_\exists(u_1, \ldots, u_m)))$ before $f_\exists(u_1, \ldots, u_m)$ we might obtain the spoiled formula $A(g_Q(t(s))) \rightarrow A(h)$, similar for the left part of the implication. This situation will not occur if we always eliminate critical formulas belonging to a $\mathbf{\Gamma}$-maximal function variable (here condition 3 of Definition 4 is applied). Furthermore, it cannot occur that critical formulas with different heads belong to one critical term. This is ensured by condition 4 of Definition 4. (Indeed, if during the elimination process different critical terms become equal, the heads have to become equal too as the unifying substitution is instance of the most general unifier existing by condition 4 of Definition 4). This ensures the soundness of the elimination procedure. To ensure the termination of the elimination procedure we will order the critical terms $f_Q(t)$ belonging to a maximal function variable f_Q according to the subterm property and always eliminate maximal terms $f_Q(t)$ under this partial order.

Example 5. (Example 1.3 continued.) The critical formulas cannot be eliminated from $A(c) \rightarrow A(e_Q), \neg A(d) \rightarrow \neg A(e_Q) \vdash A(c) \rightarrow A(d)$ with critical function variable e_Q as $A(c) \rightarrow A(d)$ is the only possible Herbrand disjunction, but $A(c) \rightarrow A(d)$ is not valid. This shows that condition 4 of Definition 4 is essential for the application of variants of Hilbert's Ansatz.

Corollary 1. *The function variable elimination theorem constructs a Herbrand sequent for $\forall x_1 A_1, \ldots, \forall x_m A_m \vdash \exists x_{m+1} A_{m+1}, \ldots, \exists x_n A_n$, where A_i for $1 \leq i \leq n$ is quantifier-free.*

Note that the result of the function variable elimination theorem can also be used to construct Herbrand expansions in case the weak quantifiers are positioned infix. This is done by considering the resulting sequent as an iterated conjunction implying an iterated disjunction and moving conjunctions and disjunctions inside if necessary. The size of each resulting conjunction and disjunction is obviously limited by the number of formulas in the original sequent (cf. [1]).

Corollary 2. *Let $F = C_1, \ldots C_n, \Pi^* \vdash \Delta^*$ be a function variable proof with $\mathbf{\Delta}(F)$ and $\mathbf{\Gamma}(F)$, where Π^* and Δ^* are function variable translations of instances by restricted function unifiers of Π and Δ, Π and Γ contain only weak quantifiers. Then $\Pi \vdash \Delta$ is valid.*

Theorem 2 induces the following non-deterministic algorithm.

Definition 8

```
begin % algorithm AlgNew
1. let $F = C_1, \ldots, C_n, \Pi \vdash \Delta$ be a function variable proof with $\Delta_0^*(F) = \mathbf{\Delta}(F)$
   and $\mathbf{\Gamma}(F)$ and define $S_0 = C_1, \ldots, C_n, F_0' = \Pi, F_0'' = \Delta$ and let $k = 0$;
2. check whether $F_k' \vdash F_k''$ is a tautology, if yes substitute all remaining
   function variable terms by the fixed constant $c$ and terminate AlgNew;
3. delete $A \to A$ from $S_k$. Let $S_k'$ be the result.;
4. choose $\Gamma$-maximal function symbols $f_1, \ldots, f_n$ in $\Delta_k^*(F)$;
5. order the critical terms $e_1, \ldots, e_r$ (which have an outermost function
   symbol among $f_1, \ldots, f_n$) by the subterm property and eliminate a maximal
   $e_i$ according to the modification in Hilbert's Ansatz. Delete the critical
   formulas $S(e_i)$ belonging to $e_i$. Let $\sigma_{i_1} \ldots \sigma_{i_l}$ be the corresponding
   substitutions. $S_{k+1} = (S_k' - S(e_i))\sigma_{i_1}\lambda \ldots (S_k' - S(e_i))\sigma_{i_l}\lambda$,
   $F_{k+1}' = F_k'\sigma_{i_1}\lambda \ldots F_k'\sigma_{i_l}\lambda, F_k'\lambda,\ F_{k+1}'' = F_k''\sigma_{i_1}\lambda \ldots F_k''\sigma_{i_l}\lambda, F_k''\lambda$,
   where $\lambda$ is $e_i \leftarrow c$ for the fixed constant $c$.
   Repeat this operation until there are no critical terms with outermost
   function symbols $f_1 \ldots f_n$. Let $\Delta_{k+1}^* = \Delta_k^* - \{f_1, \ldots, f_n\}$. Proceed with 2.;
end.
```

Note that the original elimination algorithm for ε/τ terms can be considered as instance of this algorithm using a more deterministic order of terms.

Example 6. Example 4 continued. We have $f < d$ and proceed with the elimination procedure:

1. Eliminate d by eliminating C_3 and C_4 via $\{d \leftarrow \sqrt{2}\}$ and $\{d \leftarrow \sqrt{2}^{\sqrt{2}}\}$: $C_1, C_2, \neg R(\sqrt{2}), R(\sqrt{2}^{\sqrt{2}^{\sqrt{2}}}) \vdash \neg R(\sqrt{2}) \wedge \neg R(f(\sqrt{2})) \wedge R(\sqrt{2}^{f(\sqrt{2})}), \neg R(\sqrt{2}^{\sqrt{2}}) \wedge \neg R(f(\sqrt{2}^{\sqrt{2}})) \wedge R(\sqrt{2}^{\sqrt{2}^{f(\sqrt{2}^{\sqrt{2}})}})$.
2. Eliminate f by eliminating C_1, C_3:
   (a) Eliminate C_3 via $\{f(\sqrt{2}) \leftarrow \sqrt{2}\}$: $C_2, \neg R(\sqrt{2}), R(\sqrt{2}^{\sqrt{2}^{\sqrt{2}}}) \vdash \neg R(\sqrt{2}) \wedge \neg R(\sqrt{2}) \wedge R(\sqrt{2}^{\sqrt{2}}), \neg R(\sqrt{2}^{\sqrt{2}}) \wedge \neg R(f(\sqrt{2}^{\sqrt{2}})) \wedge R(\sqrt{2}^{\sqrt{2}^{f(\sqrt{2}^{\sqrt{2}})}})$.
   (b) Eliminate C_2 via $\{f(\sqrt{2}^{\sqrt{2}}) \leftarrow \sqrt{2}\}$: $\neg R(\sqrt{2}), R(\sqrt{2}^{\sqrt{2}^{\sqrt{2}}}) \vdash \neg R(\sqrt{2}) \wedge \neg R(\sqrt{2}) \wedge R(\sqrt{2}^{\sqrt{2}}), \neg R(\sqrt{2}^{\sqrt{2}}) \wedge \neg R(\sqrt{2}) \wedge R(\sqrt{2}^{\sqrt{2}^{\sqrt{2}}})$.

Example 7 illustrates the advantage of the nonextensionality of the new method, which in fact may lead to shorter Herbrand disjunctions and a nonelementary speed-up without using the effects of the partial order of our algorithm.

Example 7. Let c, d be constants and a, u variables. Consider the proof $\Psi =$

$$\dfrac{\dfrac{\overset{(\Psi_1)}{A \vdash B, C} \qquad \overset{(\Psi_2)}{B \vdash \exists x(P(x) \lor E)}}{A \vdash C, \exists x(P(x) \lor E)}\,\mathit{cut} \qquad \overset{(\Psi_3)}{C \vdash \exists x(P(x) \lor F)}}{A \vdash \exists x(P(x) \lor E), \exists x(P(x) \lor F)}\,\mathit{cut}$$

where $A = P(c) \lor P(d)$, $B = \exists x P(x) \lor E$, $C = \exists x P(x) \lor F$ and $\Psi_1 =$

$$\dfrac{\dfrac{\dfrac{P(c) \vdash P(c)}{P(c) \vdash \exists x P(x)}\,\exists_r}{P(c) \vdash \exists x P(x) \lor E}\,\lor_r \qquad \dfrac{\dfrac{P(d) \vdash P(d)}{P(d) \vdash \exists x P(x)}\,\exists_r}{P(d) \vdash \exists x P(x) \lor F}\,\lor_r}{P(c) \lor P(d) \vdash \exists x P(x) \lor E, \exists x P(x) \lor F}\,\lor_l$$

$\Psi_2 =$

$$\dfrac{\dfrac{\dfrac{\dfrac{P(a) \vdash P(a)}{P(a) \vdash P(a) \lor E}\,\lor_r}{P(a) \vdash \exists x(P(x) \lor E)}\,\exists_r}{\exists x P(x) \vdash \exists x(P(x) \lor E)}\,\exists_l \qquad \dfrac{\dfrac{E \vdash E}{E \vdash P(a) \lor E}\,\lor_r}{E \vdash \exists x(P(x) \lor E)}\,\exists_r}{\exists x P(x) \lor E \vdash \exists x(P(x) \lor E)}\,\lor_l$$

$\Psi_3 =$

$$\dfrac{\dfrac{\dfrac{\dfrac{P(u) \vdash P(u)}{P(u) \vdash P(u) \lor F}\,\lor_r}{P(u) \vdash \exists x(P(x) \lor F)}\,\exists_r}{\exists x P(x) \vdash \exists x(P(x) \lor F)}\,\exists_l \qquad \dfrac{\dfrac{F \vdash F}{F \vdash P(u) \lor F}\,\lor_r}{F \vdash \exists x(P(x) \lor F)}\,\exists_r}{\exists x P(x) \lor F \vdash \exists x(P(x) \lor F)}\,\lor_l$$

After the minimal transformation a function variable proof Ψ' is obtained: $\Psi' = C_1, C_2, C_3, C_4, P(c) \lor P(d) \vdash P(e_1) \lor E, P(e_2) \lor F$, where $C_1 = P(c) \to P(e_3)$, $C_2 = P(d) \to P(e_4)$, $C_3 = P(e_3) \lor E \to P(e_1) \lor E$ and $C_4 = P(e_4) \lor F \to P(e_2) \lor F$ are the critical formulas and $\mathbf{\Delta}(\Psi') = \{e_1, e_2, e_3, e_4\}$. After elimination and contraction we obtain $P(c) \lor P(d) \vdash P(c) \lor E, P(d) \lor F$. Note that this result is obtained with any elimination procedure.

In the usual ε-formalism e_3 and e_4 coincide. The ε/τ-proof (written in sequent notation) is $D_1, D_2, D_3, D_4, P(c) \lor P(d) \vdash P(f_1) \lor E, P(f_2) \lor F$, where $D_1 = P(c) \to P(f)$, $D_2 = P(d) \to P(f)$, $D_3 = P(f) \lor E \to P(f_1) \lor E$ and $D_4 = P(f) \lor F \to P(f_2) \lor F$, where $f \sim \varepsilon_x P(x)$, $f_1 \sim \varepsilon_x(P(x) \lor E)$ and $f_2 \sim \varepsilon_x(P(x) \lor F)$. After elimination and contraction we obtain the longer Herbrand sequent $P(c) \lor P(d) \vdash P(c) \lor E, P(d) \lor E, P(c) \lor F, P(d) \lor F$. Again, this result is obtained with any elimination procedure.

6 Complexity Analysis

In this section we analyze the computational complexity of the elimination of ε/τ-terms and compare the algorithms AlgOld and AlgNew. The worst-case complexity of the elimination procedures is inherently nonelementary (like for cut-elimination) because the size of the shortest Herbrand sequents S_n corresponding to proofs φ_n cannot be bounded in the length of φ_n.

Definition 9. *The complexity of a formula is the number of subformula occurrences in it. The complexity of a sequent is the sum of the complexities of the formula occurrences in it. The complexity of a proof is the sum of the complexities of the sequents occurring in it.*

Theorem 3. *There exists a sequence of proofs ψ_n s.t. the ε/τ-elimination via AlgOld requires a computing time $> s(n)/2$ (where $s(0) = 1$, $s(k+1) = 2^{s(k)}$ for al $k \geq 0$) while the computing time via AlgNew is bounded by Mn^k for constants M, k.*

Proof. Consider the Statman sequence φ_n of $\varGamma_n \vdash A_n$ as defined in [5]. Replace all atoms $s = t$ in φ_n by a formula $\exists x \exists y P(c, x, y, s, t)$ where c is a new constant symbol and P is a 5-ary predicate symbol. We obtain a proof sequence φ'_n of $\varGamma'_n \vdash A'_n$. In the next step replace all axioms of the form $\exists x \exists y P(c, x, y, s, t) \vdash \exists x \exists y P(c, x, y, s, t)$ by their obvious LK-derivation from other atomic axioms $P(c, x_1, x_2, s, t) \vdash P(c, x_1, x_2, s, t)$:

$$\dfrac{\dfrac{\dfrac{\dfrac{P(c, x_1, x_2, s, t) \vdash P(c, x_1, x_2, s, t)}{P(c, x_1, x_2, s, t) \vdash \exists y P(c, x_1, y, s, t)}\,\exists_r}{P(c, x_1, x_2, s, t) \vdash \exists x \exists y P(c, x, y, s, t)}\,\exists_r}{\exists y P(c, x_1, y, s, t) \vdash \exists x \exists y P(c, x, y, s, t)}\,\exists_l}{\exists x \exists y P(c, x, y, s, t) \vdash \exists x \exists y P(c, x, y, s, t)}\,\exists_l$$

We thus obtain a sequence φ''_n of $\varGamma'_n \vdash A'_n$. It is obvious that the Herbrand complexity of ψ_n is the same as that of φ_n itself. Now we define the following LK-proof sequence ψ_n:

$$\dfrac{\dfrac{\dfrac{\begin{matrix}(\varphi''_n)\\ \varGamma'_n \vdash A'_n\end{matrix}}{\varGamma'_n \vdash A'_n, P(c, c, c, c, c)}\,w_r}{\varGamma'_n \vdash A'_n, \exists x P(x, x, x, x, x)}\,\exists_r}{P(c, c, c, c, c), \varGamma'_n \vdash A'_n, \exists x P(x, x, x, x, x)}\,w_l$$

Then we transform ψ_n to its LK*-version ψ^*

$$\dfrac{\dfrac{\dfrac{\begin{matrix}(\varphi''^*_n)\\ \varGamma^*_n \vdash A^*_n\end{matrix}}{\varGamma^*_n \vdash A^*_n, P(c, c, c, c, c)}\,w_r}{\varGamma^*_n \vdash A^*_n, P(d, d, d, d, d)}\,fv_r}{P(c, c, c, c, c), \varGamma^*_n \vdash A^*_n, P(d, d, d, d, d)}\,w_l$$

where φ''^*_n is the LK*-version of φ''_n. Now $\mathbf{\Delta}(\psi^*_n) = \mathbf{\Delta}(\varphi''^*_n) \cup \{d\}$ and the set of critical formulas $\mathcal{S}(\psi^*_n) = \mathcal{S}(\varphi''^*_n) \cup \{P(c, c, c, c, c) \rightarrow P(d, d, d, d, d)\}$. Note that in our partial ordering $d \not< g$ for all $g \in \mathbf{\Delta}(\varphi''^*_n)$ and d is locally maximal. So our (nondeterminisitc) algorithm AlgNew may eliminate the term d, thus obtaining the valid sequent $P(c,c,c,c,c), \Gamma_n^* \vdash A_n^*, P(c,c,c,c,c)$. In order to obtain a sequent without function variable symbols we may replace all remaining function terms by c. The whole procedure works in polynomial time in $||\psi_n||$. As $||\psi_n|| \leq 2^{kn}$ for a constant k we get a total time bound of $\leq 2^{k'n}$ for a constant k'.

AlgOld must first eliminate all ε-terms of rank >1 before eliminating d. But the elimination down to the innermost quantifiers $\exists x \exists y P(c,x,y,s,t)$ already produces a sequence of Herbrand sequents S_n (still containing ε-terms) of ψ_n; this sequence S_n must be of size $>s(n)/2$ for $s(0) = 1$, $s(k+1) = 2^{s(k)}$ for all k and thus is nonelementary. Therefore there is no elementary bound of the computing time of AlgOld in n.

7 Conclusion

This paper can be considered as a first approach to an effective algorithmic formulation of the computational content of the extended first ε-theorem. Additional refinements such as a restriction of the function variables in the sense of the Andrews Skolemization (see [2,3]) are possible. Additional research may restrict the application of the tautology check in the algorithm without influencing its efficiency.

Acknowledgments. Partially supported by FWF grants P-26976-N25, I-2671-N35 and the Czech-Austrian project MOBILITY No. 7AMB17AT054.

References

1. Aguilera, J.P., Baaz, M.: Unsound inferences make proofs shorter. CoRR, abs/1608.07703 (2016)
2. Andrews, P.B.: Resolution in type theory. In: Siekmann, J.H., Wrightson, G. (eds.) Automation of Reasoning. Symbolic Computation (Artificial Intelligence), pp. 487–507. Springer, Heidelberg (1971). https://doi.org/10.1007/978-3-642-81955-1_29
3. Andrews, P.B.: Theorem proving via general matings. J. ACM (JACM) **28**(2), 193–214 (1981)
4. Baaz, M., Hetzl, S., Weller, D.: On the complexity of proof deskolemization. J. Symbolic Logic **77**(2), 669–686 (2012)
5. Baaz, M., Leitsch, A.: On skolemization and proof complexity. Fundamenta Informaticae **20**(4), 353–379 (1994)
6. Baaz, M., Leitsch, A.: Cut normal forms and proof complexity. Ann. Pure Appl. Logic **97**(1–3), 127–177 (1999)
7. Hilbert, D., Bernays, P.: Grundlagen der Mathematik II (1939)
8. Luckhardt, H.: Herbrand-analysen zweier Beweise des Satzes von Roth: Polynomiale Anzahlschranken. J. Symbolic Logic **54**, 234–263 (1989)
9. Moser, G., Zach, R.: The epsilon calculus and Herbrand complexity. Stud. Logica. **82**(1), 133–155 (2006)

Angluin Learning via Logic

Simone Barlocco and Clemens Kupke(✉)

Computer and Information Sciences, University of Strathclyde, Glasgow, Scotland
{simone.barlocco,clemens.kupke}@strath.ac.uk

Abstract. In this paper we will provide a fresh take on Dana Angluin's algorithm for learning using ideas from coalgebraic modal logic. Our work opens up possibilities for applications of tools & techniques from modal logic to automata learning and vice versa. As main technical result we obtain a generalisation of Angluin's original algorithm from DFAs to coalgebras for an arbitrary finitary set functor T in the following sense: given a (possibly infinite) pointed T-coalgebra that we assume to be regular (i.e. having an equivalent finite representation) we can learn its finite representation by asking (i) "logical queries" (corresponding to membership queries) and (ii) making conjectures to which the teacher has to reply with a counterexample. This covers (a known variant) of the original L* algorithm and the learning of Mealy/Moore machines. Other examples are bisimulation quotients of (probabilistic) transition systems.

Keywords: Automata learning · Coalgebra · Modal logic

1 Introduction

Coalgebra studies "generated behaviour" that can be observed when interacting with a system. A lot of progress has been made thus far to formalise behaviour and to create languages that allow to specify and reason about it (cf. e.g. [1]). Intriguingly little, however, has been done to formalise the process of making observations and of using these observations to learn how a system works.

This has changed thanks to a series of recent work [2–5] but the connections between coalgebra & learning are still far from being completely understood. In this paper we will describe one such connection between the above mentioned coalgebraic specification languages (aka coalgebraic modal logics) and the well-known L* algorithm [6] for learning deterministic finite automata (DFA).

This algorithm constructs a minimal DFA accepting a (at the beginning unknown) regular language by asking a teacher membership queries of the form "is word w in the language?" and by making conjectures for what the language/DFA is, to which the teacher replies with a counterexample in case the conjecture is false. A central role in the algorithm is played by so-called tables

Supported by EPSRC grant EP/N015843/1.

S. Artemov and A. Nerode (Eds.): LFCS 2018, LNCS 10703, pp. 72–90, 2018.
https://doi.org/10.1007/978-3-319-72056-2_5

that essentially consist of two sets of finite words S and E of which the first corresponds to the states of the constructed DFA and the second set corresponds to observations or tests that we are performing on states.

The use of tests shows that the connection to logic is not at all surprising. A bit more surprising was for us the observation that closed & consistent tables can be best understood using the notion of a filtration from modal logic. We will not discuss this observation in detail - instead we will describe a generalisation of the L^* algorithm to coalgebras that was made possible by it. Our "L^{co} algorithm" allows - in principle - the learning of regular coalgebras for an arbitrary finitary set functor. This generalisation of L^* is the main contribution of this paper.

1.1 What the Algorithm Learns

The classical L^* algorithm looks very much like a bottom up procedure. The starting point of the algorithm are two singleton sets that grow step-by-step until the desired DFA has been learned. It is instructive, however, to think of the algorithm as follows: The regular language $\mathcal{L} \subseteq \Sigma^*$ that we intend to learn can be thought as represented by the infinite DFA

$$\langle o, \delta \rangle : \Sigma^* \to 2 \times (\Sigma^*)^{\Sigma}$$

where $o(w) = 1$ iff $w \in \mathcal{L}$ and $\delta(w)(a) = w \cdot a$ for all $w \in \Sigma^*$, $a \in \Sigma$. The assumption that $\mathcal{L}$ is a regular language can be rephrased by stating that the pointed coalgebra $(\Sigma^*, \langle o, \delta \rangle, \lambda)$ is behaviourally equivalent to a finite well-pointed coalgebra [7]. The aim of the algorithm is to learn this finite well-pointed coalgebra using queries that can be asked concerning the given infinite coalgebra. Our generalisation of Angluin's algorithm looks thus as follows: We are given a (possibly infinite) pointed T-coalgebra (X, γ, x) (corresponding to the language $\mathcal{L}$) and we assume that (X, γ, x) is *regular*, i.e. behaviourally equivalent to a finite well-pointed T-coalgebra (Y, δ, y). The goal of the algorithm is to learn (Y, δ, y).

1.2 Means to Learn

The central device for learning in our setting is logic: we assume that we are provided with an expressive modal language that allows to characterise coalgebras up-to behavioural equivalence. The learner is able to ask two types of queries:

- Logical queries of the form "*is formula φ true at some state $x' \in X$?*"
- Conjectures of the form "*is this the correct well-pointed T-coalgebra?*"

Logical queries are answered truthfully with *Yes* or *No* by the teacher, conjectures are either confirmed or the teacher provides a counterexample in the shape of a formula $\psi \in \mathcal{F}(\Lambda)$ that can be used to distinguish the point of the conjectured coalgebra from the point x of the coalgebra that we are trying to learn.

1.3 Related Work

Within the large body of literature on Angluin learning [6], the line of research closely related to our paper is [2–5] which applies categorical techniques to automata learning. There are, however, important differences to our work. On the one hand, our logic-based methodology allows us to make the L* algorithm *parametric in the system type*, which is less clear in the cited articles. On the other hand, our results are limited to the category of sets - results such as learning linear weighted or nominal automata [2,5] are currently out of reach for our approach. Connections between modal logic and automata theory are of course well-known [8], but the link between modal logic and automata learning that we are describing is, to the best of our knowledge, new.

2 Preliminaries

We assume that the reader is familiar with basic category theory [9] and coalgebra [10]. We briefly recall some definitions from coalgebra & modal logic that will play a central role in the paper. Throughout the paper we will be working with an arbitrary finitary set functor T and we will assume - without loss of generality [7,11] - that T preserves intersections and for any sets X, Y with $X \subseteq Y$ the inclusion map $\iota_{X,Y} : X \to Y$ gets mapped to the inclusion $\iota_{TX,TY}$ (in particular, we have $TX \subseteq TY$). Under this assumption finitariness of the functor T means that for all sets X and all elements $t \in TX$ there is a finite subset $X' \subseteq X$ such that $t \in TX'$. Another functor that will be playing an important role in our paper is the contravariant power set functor $P : \mathsf{Set}^{op} \to \mathsf{Set}$ that maps a set X to the collection PX of subsets of X and a function $f : X \to Y$ to the inverse image function $Pf = f^{-1}$ where for $V \subseteq Y$ we have $f^{-1}(V) = \{x \in X \mid f(x) \in V\}$. Finally, for a set X we denote by $\#X$ the cardinality of X and we write $X' \subseteq_\omega X$ if X' is a *finite* subset of X.

2.1 Coalgebra

A T-coalgebra is a pair (X, γ) where X is a set and $\gamma : X \to TX$ is a function. A T-coalgebra morphism from a T-coalgebra (X, γ) to another T-coalgebra morphism (Y, δ) is a function $f : X \to Y$ such that the following diagram commutes.

$$\begin{tikzcd}
X \arrow[r, "f"] \arrow[d, "\gamma"'] & Y \arrow[d, "\delta"] \\
TX \arrow[r, "Tf"'] & TY
\end{tikzcd}$$

We call two states $x_1 \in X_1$ and $x_2 \in X_2$ of two T-coalgebras (X_1, γ_1) and (X_2, γ_2) *behaviourally equivalent*[1] if there exists a T-coalgebra (Y, δ) and coalgebra morphisms $f_i : (X_i, \gamma_i) \to (Y, \delta)$ for $i = 1, 2$ such that $f_1(x_1) = f_2(x_2)$.

[1] Readers should think of "behavioural equivalence" as a general notion of bisimilarity. In all concrete examples in this paper both notions of equivalence coincide.

In this case we write $x_1 \leftrightarroweq_T x_2$ or simply $x_1 \leftrightarroweq x_2$. A *pointed* T-coalgebra is a triple (X, γ, x) where (X, γ) is a T-coalgebra and $x \in X$ is a distinguished point (to be thought of as the "initial state"). A morphism $f : (X, \gamma, x) \to (Y, \delta, y)$ is a T-coalgebra morphism $f : (X, \gamma) \to (Y, \delta)$ such that $f(x) = y$. Two pointed coalgebras (X, γ, x) and (Y, δ, y) are behaviourally equivalent if $x \leftrightarroweq y$. When discussing concrete constructions on transition systems it is important to be able to express the notion of a "successor state", i.e. a state that can be reached after performing one step in the transition structure. Coalgebra does not come with such a notion from the outset, but it is well-known that successors can be formalised via the notion of base.

Definition 1. *Given a finitary set functor T and an element $t \in TX$, we define*

$$\mathrm{Base}^T_X(t) := \bigcap \{Y \subseteq_\omega X \mid t \in TY\}.$$

Whenever T and X are clear from the context we will drop the super- and subscript, respectively, and simply write $\mathrm{Base}(t)$*.*

Fact 1. *Let T be a finitary set functor, let X be a set and let $t \in TX$. Then $t \in T\mathrm{Base}^T_X(t)$ and for any proper subset $X' \subset \mathrm{Base}^T_X(t)$ we have $t \notin TX'$.*

The notion of base can be used to associate with any T-coalgebra a graph that can be used to give a concrete representation of reachability.

Definition 2 ([7]). *Let (X, γ) be a T-coalgebra. The canonical graph $(X, \mathrm{can}_\gamma : X \to \mathcal{P}_f X)$ is defined by putting $\mathrm{can}_\gamma(x) := \mathrm{Base}^T_X(\gamma(x))$ for all $x \in X$.*

Note that we can confine ourselves to finitely branching canonical graphs as we are considering only *finitary* set functors.

Example 1. 1. $T = A \times \mathrm{Id}$ where A is a set and Id is the identity functor. In this case $\mathrm{Base}^T_X((a, x)) = \{x\}$ for $(a, x) \in A \times X$.
2. $T = 2 \times \mathrm{Id}^A$ where A is a finite set (the "alphabet"). Then $\mathrm{Base}^T_X((i, f)) = \{fa \mid a \in A\}$ for $(i, f) \in TX$.
3. Let $T = \mathcal{D}_\omega$ where $\mathcal{D}_\omega X$ denotes the collection of discrete probability distributions over X with finite support, i.e. $\mathcal{D}_\omega X = \{\mu : X \to [0, 1] \mid \sum_{x \in X} \mu(x) = 1, \mathrm{supp}(\mu) \text{ finite}\}$ and for a function $f : X \to Y$ and $\mu : X \to [0, 1]$ we have $(\mathcal{D}_\omega f)(\mu)(y) = \sum_{f(x)=y} \mu(x)$, where $\mathrm{supp}(\mu) = \{x \in X \mid \mu(x) \neq 0\}$ denotes the support of μ. In this case $\mathrm{Base}^{\mathcal{D}_\omega}_X(\mu) = \mathrm{supp}(\mu)$. Note that while $\mathcal{D}_\omega$ does not satisfy our condition that $X \subseteq Y$ implies $\mathcal{D}_\omega X \subseteq \mathcal{D}_\omega Y$, an isomorphic modification of $\mathcal{D}_\omega$ does, e.g. the functor $\mathcal{D}'_\omega$ that maps a set X to the set of *partial* functions $\mu : X \rightharpoonup (0, 1]$ with $\mathrm{supp}(\mu)$ finite and $\sum_{x \in X} \mu(x) = 1$. We stick with the functor $\mathcal{D}_\omega$ as it is commonly used and as this technical subtlety is irrelevant for this specific example.

We are going to describe a learning algorithm that learns a *minimal* finite representation of the behaviour of a given pointed coalgebra. Minimal here means that the learned pointed coalgebra should not have any proper subcoalgebras or proper quotients. Let us first recall those notions.

Definition 3. *Let (X, γ, x) be a pointed T-coalgebra. We say (Y, δ, y) is a quotient of (X, γ, x) if there is a surjective morphism $q : (X, \gamma, x) \to (Y, \delta, y)$. We say (Y, δ, y) is a subobject of (X, γ, x) if there is an injection $m : (Y, \delta, y) \to (X, \gamma, x)$. A quotient or subobject (Y, δ, y) of (X, γ, x) is said to be proper if (Y, δ, y) is not isomorphic to (X, x, γ).*

Minimality of a pointed coalgebra is captured by the following notion from [7].

Definition 4. *A pointed T-coalgebra (X, γ, x) is called* well-pointed *if it is* simple *and* reachable, *i.e. if it does not have proper quotients or proper subcoalgebras.*

Example 2. 1. For $T = 2 \times (\_)^A$, the (finite) well-pointed T-coalgebras are DFAs where each state is reachable from the initial state and where no two states represent the same (regular) language.
2. For $T = \mathcal{P}_\omega$ (where $\mathcal{P}_\omega X = \{X' \mid X' \subseteq_\omega X\}$) well-pointed T-coalgebras are finitely-branching Kripke frames with a designated state such that each state is reachable from the designated state and such that no two distinct states are bisimilar.

2.2 Coalgebraic Modal Logic

Coalgebraic modal logics [12,13] are a family of logics that allow to express properties of pointed coalgebras that are invariant under behavioural equivalence. We first recall the basic definitions and state an important fact on expressivity of these logics. Later in the paper formulas will serve as tests that allow to learn a given (pointed) coalgebra. The language of coalgebraic modal logic is determined by choosing a collection of modal operators - the so-called similarity type:

Definition 5. *A* (modal) similarity type *is a set of modal operators with arities. If Λ is a similarity type, a Λ*-structure *consists of an endofunctor $T : \mathsf{Set} \to \mathsf{Set}$, together with an assignment of an n-ary* predicate lifting, *that is, a natural transformation of type $[\![\heartsuit]\!] : (P)^n \to P \circ T$ where $P : \mathsf{Set} \to \mathsf{Set}^{op}$ is the contravariant powerset functor, to every n-ary operator $\heartsuit \in \Lambda$.*

The syntax is now given as propositional modal logic, where the modal operators are the elements of Λ. To keep things simple we treat propositional variables as nullary predicate liftings. The definition of the semantics is also standard.

Definition 6. *The* language *induced by a modal similarity type Λ is the set $\mathcal{F}(\Lambda)$ of formulas given by*

$$\mathcal{F}(\Lambda) \ni \varphi, \psi ::= \top \mid \varphi \wedge \psi \mid \neg\varphi \mid \heartsuit(\varphi_1, \ldots, \varphi_n) \qquad (\heartsuit \in \Lambda \text{ } n\text{-ary})$$

Given a Λ-structure T and a T-coalgebra (X, γ), the semantics *of $\varphi \in \mathcal{F}(\Lambda)$ is inductively given by*

$$\begin{array}{ll} [\![\top]\!]_{(X,\gamma)} = X & \\ [\![\varphi \wedge \psi]\!]_{(X,\gamma)} := [\![\varphi]\!]_{(X,\gamma)} \cap [\![\psi]\!]_{(X,\gamma)} & [\![\neg\varphi]\!]_{(X,\gamma)} := X \setminus [\![\varphi]\!]_{(X,\gamma)}, \\ [\![\heartsuit(\varphi_1, \ldots, \varphi_n)]\!]_{(X,\gamma)} := P\gamma \circ [\![\heartsuit]\!]_X([\![\varphi_1]\!]_{(X,\gamma)}, \ldots, [\![\varphi_n]\!]_{(X,\gamma)}), & \end{array}$$

Instead of $x \in [\![\varphi]\!]_{(X,\gamma)}$ we will often write $(X, \gamma, x) \models \varphi$ or simply $x \models \varphi$. In words "the pointed coalgebra (X, γ, x) satisfies φ", or simply "x satisfies φ".

The logics give rise to the notion of logical equivalence.

Definition 7. *Let (X, γ) and (Y, δ) be T-coalgebras and let $\Sigma \subseteq \mathcal{F}(\Lambda)$ be a set of formulas. Two states $x \in X$ and $y \in Y$ are logically equivalent wrt Σ if for all formulas $\varphi \in \Sigma$ we have $x \models \varphi$ iff $y \models \varphi$. In this case we write $x \equiv_\Sigma y$. The equivalence classes wrt $\equiv_\Sigma$ on a coalgebra (X, γ) will be denoted by $|x|_\Sigma$, i.e. we put $|x|_\Sigma = \{x' \in X \mid x' \equiv_\Sigma x\}$. If $\Sigma = \mathcal{F}(\Lambda)$ is the set of all formulas we simply write $\equiv$ for the equivalence and denote the equivalence class of some x by $|x|$.*

The semantics of formulas of coalgebraic modal logic is invariant under behavioural equivalence. An important stronger property that coalgebraic modal logics often possess is expressivity, i.e. the property that logical equivalence implies behavioural equivalence.

Definition 8. *The language $\mathcal{F}(\Lambda)$ is called* expressive *if for all T-coalgebras (X, γ) and (Y, δ) and all $x \in X$, $y \in Y$ we have $x \equiv y$ iff $x \leftrightarroweq y$.*

For a finitary set functor T we are always able to find an expressive language, cf. e.g. [14]. Many expressive coalgebraic modal logics have been considered in the literature, we mention two examples - note that in both examples we do not need the full Boolean structure of the language to obtain expressivity. Also note that we are simply sketching the semantics of the logics without explicitly explaining how its definition can be seen as special case of Definition 6.

Example 3. 1. For $T = (\_ \times O)^I$ coalgebras correspond to so-called Mealy machines [15] and we define:

$$\mathcal{F}(\Lambda) \ni \varphi ::= \top \mid p_{i/o}, i \in I, o \in O \mid [i]\varphi, i \in I.$$

Given a T-coalgebra (X, γ), a state $x \in X$ and $i \in I$, $o \in O$, we put $x \models p_{i/o}$ if $\pi_2(\gamma(x)(i)) = o$. Intuitively, $p_{i/o}$ is true in x if the output of the coalgebra at x on input i is equal to o. Furthermore we put $x \models [i]\varphi$ if $\pi_1(\gamma(x)(i)) \models \varphi$.

2. For $T = \mathcal{P}\mathsf{At} \times \mathcal{D}_\omega\_$ with At a set of propositional variables, coalgebras correspond to probabilistic transition systems and an expressive language [16] is given by

$$\mathcal{F}(\Lambda) \ni \varphi ::= \top \mid p, p \in \mathsf{At} \mid \varphi_1 \wedge \varphi_2 \mid L_q\varphi, q \in [0, 1],$$

where for a coalgebra (X, γ) we have $x \models p$ if $p \in \pi_1(\gamma(x))$ and $x \models L_q\varphi$ if $\sum_{x' \models \varphi} \pi_2(\gamma(x))(x') \geq q$, in words, if the probability of reaching a successor of x that makes φ true is at least q.

2.3 Logical Quotients

Given an expressive modal language $\mathcal{F}(\Lambda)$ for T, it is well known that we can compute the maximal quotient of a coalgebra simply by identifying states that satisfy the same modal formulas in $\mathcal{F}(\Lambda)$.

Fact 2. *Let (X, γ) be a T-coalgebra and let $\mathcal{F}(\Lambda)$ be an expressive language, let $|X| = \{|x| \mid x \in X\}$ and define $\gamma_{\mathcal{F}\Lambda} : |X| \to T|X|$ by putting $\gamma_{\mathcal{F}(\Lambda)}(|x|) := (T|\_|)(\gamma(x))$. Then the* logical quotient *$\gamma_{\mathcal{F}\Lambda}$ is well-defined and $q : X \to |X|$ given by $x \mapsto |x|$ is a T-coalgebra map that computes the maximal quotient of (X, γ).*

The proof of well-definedness can e.g. be found in [17]. That the quotient is maximal is an immediate consequence of the fact that truth of modal formulas is preserved by coalgebra morphisms. The method also allows us to obtain the well-pointed coalgebra that is equivalent to a given pointed T-coalgebra.

Lemma 1. *Let (X, γ, x) be a pointed T-coalgebra and let (Y, δ, x) be its smallest pointed subcoalgebra. The logical quotient $(|Y|, \delta_{\mathcal{F}(\Lambda)}, |x|)$ is well-pointed.*

Proof. First note that the smallest pointed subcoalgebra (Y, δ, x) exists as T preserves intersections. It is reachable by definition. Therefore also its quotient will be reachable because (i) it is easy to see that the canonical graph of the quotient is a quotient of the canonical graph of (Y, δ, x) and (ii) by [7, Lemma 3.16] reachability of its canonical graph implies reachability of a pointed T-coalgebra. Furthermore the logical quotient of (Y, δ, x) is simple as any two distinct states can be distinguished by a formula in $\mathcal{F}(\Lambda)$ and can thus not be behaviourally equivalent. This implies that $(|Y|, \delta_{\mathcal{F}(\Lambda)}, |x|)$ is well-pointed (cf. Definition 4).

3 The L$^{\text{co}}$ Algorithm

We first describe the possible configurations of the algorithm. After that we describe the algorithm parametric in a finitary set functor T. Finally we prove termination and correctness of the algorithm. Throughout this section we assume that we are given a pointed T-coalgebra (X, γ, x) whose finite representation we are trying to learn. Furthermore we are given an expressive language $\mathcal{F}(\Lambda)$ for T.

3.1 Filtrations and Logical Tables

Following Angluin's original terminology we will refer to configurations of our algorithm as logical tables or simply as tables. First we need some terminology concerning sets of formulas from $\mathcal{F}(\Lambda)$.

Definition 9. *A set $\Sigma \subseteq \mathcal{F}(\Lambda)$ is* closed under taking subformulas *or* subformula closed *if we have*

- *$\varphi_1 \wedge \varphi_2 \in \Sigma$ implies $\varphi_i \in \Sigma$ for $i \in \{1, 2\}$*
- *$\neg\varphi \in \Sigma$ implies $\varphi \in \Sigma$, and*
- *for all $\heartsuit \in \Lambda$ we have $\heartsuit(\varphi_1, \ldots, \varphi_n) \in \Sigma$ implies $\varphi_i \in \Sigma$ for all $i \in \{1, \ldots, n\}$.*

Sets that are closed under taking subformulas are used in modal logic to compute filtrations of Kripke models and to prove a truth lemma for such filtrations [18]. It turns out that a similar idea is at work in the L* algorithm, although the relevant construction is a modification of the standard filtration. We first give a coalgebraic account of the filtrations that play a role in our algorithm.

Definition 10. *Let (X, γ) be a T-coalgebra, let $\Sigma \subseteq \mathcal{F}(\Lambda)$ be a subformula closed set of formulas and let $S \subseteq X$ be a selection of points in X such that*

1. *for all $x_1, x_2 \in S$ we have $x_1 \not\equiv_\Sigma x_2$*
2. *for all $x \in S$ and all $x' \in \mathrm{Base}(\gamma(x))$ we have $|x'|_\Sigma \in |S|_\Sigma$*

*where $|S|_\Sigma = \{|x|_\Sigma \mid x \in S\}$. The (S, Σ)-*filtration *of (X, γ) has as carrier the set $|S|_\Sigma$ and as coalgebra structure $\gamma_{S,\Sigma}$ we define the map*

$$\gamma_{S,\Sigma}(|x|_\Sigma) := (T|\text{\_}|_\Sigma)(\gamma(x))$$

where we view the operation of identifying equivalent points as a function $|\text{\_}|_\Sigma : X \to |X|_\Sigma$ to which the functor T is applied.

Lemma 2. *Under the conditions on (S, Σ) from the previous definition, the (S, Σ)-filtration of a T-coalgebra (X, γ) is well-defined.*

Proof. This follows as $x \equiv_\Sigma x'$ implies $x = x'$ and thus $\gamma(x) = \gamma(x')$ for all $x, x' \in S$ by condition 1 and as $(T|\text{\_}|_\Sigma)(\gamma(x)) \in T|S|_\Sigma$ as S satisfies condition 2.

Definition 10 is reminiscent of the definition of a logical quotient in Fact 2 - the differences are that we select only one representant for each $\equiv_\Sigma$-equivalence class and that a (S, Σ)-filtration of a coalgebra is in general not a quotient. Instead a weaker property holds: restricted to elements in S, the map $|\text{\_}|_\Sigma$ preserves truth of formulas in Σ.

Lemma 3. *Let (X, γ) be a T-coalgebra, let $\Sigma \subseteq \mathcal{F}(\Lambda)$ be a subformula closed set of formulas and let $(|S|_\Sigma, \gamma_{S,\Sigma})$ be the (S, Σ)-filtration of (X, γ) for some suitable $S \subseteq X$. Then for all $x \in S$ and all $\varphi \in \Sigma$ we have*

$$(X, \gamma, x) \models \varphi \qquad \textit{iff} \qquad (|S|_\Sigma, \gamma_{S,\Sigma}, |x|_\Sigma) \models \varphi.$$

In particular, $(|S|_\Sigma, \gamma_{S,\Sigma})$ is simple.

Proof. We prove the statement by induction on the structure of the formula φ. In case $\varphi = \top$ is obvious. Similarly, the Boolean cases $\varphi = \psi_1 \wedge \psi_2$ and $\varphi = \neg\psi$ easily follow from the induction hypothesis. Suppose now that $\varphi = \heartsuit(\psi_1, \dots, \psi_n)$. We have $x \models \heartsuit(\psi_1, \dots, \psi_n)$ iff $\gamma(x) \in [\![\heartsuit]\!]_X([\![\psi_1]\!]_{(X,\gamma)}, \dots, [\![\psi_n]\!]_{(X,\gamma)})$ iff $x \in P\gamma\left([\![\heartsuit]\!]_X([\![\psi_1]\!]_{(X,\gamma)}, \dots, [\![\psi_n]\!]_{(X,\gamma)})\right)$. The last statement is by I.H. on the ψ_i's equivalent to

$$\sigma(x) \in P\gamma\left([\![\heartsuit]\!]_X(P|\text{\_}|_\Sigma([\![\psi_1]\!]_{(|S|_\Sigma,\gamma_{S,\Sigma})}), \dots, P|\text{\_}|_\Sigma([\![\psi_n]\!]_{(|S|_\Sigma,\gamma_{S,\Sigma})}))\right)$$

which is by naturality of $[\![\heartsuit]\!]$ equivalent to

$$x \in (P\gamma \circ PT|\text{\_}|_\Sigma)\left([\![\heartsuit]\!]_{|S|_\Sigma}([\![\psi_1]\!]_{(|S|_\Sigma,\gamma_{S,\Sigma})}, \dots, [\![\psi_n]\!]_{(|S|_\Sigma,\gamma_{S,\Sigma})})\right)$$

which is equivalent to $T|\text{\_}|_\Sigma(\gamma(x)) \in [\![\heartsuit]\!]_{|S|_\Sigma}([\![\psi_1]\!]_{(|S|_\Sigma,\gamma_{S,\Sigma})}), \dots, [\![\psi_n]\!]_{(|S|_\Sigma,\gamma_{S,\Sigma})})$ which finally is equivalent to $|x|_\Sigma \models \heartsuit(\psi_1, \dots, \psi_n)$ as required.

Given our observations on filtrations we are now ready to introduce the tables that will form the configurations of our algorithm.

Definition 11. *A* (logical) table *is a pair* (S, Σ) *where* $S \subseteq X$ *and* $\Sigma \subseteq \mathcal{F}(\Lambda)$ *is a set of formulas that is closed under taking subformulas. A table* (S, Σ) *is* closed *if for all* $x \in S$ *we have* $|\mathrm{Base}(\gamma(x)|_\Sigma \subseteq |S|_\Sigma$. *Finally we call* (S, Σ) sharp *if for all* $x_1, x_2 \in S$ *we have* $x_1 \neq x_2$ *implies* $x_1 \not\equiv_\Sigma x_2$.

Remark 1. Readers familiar with the L* algorithm will recognise the closedness condition. The condition that a table is sharp implies the consistency requirement in Angluin's work. Sharpness will be maintained by adding counterexamples to the tests and not to the states, similar to what is done e.g. in [19].

3.2 Description of the Algorithm

Let (X, γ, x) be a pointed coalgebra that is behaviourally equivalent to a finite well-pointed coalgebra and let $\mathcal{F}(\Lambda)$ be an expressive language. We will now describe a procedure that allows to learn the well-pointed coalgebra by asking queries to a teacher that knows (X, γ, x). Our algorithm will compute a closed and sharp table (S, Σ)[2] such that $x \in S$ and such that the pointed T-coalgebra $(|S|_\Sigma, \gamma_{S,\Sigma}, |x|_\Sigma)$ that is based on the (S, Σ)-filtration of (X, γ, x) is the finite quotient of a subcoalgebra of (X, γ, x) that contains x. The algorithm - depicted in Algorithm 1 - makes the following steps:

1. The start table is $(\{x\}, \{\top\})$ - the first component ensures that the distinguished point of (X, γ, x) is represented, the second component equals $\{\top\}$ to keep the formulation uniform as $\top \in \mathcal{F}(\Lambda)$ independently of the choice of language.
2. At configuration (S, Σ), check whether (S, Σ) is closed. If yes, jump to Step 4. If not, proceed with the next step.
3. Given a configuration (S, Σ) that is not closed, pick an element $x' \in S$ such that $|\mathrm{Base}(\gamma(x'))|_\Sigma \not\subseteq |S|_\Sigma$. For all $x'' \in \mathrm{Base}(\gamma(x'))$ check whether $|x''|_\Sigma \in |S|_\Sigma$, i.e. whether the equivalence class of x'' is already represented by some other element of S. If not, then add x'' to S. Otherwise check the next element of $\mathrm{Base}(\gamma(x'))$. Note that we are adding elements of $\mathrm{Base}(\gamma(x'))$ one-by-one to maintain sharpness of the table.
4. Given the closed configuration (S, Σ), present the (S, Σ)-filtration $\gamma_{S,\Sigma}$ to the teacher. If teacher accepts, the algorithm terminates. If teacher rejects, she has to provide a counterexample, i.e. a formula $\varphi \in \mathcal{F}(\Lambda)$ s.t. $x \models \varphi$ and $|x|_\Sigma \not\models \varphi$ or vice versa. In this case we put $\Sigma' = \Sigma \cup \mathrm{Sub}(\varphi)$ and the algorithm continues at Step 3.2 with (S, Σ').

[2] Instead, we could use triples $(S, \Sigma, \models_S)$ to be in line with [6] but we decided to leave the third "bookkeeping" component implicit.

Algorithm 1. The L^{co} algorithm with teacher (X, γ, x) and language $\mathcal{F}(\Lambda)$

```
  Initialize table $\mathcal{T} = (S, \Sigma)$, with $S = \{x\}$ and $\Sigma = \{\top\}$
⋆ Check if $\mathcal{T} = (S, \Sigma)$ is closed
  if $Closed(\mathcal{T})$ then
      Given $\mathcal{T}$ closed
      $Conjecture$: $\gamma_{S,\Sigma} \colon |S|_\Sigma \to T|S|_\Sigma$
      $|x|_\Sigma \mapsto (T|\text{-}|_\Sigma)(\gamma(x))$
      if $Conjecture == \bot$ then
          Provide $\varphi \in \mathcal{F}(\Lambda)$ s.t. $x \vDash \varphi$ and $|x| \nvDash \varphi$
          Update $\Sigma \leftarrow \Sigma \cup \mathrm{Sub}(\varphi)$
          return $\mathcal{T}$
          Go to ⋆
      else
          return $Aut(\mathcal{T})$
  else
      Given $\mathcal{T}$ not closed
      Pick $x \in S$ s.t. $|\mathrm{Base}(\gamma(x))|_\Sigma \nsubseteq |S|_\Sigma$
      while $\mathrm{Base}(\gamma(x)) \neq \emptyset$ do
          Take $y \in \mathrm{Base}(\gamma(x))$
          if $|y| \in |S|_\Sigma$ then
              $\mathrm{Base}(\gamma(x)) \leftarrow \mathrm{Base}(\gamma(x)) \setminus \{y\}$
          else
              Update $S \leftarrow S \cup \{y\}$
              $\mathrm{Base}(\gamma(x)) \leftarrow \mathrm{Base}(\gamma(x)) \setminus \{y\}$
      return $\mathcal{T}$
      Go to ⋆
```

3.3 Termination and Correctness

In this section we are going to prove termination and correctness of our L^{co} algorithm. We first state the theorem and sketch its proof. After that we will provide the proofs for the necessary technical lemmas.

Theorem 3. *Let (X, γ, x) be a pointed T-coalgebra and suppose (X, γ, x) is behaviourally equivalent to a finite well-pointed T-coalgebra (Y, δ, y). Let $\mathcal{F}(\Lambda)$ be an expressive language for T-coalgebras. The L^{co} algorithm with teacher (X, γ, x) and test language $\mathcal{F}(\Lambda)$ terminates and returns the correct well-pointed coalgebra.*

Proof. That the algorithm terminates can be seen as follows:

- The algorithm builds tables (S, Σ) where S is a subset of the carrier of the smallest subcoalgebra of (X, γ, x) that contains x. Therefore the size of S is bound by the number of elements of Y (Lemmas 4 and 5).
- Whenever a table is not closed, the algorithm will be able to close it. Whenever the algorithm turns a table into a closed one, the size of S strictly increases (Lemma 6).

– Whenever the teacher provides a counterexample the resulting table will not be closed (Lemma 7).

Collectively these claims show that the teacher eventually will accept the conjecture that the algorithm produces. Correctness of the algorithm then follows: Let (S, Σ) be a closed & sharp table. By Lemma 3 we have that the (S, Σ)-filtration $(|S|_\Sigma, \gamma_{S,\Sigma}, |x|_\Sigma)$ of (X, γ, x) is simple. To see that it is reachable consider the following diagram

$$\begin{tikzcd}
TS \arrow[r, "\mathrm{Base}^T_S"] \arrow[d, "T|\text{-}|_\Sigma"'] & \mathcal{P}_\omega S \arrow[d, "\mathcal{P}_\omega|\text{-}|_\Sigma"] \\
T|S|_\Sigma \arrow[r, "\mathrm{Base}^T_{|S|_\Sigma}"'] & \mathcal{P}_\omega|S|_\Sigma
\end{tikzcd}$$

where $|\text{-}|_\Sigma$ is the mapping to equivalence classes restricted to $S \subseteq X$. Clearly $|\text{-}|_\Sigma$ is mono (due to sharpness of (S, Σ)) and thus we know from [7, Remark 2.49] that the square commutes (and even forms a pullback). Whenever the algorithm adds a state x' to S we know that $x' \in \mathrm{Base}(\gamma(x''))$ for some $x'' \in S$ that has been already added at an earlier stage. Commutation of the diagram implies that $|x'|_\Sigma \in \mathrm{Base}(T|\text{-}|_\Sigma(\gamma(x''))) = \mathrm{Base}(\gamma_{S,\Sigma}(|x''|_\Sigma))$ where the equality is a consequence of the definition of the filtration. A routine induction argument together with [7, Lemma 3.16] reducing reachability of the coalgebra to reachability of its canonical graph can now be used to show that the (S, Σ)-filtration is reachable and hence well-pointed. If, in addition, the teacher accepts the (S, Σ)-filtration of (X, γ, x) we have that $(X, \gamma, x) \models \varphi$ iff $(|S|_\Sigma, \gamma_{S,\Sigma}, |x|_\Sigma) \models \varphi$ for all formulas $\varphi \in \mathcal{F}(\Lambda)$. By expressivity of $\mathcal{F}(\Lambda)$ this means (X, γ, x) is behaviourally equivalent to $(|S|_\Sigma, \gamma_{S,\Sigma}, |x|_\Sigma)$, i.e. we have learned the correct well-pointed coalgebra.

3.4 Termination Lemmas

In this section we prove the claims from the proof of Theorem 3. In the following let (X, γ, x) be a pointed T-coalgebra that is behaviourally equivalent to a finite well-pointed coalgebra (Y, δ, y). We start by proving that states occurring in tables are always taken from the reachable part of (X, γ, x).

Lemma 4. *Let (S, Σ) be a table that is obtained in a run of the L^{co} algorithm with teacher (X, γ, x). Then S is contained in the subcoalgebra of (X, γ) that is generated by x.*

Proof. Let $(X', \gamma_{\restriction X'}, x)$ be the smallest pointed subcoalgebra of (X, γ, x). Clearly we have $\mathrm{Base}(\gamma(x')) \subseteq X'$ for all $x' \in X$ as the coalgebra map γ restricts to a map $X' \to TX'$. Now we can easily prove the claim on the number n of steps the L^{co} algorithm was closing a table to reach the table (S, Σ). For $n = 0$ we have $S = \{x\}$ and obviously $x \in X'$. For $n = m + 1$ there is a table (S', Σ) such that S is obtained from S by closing the table (S', Σ). By I.H. we know that $S' \subseteq X'$

and by the definition of the algorithm we have $S \subseteq S' \cup \bigcup_{x' \in S'} \mathrm{Base}(\gamma(x'))$. But as we saw at the beginning of our proof the latter is a subset of X' which shows that $S \subseteq X'$ as required.

Consequently, we obtain an upper bound for the number of elements of S for each table S.

Lemma 5. *Let $(S, \varSigma)$ be a table computed by the L^{co} algorithm with teacher (X, γ, x). Then $\#S \leq \#Y$.*

Proof. By Lemma 1 we know that (Y, δ, y) is the quotient of the smallest pointed subcoalgebra (X', γ', x) of (X, γ, x) and by the previous lemma we have $S \subseteq X'$. The quotient map from (X', γ') to (Y, δ) is a coalgebra morphism and preserves the truth of modal formulas. Therefore, for any element of $x' \in S \subseteq X'$ there exists a $y \in Y$ such that $x' \equiv y$, in other words there is a function $f : S \to Y$ such that $x \equiv f(x)$. We also know that $(S, \varSigma)$ is sharp which means that $x_1 \equiv y$ and $x_2 \equiv y$ implies $x_1 = x_2$. Therefore the function $f : S \to Y$ has to be injective which shows that $\#S \leq \#Y$ as required.

This finishes the argument for why there is a finite upper bound on the number of states stored in a table. Let us now turn to what happens in case the current table is not closed.

Lemma 6. *Let $(S, \varSigma)$ be a table computed by the L^{co} algorithm with teacher (X, γ, x). If $(S, \varSigma)$ is not closed then L^{co} computes a closed table $(S', \varSigma)$ such that $\#S < \#S'$.*

Proof. Suppose $(S, \varSigma)$ is a table that is not closed. While $(S, \varSigma)$ is not closed, the L^{co} algorithm extends the collection of states S by elements of X as described in the previous section. By Lemma 5 this can only happen finitely often which implies that after finitely many additions to S the table will be closed as required.

Finally we need to investigate what happens in case the teacher is providing a counterexample.

Lemma 7. *Let $(S, \varSigma)$ be a closed and sharp table computed by the L^{co} algorithm with teacher (X, γ, x). If the teacher provides a counterexample φ, then $(S, \varSigma')$ is not closed where $\varSigma'$ is the smallest set of formulas that contains $\varSigma \cup \{\varphi\}$ and that is closed under subformulas.*

Proof. Suppose for a contradiction that $(S, \varSigma')$ is closed (obviously it will be sharp). For every set $Z \subseteq X$ let $f_Z : |Z|_{\varSigma'} \to |Z|_{\varSigma}$ given by $f_Z(|x'|_{\varSigma'}) := |x'|_{\varSigma}$ for all $x' \in Z$. Note this is well-defined as $\varSigma' \supseteq \varSigma$ and thus $x_1 \equiv_{\varSigma'} x_2$ implies $x_1 \equiv_{\varSigma} x_2$. We now claim that f_S is a pointed T-coalgebra morphism from the $(S, \varSigma')$-filtration of (X, γ, x) to its $(S, \varSigma)$ filtration. To check this we first note that $f_X \circ |\_|_{\varSigma'} = |\_|_{\varSigma}$. Let $x' \in S$ be arbitrary. We calculate:

$$\begin{aligned} \gamma_{S,\varSigma}(f_S(|x'|_{\varSigma'})) &= \gamma_{S,\varSigma}(|x'|_{\varSigma}) = (T|\_|_{\varSigma})(\gamma(x')) = T(f_X \circ |\_|_{\varSigma'})(\gamma(x')) \\ &= Tf_X(T|\_|_{\varSigma'}(\gamma(x'))) \overset{T|\_|_{\varSigma'}(\gamma(x')) \,\in\, T|S|_{\varSigma'}}{=} Tf_S(T|\_|_{\varSigma'}(\gamma(x'))) \end{aligned}$$

which shows that f_S is indeed a pointed T-coalgebra morphism between the filtrations. This is, however, a contradiction to Σ' containing a counterexample, i.e. a formula $\varphi \in \mathcal{F}(\Lambda)$ such that w.l.o.g. $|x|_{\Sigma'} \models \varphi$ and $|x|_{\Sigma} \not\models \varphi$.

This finishes the proofs of the claims that are necessary to prove Theorem 3. In summary, we have proved termination and correctness of our algorithm. Remarkably, if we measure the complexity of the algorithm in the number of logical and equivalence queries, our algorithm meets similar bounds as Angluin's L^*-algorithm. The main difference is that the run-time of our algorithm also depends on the maximal number of successors a state of the original coalgebra has - in the case of finite automata the branching is bounded by a constant, namely the size of the input alphabet. We need the following definitions for our complexity considerations:

- The *size of a formula* is the number of its distinct subformulas.
- A T-coalgebra (X, γ) is *k-branching* if there exists some $k \in \omega$ such that for all $x \in X$ we have $\#\text{Base}(\gamma(x)) \leq k$.

Proposition 1. *Let (X, γ, x) be a k-branching pointed T-coalgebra, let n be the size of the behavioural equivalent well-pointed T-coalgebra that the algorithm learns and let m be the maximal size of the counterexample provided by the Teacher. Then the algorithm terminates after asking $\mathcal{O}(k \cdot m \cdot n^2)$ logical queries and at most n equivalence queries.*

Proof. Let (S, Σ) be a table occurring during the run of the algorithm. By Lemma 5 we have $\#S \leq n$. Furthermore, the size of Σ is bound by $m \cdot (n-1)$ as each counterexample is of size at most m and at most $n-1$ counterexamples can be added to Σ as the addition of a counterexample always results in an increase of the size of S by Lemmas 6 and 7. To check closedness of a table (S, Σ) we need to ask logical queries about the elements of S and their successors. Therefore, we need at most $(k+1) \cdot n \cdot m \cdot (n-1)$ such queries. This shows that we need $\mathcal{O}(k \cdot m \cdot n^2)$ logical queries. The upper bound on equivalence queries is obvious.

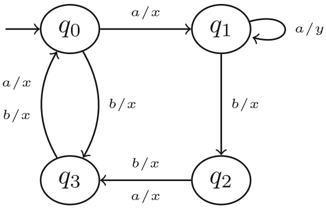
Our discussion demonstrates that the move from deterministic finite automata to T-coalgebras does not essentially alter the number of queries. At this stage we cannot predict how fast queries can be answered in practice. To answer this question we plan to implement our algorithm in the near future.

4 Examples

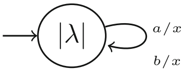
We will illustrate our algorithm with two examples: One known example on learning Mealy machines and one example that is to the best of our knowledge new, an Angluin learning algorithm for discrete Markov chains. While learning probabilistic automata is an active area [20–22], existing work uses other learning paradigms and focuses on constructing minimal automata for a probabilistic language rather than - as our algorithm does - on *bisimulation* quotients of transition systems.

Example 4 (Mealy machines). In the first example we show how the algorithm learns functions $L\colon I^+ \to O$, i.e. maps from finite sequences over I to elements of O where I and O are finite sets that constitute the input and output alphabet, respectively. The "regularity" assumption on L is that L can be represented by a finite Mealy machine. The latter are a generalization of deterministic automata where each transition has an associated input and output letter. Such Mealy machines correspond to coalgebras for the Mealy functor $T = (\_ \times O)^I$ [15]. Learning L is equivalent to learning a finite representation of the coalgebra $I^* \xrightarrow{\gamma} (I^* \times O)^I$ with designated point $\lambda \in I^*$, where $\gamma(w)(a) = \langle wa, L(wa)\rangle$. Recall the expressive language from Example 3 for the Mealy functor.

As a concrete example, we want to learn the function L represented by the Mealy machine in the following diagram, where the input alphabet is $I = \{a, b\}$ and the output alphabet is $O = \{x, y\}$:



According to the Algorithm 1, the first thing to do is checking if the starting table $\mathcal{T} = (\{\lambda\}, \{\top\})$ is closed. The table is trivially closed, as $\top$ is always true and the first conjecture of the algorithm is as follows:



As the conjecture is incorrect the teacher returns a counterexample, e.g. the formula $[a]p_{a/y}$. We update the set Σ of formulas with this new formula together with all its subformulas. Note that this process is a well-known variant of the one described in [6] where the counterexample is usually added to the set S of states and not to Σ. We use this variant that allows to automatically maintain the table consistent (cf. [19]), in order to be sure to have a sharp table at every step.

The table is now $\mathcal{T} = (\{\lambda\}, \{\top, p_{a/y}, [a]p_{a/y}\})$ and it is not closed. Therefore, we add another state to the set of states according to the **else** part of the **if**-statement 1. We pick a state $x' \in |S|_\Sigma$ such that its successors' equivalence classes are not represented in S, in symbols: $|\mathrm{Base}(\gamma(x'))|_\Sigma \not\subseteq |S|_\Sigma$. In our particular case, we can pick only λ; the two successors of λ are $\langle a, x\rangle$ and $\langle b, x\rangle$ and we have $\mathrm{Base}(\gamma(\lambda)) = \{a, b\}$. We take one element of this set, say a, the equivalence class of a is not equal to $|\lambda|_\Sigma$, therefore we add a to S and we remove it from $\mathrm{Base}(\gamma(\lambda))$. We do the same for b. Because $|b|_\Sigma \notin |\{\lambda, a\}|_\Sigma$, b is also added to S.

This procedure can be seen as the standard table filling process in the L*-algorithm: in the lower part of the table, we identify those states that do not have a corresponding representative in the set of states S and we add them in the upper part of the table; in our case both a and b have to be added. The entries of the table are 1 and 0, according to the truth values which the formula assumes in a particular state. We can schematically see this process in Table 1, where the values in the first column are shorthands for $\pi_1(\gamma(w)(a))$, i.e., we write wa for the state reached from w after reading input a.

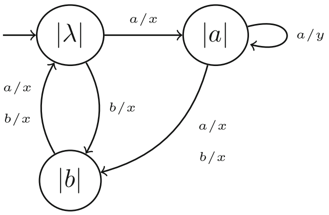
Table 1. From a not closed table to a closed one

	$\top$	$p_{a/y}$	$[a]p_{a/y}$
λ	1	0	1
a	1	1	1
b	1	0	0

$\Rightarrow$

	$\top$	$p_{a/y}$	$[a]p_{a/y}$
λ	1	0	1
a	1	1	1
b	1	0	0
aa	1	1	1
ab	1	0	0
ba	1	0	1
bb	1	0	1

Having added a and b to S, we have a new table: $\mathcal{T} = (\{\lambda, a, b\}, \{\top, p_{a/y}, [a]p_{a/y}\})$, that is closed. Now, we can make our conjecture:



But the conjecture is incorrect and we again receive a counterexample, e.g. $[a][b][b][a]p_{a/x}$. The resulting table is not closed, as $|\text{Base}(\gamma(a))|_\Sigma \not\subseteq |S|_\Sigma$. Indeed, the successor of a, aa, should satisfy $[a][b][b][a]p_{a/x}$, whereas the formula is false in a. We only pick the element aa from $\text{Base}(\gamma(a)) = \{aa, ab\}$ as the equivalence class of ab is already represented. The new table is: $\mathcal{T} = (\{\lambda, a, b, aa\}, \Sigma)$.

Closing the table with aa, we go to **else** 2 and we compute our conjecture. Table 2 on page 16 represents this step. No counterexamples are given back, therefore the conjecture is the right one and we have the same automaton $Aut(\mathcal{T})$ describing the Mealy machine we wanted to learn.

Note that there is a difference with the standard Angluin algorithm adapted for Mealy machines. Usually the entries of the tables are the output values that the Mealy automata produce. This is not the case in our algorithm. The entries are always the values 1 or 0 according to the truth values of the formulas, whereas the output values are given by the function γ.

Example 5 (Discrete Markov Chain). In this example we show that our algorithm can learn bisimulation quotients of (discrete) probabilistic transition systems. We confine ourselves to a finite example and we model probabilistic transition systems as $\mathcal{P}\mathsf{At} \times \mathcal{D}_\omega$-coalgebras where At is a set of propositional variables - in our example we will assume $\mathsf{At} = \{p\}$ for a single proposition p. Note, however, that the

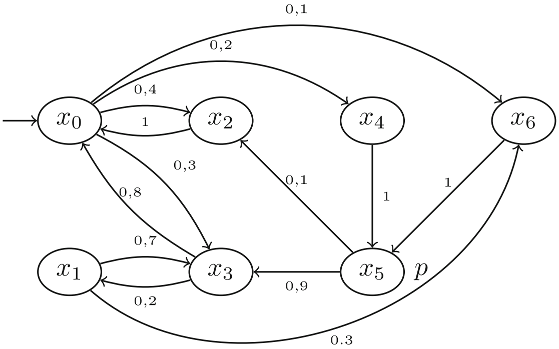
Table 2. Changing of the table according to the Base-step of the algorithm

	$\top$	$p_{a/y}$	$[a]p_{a/y}$	$p_{a/x}$	$[a]p_{a/x}$	$[b][a]p_{a/x}$	$[b][b][a]p_{a/x}$	$[a][b][b][a]p_{a/x}$
λ	1	0	1	1	0	1	0	1
a	1	1	1	0	0	1	1	1
b	1	0	0	1	1	0	1	0
aa	1	1	1	0	0	1	1	1
ab	1	0	0	1	1	1	0	1
ba	1	0	1	1	0	1	0	1
bb	1	0	1	1	0	1	0	1

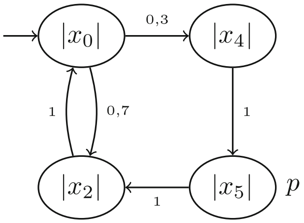
⇓

	$\top$	$p_{a/y}$	$[a]p_{a/y}$	$p_{a/x}$	$[a]p_{a/x}$	$[b][a]p_{a/x}$	$[b][b][a]p_{a/x}$	$[a][b][b][a]p_{a/x}$
λ	1	0	1	1	0	1	0	1
a	1	1	1	0	0	1	1	1
b	1	0	0	1	1	0	1	0
aa	1	1	1	0	0	1	1	1
ab	1	0	0	1	1	1	0	1
ba	1	0	1	1	0	1	0	1
bb	1	0	1	1	0	1	0	1
aaa	1	1	1	0	0	1	1	1
aab	1	1	1	0	0	1	1	1

input model of our learning algorithm could be infinite and that our coalgebraic framework is general enough to cover a large variety of probabilistic systems [23]. Recall an expressive language for this example from Example 3. Consider now the following pointed $\mathcal{P}\mathsf{At} \times \mathcal{D}_{\omega\_}$-coalgebra based on $X = \{x_0, x_1, \ldots, x_6\}$ with initial state x_0 and transition map γ, where transitions are labelled with their likelihoods and proposition p holds exclusively at state x_5:



The algorithm tries to learn a minimal pointed coalgebra that is behaviourally equivalent to (X, γ, x_0). As always the algorithm starts with table $(\{x_0\}, \{\top\})$. This table is trivially closed and our first conjecture will be a state at which p does not hold and with a loop of probability 1. The teacher has to reject the conjecture and provides a counter example, e.g. the formula $\varphi = L_{0.2}L_1 p$ as $x_0 \models \varphi$ and $|x_0| \not\models \varphi$. This leads to the new table $(\{x_0\}, \Sigma_1)$ with $\Sigma_1 = \{L_{0.2}L_1 p, L_1 p, p, \top\})$. We have $\mathrm{Base}(\gamma(x_0)) = \{x_2, x_3, x_4, x_6\}$ and it is easy to see that $|\{x_2, x_3, x_4, x_6\}|_{\Sigma_1} \not\subseteq |\{x_0\}|_{\Sigma_1}$. Therefore the table is not closed. It can easily be checked that $x_2 \equiv_{\Sigma_1} x_3$ and $x_4 \equiv_{\Sigma_1} x_6$ and closing the table leads to $(\{x_0, x_2, x_4\}, \Sigma_1)$. We have $\mathrm{Base}(\gamma(x_4)) = \{x_5\}$ and obviously $|x_5|_{\Sigma_1} \notin |\{x_0, x_2, x_4\}|_{\Sigma_1}$ as x_5 is the only state satisfying proposition p. Closing the table we arrive at $(\{x_0, x_2, x_4, x_5\}, \Sigma_1)$ and this table is closed as readers can easily convince themselves. The table leads to the correct conjecture:



5 Conclusions

While the last example was simple it demonstrates how our algorithm can be used to determine the quotient of a discrete Markov chain modulo behavioural equivalence, i.e. bisimilarity. Other transition systems can be quotiented in the same way. This is interesting as bisimulation quotients play an important role in verification [24] and as the complexity of standard algorithms is determined by the size of the system before taking the quotient whereas the complexity of the learning algorithm depends on the size of the potentially much smaller quotient. To explore the practical usefulness of our algorithm in these cases we are planning to provide an implementation in the near future. The biggest challenge will in our view be to implement a teacher that provides "good" counterexamples.

There are many more questions concerning our work that we would like to clarify: Our approach is currently very much based on the category of sets. This excludes for example the important example of linear weighted automata [2]. We believe that our arguments can be extended by (i) building on a duality between algebras and coalgebras and (ii) understanding how filtrations translate via this duality. There are some partial answers to the latter question in the modal logic literature (cf. e.g. [25,26]) but modal logicians usually focus on properties of filtrations that do not play a role in learning. Closely related to this question is a more diagrammatic understanding of termination and correctness of our algorithm: we believe that all essential ingredients for this are contained in our work but we need to understand filtrations on a more abstract, categorical level.

Acknowledgements. The authors would like to thank Nick Bezhanishvili and Alexandra Silva for helpful discussions and pointers to the literature.

References

1. Jacobs, B.: Introduction to Coalgebra: Towards Mathematics of States and Observation. Cambridge Tracts in TCS. Cambridge University Press, New York (2016)
2. Jacobs, B., Silva, A.: Automata learning: a categorical perspective. In: van Breugel, F., Kashefi, E., Palamidessi, C., Rutten, J. (eds.) Horizons of the Mind. A Tribute to Prakash Panangaden. LNCS, vol. 8464, pp. 384–406. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-06880-0_20
3. van Heerdt, G.: An abstract automata learning framework. Master's thesis, Radboud Universiteit Nijmegen (2016)
4. Moerman, J., Sammartino, M., Silva, A., Klin, B., Szynwelski, M.: Learning nominal automata. In: POPL 2017 (2017)
5. van Heerdt, G., Sammartino, M., Silva, A.: Learning automata with side-effects. CoRR abs/1704.08055 (2017)
6. Angluin, D.: Learning regular sets from queries and counter examples. Inf. Comput. **75**(2), 87–106 (1987)
7. Adámek, J., Milius, S., Moss, L.S., Sousa, L.: Well-pointed coalgebras. Logical Methods Comput. Sci. **9**(3) (2013)
8. Grädel, E., Thomas, W., Wilke, T. (eds.): Automata Logics, and Infinite Games. LNCS, vol. 2500. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-36387-4
9. Mac Lane, S.: Categories for the Working Mathematician. Graduate Texts in Mathematics, vol. 5. Springer, New York (1971). https://doi.org/10.1007/978-1-4757-4721-8
10. Jacobs, B., Rutten., J.: An introduction to (co)algebras and (co)induction. In: Advanced Topics in Bisimulation and Coinduction. Cambridge Tracts in Theoretical Computer Science, vol. 5, pp. 38–99. Cambridge University Press (2011)
11. Adámek, J., Trnková, V.: Automata and Algebras in Categories. Kluwer Academic Publishers, Dordrecht (1990)
12. Cirstea, C., Kurz, A., Pattinson, D., Schröder, L., Venema, Y.: Modal logics are coalgebraic. Comput. J. **54**(1), 31–41 (2009)
13. Kupke, C., Pattinson, D.: Coalgebraic semantics of modal logics: an overview. Theoret. Comput. Sci. **412**(38), 5070–5094 (2011)
14. Schröder, L.: Expressivity of coalgebraic modal logic: the limits and beyond. Theoret. Comput. Sci. **390**(2), 230–247 (2008)
15. Hansen, H.H., Rutten, J.J.M.M.: Symbolic synthesis of mealy machines from arithmetic bitstream functions. Sci. Ann. Comp. Sci. **20**, 97–130 (2010)
16. Desharnais, J., Edalat, A., Panangaden, P.: Bisimulation for labelled markov processes. Inf. Comput. **179**(2), 163–193 (2002)
17. Kupke, C., Leal, R.A.: Characterising behavioural equivalence: three sides of one coin. In: Kurz, A., Lenisa, M., Tarlecki, A. (eds.) CALCO 2009. LNCS, vol. 5728, pp. 97–112. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03741-2_8
18. Blackburn, P., de Rijke, M., Venema, Y.: Modal Logic. Cambridge Tracts in Theoretical Computer Science, vol. 53. Cambridge University Press, New York (2001)
19. Maler, O., Pnueli, A.: On the learnability of infinitary regular sets. Inf. Comput. **118**(2), 316–326 (1995)

20. Balle, B., Castro, J., Gavald, R.: Learning probabilistic automata: a study in state distinguishability. TCS **473**, 46–60 (2013)
21. Mao, H., Chen, Y., Jaeger, M., Nielsen, T.D., Larsen, K.G., Nielsen, B.: Learning probabilistic automata for model checking. In: 2011 Eighth International Conference on Quantitative Evaluation of Systems, pp. 111–120 (2011)
22. Tzeng, W.G.: Learning probabilistic automata and markov chains via queries. Mach. Learn. **8**(2), 151–166 (1992)
23. Sokolova, A.: Probabilistic systems coalgebraically: a survey. TCS **412**(38), 5095–5110 (2011)
24. Glück, R., Möller, B., Sintzoff, M.: Model refinement using bisimulation quotients. In: Johnson, M., Pavlovic, D. (eds.) AMAST 2010. LNCS, vol. 6486, pp. 76–91. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-17796-5_5
25. Ghilardi, S.: Continuity, freeness, and filtrations. J. Appl. Non Class. Logics **20**(3), 193–217 (2010)
26. Bezhanishvili, G., Bezhanishvili, N., Iemhoff, R.: Stable canonical rules. J. Symbol. Logic **81**(1), 284–315 (2016)

A Universal Algebra for the Variable-Free Fragment of RC^{∇}

Lev D. Beklemishev(✉)

Steklov Mathematical Institute of the Russian Academy of Sciences, Moscow, Russia
bekl@mi.ras.ru

Abstract. The language of Reflection Calculus RC consists of implications between formulas built up from propositional variables and the constant 'true' using only conjunction and the diamond modalities which are interpreted in Peano arithmetic as restricted uniform reflection principles. In [6] we introduced RC^{∇}, an extension of RC by a series of modalities representing the operators associating with a given arithmetical theory T its fragment axiomatized by all theorems of T of arithmetical complexity Π^0_n, for all $n > 0$. In this paper we continue the study of the variable-free fragment of RC^{∇} and characterize its Lindenbaum–Tarski algebra in several natural ways.

Keywords: Strictly positive logics · Reflection principle · Provability GLP

1 Introduction

A system, called *Reflection Calculus* and denoted RC, was introduced in [9] and, in the present format, in [5]. From the point of view of modal logic, RC can be seen as a fragment of Japaridze's polymodal provability logic GLP [8,16,23] consisting of the implications of the form $A \to B$, where A and B are formulas built-up from $\top$ and propositional variables using just $\wedge$ and the diamond modalities. We call such formulas A and B *strictly positive* (s.p.).

Strictly positive modal logics earlier appeared in two other contexts: in the work on description logic, on the one hand, and in universal algebra, as a study of the varieties of semilattices with monotone operators, on the other (see e.g. [15,24]). The recent paper [19] and its predecessors [18,20] is a systematic study of general s.p. logics and contains comprehensive references.

The main advantage of the use of reflection calculus in provability logic is its simplicity combined with sufficient expressivity for main proof-theoretic applications (see [1,3–5]). Unlike GLP, the system RC is complete with respect to a natural class of finite Kripke frames [9]. Also, RC is decidable in polynomial

L. D. Beklemishev—This work is supported by the Russian Science Foundation under grant 16–11–10252.

S. Artemov and A. Nerode (Eds.): LFCS 2018, LNCS 10703, pp. 91–106, 2018.
https://doi.org/10.1007/978-3-319-72056-2_6

time [9], whereas most of the standard modal logics are PSpace-complete and the same holds even for the variable-free fragment of GLP [21].

A system RC^{∇} with modalities $\Diamond_n$ representing uniform reflection principles of arithmetical complexity Σ^0_n, and ∇_n representing Π^0_{n+1}-conservativity operators was introduced in [6]. Provability interpretation of RC^{∇} is formulated in terms of the semilattice $\mathfrak{G}_{\mathrm{EA}}$ of (numerated) arithmetical r.e. theories extending elementary arithmetic EA. Under this semantics, propositional variables are interpreted as the elements of the lattice (i.e., theories extending EA modulo provable deductive equivalence), operators $\Diamond_n$ associate with a theory $T \in \mathfrak{G}_{\mathrm{EA}}$ the extension of EA by the uniform Σ_n-reflection principle for T, whereas the operators ∇_n associate with a theory T its Π^0_{n+1}-fragment, i.e., the extension of EA by all Π^0_{n+1}-theorems of T. We refer to [6] for a more comprehensive discussion and motivation.

The following results on RC^{∇} were established in [6]. Firstly, RC^{∇} can express α-iterations of modalities $\Diamond_n$, for each $n < \omega$ and ordinals $\alpha < \varepsilon_0$. A variable-free s.p. logic where such iterations are explicitly present in the language has been introduced by Joosten and Hermo Reyes [11,12] which is, thereby, contained in RC^{∇}. Secondly, unique normal forms for the formulas of the variable-free fragment of the system RC^{∇} were provided. As a corollary, this fragment was shown to be decidable and arithmetically complete [6].

In this paper we consider algebraic semantics for the variable-free fragment of RC^{∇}. We characterize its Lindenbaum–Tarski algebra in several natural ways. It turns out that this structure is tightly related to the so-called *Ignatiev frame* $\mathcal{I}$ for modal logic GLP [14], a Kripke frame whose points are certain sequences of ordinals below ε_0. We define on the domain of $\mathcal{I}$ the structure of a lower semilattice $\mathfrak{I}$ with the appropriate operations $\Diamond_n$ and ∇_n, for each $n < \omega$. This algebra is shown to be isomorphic to the Lindenbaum–Tarski algebra of the variable-free fragment of RC^{∇}; moreover, every sequence in $\mathfrak{I}$ corresponds to a variable-free formula of RC^{∇} in the normal form.

Another characterization of the same algebra is obtained by identifying the points of $\mathcal{I}$ with bounded variable-free RC-theories. Via the isomorphic embedding of $\mathfrak{I}$ into $\mathfrak{G}_{\mathrm{EA}}$, such RC-theories, in turn, correspond to natural arithmetical theories related to Turing–Feferman progressions of iterated reflection principles up to the ordinals below ε_0.

The points of $\mathfrak{I}$ have a natural proof-theoretic interpretation in terms of sequences of proof-theoretic ordinals of (bounded) arithmetical theories for each complexity level Π^0_{n+1}. Such collections, called *conservativity spectra*, appeared for the first time in the work of Joost Joosten [17]. He established a one-to-one correspondence between conservativity spectra (for a certain class of theories) and the points of Ignatiev's model. Our results show that conservativity spectra can be naturally seen as the points of the RC^{∇}-algebra $\mathfrak{I}$.

2 Strictly Positive Logics and Reflection Calculi

2.1 Normal Strictly Positive Logics

Consider a modal language $\mathcal{L}_{\Sigma}$ with propositional variables $p, q, \ldots$, a constant $\top$, conjunction $\wedge$, and a possibly infinite set of symbols $\Sigma = \{a_i : i \in J\}$ understood as diamond modalities. The family Σ is called the *signature* of the language $\mathcal{L}_{\Sigma}$. Strictly positive formulas (or simply *formulas*) are built up by the grammar:

$$A ::= p \mid \top \mid (A \wedge A) \mid aA, \quad \text{where } a \in \Sigma.$$

Sequents are expressions of the form $A \vdash B$ where A, B are strictly positive formulas.

Basic sequent-style system, denoted K$^+$, is given by the following axioms and rules:

1. $A \vdash A$; $\quad A \vdash \top$; $\quad$ from $A \vdash B$ and $B \vdash C$ infer $A \vdash C$;
2. $A \wedge B \vdash A$; $\quad A \wedge B \vdash B$; $\quad$ from $A \vdash B$ and $A \vdash C$ infer $A \vdash B \wedge C$;
3. from $A \vdash B$ infer $aA \vdash aB$, for each $a \in \Sigma$.

It is well-known that K$^+$ axiomatizes the strictly positive fragment of a polymodal version of basic modal logic K. All our systems will also contain the following principle corresponding to the transitivity axiom in modal logic:

4. $aaA \vdash aA$.

The extension of K$^+$ by this axiom will be denoted K4$^+$.

Let $C[A/p]$ denote the result of replacing in C all occurrences of a variable p by A. A set of sequents L is called a *normal strictly positive logic* if it contains the axioms and is closed under the rules of K$^+$ and under the following *substitution rule*: if $(A \vdash B) \in L$ then $(A[C/p] \vdash B[C/p]) \in L$. We will only consider normal strictly positive logics below. We write $A \vdash_L B$ for the statement that $A \vdash B$ is provable in L (or belongs to L). $A =_L B$ means $A \vdash_L B$ and $B \vdash_L A$.

2.2 Algebraic Semantics

Algebraic semantics for normal strictly positive logics is given by *semilattices with monotone operators* (SLOs), that is, structures of the form $\mathfrak{M} = (M; \wedge^{\mathfrak{M}}, \{a^{\mathfrak{M}} : a \in \Sigma\})$ where $(M, \wedge^{\mathfrak{M}})$ is a semilattice with top and each $a^{\mathfrak{M}} : M \to M$ is a monotone operator on $\mathfrak{M}$: $x \leqslant y$ implies $a^{\mathfrak{M}}(x) \leqslant a^{\mathfrak{M}}(y)$, for all $x, y \in M$. Every strictly positive formula A of $\mathcal{L}_{\Sigma}$ represents a term $A^{\mathfrak{M}}$ of $\mathfrak{M}$. We say that $A \vdash B$ *holds in* $\mathfrak{M}$ (or $\mathfrak{M}$ *satisfies* $A \vdash B$) if $\mathfrak{M} \vDash \forall \boldsymbol{x}\, A^{\mathfrak{M}}(\boldsymbol{x}) \leqslant B^{\mathfrak{M}}(\boldsymbol{x})$. It is easy to see that $A \vdash_{\mathrm{K}^+} B$ if and only if $A \vdash B$ holds in each SLO $\mathfrak{M}$. The SLOs satisfying all the theorems of a normal s.p. logic L are called *L-algebras.*

Given a normal s.p. logic L in a signature Σ and an alphabet of variables V, its *Lindenbaum–Tarski algebra* is a SLO $\mathfrak{L}_L^V$ whose domain consists of the equivalence classes of formulas of the language of L modulo $=_L$. Let $[A]_L$ denote the equivalence class of A. The operations are defined in a standard way as follows: $[A]_L \wedge^{\mathfrak{L}} [B]_L := [A \wedge B]_L$, $a^{\mathfrak{L}}([A]_L) := [aA]_L$, for each $a \in \Sigma$. It is well-known that $A \vdash_L B$ iff $A \vdash B$ holds in $\mathfrak{L}_L^V$. Hence, any normal s.p. logic L is complete w.r.t. its algebraic semantics, that is, w.r.t. the class of all L-algebras.

The algebra $\mathfrak{L}_L^V$ is also called the *free V-generated L-algebra*. In this paper we will be particularly interested in the algebras $\mathfrak{L}_L^V$ where V is empty. In this case we denote the algebra $\mathfrak{L}_L^V$ by $\mathfrak{L}_L^0$.

2.3 The System RC

Reflection calculus. RC is a normal strictly positive logic formulated in the signature $\{\Diamond_n : n \in \omega\}$. It is obtained by adjoining to the axioms and rules of K4$^+$ (stated for each $\Diamond_n$) the following principles:

5. $\Diamond_n A \vdash \Diamond_m A$, for all $n > m$;
6. $\Diamond_n A \wedge \Diamond_m B \vdash \Diamond_n(A \wedge \Diamond_m B)$, for all $n > m$.

We notice that the converse of Axiom 6 is also provable in RC, so that in fact

$$\Diamond_n(A \wedge \Diamond_m B) =_{\mathrm{RC}} \Diamond_n A \wedge \Diamond_m B. \tag{1}$$

Dashkov [9] showed that RC axiomatizes the set of all sequents $A \vdash B$ such that the implication $A \to B$ is provable in the polymodal logic GLP.

We recall a correspondence between variable-free RC-formulas and ordinals [2]. Let $\mathbb{F}$ denote the set of all variable-free RC-formulas, and let $\mathbb{F}_n$ denote its restriction to the signature $\{\Diamond_i : i \geqslant n\}$, so that $\mathbb{F} = \mathbb{F}_0$. For each $n \in \omega$ we define binary relations $<_n$ on $\mathbb{F}$ by

$$B <_n A \overset{\text{def}}{\iff} A \vdash_{\mathrm{RC}} \Diamond_n B.$$

Obviously, $<_n$ is a transitive relation invariantly defined on the equivalence classes w.r.t. provable equivalence in RC (denoted $=_{\mathrm{RC}}$). Since RC is polytime decidable, so are both $=_{\mathrm{RC}}$ and all of $<_n$.

An RC-formula without variables and $\wedge$ is called a *word*. In fact, any such formula syntactically is a finite sequence of letters $\Diamond_i$ (followed by $\top$). If A, B are words then AB will denote $A[B/\top]$, that is, the word corresponding to the concatenation of these sequences. $A \doteq B$ denotes the graphical identity of formulas (words). The set of all words will be denoted $\mathbb{W}$, and $\mathbb{W}_n$ will denote its restriction to the signature $\{\Diamond_i : i \geqslant n\}$. The following facts are from [2,5]:

- Every $A \in \mathbb{F}_n$ is RC-equivalent to a word in $\mathbb{W}_n$;
- $(\mathbb{W}_n/{=_{\mathrm{RC}}}, <_n)$ is isomorphic to $(\varepsilon_0, <)$.

Here, ε_0 is the first ordinal α such that $\omega^\alpha = \alpha$. Thus, the set $\mathbb{W}_n/{=_{\mathrm{RC}}}$ is well-ordered by the relation $<_n$. The isomorphism can be established by an onto and order preserving function $o_n : \mathbb{W}_n \to \varepsilon_0$ such that, for all $A, B \in \mathbb{W}_n$,

$$A =_{\mathrm{RC}} B \iff o_n(A) = o_n(B).$$

Then $o_n(A)$ is the order type of $\{B \in \mathbb{W}_n : B <_n A\}/{=_{\mathrm{RC}}}$.

The function $o(A) := o_0(A)$ can be inductively calculated as follows: If $A \mathrel{\mathring{=}} \diamond_0^k \top$ then $o(A) = k$. If $A \mathrel{\mathring{=}} A_1 \diamond_0 A_2 \diamond_0 \cdots \diamond_0 A_n$, where all $A_i \in \mathbb{W}_1$ and not all of them are empty, then

$$o(A) = \omega^{o(A_n^-)} + \cdots + \omega^{o(A_1^-)}.$$

Here, B^- is obtained from $B \in \mathbb{W}_1$ replacing every $\diamond_{m+1}$ by $\diamond_m$. For $n > 0$ and $A \in \mathbb{W}_n$ we let $o_n(A) = o_{n-1}(A^-)$.

2.4 The System $\mathbf{RC}^{\nabla}$ and Fat Normal Forms

The signature of RC^{∇} consists of modalities $\diamond_n$ and ∇_n, for each $n < \omega$. The system RC^{∇} is a normal s.p. logic given by the following axioms and rules, for all $m, n < \omega$:

1. RC for $\diamond_n$; RC for ∇_n;
2. $A \vdash \nabla_n A$;
3. $\diamond_n A \vdash \nabla_n A$;
4. $\diamond_m \nabla_n A \vdash \diamond_m A$; $\nabla_n \diamond_m A \vdash \diamond_m A$ if $m \leqslant n$.

A formula A is called *ordered* if no modality (be it $\diamond_i$ or ∇_i) occurs in A within the scope of a modality with a strictly larger index. The following lemma is easy [6].

Lemma 1. *Every formula A of* RC^{∇} *is equivalent to an ordered one.*

Ordered formulas equivalent to a given formula need not be unique. Normal forms for the variable-free fragment of RC^{∇} were introduced in [6]. Let $\mathbb{F}^{\nabla}$ denote the set of variable-free formulas of RC^{∇}.

A formula $A \in \mathbb{F}^{\nabla}$ is in the *fat normal form* if either $A \mathrel{\mathring{=}} \top$ or A has the form $\nabla_0 A_0 \wedge \nabla_1 A_1 \wedge \cdots \wedge \nabla_k A_k$, where for all $i = 0, \ldots, k$, $A_i \in \mathbb{W}_i$, $A_k \not\mathrel{\mathring{=}} \top$, and

$$\nabla_i A_i \vdash_{\mathrm{RC}^{\nabla}} \nabla_i(\nabla_i A_i \wedge \cdots \wedge \nabla_k A_k).$$

One of the main results in [6] is the following proposition.

Proposition 1

(i) *Every $A \in \mathbb{F}^{\nabla}$ is equivalent to a formula in the fat normal form.*
(ii) *For all $A \in \mathbb{F}^{\nabla}$, the words A_i in the fat normal form of A are unique modulo equivalence in* RC*.*

A corollary of this theorem is that the variable-free fragment of RC^{∇} is decidable and arithmetically complete [6].

3 Ignatiev Frame and Ignatiev RC^∇-Algebra

In this and the following section we characterize in several ways the Lindenbaum–Tarski algebra of the variable-free fragment of RC^∇. It turns out that this structure is tightly related to the so-called *Ignatiev's* Kripke frame. This frame, denoted here $\mathcal{I}$, has been introduced by Konstantin Ignatiev [14] as a universal frame for the variable-free fragment of Japaridze's logic GLP. Later this frame has been slightly modified and studied in more detail in [7,13]. In particular, Thomas Icard established a detailed relationship between $\mathcal{I}$ and the canonical frame for the variable-free fragment of GLP and used it to define a complete topological semantics for this fragment. David Fernández and Joost Joosten [10] generalized $\mathcal{I}$ to a version of GLP with transfinitely many modalities. Ignatiev's frame is defined constructively ('coordinatewise') as follows.

Let $\bar{\mathrm{I}}$ denote the set of all ω-sequences of ordinals $\boldsymbol{\alpha} = (\alpha_0, \alpha_1, \dots)$ such that $\alpha_i \leqslant \varepsilon_0$ and $\alpha_{i+1} \leqslant \ell(\alpha_i)$, for all $i \in \omega$. Here, the function ℓ is defined by: $\ell(\beta) = 0$ if $\beta = 0$, and $\ell(\beta) = \gamma$ if $\beta = \delta + \omega^\gamma$, for some δ, γ. Thus, all sequences of $\bar{\mathrm{I}}$, with the exception of identically ε_0, are eventually zero. Relations R_n on $\bar{\mathrm{I}}$ are defined by:

$$\boldsymbol{\alpha} R_n \boldsymbol{\beta} \iff (\forall i < n\, \alpha_i = \beta_i \text{ and } \alpha_n > \beta_n).$$

The structure $\overline{\mathcal{I}} = (\bar{\mathrm{I}}, (R_n)_{n \in \omega})$ is called *the extended Ignatiev frame* (see [13]). The Ignatiev frame is its restriction to the subset I of all sequences $\boldsymbol{\alpha} \in \bar{\mathrm{I}}$ such that $\forall i \in \omega\, \alpha_i < \varepsilon_0$. This subset is upwards closed w.r.t. all relations R_n, hence the evaluation of the variable-free RC-formulas (and GLP-formulas) in $\mathcal{I}$ and in $\overline{\mathcal{I}}$ coincide. We denote by $\mathcal{I}, \boldsymbol{\alpha} \Vdash \varphi$ the truth of a GLP-formula φ at a node $\boldsymbol{\alpha}$ of $\mathcal{I}$. The following important theorem is a corollary of the results of Ignatiev but, in fact, has an easier direct proof (which we omit for the reasons of brevity).

Proposition 2. *For any variable-free formulas A, B of* RC*, $A \vdash_{\mathrm{RC}} B$ iff $\mathcal{I}, \boldsymbol{\alpha} \Vdash A \to B$, for all $\boldsymbol{\alpha} \in \mathrm{I}$.*

The set of sequences $\boldsymbol{\alpha} \in \mathrm{I}$ such that $\forall i < \omega\, \alpha_{i+1} = \ell(\alpha_i)$ is called *the main axis* of $\mathcal{I}$ and is denoted O. Obviously, a sequence in O is uniquely determined by its initial element α_0, hence O naturally corresponds to the ordinals up to ε_0. We can also associate with every word $A \in \mathbb{W}$ an element $\iota(A) \in \mathrm{O}$ by letting

$$\iota(A) := (o(A), \ell(o(A)), \dots, \ell^{(n)}(o(A)), \dots).$$

The following lemma, explicitly stated by Thomas Icard [13, Lemma 3.8], describes all the subsets of $\overline{\mathcal{I}}$ definable by words (and hence by all variable-free s.p. formulas of RC).

Lemma 2. *Suppose $A \in \mathbb{W}$ and $\boldsymbol{\alpha} = \iota(A)$. Then, for all $\boldsymbol{\beta} \in \overline{\mathcal{I}}$, $\overline{\mathcal{I}}, \boldsymbol{\beta} \Vdash A$ iff $\forall i \in \omega\, \alpha_i \leqslant \beta_i$.*

Our goal is to transform $\mathcal{I}$ into an RC^∇-algebra $\mathfrak{I}$ with the same domain I, that is, into an SLO satisfying RC^∇. We consider the set $\bar{\mathrm{I}}$ equipped with the ordering

$$\boldsymbol{\alpha} \leqslant_{\mathfrak{I}} \boldsymbol{\beta} \overset{\text{def}}{\iff} \forall n \in \omega\, \alpha_n \geqslant \beta_n.$$

The structure $(\bar{\mathrm{I}}, \leqslant_{\mathfrak{I}})$ can be seen as a subordering of the product ordering on the set of all ω-sequences of ordinals $\leqslant \varepsilon_0$, which we denote $\mathcal{E}$.

A *cone in* $\mathcal{E}$ is the set of points $E_{\boldsymbol{\alpha}} := \{\boldsymbol{\beta} \in \mathcal{E} : \boldsymbol{\beta} \leqslant_{\mathfrak{I}} \boldsymbol{\alpha}\}$, for some $\boldsymbol{\alpha} \in \mathcal{E}$. A sequence $\boldsymbol{\alpha} \in \mathcal{E}$ is called *bounded* if $\forall i \in \omega\ \alpha_i < \varepsilon_0$ and $\alpha_i \neq 0$ for only finitely many $i \in \omega$. Obviously, each $\boldsymbol{\alpha} \in \mathrm{I}$ is bounded.

Lemma 3. *Suppose $\boldsymbol{\alpha} \in \mathcal{E}$ is bounded. Then $E_{\boldsymbol{\alpha}} \cap \mathrm{I}$ is not empty and has a greatest point $\boldsymbol{\beta}$ w.r.t. $\leqslant_{\mathfrak{I}}$.*

Proof. Let $n \in \omega$ be the largest number such that $\alpha_n \neq 0$. Consider the sequence $\boldsymbol{\beta}$ such that $\beta_i = 0$ for all $i > n$, $\beta_n := \alpha_n$, and, for all $i < n$:

$$\beta_i := \begin{cases} \alpha_i, & \text{if } \ell(\alpha_i) \geqslant \beta_{i+1}, \\ \alpha_i + \omega^{\beta_{i+1}}, & \text{otherwise.} \end{cases}$$

It is easy to see that $\boldsymbol{\beta}$ is the greatest point of $E_{\boldsymbol{\alpha}} \cap \mathrm{I}$. Also notice that $\boldsymbol{\beta}$ can be effectively computed from $\boldsymbol{\alpha}$.

Corollary 1. $(\mathrm{I}, \leqslant_{\mathfrak{I}})$ *is a meet-semilattice with top.*

Proof. Let $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathrm{I}$. The sequence $\boldsymbol{\gamma} := (\max(\alpha_i, \beta_i))_{i<\omega}$ is the g.l.b. of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ in $\mathcal{E}$ and is bounded. By Lemma 3, $E_{\boldsymbol{\gamma}} \cap \mathrm{I}$ has a greatest point, which has to be the g.l.b. of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ in I.

We denote by $\wedge_{\mathfrak{I}}$ the meet operation of this semilattice. A nonempty set $C_\alpha := E_{\boldsymbol{\alpha}} \cap \mathrm{I}$ is called a *cone in* $\mathcal{I}$. The set of all cones in $\mathcal{I}$ ordered by inclusion is denoted $\mathfrak{C}(\mathcal{I})$. The orderings $(\mathfrak{C}(\mathcal{I}), \subseteq)$ and $(\mathrm{I}, \leqslant_{\mathfrak{I}})$ are isomorphic by the map $\boldsymbol{\alpha} \mapsto C_{\boldsymbol{\alpha}}$. So, we have

Corollary 2. *For all $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathrm{I}$, $C_{\boldsymbol{\alpha} \wedge_{\mathfrak{I}} \boldsymbol{\beta}} = C_{\boldsymbol{\alpha}} \cap C_{\boldsymbol{\beta}}$.*

Let C(O) denote the set $\{C_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \mathrm{O}\}$ of all cones in $\mathcal{I}$ generated by the points of the main axis. For all $X \subseteq \bar{\mathrm{I}}$ define $R_n^{-1}(X) := \{y \in X : xR_n y\}$. We claim that the operations $\cap$ and R_n^{-1} map cones of C(O) to cones of C(O). Moreover, the following proposition holds.[1]

Proposition 3. *The algebra $\mathfrak{C}(\mathrm{O}) = (\mathrm{C}(\mathrm{O}); \cap, \{R_n^{-1} : n \in \omega\})$ is isomorphic to the Lindenbaum–Tarski algebra $\mathfrak{L}^0_{\mathrm{RC}}$.*

Proof. Let $v : \mathbb{F} \to \mathcal{P}(\mathrm{I})$ denote the map associating with every variable-free formula A of RC the set of points $v(A) := \{\boldsymbol{\alpha} \in \mathrm{I} : \mathcal{I}, \boldsymbol{\alpha} \Vdash A\}$. By the soundness and completeness of RC w.r.t. the Ignatiev model we have $v(A) = v(B)$ iff $A =_{\mathrm{RC}} B$. Moreover, by Lemma 2 the range of v consists of all the cones of C(O). So, v factors to a bijective map $\bar{v} : \mathfrak{L}^0_{\mathrm{RC}} \to \mathrm{C}(\mathrm{O})$. The operations $\cap$ and R_n^{-1} correspond to the definition of truth in a Kripke model, hence C(O) is closed under these operations and $\bar{v}$ is an isomorphism of the respective algebras.

[1] We do not distinguish notationally an operation on a set and its restriction to a subset.

We remark that the work of Pakhomov [22] shows that the elementary theory of the algebra $\mathfrak{L}^0_{\mathrm{RC}}$ is undecidable. We now define the structure of an RC^∇-algebra on I.

Definition 1. For all $n \in \omega$ we define the functions $\nabla^{\mathfrak{I}}_n, \diamond^{\mathfrak{I}}_n : \mathrm{I} \to \mathrm{I}$. For each element $\boldsymbol{\alpha} = (\alpha_0, \alpha_1, \dots, \alpha_n, \dots) \in \mathrm{I}$ let:

$\nabla^{\mathfrak{I}}_n(\boldsymbol{\alpha}) := (\alpha_0, \alpha_1, \dots, \alpha_n, 0, \dots)$;

$\diamond^{\mathfrak{I}}_n(\boldsymbol{\alpha}) := (\beta_0, \beta_1, \dots, \beta_n, 0, \dots)$, where $\beta_{n+1} := 0$ and $\beta_i := \alpha_i + \omega^{\beta_{i+1}}$, for all $i \leqslant n$.

The algebra $\mathfrak{I} = (\mathrm{I}, \wedge_{\mathfrak{I}}, \{\diamond^{\mathfrak{I}}_n, \nabla^{\mathfrak{I}}_n : n \in \omega\})$ is called the *Ignatiev* RC^∇*-algebra.*

The definition of the operations $\diamond^{\mathfrak{I}}_n$ is motivated by the following lemma and its corollary.

Lemma 4. *Suppose* $\boldsymbol{\alpha} \in \mathrm{I}$ *and* $\boldsymbol{\beta} = \diamond^{\mathfrak{I}}_n(\boldsymbol{\alpha})$*. Then* $\boldsymbol{\beta} \in \mathrm{O}$ *and*

(i) $C_{\boldsymbol{\beta}} = \bigcap_{i \leqslant n} R_i^{-1}(C_{\boldsymbol{\alpha}})$*;*
(ii) *If* $\boldsymbol{\alpha} \in \mathrm{O}$ *then* $C_{\boldsymbol{\beta}} = R_n^{-1}(C_{\boldsymbol{\alpha}})$*.*

Proof. (i) It is easy to see that each of the sets $R_i^{-1}(C_{\boldsymbol{\alpha}})$, for $i \leqslant n$, is a cone in $\mathcal{I}$ generated by the bounded sequence $(\alpha_0, \dots, \alpha_{i-1}, \alpha_i + 1, 0, \dots)$ from $\mathcal{E}$. Hence, the intersection of these cones is a cone generated by $(\alpha_0 + 1, \dots, \alpha_{n-1} + 1, \alpha_n + 1, 0, \dots)$. Its greatest element in I obviously coincides with $\diamond^{\mathfrak{I}}_n(\boldsymbol{\alpha})$.

(ii) Clearly, $\boldsymbol{\beta} \in R_n^{-1}(C_{\boldsymbol{\alpha}})$, since $\boldsymbol{\beta}' := (\beta_0, \beta_1, \dots, \beta_{n-1}, \alpha_n, \alpha_{n+1}, \dots)$ satisfies $\boldsymbol{\beta} R_n \boldsymbol{\beta}'$ and $\boldsymbol{\beta}' \leqslant_{\mathfrak{I}} \boldsymbol{\alpha}$. In the opposite direction, show by downward induction on $i \leqslant n$ that if $\boldsymbol{\gamma} \in R_n^{-1}(C_{\boldsymbol{\alpha}})$ then $\gamma_i \geqslant \beta_i$. For $i = n$ the claim is obvious. Assume $i < n$, then $\gamma_i \geqslant \alpha_i$. Since $\ell(\gamma_i) \geqslant \gamma_{i+1} \geqslant \beta_{i+1}$ and $\ell(\alpha_i) = \alpha_{i+1} < \beta_{i+1}$, we must also have $\gamma_i \geqslant \alpha_i + \omega^{\beta_{i+1}} = \beta_i$.

Corollary 3. $\mathfrak{C}(\mathrm{O})$ *is isomorphic to the algebra* $\mathfrak{O} = (\mathrm{O}, \wedge_{\mathfrak{I}}, \{\diamond^{\mathfrak{I}}_n : n \in \omega\})$*.*

Proof. Consider the bijection $c : \boldsymbol{\alpha} \longmapsto C_{\boldsymbol{\alpha}}$ from O to C(O). By Corollary 2 this map preserves the meet, and by Lemma 4 it preserves the diamond modalities.

We summarize the previous results in the following theorem characterizing the Lindenbaum–Tarski algebra of the variable-free fragment of RC.

Theorem 1. *The algebras* $\mathfrak{L}^0_{\mathrm{RC}}$*,* $\mathfrak{C}(\mathrm{O})$*,* $\mathfrak{O}$ *are naturally isomorphic by the following maps:*

(i) $\bar{v} : \mathfrak{L}^0_{\mathrm{RC}} \to \mathfrak{C}(\mathrm{O})$*;*
(ii) $c : \mathfrak{O} \to \mathfrak{C}(\mathrm{O})$*;*
(iii) $\bar{\iota} : \mathfrak{L}^0_{\mathrm{RC}} \to \mathfrak{O}$*.*

Here, for any $A \in \mathbb{F}$, $\bar{\iota}([A]_{\mathrm{RC}}) := \iota(A')$, where $A' \in \mathbb{W}$ is a word such that $A =_{\mathrm{RC}} A'$. This definition is invariant, since, for any words A', A'', if $A' =_{\mathrm{RC}} A''$ then $o(A') = o(A'')$ and hence $\iota(A') = \iota(A'')$. For a proof that (iii) is an isomorphism it is sufficient to remark that $v(A) = c(\iota(A))$, for each $A \in \mathbb{W}$, by Lemma 2.

Our next goal is to show that $\mathfrak{I}$ is isomorphic to the Lindenbaum–Tarski algebra of RC^∇. First, we need an auxiliary lemma.

Lemma 5. *For every* $\boldsymbol{\alpha} \in \mathrm{I}$ *and* $n \in \omega$*, there is an* $\boldsymbol{\alpha}' \in \mathrm{O}$ *such that* $\boldsymbol{\alpha}' \leqslant_{\mathfrak{I}} \boldsymbol{\alpha}$ *and* $\Diamond_n^{\mathfrak{I}}(\boldsymbol{\alpha}) = \Diamond_n^{\mathfrak{I}}(\boldsymbol{\alpha}')$*.*

Proof. Let $\alpha'_n := \alpha_n$, $\forall i \geqslant n\ \alpha'_{i+1} := \ell(\alpha'_i)$, and $\forall i < n\ \alpha'_i := \alpha_i + \omega^{\alpha'_{i+1}}$. It is easy to check that $\boldsymbol{\alpha}'$ is as required.

Let $A^{\mathfrak{I}}$ denote the value of a variable-free RC$^\nabla$-formula A in $\mathfrak{I}$. The following lemma shows that $\mathfrak{I}$ satisfies the variable-free fragment of RC$^\nabla$.

Lemma 6. *For any* $A, B \in \mathbb{F}^\nabla$*,* $A \vdash_{\mathrm{RC}^\nabla} B$ *implies* $A^{\mathfrak{I}} \leqslant_{\mathfrak{I}} B^{\mathfrak{I}}$*.*

Proof. We argue by induction on the length of RC$^\nabla$-derivation. In almost all the cases the proof is routine. We consider the nontrivial case of the axiom $\Diamond_n A \wedge \Diamond_m B \vdash \Diamond_n(A \wedge \Diamond_m B)$ for $m < n$. Let $\boldsymbol{\alpha} = A^{\mathfrak{I}}$ and $\boldsymbol{\beta} = B^{\mathfrak{I}}$. Using Lemma 5 we obtain $\boldsymbol{\alpha}', \boldsymbol{\beta}' \in \mathrm{O}$ such that $\boldsymbol{\alpha}' \leqslant_{\mathfrak{I}} \boldsymbol{\alpha}$, $\boldsymbol{\beta}' \leqslant_{\mathfrak{I}} \boldsymbol{\beta}$ and $\Diamond_n^{\mathfrak{I}}\boldsymbol{\alpha} = \Diamond_n^{\mathfrak{I}}\boldsymbol{\alpha}'$, $\Diamond_m^{\mathfrak{I}}\boldsymbol{\beta} = \Diamond_m^{\mathfrak{I}}\boldsymbol{\beta}'$. By Theorem 1 the algebra $\mathfrak{O}$ satisfies RC, hence

$$\Diamond_n^{\mathfrak{I}}\boldsymbol{\alpha}' \wedge_{\mathfrak{I}} \Diamond_m^{\mathfrak{I}}\boldsymbol{\beta}' \leqslant_{\mathfrak{I}} \Diamond_n^{\mathfrak{I}}(\boldsymbol{\alpha}' \wedge_{\mathfrak{I}} \Diamond_m^{\mathfrak{I}}\boldsymbol{\beta}').$$

Therefore, $\Diamond_n^{\mathfrak{I}}\boldsymbol{\alpha} \wedge_{\mathfrak{I}} \Diamond_m^{\mathfrak{I}}\boldsymbol{\beta} \leqslant_{\mathfrak{I}} \Diamond_n^{\mathfrak{I}}(\boldsymbol{\alpha}' \wedge_{\mathfrak{I}} \Diamond_m^{\mathfrak{I}}\boldsymbol{\beta}) \leqslant_{\mathfrak{I}} \Diamond_n^{\mathfrak{I}}(\boldsymbol{\alpha} \wedge_{\mathfrak{I}} \Diamond_m^{\mathfrak{I}}\boldsymbol{\beta})$. The second inequality holds by the monotonicity of $\wedge_{\mathfrak{I}}$ and $\Diamond_n^{\mathfrak{I}}$.

Lemma 7. *Suppose* $A \doteq \nabla_0 A_0 \wedge \nabla_1 A_1 \wedge \cdots \wedge \nabla_n A_n$ *is in the fat normal form. Then* $A^{\mathfrak{I}} = (o_0(A_0), o_1(A_1), \ldots, o_n(A_n), 0, \ldots)$*.*

Proof. Firstly, since each $A_i \in \mathbb{W}_i$ we obtain from Theorem 1 that

$$(A_i)^{\mathfrak{I}} = \iota(A_i) = (\omega_i(o_i(A_i)), \omega_{i-1}(o_i(A_i)), \ldots, \omega^{o_i(A_i)}, o_i(A_i), \ell(o_i(A_i)), \ldots),$$

where by definition $\omega_0(\alpha) = \alpha$ and $\omega_{k+1}(\alpha) = \omega^{\omega_k(\alpha)}$. Hence,

$$(\nabla_i A_i)^{\mathfrak{I}} = (\omega_i(o_i(A_i)), \omega_{i-1}(o_i(A_i)), \ldots, \omega^{o_i(A_i)}, o_i(A_i), 0, \ldots).$$

Denote $\overline{A_i} := \nabla_i A_i \wedge \nabla_{i+1} A_{i+1} \wedge \cdots \wedge \nabla_n A_n$. By downwards induction on $i \leqslant n$ we show that $(\overline{A_i})^{\mathfrak{I}}$ equals

$$(\omega_i(o_i(A_i)), \omega_{i-1}(o_i(A_i)), \ldots, o_i(A_i), o_{i+1}(A_{i+1}), \ldots, o_n(A_n), 0, \ldots). \quad (2)$$

For $i = n$ the claim follows from the above. Assume $i < n$ and that the claim holds for $i + 1$. Since in a fat normal form

$$\nabla_i A_i \vdash_{\mathrm{RC}^\nabla} \nabla_i(\nabla_i A_i \wedge \nabla_{i+1} A_{i+1}),$$

by Lemma 6 we obtain that the sequence $(\nabla_i A_i)^{\mathfrak{I}}$ coordinatewise majorizes the sequence $(\nabla_i(\nabla_i A_i \wedge \nabla_{i+1} A_{i+1}))^{\mathfrak{I}}$. The former has the ordinal $o_i(A_i)$ at i-th position, and the latter has at the same place the least ordinal α such that $\alpha \geqslant o_i(A_i), \omega^{o_{i+1}(A_{i+1})}$ and $\ell(\alpha) \geqslant o_{i+1}(A_{i+1})$. Therefore, $o_i(A_i) = \alpha$ and $\ell(o_i(A_i)) \geqslant o_{i+1}(A_{i+1})$.

Now consider the sequence $(\overline{A_i})^{\mathfrak{I}} = (\nabla_i A_i \wedge \overline{A_{i+1}})^{\mathfrak{I}}$. By the induction hypothesis its tail coincides with that of (2) starting from position $i + 1$. Since $\ell(o_i(A_i)) \geqslant o_{i+1}(A_{i+1})$, the ordinal $o_i(A_i)$ occurs in it on i-th position. Also, for each $k < i$ we have $\omega_k(o_i(A_i)) \geqslant \omega_k(\omega^{o_{i+1}(A_{i+1})})$. It follows that the sequence $(\overline{A_i})^{\mathfrak{I}}$ coincides with (2).

The following corollary will be useful later on.

Corollary 4. *For any $A, B \in \mathbb{W}$ and $n \in \omega$, if $\mathfrak{I} \vDash \nabla_n A = \nabla_n B$ then $A =_{\mathrm{RC}} B$.*

Proof. Firstly, we infer: $\mathfrak{I} \vDash \nabla_0 A = \nabla_0 \nabla_n A = \nabla_0 \nabla_n B = \nabla_0 B$. By Lemma 7 we conclude $o(A) = o(B)$, therefore $A =_{\mathrm{RC}} B$.

Theorem 2. *For all $A, B \in \mathbb{F}^{\nabla}$, $A \vdash_{\mathrm{RC}^{\nabla}} B$ iff $A^{\mathfrak{I}} \leqslant_{\mathfrak{I}} B^{\mathfrak{I}}$.*

Proof. We must only prove the 'only if' part. Moreover, it is sufficient to prove it for fat normal forms $A \overset{\circ}{=} \nabla_0 A_0 \wedge \nabla_1 A_1 \wedge \cdots \wedge \nabla_n A_n$ and $B \overset{\circ}{=} \nabla_0 B_0 \wedge \nabla_1 B_1 \wedge \cdots \wedge \nabla_m B_m$. If $A^{\mathfrak{I}} \leqslant_{\mathfrak{I}} B^{\mathfrak{I}}$ then by Lemma 7 we have $n \geqslant m$ and $o_i(A_i) \geqslant o_i(B_i)$, for each $i \leqslant m$. Since $A_i, B_i \in \mathbb{W}_i$, this means that $A_i \vdash_{\mathrm{RC}} \Diamond_i B_i$ or $A_i =_{\mathrm{RC}} B_i$. In either case we can infer $\nabla_i A_i \vdash_{\mathrm{RC}^{\nabla}} \nabla_i B_i$ for each $i \leqslant m$. It follows that $A \vdash_{\mathrm{RC}^{\nabla}} B$.

Theorem 2 essentially means the following.

Corollary 5. *The Ignatiev RC^{∇}-algebra $\mathfrak{I}$ is isomorphic to the Lindenbaum–Tarski algebra of the variable-free fragment of RC^{∇}.*

4 $\mathfrak{I}$ as the Algebra of Variable-Free RC-Theories

Another, perhaps even more natural, view of the Ignatiev RC^{∇}-algebra is via an interpretation of the points of $\mathcal{I}$ as variable-free RC-theories. It nicely agrees with the arithmetical interpretation in that we can also view such a theory as an arithmetical theory (every variable-free RC-formula corresponds to an arithmetical sentence). In this section we will presuppose that the language is variable-free and will only consider variable-free formulas and theories.

A set of strictly positive formulas T is called an RC-*theory* if $B \in T$ whenever there are $A_1, \ldots, A_n \in T$ such that $A_1 \wedge \cdots \wedge A_n \vdash_{\mathrm{RC}} B$. A theory T is called *improper* if T coincides with the set of all strictly positive formulas, otherwise it is called *proper*.[2] A theory is called *bounded* if there is a strictly positive formula A such that $T \subseteq \{B : A \vdash_{\mathrm{RC}} B\}$. We will use the following basic fact.

The set $\overline{\mathrm{I}}$ bears a natural topology generated as a subbase by the set of all cones in $\overline{\mathcal{I}}$ and their complements. By [13, Theorem 3.12], this topology coincides with the product topology of the space $\mathcal{E}$ induced on $\overline{\mathrm{I}}$. Obviously, for each RC-formula A, the set $v(A)$ is clopen. Moreover, this topology is compact and totally disconnected on $\overline{\mathrm{I}}$, since $\overline{\mathrm{I}}$ is closed in $\mathcal{E}$ and $\mathcal{E}$ is compact by Tychonoff theorem. As a corollary we obtain the following *strong completeness* result.

Proposition 4. *Let T be an RC-theory and A an RC-formula.*

(i) *$T \nvdash_{\mathrm{RC}} A$ iff there is an $\boldsymbol{\alpha} \in \overline{\mathcal{I}}$ such that $\overline{\mathcal{I}}, \boldsymbol{\alpha} \Vdash T$ and $\overline{\mathcal{I}}, \boldsymbol{\alpha} \nVdash A$;*
(ii) *If T is bounded then $T \nvdash_{\mathrm{RC}} A$ iff there is an $\boldsymbol{\alpha} \in \mathcal{I}$ such that $\mathcal{I}, \boldsymbol{\alpha} \Vdash T$ and $\mathcal{I}, \boldsymbol{\alpha} \nVdash A$.*

[2] We avoid the term 'consistent', for even the improper theory corresponds to a consistent set of arithmetical sentences.

Proof

(i) The nontrivial implication is from left to right. Assume $T \nvdash_{\mathrm{RC}} A$. There is an increasing sequence of finite theories $(T_n)_{n\in\omega}$ such that $T = \bigcup_{n\in\omega} T_n$. By the completeness of the variable-free fragment of RC w.r.t. $\mathcal{I}$ each of the sets $v(T_n) \setminus v(A)$ is nonempty and clopen. By the compactness of $\bar{\mathrm{I}}$ there is a point $\boldsymbol{\alpha} \in \bigcap_{n\in\omega} v(T_n) \setminus v(A) = v(T) \setminus v(A)$.

(ii) In case T is bounded we have $v(T) \supseteq v(B)$, for some word B. There is a bounded sequence $\boldsymbol{\beta} \in \mathcal{E}$ such that $v(T) = E_{\boldsymbol{\beta}} \cap \bar{\mathrm{I}}$: consider the pointwise supremum of the generating points of the cones $v(T_n)$ in $\overline{\mathcal{I}}$, each of which is pointwise majorized by the greatest element $B^{\mathfrak{I}}$ of $v(B)$. By Lemma 3, the set $v(T)$ has a greatest point, say $\boldsymbol{\gamma} \in \mathrm{I}$. Since $\boldsymbol{\alpha} \in v(T)$ we have $\boldsymbol{\alpha} \leqslant_{\mathfrak{I}} \boldsymbol{\gamma}$, hence $\mathcal{I}, \boldsymbol{\gamma} \nVdash A$.

For any RC-theories T, S define $T \leqslant_{\mathrm{RC}} S$ iff $T \supseteq S$. The g.l.b. of T and S in this ordering, denoted $T \wedge_{\mathrm{RC}} S$, is the theory generated by the union $T \cup S$. Thus, the set $\mathfrak{T}^0_{\mathrm{RC}}$ of all bounded variable-free RC-theories is a semilattice (it is, in fact, a lattice with $T \cap S$ the l.u.b. of T and S). The set $\{A \in \mathbb{F} : \top \vdash_{\mathrm{RC}} A\}$ corresponds to the top of this lattice and is denoted $\top_{\mathrm{RC}}$.

For each $\boldsymbol{\alpha} \in \overline{\mathcal{I}}$ define an RC-theory $[\boldsymbol{\alpha}] := \{A : \overline{\mathcal{I}}, \boldsymbol{\alpha} \Vdash A\}$. Clearly, $[\boldsymbol{\alpha}]$ is bounded if $\boldsymbol{\alpha} \in \mathrm{I}$. For each RC-theory T define $v(T) := \{\boldsymbol{\alpha} \in \mathrm{I} : \mathcal{I}, \boldsymbol{\alpha} \Vdash T\}$.

Proposition 5

(i) *The map* $\boldsymbol{\alpha} \mapsto [\boldsymbol{\alpha}]$ *is an isomorphism between* $(\mathcal{I}, \leqslant_{\mathfrak{I}})$ *and the ordered set* $\mathfrak{T}^0_{\mathrm{RC}}$ *of bounded* RC*-theories.*

(ii) *The map* v *is an isomorphism between* $\mathfrak{T}^0_{\mathrm{RC}}$ *and the ordered set* $(\mathrm{C}(\mathcal{I}), \subseteq)$ *of cones in* $\mathcal{I}$.

Proof. It is sufficient to prove that

(a) The maps $\boldsymbol{\alpha} \mapsto [\boldsymbol{\alpha}]$ and $T \mapsto v(T)$ are order-preserving;
(b) $\forall \boldsymbol{\alpha} \in \mathrm{I}\; v([\boldsymbol{\alpha}]) = C_{\boldsymbol{\alpha}}$;
(c) If $v(T) = C_{\boldsymbol{\alpha}}$ then $T = [\boldsymbol{\alpha}]$.

Item (a) is obvious. For (b) we observe:

$$\boldsymbol{\beta} \in v([\boldsymbol{\alpha}]) \iff \forall A\, (\mathcal{I}, \boldsymbol{\alpha} \Vdash A \Rightarrow \mathcal{I}, \boldsymbol{\beta} \Vdash A).$$

The right hand side is equivalent to $\boldsymbol{\beta} \leqslant_{\mathfrak{I}} \boldsymbol{\alpha}$: If $\boldsymbol{\beta} \leqslant_{\mathfrak{I}} \boldsymbol{\alpha}$ and $\mathcal{I}, \boldsymbol{\alpha} \Vdash A$ then $\mathcal{I}, \boldsymbol{\beta} \Vdash A$ by Proposition 2. If $\boldsymbol{\beta} \nleqslant_{\mathfrak{I}} \boldsymbol{\alpha}$ then there is a word A such that $\mathcal{I}, \boldsymbol{\alpha} \Vdash A$ and $\mathcal{I}, \boldsymbol{\beta} \nVdash A$, by [13, Corollary 3.9]. Hence, $\boldsymbol{\beta} \in v([\boldsymbol{\alpha}])$ iff $\boldsymbol{\beta} \in C_{\boldsymbol{\alpha}}$.

For (c) we use Proposition 4. Suppose $\boldsymbol{\alpha} \in \mathrm{I}$ and $v(T) = C_{\boldsymbol{\alpha}}$. Then $\mathcal{I}, \boldsymbol{\alpha} \Vdash T$ and thus $T \subseteq [\boldsymbol{\alpha}]$. For the opposite inclusion assume $A \in [\boldsymbol{\alpha}]$ and $A \notin T$. By Proposition 4 there is a node $\boldsymbol{\beta} \in \mathrm{I}$ such that $\mathcal{I}, \boldsymbol{\beta} \Vdash T$ and $\mathcal{I}, \boldsymbol{\beta} \nVdash A$. Thus, $\boldsymbol{\beta} \in v(T)$ and, since $v(A)$ is downwards persistent, $\boldsymbol{\beta} \nleqslant_{\mathfrak{I}} \boldsymbol{\alpha}$. It follows that $v(T) \nsubseteq C_{\boldsymbol{\alpha}}$.

The operations of the Ignatiev RC^{∇}-algebra can be interpreted in terms of the semilattice of bounded theories as follows. For each $T \in \mathfrak{T}^0_{\mathrm{RC}}$ let $\nabla^{\mathrm{RC}}_n T$ denote the RC-theory axiomatized by $\{\Diamond_m A : \Diamond_m A \in T \text{ and } m \leqslant n\}$.

Lemma 8. *For all $\boldsymbol{\alpha} \in \mathfrak{I}$, $\nabla^{\mathrm{RC}}_n([\boldsymbol{\alpha}]) = [\nabla^{\mathfrak{I}}_n \boldsymbol{\alpha}]$.*

Proof. For the inclusion ($\subseteq$) we need to show: if $m \leqslant n$ and $\Diamond_m A \in [\boldsymbol{\alpha}]$ then $\Diamond_m A \in [\nabla^{\mathfrak{I}}_n \boldsymbol{\alpha}]$. If $\Diamond_m A \in [\boldsymbol{\alpha}]$ then $\mathcal{I}, \boldsymbol{\alpha} \Vdash \Diamond_m A$, hence there is a $\boldsymbol{\beta}$ such that $\boldsymbol{\alpha} R_m \boldsymbol{\beta}$ and $\mathcal{I}, \boldsymbol{\beta} \Vdash A$. So, we have $\forall i < m\ \alpha_i = \beta_i$ and $\alpha_m > \beta_m$. Since $m \leqslant n$, the node $\nabla^{\mathfrak{I}}_n \boldsymbol{\alpha}$ has the same coordinates as $\boldsymbol{\alpha}$ for all $i \leqslant m$. Therefore, $(\nabla^{\mathfrak{I}}_n \boldsymbol{\alpha}) R_m \boldsymbol{\beta}$ and $\mathcal{I}, (\nabla^{\mathfrak{I}}_n \boldsymbol{\alpha}) \Vdash \Diamond_m A$.

For the inclusion ($\supseteq$) we consider any node $\boldsymbol{\gamma} \in \mathrm{I}$ such that $\mathcal{I}, \boldsymbol{\gamma} \Vdash \nabla^{\mathrm{RC}}_n [\boldsymbol{\alpha}]$ and show that $\mathcal{I}, \boldsymbol{\gamma} \Vdash [\nabla^{\mathfrak{I}}_n \boldsymbol{\alpha}]$. This means that $v(\nabla^{\mathrm{RC}}_n [\boldsymbol{\alpha}]) \subseteq v([\nabla^{\mathfrak{I}}_n \boldsymbol{\alpha}])$ and hence $\nabla^{\mathrm{RC}}_n([\boldsymbol{\alpha}]) \supseteq [\nabla^{\mathfrak{I}}_n \boldsymbol{\alpha}]$ by Proposition 5.

Assume $\mathcal{I}, \boldsymbol{\gamma} \nVdash [\nabla^{\mathfrak{I}}_n \boldsymbol{\alpha}]$. Since $v(\nabla^{\mathfrak{I}}_n \boldsymbol{\alpha}) = C_{\nabla^{\mathfrak{I}}_n \boldsymbol{\alpha}}$ we have $\boldsymbol{\gamma} \notin C_{\nabla^{\mathfrak{I}}_n \boldsymbol{\alpha}}$, hence there is an $m \leqslant n$ such that $\gamma_m < \alpha_m$. Consider a word $A \in \mathbb{W}_m$ such that $o_m(A) = \gamma_m$. Recall that that the point on the main axis corresponding to A is $\iota(A) = (\omega_m(\gamma_m), \dots, \omega^{\gamma_m}, \gamma_m, \ell(\gamma_m), \dots)$.

We claim that $\mathcal{I}, \boldsymbol{\gamma} \nVdash \Diamond_m A$, whereas $\mathcal{I}, \boldsymbol{\alpha} \Vdash \Diamond_m A$. The former holds, since for all $\boldsymbol{\delta}$ such that $\boldsymbol{\gamma} R_m \boldsymbol{\delta}$ one has $\delta_m < \gamma_m$, hence $\boldsymbol{\delta} \nleqslant_{\mathfrak{I}} \iota(A)$ and $\mathcal{I}, \boldsymbol{\delta} \nVdash A$. On the other hand, $\mathcal{I}, \boldsymbol{\alpha} \Vdash \Diamond_m A$ holds, since there is a sequence $\boldsymbol{\alpha}' := (\alpha_0, \dots, \alpha_{m-1}, \gamma_m, \gamma_{m+1}, \dots)$ such that $\boldsymbol{\alpha} R_m \boldsymbol{\alpha}'$ and $\mathcal{I}, \boldsymbol{\alpha}' \Vdash A$.

To show that $\boldsymbol{\alpha}' \leqslant_{\mathfrak{I}} \iota(A)$ we prove that $\forall i \leqslant m\, \omega_{m-i}(\gamma_m) \leqslant \alpha_i$ by downward induction on $i \leqslant m$. Assume the claim holds for some i such that $0 < i \leqslant m$. Then $\alpha_{i-1} \geqslant \omega^{\ell(\alpha_{i-1})} \geqslant \omega^{\alpha_i} \geqslant \omega^{\gamma_i} = \gamma_{i-1}$.

In order to define the operations $\Diamond^{\mathrm{RC}}_n$ on the set of bounded RC-theories we need a few definitions. An RC-theory T is of *level n* if T is generated by a (nonempty) set of formulas $\Diamond_n A$ such that $A \in \mathbb{W}_n$. A theory T is *of level at least n* if it is generated by a (nonempty) subset of $\mathbb{W}_n \setminus \{\top\}$.

Lemma 9. *Every bounded* RC*-theory T is representable in the form $T = T_0 \wedge_{\mathrm{RC}} T_1 \wedge_{\mathrm{RC}} \cdots \wedge_{\mathrm{RC}} T_n$ where each T_i is of level i or $T_i = \top_{\mathrm{RC}}$.*

Proof. Recall that every RC-formula is RC-equivalent to an ordered formula. Moreover, every variable-free RC-formula in which only the modalities $\Diamond_i$ with $i \geqslant m$ occur is equivalent to a word in $\mathbb{W}_m$. Hence, every formula is equivalent to a conjunction of formulas of the form $\Diamond_i A$ with $A \in \mathbb{W}_i$. Since T is bounded, the set of indices of modalities occurring in the axioms of T is bounded, say by n. Hence, each axiom of T can be replaced by a finite set of formulas of various levels below n and one can partition the union of all these axioms into the disjoint subsets of the same level.

Lemma 10. *For each $\boldsymbol{\alpha} \in \mathrm{I}$ such that $\alpha_n > 0$, the theory generated by $[\boldsymbol{\alpha}] \cap \mathbb{W}_n$ corresponds to the sequence $\boldsymbol{\alpha}' := (\omega_n(\alpha_n), \dots, \omega^{\alpha_n}, \alpha_n, \alpha_{n+1}, \dots)$.*

We remark that if $\alpha_n = 0$ then the theory generated by $[\boldsymbol{\alpha}] \cap \mathbb{W}_n$ is $\top_{\mathrm{RC}}$.

Proof. Let T be the theory generated by $[\boldsymbol{\alpha}] \cap \mathbb{W}_n$. We consider a $\boldsymbol{\beta} \in \mathrm{I}$ such that $[\boldsymbol{\beta}] = T$ and show that $\boldsymbol{\beta} = \boldsymbol{\alpha}'$. It is easy to see that $\boldsymbol{\alpha} \leqslant_{\mathfrak{I}} \boldsymbol{\alpha}'$ and that the submodel of $\mathcal{I}$ generated from $\boldsymbol{\alpha}$ by the relations R_k, for all $k \geqslant n$, is isomorphic to the submodel generated by these relations from $\boldsymbol{\alpha}'$. Hence, if B is a formula in which only the modalities $\diamond_k$ with $k \geqslant n$ occur, then $\mathcal{I}, \boldsymbol{\alpha} \Vdash B$ holds iff $\mathcal{I}, \boldsymbol{\alpha}' \Vdash B$. It follows that $[\boldsymbol{\alpha}] \cap \mathbb{W}_n \subseteq [\boldsymbol{\alpha}']$, that is, $\boldsymbol{\alpha}' \leqslant_{\mathfrak{I}} \boldsymbol{\beta}$.

Now assume $\boldsymbol{\alpha}' <_{\mathfrak{I}} \boldsymbol{\beta}$, so there is a $k \in \omega$ such that $\beta_k < \alpha'_k$. If $k < n$ then $\beta_k < \omega_{n-k}(\alpha_n)$. For all ordinals γ, δ, if $\gamma < \omega^{\delta}$ then $\ell(\gamma) < \delta$. Then, by induction, for all $i = k, \ldots, n$ we obtain $\beta_i < \omega_{n-i}(\alpha_n)$. Ergo $\beta_n < \alpha_n$.

So, we may assume that $k \geqslant n$. In this case consider a word $B \in \mathbb{W}_k$ such that $o_k(B) = \beta_k + 1$. Then,

$$\iota(B) = (\omega_k(\beta_k + 1), \ldots, \omega_1(\beta_k + 1), \beta_k + 1, 0, \ldots).$$

We have $\mathcal{I}, \boldsymbol{\beta} \nVdash B$, since $\beta_k + 1 > \beta_k$. On the other hand,

$$\forall i \leqslant k \, \omega_i(\beta_k + 1) \leqslant \alpha_{k-i},$$

which is easy to see by induction on i. It follows that $\mathcal{I}, \boldsymbol{\alpha} \Vdash B$, therefore $[\boldsymbol{\beta}] \neq T$, a contradiction.

Corollary 6. *For each $\boldsymbol{\alpha} \in \mathrm{I}$, $[\boldsymbol{\alpha}]$ is of level at least n iff $\alpha_n > 0$ and*

$$\forall i < n \, \alpha_i = \omega_{n-i}(\alpha_n). \tag{3}$$

Lemma 11. *For each bounded* RC*-theory T of level at least n, there is an* RC*-formula $A \in \mathbb{W}_n$ such that $\nabla_n^{\mathrm{RC}} A = \nabla_n^{\mathrm{RC}} T$ holds in $\mathfrak{T}^0_{\mathrm{RC}}$.*

Proof. Suppose $T = [\boldsymbol{\alpha}]$ is of level at least n. Let $A \in \mathbb{W}_n$ be such that $o_n(A) = \alpha_n > 0$. Then, by Lemma 8, $\nabla_n^{\mathrm{RC}}(T) = \nabla_n^{\mathrm{RC}}([\boldsymbol{\alpha}]) = [\nabla_n^{\mathfrak{I}} \boldsymbol{\alpha}]$. By (3) we have

$$\nabla_n^{\mathfrak{I}} \boldsymbol{\alpha} = (\omega_n(\alpha_n), \omega_{n-1}(\alpha_n), \ldots, \alpha_n, 0, \ldots).$$

On the other hand, $\iota(A) = (\omega_n(\alpha_n), \omega_{n-1}(\alpha_n), \ldots, \alpha_n, \ell(\alpha_n), \ldots)$, and we obtain $\nabla_n^{\mathrm{RC}} A = [\nabla_n^{\mathfrak{I}}(\iota(A))] = [(\omega_n(\alpha_n), \omega_{n-1}(\alpha_n), \ldots, \alpha_n, 0, \ldots)]$. Proposition 5 yields the result.

Now we can give the following definition of the theory $\diamond_n^{\mathrm{RC}} T$, for each bounded RC-theory T.

If T is of level at least n or $T = \top_{\mathrm{RC}}$, we let $\diamond_n^{\mathrm{RC}} T$ be the theory generated by the formula $\diamond_n A$, where $A \in \mathbb{W}_n$ is such that $\nabla_n^{\mathrm{RC}} A = \nabla_n^{\mathrm{RC}} T$ in $\mathfrak{T}^0_{\mathrm{RC}}$. (Notice that this definition is correct, since any two words A_1, A_2 satisfying $\nabla_n^{\mathrm{RC}} A_1 = \nabla_n^{\mathrm{RC}} A_2$ in $\mathfrak{T}^0_{\mathrm{RC}}$ also satisfy $\diamond_n A_1 =_{\mathrm{RC}} \diamond_n A_2$ by Corollary 4.)

For each $i \leqslant n$, let T_i denote the theory generated by $T \cap \mathbb{W}_i$. We define

$$\diamond_n^{\mathrm{RC}}(T) := \diamond_0^{\mathrm{RC}}(T_0) \wedge_{\mathrm{RC}} \diamond_1^{\mathrm{RC}}(T_1) \wedge_{\mathrm{RC}} \cdots \wedge_{\mathrm{RC}} \diamond_n^{\mathrm{RC}}(T_n).$$

The following lemma shows that this definition agrees with the operations on the Ignatiev algebra.

Lemma 12. *For all $\boldsymbol{\alpha} \in \mathfrak{I}$, $\Diamond_n^{\mathrm{RC}}([\boldsymbol{\alpha}]) = [\Diamond_n^{\mathfrak{I}}(\boldsymbol{\alpha})]$.*

Proof. If $T = [\boldsymbol{\alpha}]$ then by Lemma 10, for each $i \leqslant n$, either the theory $T_i := T \cap \mathbb{W}_i$ is $\top_{\mathrm{RC}}$ or corresponds to the sequence $\boldsymbol{\alpha}' := (\omega_i(\alpha_i), \dots, \omega^{\alpha_i}, \alpha_i, \alpha_{i+1}, \dots)$ with $\alpha_i > 0$. If $T_i = \top_{\mathrm{RC}}$ we have $\Diamond_i^{\mathrm{RC}} T_i = \Diamond_i \top$. Otherwise, $\Diamond_i^{\mathrm{RC}} T_i = \Diamond_i A_i$ where A_i corresponds to $(\omega_i(\alpha_i), \dots, \omega^{\alpha_i}, \alpha_i, \ell(\alpha_i), \dots)$. In both cases

$$\Diamond_i^{\mathrm{RC}} T_i = [(\omega_i(\alpha_i + 1), \dots, \omega^{\alpha_i+1}, \alpha_i + 1, 0, \dots)].$$

Then we observe that $\Diamond_n^{\mathrm{RC}}(T) = \Diamond_0^{\mathrm{RC}}(T_0) \wedge_{\mathrm{RC}} \Diamond_1^{\mathrm{RC}}(T_1) \wedge_{\mathrm{RC}} \cdots \wedge_{\mathrm{RC}} \Diamond_n^{\mathrm{RC}}(T_n)$ corresponds to the cone generated by $(\alpha_0 + 1, \alpha_1 + 1, \dots, \alpha_n + 1, 0, \dots)$ in $\mathcal{E}$ which coincides with the cone of $\Diamond_n^{\mathfrak{I}}(\boldsymbol{\alpha})$ (cf Lemma 4).

Using Lemma 4 we can also isomorphically represent $\mathfrak{I}$ as an algebra of cones in $\mathcal{I}$. Given a cone $C \in \mathrm{C}(\mathcal{I})$ let $\Diamond_n^{\mathfrak{C}}(C) := \bigcap_{i \leqslant n} R_i^{-1}(C)$. We also define

$$\nabla_n^{\mathfrak{C}}(C) := \bigcap \{R_i^{-1}(D) : D \in \mathfrak{C}(\mathcal{I}),\ i \leqslant n,\ R_i^{-1}(D) \supseteq C\}.$$

We summarize the information in the following theorem.

Theorem 3. *The following structures are isomorphic:*

(i) *The Lindenbaum–Tarski algebra of the variable-free fragment of* RC^{∇}*;*
(ii) $\mathfrak{I} = (\mathrm{I}, \wedge_{\mathfrak{I}}, \{\Diamond_n^{\mathfrak{I}}, \nabla_n^{\mathfrak{I}} : n \in \omega\})$*;*
(iii) $(\mathfrak{T}_{\mathrm{RC}}^0, \wedge_{\mathrm{RC}}, \{\Diamond_n^{\mathrm{RC}}, \nabla_n^{\mathrm{RC}} : n \in \omega\})$*;*
(iv) $\mathfrak{C}(\mathcal{I}) = (\mathrm{C}(\mathcal{I}), \cap, \{\Diamond_n^{\mathfrak{C}}, \nabla_n^{\mathfrak{C}} : n \in \omega\})$*.*

Proof. We only need to prove the isomorphism of (iv) with either (ii) or (iii). Proposition 5 provides the isomorphisms of the semilattice reducts. Further, for all $\boldsymbol{\alpha} \in \mathrm{I}$, $\Diamond_n^{\mathfrak{C}}(C_{\boldsymbol{\alpha}}) = C_{\Diamond_n^{\mathfrak{I}}(\boldsymbol{\alpha})}$ by Lemma 4(i). Hence, $\Diamond_n^{\mathfrak{C}}$ corresponds to $\Diamond_n^{\mathfrak{I}}$ of (ii). On the other hand, $\nabla_n^{\mathfrak{C}}(C_{\boldsymbol{\alpha}}) = v(\nabla_n^{\mathrm{RC}}([\boldsymbol{\alpha}]))$. Hence, $\nabla_n^{\mathfrak{C}}$ corresponds to ∇_n^{RC} of (iii).

We remark that the algebra $\mathfrak{C}(\mathcal{I})$ has rather simple definitions of meet and diamonds, but somewhat convoluted nablas. In contrast, $\mathfrak{T}_{\mathrm{RC}}^0$ has simple meet and nablas but somewhat convoluted diamonds. The algebra $\mathfrak{I}$, perhaps the most elegant of all three, has a more complicated meet operation (though the order relation $\leqslant_{\mathfrak{I}}$ is simple).

5 A Universal Kripke Frame for RC^{∇}

In view of Theorem 3 it is natural to ask if one can describe a universal Kripke frame for the variable-free fragment of RC^{∇}. There is a general construction associating with a SLO $\mathfrak{B} = (B, \wedge^{\mathfrak{B}}, \{a^{\mathfrak{B}} : a \in \Sigma\})$ its 'dual' Kripke frame, similar to the way the canonical model of an s.p. logic L is obtained from its Lindenbaum–Tarski algebra.

Recall that a *filter in* $\mathfrak{B}$ is a nonempty subset $F \subseteq B$ such that

1. If $x \leqslant_{\mathfrak{B}} y$ and $x \in F$ then $y \in F$;
2. If $x, y \in F$ then $x \wedge^{\mathfrak{B}} y \in F$.

The set of filters of $\mathfrak{B}$ will be denoted $\mathcal{F}(\mathfrak{B})$. On $\mathcal{F}(\mathfrak{B})$ one can define binary relations $\{R_a : a \in \Sigma\}$ as follows: For all $F, G \in \mathcal{F}(\mathfrak{B})$,

$$F R_a G \overset{\text{def}}{\iff} \forall x \in G\, a^{\mathfrak{B}}(x) \in F.$$

Let $\mathfrak{B}^*$ denote the Kripke frame $(\mathcal{F}(\mathfrak{B}), \{R_a : a \in \Sigma\})$ together with the canonical valuation $v : \mathfrak{B} \to \mathcal{P}(\mathcal{F}(\mathfrak{B}))$, where $v(x) := \{F \in \mathcal{F}(\mathfrak{B}) : x \in F\}$. It is then easy to see that, for all $x \in \mathfrak{B}$ and $a \in \Sigma$, $R_a^{-1}(v(x)) = v(a^{\mathfrak{B}}(x))$. Hence, we obtain the following corollaries.

Proposition 6

(i) *The map* $v : \mathfrak{B} \to \mathcal{P}(\mathfrak{B}^*)$ *is an embedding of* $\mathfrak{B}$ *into the algebra* $(\mathcal{P}(\mathfrak{B}^*), \cap, \{R_a^{-1} : a \in \Sigma\})$.
(ii) *If* A, B *in* $\mathcal{L}_\Sigma$ *are variable-free, then* $A \vdash B$ *holds in* $\mathfrak{B}$ *iff* $\mathfrak{B}^*, F \Vdash A \to B$ *for all* $F \in \mathfrak{B}^*$.

Corollary 7. $\mathfrak{I}^*$ *is complete for the variable-free fragment of* RC^{∇}.

It is possible to give a more explicit description of the set of filters in $\mathfrak{I}$ in terms of sequences of ordinals. With each filter F in $\mathfrak{I}$ we associate a sequence $\boldsymbol{\alpha}_F \in \mathcal{E}$ by letting $\alpha_i := \sup\{\beta_i + 1 : \boldsymbol{\beta} \in F\}$, for each $i \in \omega$.

Lemma 13. *For each filter* F, *the sequence* $\boldsymbol{\alpha} = \boldsymbol{\alpha}_F$ *satisfies the following condition: For all* $i \in \omega$, *either* α_i *is a limit ordinal and* $\alpha_{i+1} \leqslant \ell(\alpha_i)$, *or* $\alpha_i = \alpha_i' + 1$ *and* $\alpha_{i+1} \leqslant \ell(\alpha_i') + 1$, *for some* α_i'.

Vice versa, if a sequence $\boldsymbol{\alpha}$ satisfies the condition of Lemma 13, then the set $F_{\boldsymbol{\alpha}} := \{\boldsymbol{\beta} \in \mathrm{I} : \forall i \in \omega\ \alpha_i > \beta_i\}$ is a filter in $\mathfrak{I}$. However, at present we lack a characterization of the relations R_n and S_n on $\mathfrak{I}^*$ corresponding to the modalities $\Diamond_n$ and ∇_n by formulas that would be as nice as the definition of these operations in $\mathfrak{I}$ itself. So, we prefer not to go into the details here and to leave Lemma 13 without proof.

References

1. Beklemishev, L.D., Onoprienko, A.A.: On some slowly terminating term rewriting systems. Sbornik Math. **206**, 1173–1190 (2015)
2. Beklemishev, L.D.: Provability algebras and proof-theoretic ordinals. Ann. Pure Appl. Logic **128**, 103–123 (2004)
3. Beklemishev, L.D.: Reflection principles and provability algebras in formal arithmetic. Russ. Math. Surv. **60**(2), 197–268 (2005). Russian original: Uspekhi Matematicheskikh Nauk, 60(2): 3–78 (2005)
4. Beklemishev, L.D.: The Worm principle. In: Chatzidakis, Z., Koepke, P., Pohlers, W. (eds.) Logic Colloquium 2002. Lecture Notes in Logic, vol. 27, pp. 75–95. AK Peters (2006). Preprint: Logic Group Preprint Series 219, Utrecht University, March 2003

5. Beklemishev, L.D.: Calibrating provability logic: from modal logic to reflection calculus. In: Bolander, T., Braüner, T., Ghilardi, S., Moss, L. (eds.) Advances in Modal Logic, vol. 9, pp. 89–94. College Publications, London (2012)
6. Beklemishev, L.D.: On the reflection calculus with partial conservativity operators. In: Kennedy, J., de Queiroz, R.J.G.B. (eds.) WoLLIC 2017. LNCS, vol. 10388, pp. 48–67. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-662-55386-2_4
7. Beklemishev, L.D., Joosten, J., Vervoort, M.: A finitary treatment of the closed fragment of Japaridze's provability logic. J. Logic Comput. **15**(4), 447–463 (2005)
8. Boolos, G.: The Logic of Provability. Cambridge University Press, Cambridge (1993)
9. Dashkov, E.V.: On the positive fragment of the polymodal provability logic GLP. Matematicheskie Zametki **91**(3), 331–346 (2012). English translation: Mathematical Notes 91(3):318–333, 2012
10. Fernández-Duque, D., Joosten, J.: Models of transfinite provability logic. J. Symbol. Logic **78**(2), 543–561 (2013)
11. Reyes, E.H., Joosten, J.J.: The logic of Turing progressions. arXiv:1604.08705v2 [math.LO] (2016)
12. Reyes, E.H., Joosten, J.J.: Relational semantics for the Turing Schmerl calculus. arXiv:1709.04715 [math.LO] (2017)
13. Icard III, T.F.: A topological study of the closed fragment of GLP. J. Logic Comput. **21**(4), 683–696 (2011)
14. Ignatiev, K.N.: On strong provability predicates and the associated modal logics. J. Symbol. Logic **58**, 249–290 (1993)
15. Jackson, M.: Semilattices with closure. Algebra Universalis **52**, 1–37 (2004)
16. Japaridze, G.K.: The modal logical means of investigation of provability. Thesis in Philosophy, in Russian, Moscow (1986)
17. Joosten, J.J.: Turing–Taylor expansions of arithmetical theories. Studia Logica **104**, 1225–1243 (2015). https://doi.org/10.1007/s11225-016-9674-z
18. Kikot, S., Kurucz, A., Tanaka, Y., Wolter, F., Zakharyaschev, M.: On the completeness of EL-equations: first results. In: 11th International Conference on Advances in Modal Logic, Short Papers (Budapest, 30 August – 2 September, 2016), pp. 82–87 (2016)
19. Kikot, S., Kurucz, A., Tanaka, Y., Wolter, F., Zakharyaschev, M.: Kripke Completeness of strictly positive modal logics over meet-semilattices with operators. ArXiv e-prints, August 2017
20. Kurucz, A., Tanaka, Y., Wolter, F., Zakharyaschev, M.: Conservativity of Boolean algebras with operators over semi lattices with operators. In: Proceedings of TACL 2011, pp. 49–52 (2011)
21. Pakhomov, F.: On the complexity of the closed fragment of Japaridze's provability logic. Archive Math. Logic **53**(7), 949–967 (2014)
22. Pakhomov, F.: On elementary theories of ordinal notation systems based on reflection principles. Proc. Steklov Inst. Math. **289**, 194–212 (2015)
23. Shamkanov, D.: Nested sequents for provability logic GLP. Logic J. IGPL **23**(5), 789–815 (2015)
24. Sofronie-Stokkermans, V.: Locality and subsumption testing in EL and some of its extensions. In: Areces, C., Goldblatt, R. (eds.) Advances in Modal Logic, vol. 7, pp. 315–339. College Publications, London (2008)

A Logic of Blockchain Updates

Kai Brünnler[1], Dandolo Flumini[2], and Thomas Studer[3(✉)]

[1] Bern University of Applied Sciences, Bern, Switzerland
kai.bruennler@bfh.ch
[2] ZHAW School of Engineering, Winterthur, Switzerland
dandolo.flumini@zhaw.ch
[3] University of Bern, Bern, Switzerland
tstuder@inf.unibe.ch

Abstract. Blockchains are distributed data structures that are used to achieve consensus in systems for cryptocurrencies (like Bitcoin) or smart contracts (like Ethereum). Although blockchains gained a lot of popularity recently, there are only few logic-based models for blockchains available. We introduce BCL, a dynamic logic to reason about blockchain updates, and show that BCL is sound and complete with respect to a simple blockchain model.

Keywords: Blockchain · Modal logic · Dynamic epistemic logic

1 Introduction

Bitcoin [16] is a cryptocurrency that uses peer-to-peer technology to support direct user-to-user transactions without an intermediary such as a bank or credit card company. In order to prevent double spending, which is a common issue in systems without central control, Bitcoin maintains a complete and public record of all transactions at each node in the network. This ledger is called the *blockchain*.

The blockchain is essentially a growing sequence of blocks, which contain approved transactions and a cryptographic hash of the previous block in the sequence. Because the blockchain is stored locally at each node, any update to it has to be propagated to the entire network. Nodes that receive a transaction [1,18]

1. first verify its validity (i.e., whether it is compatible with all preceeding transactions);
2. if it is valid, then it is added to the blockchain and
3. sent to all other nodes.

Blockchain technology, as a general solution to the Byzantine Generals' Problem [15], is now not only used for financial transactions but also for many other applications like, e.g., smart contracts [5].

T. Studer—Supported by the Swiss National Science Foundation grant 200021_165549.

S. Artemov and A. Nerode (Eds.): LFCS 2018, LNCS 10703, pp. 107–119, 2018.
https://doi.org/10.1007/978-3-319-72056-2_7

Herlihy and Moir [11] propose to develop a logic of accountability to design and verify blockchain systems. In particular, they discuss blockchain scenarios to test (i) logics of authorization, (ii) logics of concurrency, and (iii) logics of incentives.

Halpern and Pass [10] provide a characterization of agents' knowledge when running a blockchain protocol using a variant of common knowledge.

In the present paper, we are not interested in accountability or aspects of common knowledge. We study the local, single agent perspective of a blockchain. That is we investigate steps 1 and 2 of the above procedure for receiving a transaction. Our approach is inspired by dynamic epistemic logic [7]. A given state of the local blockchain entails knowledge about the transactions that have taken place. We ask: *how does this knowledge change when a new block is received that might be added to the blockchain?* We develop a dynamic logic, BCL, with a semantics that is based on a blockchain model. The update operators of BCL are interpreted as receiving new blocks. It is the aim of this paper to investigate the dynamics of local blockchain updates.

The deductive system for BCL includes reduction axioms that make it possible to establish completeness by a reduction to the update-free case [13]. However, since blockchain updates are only performed if certain consistency conditions are satisfied, we use conditional reduction axioms similar to the ones developed by Steiner to model consistency preserving updates [19]. Moreover, unlike traditional public announcements [7], blockchain updates cannot lead to an inconsistent state, i.e., updates are total, like in [20].

We do not base BCL on an existing blockchain implementation but use a very simple model. First of all, the blockchain is a sequence of propositional formulas. Further, we maintain a list of provisional updates. Our blocks consist of two parts: a sequence number (called the index of the block) and a propositional formula. If a block is received, then the following case distinction is performed where i is the index of the block and l is the current length of the blockchain:

1. $i \leq l$. The block is ignored.
2. $i = l + 1$. If the formula of the block is consistent with the blockchain, then it is added to the blockchain; otherwise the block is ignored. If the blockchain has been extended, then this procedure is performed also with the blocks stored in the list of provisional updates.
3. $i > l + 1$. The block is added to the list of provisional updates.

Although this is a simple model, it features two important logical properties of blockchains: consistency must be preserved and blocks may be received in the wrong order, in which case they are stored separately until the missing blocks have been received.

The main contribution of our paper from the point of view of dynamic epistemic logic is that we maintain a list of provisional updates. That means we support updates that do not have an immediate effect but that may lead to a belief change later only after certain other updates have been performed. BCL is the first logic that features provisional updates of this kind.

The paper is organized as follows. The next section introduces our blockchain model, the language of BCL, and its semantics. In Sect. 3, we introduce a deductive system for BCL. We establish soundness of BCL in Sect. 4. In Sect. 5, we show a normal form theorem for BCL, which is used in Sect. 6 to prove completeness of BCL. The final section studies some key principles of the dynamics of our blockchain logic and discusses future work.

We will only mention the lemmas that are needed to establish the main theorems of this paper; but we do not give any proofs because of lack of space. Detailed proofs can be found in the accompanying arXiv paper [3].

2 A Simple Blockchain Logic

The set of all natural numbers is denoted by $\mathbb{N} := \{0, 1, 2, \ldots\}$. The set of positive natural numbers is denoted by $\mathbb{N}^+ := \{1, 2, \ldots\}$. We use ω for the least ordinal such that $\omega > n$, for all $n \in \mathbb{N}$.

Let $\sigma = \langle \sigma_1, \ldots, \sigma_n \rangle$ be a finite sequence. We define its *length* by $\mathsf{len}(\sigma) := n$. For an infinite sequence $\sigma = \langle \sigma_1, \sigma_2, \ldots \rangle$ we set $\mathsf{len}(\sigma) := \omega$. For a (finite or infinite) sequence $\sigma = \langle \sigma_1, \sigma_2, \ldots, \sigma_i, \ldots \rangle$ we set $(\sigma)_i := \sigma_i$ for $i \leq \mathsf{len}(\sigma)$. The case $i > \mathsf{len}(\sigma)$ can be safely ingored. The *empty sequence* is denoted by $\langle\rangle$ and we set $\mathsf{len}(\langle\rangle) := 0$. We can append x to a finite sequence $\sigma := \langle \sigma_1, \ldots, \sigma_n \rangle$, in symbols we set $\sigma \circ x := \langle \sigma_1, \ldots, \sigma_n, x \rangle$. We will also need the set of all components of a sequence σ and define

$$\mathsf{set}(\sigma) := \{x \mid \text{there is an } i \text{ such that } x = \sigma_i\}.$$

In particular, we have $\mathsf{set}(\langle\rangle) := \emptyset$. Moreover, we use the shorthand $x \in \sigma$ for $x \in \mathsf{set}(\sigma)$.

We start with a countable set of atomic propositions $\mathcal{AP} := \{P0, P1, \ldots\}$. The set of formulas $\mathcal{L}_{\mathsf{cl}}$ of classical propositional logic is given by the following grammar

$$A ::= \bot \mid P \mid A \to A,$$

where $P \in \mathcal{AP}$.

In order to introduce the language $\mathcal{L}_{\mathsf{B}}$ for blockchain logic, we need another countable set of special atomic propositions $\mathcal{AQ} := \{Q1, Q2, \ldots\}$ that is disjoint with $\mathcal{AP}$. We will use these special propositions later to keep track of the length of the blockchain. The formulas of $\mathcal{L}_{\mathsf{B}}$ are now given by the grammar

$$F ::= \bot \mid P \mid Q \mid F \to F \mid \Box A \mid [i, A]F,$$

where $P \in \mathcal{AP}$, $Q \in \mathcal{AQ}$, $A \in \mathcal{L}_{\mathsf{cl}}$, and $i \in \mathbb{N}^+$. The operators of the form $[i, A]$ are called *blockchain updates* (or simply *updates*).

Note that in $\mathcal{L}_{\mathsf{B}}$ we cannot express higher-order knowledge, i.e., we can only express knowledge about propositional facts but not knowledge about knowledge of such facts.

For all languages in this paper, we define further Boolean connectives (e.g. for negation, conjunction, and disjunction) as usual. Moreover, we assume that unary connectives bind stronger than binary ones.

For $\mathcal{L}_{\mathsf{cl}}$ we use the semantics of classical propositional logic. A *valuation* v is a subset of $\mathcal{AP}$ and we define the truth of an $\mathcal{L}_{\mathsf{cl}}$-formula A under v, in symbols $\mathsf{v} \models A$ as usual. For a set Γ of $\mathcal{L}_{\mathsf{cl}}$-formulas, we write $\mathsf{v} \models \Gamma$ if $\mathsf{v} \models A$ for all $A \in \Gamma$. The set Γ is *satisfiable* if there is a valuation v such that $\mathsf{v} \models \Gamma$. We say Γ *entails* A, in symbols $\Gamma \models A$, if for each valuation v we have

$$\mathsf{v} \models \Gamma \quad \text{implies} \quad \mathsf{v} \models A.$$

Now we introduce the blockchain semantics for $\mathcal{L}_{\mathsf{B}}$.

Definition 1. *A* block *is a pair* $[i, A]$ *where* A *is an* $\mathcal{L}_{\mathsf{cl}}$*-formula and* $i \in \mathbb{N}^+$*. We call* i *the* index *and* A *the* formula *of the block* $[i, A]$*. We define functions* ind *and* fml *by* $\mathsf{ind}[i, A] := i$ *and* $\mathsf{fml}[i, A] := A$*.*

Definition 2. *A* model $\mathsf{M} := (\mathsf{I}, \mathsf{BC}, \mathsf{PU}, \mathsf{v})$ *is a quadruple where*

1. I *is a set of* $\mathcal{L}_{\mathsf{cl}}$*-formulas*
2. BC *is a sequence of* $\mathcal{L}_{\mathsf{cl}}$*-formulas*
3. PU *is a finite sequence of blocks*
4. v *is a valuation, i.e.* $\mathsf{v} \subseteq \mathcal{AP}$

such that

$$\mathsf{I} \cup \mathsf{set}(\mathsf{BC})\, is\ satisfiable \tag{1}$$

and

$$for\ each\ block\, [i, A] \in \mathsf{PU}\, we\ have\, i > \mathsf{len}(\mathsf{BC}) + 1. \tag{2}$$

The components of a model $(\mathsf{I}, \mathsf{BC}, \mathsf{PU}, \mathsf{v})$ have the following meaning:

1. I models initial background knowledge.
2. BC is the blockchain.
3. PU stands for *provisional updates.* The sequence PU consists of those blocks that have been announced but that could not yet be added to the blockchain because their index is too high. Maybe they will be added to BC later (i.e., after the missing blocks have been added).
4. v states which atomic propositions are true.

We need some auxiliary definitions in order to precisely describe dynamics of the blockchain.

Definition 3

1. *Let* PU *be a finite sequence of blocks. Then we let* $\mathsf{find}(i, \mathsf{PU})$ *be the least* $j \in \mathbb{N}^+$ *such that there is an* $\mathcal{L}_{\mathsf{cl}}$*-formula* A *with* $[i, A] = (\mathsf{PU})_j$*.*
2. *Let* $\sigma = \langle \sigma_1, \ldots, \sigma_{i-1}, \sigma_i, \sigma_{i+1}, \ldots \rangle$ *be a sequence. We set*

$$\mathsf{remove}(i, \sigma) := \langle \sigma_1, \ldots, \sigma_{i-1}, \sigma_{i+1}, \ldots \rangle.$$

3. *Given a set of $\mathcal{L}_{\mathsf{cl}}$-formulas* I*, a sequence of $\mathcal{L}_{\mathsf{cl}}$-formulas* BC*, and a finite sequence of blocks* PU*, then the* chain completion $\mathsf{complete}(\mathsf{I}, \mathsf{BC}, \mathsf{PU})$ *is computed according to Algorithm 1.*

Algorithm 1. Chain Completion Algorithm: complete

```
Input: $(\mathsf{I}, \mathsf{BC}, \mathsf{PU})$
 1: $n \leftarrow \mathsf{len}(\mathsf{BC}) + 1$
 2: while $[n, A] \in \mathsf{PU}$ for some formula $A$ do
 3:     $i \leftarrow \mathsf{find}(n, \mathsf{PU})$
 4:     $B \leftarrow \mathsf{fml}((\mathsf{PU})_i)$
 5:     $\mathsf{remove}(i, \mathsf{PU})$
 6:     if $\mathsf{I} \cup \mathsf{set}(\mathsf{BC}) \cup \{B\}$ is satisfiable then
 7:         $\mathsf{BC} \leftarrow \mathsf{BC} \circ B$
 8:         $n \leftarrow \mathsf{len}(\mathsf{BC}) + 1$
 9:     end if
10: end while
11: for $i \in \mathsf{len}(\mathsf{PU}), \ldots, 1$ do
12:     if $\mathsf{ind}((\mathsf{PU})_i) < n$ then
13:         $\mathsf{remove}(i, \mathsf{PU})$
14:     end if
15: end for
16: return $(\mathsf{BC}, \mathsf{PU})$
```

Let us comment on the chain completion procedure. The numbers refer to the lines in Algorithm 1.

1: n is the index a block must contain so that it could be added to the blockchain BC.
2: '$[n, A] \in \mathsf{PU}$ for some formula A' means that PU contains a block that could be added to BC.
3–5: Find the next formula B that could be added to BC and remove the corresponding block from PU.
6: '$\mathsf{I} \cup \mathsf{set}(\mathsf{BC}) \cup \{B\}$ is satisfiable' means that B is consistent with the current belief. This test guarantees that (1) will always be satisfied.
7, 8: Update the blockchain BC with B.
11–15: Remove all blocks from PU whose index is less than or equal to the current length of the blockchain BC. Because the blockchain never gets shorter, these block will never be added. Removing them guarantees that (2) will always be satisfied.

Note if BC and PU satisfy condition (2) in the definition of a model, then the chain completion algorithm will return BC and PU unchanged.

Lemma 1. *Let* I *be a set of $\mathcal{L}_{\mathsf{cl}}$-formulas and let* BC *be a sequence of $\mathcal{L}_{\mathsf{cl}}$-formulas such that* $\mathsf{I} \cup \mathsf{set}(\mathsf{BC})$ *is satisfiable. Let* PU *be an arbitrary finite sequence of blocks. For* $(\mathsf{BC}', \mathsf{PU}') := \mathsf{complete}(\mathsf{I}, \mathsf{BC}, \mathsf{PU})$ *we find that*

1. $\mathsf{I} \cup \mathsf{set}(\mathsf{BC}')$ *is satisfiable and*
2. *for each block* $[i, A] \in \mathsf{PU}'$ *we have* $i > \mathsf{len}(\mathsf{BC}') + 1$.

Definition 4. *Let* $\mathsf{M} := (\mathsf{I}, \mathsf{BC}, \mathsf{PU}, \mathsf{v})$ *be a model and* $[i, A]$ *be a block. The* updated model $\mathsf{M}^{[i,A]}$ *is defined as* $(\mathsf{I}, \mathsf{BC}', \mathsf{PU}', \mathsf{v})$ *where*

$$(\mathsf{BC}', \mathsf{PU}') := \mathsf{complete}(\mathsf{I}, \mathsf{BC}, \mathsf{PU} \circ [i, A]).$$

Remark 1. Note that $\mathsf{M}^{[i,A]}$ is well-defined: by Lemma 1 we know that $\mathsf{M}^{[i,A]}$ is indeed a model.

Definition 5. *Let* $\mathsf{M} := (\mathsf{I}, \mathsf{BC}, \mathsf{PU}, \mathsf{v})$ *be a model. We define the* truth *of an* $\mathcal{L}_\mathsf{B}$*-formula* F *in* M*, in symbols* $\mathsf{M} \models F$*, inductively by:*

1. $\mathsf{M} \not\models \bot$;
2. $\mathsf{M} \models P$ *if* $P \in \mathsf{v}$ *for* $P \in \mathcal{AP}$;
3. $\mathsf{M} \models Qi$ *if* $i \leq \mathsf{len}(\mathsf{BC})$ *for* $Qi \in \mathcal{AQ}$;
4. $\mathsf{M} \models F \to G$ *if* $\mathsf{M} \not\models F$ *or* $\mathsf{M} \models G$;
5. $\mathsf{M} \models \Box A$ *if* $\mathsf{I} \cup \mathsf{set}(\mathsf{BC}) \models A$;
6. $\mathsf{M} \models [i, A]F$ *if* $\mathsf{M}^{[i,A]} \models F$.

A formula $\Box A$ means that A follows from the blockchain, i.e. A is a logical consequence from the propositions stored in the blockchain. We can consider $\Box$ to be an epistemic operator since the blockchain represents our *knowledge* about which transactions have happened.

We define validity only with respect to the class of models that do not have provisional updates.

Definition 6. *We call a model* $\mathsf{M} = (\mathsf{I}, \mathsf{BC}, \mathsf{PU}, \mathsf{v})$ initial *if* $\mathsf{PU} = \langle\rangle$. *A formula* F *is called* valid *if* $\mathsf{M} \models F$ *for all initial models* M.

3 The Deductive System BCL

In order to present an axiomatic system for our blockchain logic, we need to formalize an *acceptance condition* stating whether a received block can be added to the blockchain. That is we need a formula $\mathsf{Acc}(i, A)$ expressing that the formula A is consistent with the current beliefs and the current length of the blockchain is $i-1$. Thus if $\mathsf{Acc}(i, A)$ holds, then the block $[i, A]$ will be accepted and added to the blockchain. The truth definition for the atomic propositions $Qi \in \mathcal{AQ}$ says that Qi is true if the blockchain contains at least i elements. That means the formula $Q(i-1) \land \neg Qi$ is true if the blockchain contains exactly $i-1$ elements. This leads to the following definition of $\mathsf{Acc}(i, A)$ for $i \in \mathbb{N}^+$:

$$\mathsf{Acc}(i, A) := \begin{cases} \neg Qi \land \neg\Box\neg A & \text{if } i = 1 \\ Q(i-1) \land \neg Qi \land \neg\Box\neg A & \text{if } i > 1 \end{cases}$$

As desired, we find that if $\mathsf{Acc}(i, A)$ is true, then the chain completion algorithm can append the formula A to the blockchain (see Lemma 2 later).

An $\mathcal{L}_\mathsf{B}$-formula is called compliant if the blockchain updates occur in the correct order. Formally, we use the following definition.

Definition 7. *An $\mathcal{L}_{\mathsf{B}}$-formula F is called* compliant *if no occurrence of a $[i, A]$-operator in F is in the scope of some $[j, B]$-operator with $j > i$.*

Now we can define a deductive system for BCL. It is formulated in the language $\mathcal{L}_{\mathsf{B}}$ and consists of the following axioms:

(PT) Every instance of a propositional tautology
(K) $\Box(F \to G) \to (\Box F \to \Box G)$
(D) $\neg\Box\bot$
(Q) $Qi \to Qj$ if $i > j$
(A1) $[i, A]\bot \to \bot$
(A2) $[i, A]P \leftrightarrow P$ for $P \in \mathcal{AP}$
(A3.1) $\mathsf{Acc}(i, A) \to ([i, A]Qi \leftrightarrow \top)$ for $Qi \in \mathcal{AQ}$
(A3.2) $\neg\mathsf{Acc}(i, A) \to ([i, A]Qi \leftrightarrow Qi)$ for $Qi \in \mathcal{AQ}$
(A3.3) $[i, A]Qj \leftrightarrow Qj$ for $Qj \in \mathcal{AQ}$ and $i \neq j$
(A4) $[i_1, A_1] \dots [i_k, A_k](F \to G) \leftrightarrow$
$([i_1, A_1] \dots [i_k, A_k]F \to [i_1, A_1] \dots [i_k, A_k]G)$
(A5.1) $\mathsf{Acc}(i, A) \to ([i, A]\Box B \leftrightarrow \Box(A \to B))$
(A5.2) $\neg\mathsf{Acc}(i, A) \to ([i, A]\Box B \leftrightarrow \Box B)$
(A6) $[h_1, C_1] \dots [h_k, C_k][i, A][j, B]F \leftrightarrow$
$[h_1, C_1] \dots [h_k, C_k][j, A][i, B]F$ for $i \neq j$

We need a little arithmetic: Axiom (Q) is used to compare indexes. But we do not need anything else.

Note that in (A6), we may choose k to be 0, in which case the axiom has the form $[i, A][j, B]F \leftrightarrow [j, A][i, B]F$ for $i \neq j$.

In order to formulate the rules of BCL, we need the following notation. Let $H(P)$ be a formula that may contain occurrences of the atomic proposition P. By $H(F)$, we denote the result of simultaneously replacing each occurrence of P in $H(P)$ with the formula F. The rules of BCL are:

$$(\mathsf{MP})\,\frac{F \qquad F \to G}{G} \qquad (\mathsf{NEC})\,\frac{A}{\Box A} \qquad (\mathsf{SUB})\,\frac{F \leftrightarrow G}{H(F) \leftrightarrow H(G)}$$

where (SUB) can only be applied if $H(F) \leftrightarrow H(G)$ is a compliant formula.

Remark 2. Our semantics includes the case of infinite blockchains: in a given model $(\mathsf{I}, \mathsf{BC}, \mathsf{PU}, \mathsf{v})$, the sequence BC may have infinite length. If we want to exclude such models, then we have to add an infinitary rule

$$\frac{Qi \quad \text{for all } i \in \mathbb{N}^+}{\bot}$$

to BCL. This rule states that some Qi must be false, which means that BC has finite length.

4 Soundness

Before we can establish soundness of BCL, we have to show some preparatory lemmas.

Lemma 2. *Let* $\mathsf{M} := (\mathsf{I}, \mathsf{BC}, \langle\rangle, \mathsf{v})$ *be an initial model. Further let*

$$(\mathsf{I}, \mathsf{BC}', \mathsf{PU}', \mathsf{v}) := \mathsf{M}^{[i,A]}$$

for some block $[i, A]$*.*

1. *If* $\mathsf{M} \models \mathsf{Acc}(i, A)$*, then* $\mathsf{BC}' = \mathsf{BC} \circ A$*. In particular, this yields* $\mathsf{len}(\mathsf{BC}') = i$ *and for each* j *with* $j \neq i$*,*

$$M \models Qj \quad \textit{if and only if} \quad \mathsf{M}^{[i,A]} \models Qj.$$

2. *If* $\mathsf{M} \not\models \mathsf{Acc}(i, A)$*, then* $\mathsf{BC}' = \mathsf{BC}$*.*

Lemma 3. *Each axiom of* BCL *is valid.*

Lemma 4. *Let* $\mathsf{M} = (\mathsf{I}, \mathsf{BC}, \mathsf{PU}, \mathsf{v})$ *be an arbitrary model and let* $[i, A]$ *be a block such that* $i > \mathsf{len}(\mathsf{BC}) + 1$*. Then we have* $\mathsf{M}^{[i,A]} = (\mathsf{I}, \mathsf{BC}, \mathsf{PU} \circ [i, A], \mathsf{v})$*.*

Lemma 5. *Let* $\mathsf{M} = (\mathsf{I}, \mathsf{BC}, \langle\rangle, \mathsf{v})$ *be an initial model and let* $[i, A]$ *be a block such that* $i \leq \mathsf{len}(\mathsf{BC}) + 1$*. Then* $\mathsf{M}^{[i,A]}$ *is an initial model, too.*

Lemma 6. *Let* $(\mathsf{I}, \mathsf{BC}, \mathsf{PU}, \mathsf{v})$ *be a model and* F *be an* $\mathcal{L}_{\mathsf{B}}$*-formula such that for each* $[i, A]$ *occurring in* F *we have* $i > \mathsf{len}(\mathsf{BC}) + 1$*. Then*

$$(\mathsf{I}, \mathsf{BC}, \mathsf{PU}, \mathsf{v}) \models F \quad \textit{if and only if} \quad (\mathsf{I}, \mathsf{BC}, \langle\rangle, \mathsf{v}) \models F.$$

Now we can show that the rule (SUB) preserves validity.

Lemma 7. *Let* $H(P), F, G$ *be* $\mathcal{L}_{\mathsf{B}}$*-formulas such that* $H(F) \leftrightarrow H(G)$ *is compliant. We have that*

$$\textit{if } F \leftrightarrow G \textit{ is valid, then } H(F) \leftrightarrow H(G) \textit{ is valid, too.}$$

We have established that the axioms of BCL are valid and that (SUB) preserves validity. It is easy to see that the rules (MP) and (NEC) also preseve validity. Soundness of BCL follows immediately.

Corollary 1. *For each formula* F *we have*

$$\vdash F \textit{ implies } F \textit{ is valid.}$$

Remark 3. The reduction axiom $(\mathsf{A3.3})$ does not hold in non-initial models. Indeed, let $\mathsf{M} := (\emptyset, \langle\rangle, \langle[2, \top]\rangle, \emptyset)$. We find that $\mathsf{M}^{[1,P]} = (\emptyset, \langle P, \top\rangle, \langle\rangle, \emptyset)$. Hence $\mathsf{M}^{[1,P]} \models Q2$, which is $\mathsf{M} \models [1, P]Q2$. But we also have $\mathsf{M} \not\models Q2$.

Remark 4. The above remark also implies that a block necessitation rule would not be sound, that is the validity of F does not entail the validity of $[i, A]F$. Indeed, the axiom $[1, P]Q2 \leftrightarrow Q2$ is valid; but the formula $[2, \top]([1, P]Q2 \leftrightarrow Q2)$ is not valid as shown in the previous remark.

Remark 5. The rule (SUB) would not preserve validity if we drop the condition that the conclusion must be compliant. Indeed, let us again consider the valid formula $[1, P]Q2 \leftrightarrow Q2$. Without the compliance condition, the rule (SUB) would derive $[2, P'][1, P]Q2 \leftrightarrow [2, P']Q2$, which is not a valid formula.

5 Normal Form

Remember that a formula is compliant if the blockchain updates occur in the correct order. In this section, we establish a normal form theorem for our simple blockchain logic.

Definition 8. *A* base formula *is a formula that has one of the following forms (which include the case of no blockchain updates):*

1. $[i_1, A_1] \ldots [i_m, A_m]\bot$
2. $[i_1, A_1] \ldots [i_m, A_m]P$ *with* $P \in \mathcal{AP} \cup \mathcal{AQ}$
3. $[i_1, A_1] \ldots [i_m, A_m]\Box B$

Formulas in normal form *are given as follows:*

1. *each compliant base formula is in normal form*
2. *if F and G are in normal form, then so is $F \to G$.*

Remark 6. As an immediate consequence of this definition, we obtain that for each formula F,

if F is in normal form, then F is compliant.

The following theorem states that for each formula, there is a provably equivalent formula in normal form. The proof is by induction on the structure of F.

Theorem 1. *For each $\mathcal{L}_{\mathsf{B}}$-formula F, there is an $\mathcal{L}_{\mathsf{B}}$-formula G in normal form such that $\vdash F \leftrightarrow G$.*

6 Completeness

We first show that BCL is complete for modal formulas. The modal language $\mathcal{L}_{\mathsf{M}}$ consists of all update-free $\mathcal{L}_{\mathsf{B}}$-formulas. Formally, $\mathcal{L}_{\mathsf{M}}$ is given by the following grammar

$$F ::= \bot \mid P \mid Q \mid F \to F \mid \Box A,$$

where $P \in \mathcal{AP}$, $Q \in \mathcal{AQ}$, and $A \in \mathcal{L}_{\mathsf{cl}}$.

We need the collection $\mathsf{BCL}^{\Box}$ of all BCL axioms that are given in $\mathcal{L}_{\mathsf{M}}$. The usual satisfaction relation for Kripke models is denoted by $\models_{\Box}$.

Lemma 8. *For each $\mathcal{L}_{\mathsf{M}}$-formula F we have*

F is valid implies $\vdash F$.

We establish completeness for compliant formulas using a translation from compliant formulas to provably equivalent update-free formulas. We start with defining a mapping h that eliminates update operators.

Definition 9. *The mapping* h *from* $\{[i,A]F \mid F \in \mathcal{L}_{\mathsf{M}}\}$ *to* $\mathcal{L}_{\mathsf{M}}$ *is inductively defined by:*

$$\begin{aligned}
\mathsf{h}([i,A]\bot) &:= \bot \\
\mathsf{h}([i,A]P) &:= P \quad \textit{for } P \in \mathcal{AP} \\
\mathsf{h}([i,A]Qi) &:= \mathsf{Acc}(i,A) \lor Qi \\
\mathsf{h}([i,A]Qj) &:= Qj \quad \textit{for } Qj \in \mathcal{AQ} \textit{ and } i \neq j \\
\mathsf{h}([i,A](F \to G)) &:= \mathsf{h}([i,A]F) \to \mathsf{h}([i,A]G) \\
\mathsf{h}([i,A]\Box B) &:= (\mathsf{Acc}(i,A) \land \Box(A \to B)) \lor (\neg\mathsf{Acc}(i,A) \land \Box B)
\end{aligned}$$

The mapping h corresponds to the reduction axioms of BCL. Thus it is easy to show the following lemma by induction on the structure of F.

Lemma 9. *Let* F *be an* $\mathcal{L}_{\mathsf{B}}$*-formula of the form* $[i,A]G$ *such that* $G \in \mathcal{L}_{\mathsf{M}}$*. We have that* $\vdash F \leftrightarrow \mathsf{h}(F)$*.*

We define a translation t from $\mathcal{L}_{\mathsf{B}}$ to $\mathcal{L}_{\mathsf{M}}$

Definition 10. *The mapping* $\mathsf{t} : \mathcal{L}_{\mathsf{B}} \to \mathcal{L}_{\mathsf{M}}$ *is inductively defined by:*

$$\begin{aligned}
\mathsf{t}(\bot) &:= \bot \\
\mathsf{t}(P) &:= P \quad \textit{for } P \in \mathcal{AP} \cup \mathcal{AQ} \\
\mathsf{t}(F \to G) &:= \mathsf{t}(F) \to \mathsf{t}(G) \\
\mathsf{t}(\Box A) &:= \Box A \\
\mathsf{t}([i,A]F) &:= \mathsf{h}([i,A]\mathsf{t}(F))
\end{aligned}$$

Lemma 10. *For each compliant formula* F*, we have*

$$\vdash F \leftrightarrow \mathsf{t}(F).$$

Theorem 2. *For each compliant* $\mathcal{L}_{\mathsf{B}}$*-formula* F *we have*

$$F \textit{ is valid implies } \quad \vdash F.$$

Combining Theorems 1 and 2 easily yields completeness for the full language.

Theorem 3. *For each* $\mathcal{L}_{\mathsf{B}}$*-formula* F *we have*

$$F \textit{ is valid implies } \quad \vdash F.$$

7 Conclusion

We have presented BCL, a dynamic logic to reason about updates in a simple blockchain model. Our semantics does not have the full complexity of the blockchains used in Bitcoin or Ethereum, yet it exhibits two key properties of blockchains: blockchain extensions must preserve consistency and blocks may be received in the wrong order. Note, however, that although receiving blocks in the wrong order is an important logical possibility, it only happens rarely in practice: in the Bitcoin protocol the average generation time of a new block is 10 min; the average time until a node receives a block is only 6.5 s [6].

In order to illustrate the dynamics of our simple blockchain logic, we state some valid principles of BCL:

Persistence: $\Box A \to [i,B]\Box A$. Beliefs are persistent, i.e., receiving a new block cannot lead to a retraction of previous beliefs.

Consistency: $[i,B]\neg\Box\bot$. Receiving a new block cannot result in inconsistent beliefs.

Success: $\mathsf{Acc}(i,A) \to [i,A]\Box A$. If a block $[i,A]$ is acceptable, then A is believed after receiving $[i,A]$.[1]

Failure: $(Qi \vee \neg Q(i-1)) \to ([i,B]\Box A \leftrightarrow \Box A)$. If the current length of the blockchain is not $i-1$, then receiving a block $[i,B]$ will not change the current beliefs.

Proof. 1. Persistence: $\Box A \to [i,B]\Box A$. Let $\mathsf{M} := (\mathsf{I}, \mathsf{BC}, \langle\rangle, \mathsf{v})$ be an initial model and assume $\mathsf{M} \models \Box A$. That is $\mathsf{I} \cup \mathsf{set}(\mathsf{BC}) \models A$. Let $(\mathsf{I}, \mathsf{BC}', \mathsf{PU}', \mathsf{v}) := \mathsf{M}^{[i,B]}$. We find that $\mathsf{set}(\mathsf{BC}) \subseteq \mathsf{set}(\mathsf{BC}')$. Therefore, $\mathsf{I} \cup \mathsf{set}(\mathsf{BC}') \models A$, hence we have $\mathsf{M}^{[i,B]} \models \Box A$ and $\mathsf{M} \models [i,B]\Box A$.

2. Consistency: $[i,B]\neg\Box\bot$. We let $\mathsf{M} := (\mathsf{I}, \mathsf{BC}, \langle\rangle, \mathsf{v})$ be an initial model. Further, we set $(\mathsf{I}, \mathsf{BC}', \mathsf{PU}', \mathsf{v}) := \mathsf{M}^{[i,B]}$. By Lemma 1 we know that $\mathsf{I} \cup \mathsf{set}(\mathsf{BC}')$ is satisfiable, i.e., $\mathsf{I} \cup \mathsf{set}(\mathsf{BC}') \not\models \bot$. Hence we have $\mathsf{M}^{[i,B]} \models \neg\Box\bot$, which is $\mathsf{M} \models [i,B]\neg\Box\bot$.

3. Success: $\mathsf{Acc}(i,A) \to [i,A]\Box A$. Let $\mathsf{M} := (\mathsf{I}, \mathsf{BC}, \langle\rangle, \mathsf{v})$ be an initial model and assume $\mathsf{M} \models \mathsf{Acc}(i,A)$. Let $(\mathsf{I}, \mathsf{BC}', \mathsf{PU}', \mathsf{v}) := \mathsf{M}^{[i,A]}$. By Lemma 2, we know $\mathsf{BC}' = \mathsf{BC} \circ A$. Thus $\mathsf{I} \cup \mathsf{set}(\mathsf{BC}') \models A$ and, therefore $\mathsf{M}^{[i,A]} \models \Box A$, which is $\mathsf{M} \models [i,A]\Box A$.

4. Failure: $(Qi \vee \neg Q(i-1)) \to ([i,B]\Box A \leftrightarrow \Box A)$. Again, let $\mathsf{M} := (\mathsf{I}, \mathsf{BC}, \langle\rangle, \mathsf{v})$ be an initial model and assume $\mathsf{M} \models Qi \vee \neg Q(i-1)$. We find that $\mathsf{M} \not\models \mathsf{Acc}(i,B)$. Indeed,
$$\mathsf{M} \models Qi \text{ implies } \mathsf{M} \not\models \mathsf{Acc}(i,B)$$
and
$$\mathsf{M} \models \neg Q(i-1) \text{ implies } i > 1 \text{ and } \mathsf{M} \not\models \mathsf{Acc}(i,B).$$
Let $(\mathsf{I}, \mathsf{BC}', \mathsf{PU}', \mathsf{v}) := \mathsf{M}^{[i,B]}$. By Lemma 2, we know $\mathsf{BC}' = \mathsf{BC}$. Therefore, $\mathsf{M}^{[i,B]} \models \Box A$ if and only if $\mathsf{M} \models \Box A$, which yields $\mathsf{M} \models [i,B]\Box A \leftrightarrow \Box A$. □

There are several open issues for future work. Let us only mention two of them. Although blockchains are called *chains*, the data structure that is actually used is more tree-like and there are different options how to choose the valid branch: Bitcoin currently uses the branch that has the greastest proof-of-work effort invested in it [16] (for simplicity we can think of it as the longest branch); but recent research shows that the GHOST rule [18] (used, e.g., in Ethereum [21])

[1] We call this prinicple *success*; but it is not related to the notion of a *successful formula* as studied in dynamic epistemic logic, see, e.g., [8].

provides better security at higher transaction throughput. We plan to extend BCL so that it can handle tree-like structures and the corresponding forks of the chain. In particular, this requires some form of probability logic to model the fact that older transactions are less likely reversed [9,16,18].

In a multi-agent setting, each agent (node) has her own instance of the blockchain. Justification logics [2] could provide a formal approach to handle this. Evidence terms could represent blockchain instances and those instances can be seen as justifying the agents' knowledge about the accepted transactions. This approach would require to develop new dynamic justification logics [4,14,17]. Moreover, if the underlying blockchain model supports forks of the chain, then we need justification logics with probability operators [12].

Acknowledgements. We would like to thank Eveline Lehmann and Nenad Savic for carefully reading a previous version of this paper. We also thank the anonymous referees for many valuable comments that helped to improve this paper.

References

1. Antonopoulos, A.M.: Mastering Bitcoin: Unlocking Digital Crypto-Currencies. O'Reilly Media, Inc., Sebastopol (2014)
2. Artemov, S.N.: Explicit provability and constructive semantics. Bullet. Symbolic Logic **7**(1), 1–36 (2001)
3. Brünnler, K., Flumini, D., Studer, T.: A logic of blockchain updates. E-print 1707.01766. arXiv.org (2017)
4. Bucheli, S., Kuznets, R., Studer, T.: Realizing public announcements by justifications. J. Comput. Syst. Sci. **80**(6), 1046–1066 (2014)
5. Buterin, V.: Ethereum: a next-generation smart contract and decentralized application platform (2013). https://github.com/ethereum/wiki/wiki/White-Paper. Accessed 2 Feb 2017
6. Decker, C., Wattenhofer, R.: Information propagation in the Bitcoin network. In: 13th IEEE International Conference on Peer-to-Peer Computing, pp. 1–10 (2013)
7. van Ditmarsch, H., van der Hoek, W., Kooi, B.: Dynamic Epistemic Logic. Synthese Library, vol. 337. Springer, Dordrecht (2008). https://doi.org/10.1007/978-1-4020-5839-4
8. van Ditmarsch, H., Kooi, B.: The secret of my success. Synthese **151**(2), 201–232 (2006)
9. Grunspan, C., Pérez-Marco, R.: Double spend races. ArXiv e-prints 1702.02867 (2017)
10. Halpern, J.H., Rafael, P.: A knowledge-based analysis of the blockchain protocol. In: Lang, K. (ed.) TARK 2017, pp. 324–335, no. 251 in EPTCS (2017)
11. Herlihy, M., Moir, M.: Blockchains and the logic of accountability: keynote address. In: LICS 2016, pp. 27–30 (2016)
12. Kokkinis, I., Maksimović, P., Ognjanović, Z., Studer, T.: First steps towards probabilistic justification logic. Logic J. IGPL **23**(4), 662–687 (2015)
13. Kooi, B.: Expressivity and completeness for public update logics via reduction axioms. J. Appl. Non Classical Logics **17**(2), 231–253 (2007)
14. Kuznets, R., Studer, T.: Update as evidence: belief expansion. In: Artemov, S., Nerode, A. (eds.) LFCS 2013. LNCS, vol. 7734, pp. 266–279. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-35722-0_19

15. Lamport, L., Shostak, R., Pease, M.: The byzantine generals problem. ACM Trans. Program. Lang. Syst. **4**(3), 382–401 (1982)
16. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system (2009)
17. Renne, B.: Public communication in justification logic. J. Logic Comput. **21**(6), 1005–1034 (2011). Published online July 2010
18. Sompolinsky, Y., Zohar, A.: Secure high-rate transaction processing in bitcoin. In: Böhme, R., Okamoto, T. (eds.) FC 2015. LNCS, vol. 8975, pp. 507–527. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-47854-7_32
19. Steiner, D.: A system for consistency preserving belief change. In: Artemov, S., Parikh, R. (eds.) Proceedings of Rationality and Knowledge, 18th ESSLLI, pp. 133–144. Association for Logic, Language and Information (2006)
20. Steiner, D., Studer, T.: Total public announcements. In: Artemov, S.N., Nerode, A. (eds.) LFCS 2007. LNCS, vol. 4514, pp. 498–511. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72734-7_35
21. Wood, G.: Ethereum: a secure decentralised generalised transaction ledger, EIP-150 revision (2017). https://ethereum.github.io/yellowpaper/paper.pdf. Accessed 2 Feb 2017

From Display to Labelled Proofs for Tense Logics

Agata Ciabattoni, Tim Lyon(✉), and Revantha Ramanayake

Institut für Computersprachen, Technische Universität Wien, 1040 Wien, Austria
{agata,lyon,revantha}@logic.at

Abstract. We introduce an effective translation from proofs in the display calculus to proofs in the labelled calculus in the context of tense logics. We identify the labelled calculus proofs in the image of this translation as those built from labelled sequents whose underlying directed graph possesses certain properties. For the basic normal tense logic Kt, the image is shown to be the set of all proofs in the labelled calculus G3Kt.

Keywords: Display calculus · Labelled calculus
Structural proof theory · Tense logic · Modal logic

1 Introduction

The widespread application of logical methods in several areas of computer science, epistemology, and artificial intelligence has resulted in an explosion of new logics — each requiring an analytic proof calculus to facilitate study and applications. The reason is that the rules in an analytic calculus (de)compose the formula to be proved in a stepwise manner. This systematic decomposition can be exploited to prove important metalogical properties of the formalized logics and is central to developing automated reasoning methods. Being relatively simple and not requiring much technical machinery (‘bureaucracy’), the sequent calculus has always been the most popular formalism to use and try to construct analytic calculi. However, its simplicity means that it is also limited in its expressive power, and is hence unable to support analytic calculi for the many logics of interest. This has motivated the search for other, more expressive formalisms. Many proof formalisms generalizing the sequent calculus have been introduced in the last 30 years; each of them incorporates the bureaucracy in a distinct way and hence possesses distinct strengths, weaknesses, and expressive power. In particular, certain formalisms are more helpful than others for proving certain computational or metalogical properties. For this reason, it is fruitful to study logics in a number of different formalisms. For example, a large class of extensions of the minimal tense logic Kt have been presented as instances of the labelled calculus (e.g., [17,21]) and of the display calculus [10,14,22]. The former is an extension of the sequent calculus in which the relational semantics of the formalized logics is made an explicit part of the syntax; the latter extends Gentzen's language of sequents with new structural connectives that allow each

S. Artemov and A. Nerode (Eds.): LFCS 2018, LNCS 10703, pp. 120–139, 2018.
https://doi.org/10.1007/978-3-319-72056-2_8

formula in a sequent to be "displayed" as the whole of the antecedent or the whole of the succedent.

Labelled and display calculi substantially differ in their nature. Display calculi are typically *internal* in the sense that each step in a proof can be read as a formula of the logic.[1] In general, labelled calculi appear to manipulate formulae from a more expressive language which partially encodes the logic's semantics, and are hence termed *external*. Internal and external calculi have been introduced and studied within two essentially independent–and sometimes competing–streams in proof theory. These calculi possess different properties and lead to distinct proofs.

An effective way to relate calculi is by defining *embeddings*, i.e. functions that stepwise transform any proof in a calculus into a proof of the same formula in another calculus. A crucial feature of such a function is that the structural properties of the derivation are preserved in the translation. Such embeddings permit the transfer of certain proof theoretic results, thus alleviating the need for independent proofs in each system (see [9,11,18]). Moreover they shed light on the role of bureaucracy in proof calculi, and on the relationships between different syntactic and semantic presentations of a logic.

In this paper we investigate the relationships between display and labelled proofs for a well known class of tense logics obtained by extending Kt with Scott-Lemmon [15] axioms $\Diamond^h\Box^i p \rightarrow \Box^j \Diamond^k p$ $(h, i, j, k \geq 0)$. This class is an adequate case study as it includes many interesting/well-known logics, its display calculi are all internal, and the display and labelled rules capturing the Scott-Lemmon axioms[2] have a simple form. Due to their distinct foundational origins–the algebraic semantics for display calculi [14] and Kripke semantics for labelled calculi [17]–the relationship between their proofs is *prima facie* unclear; this is particularly true for the direction from labelled to display proofs (e.g., [19] contains a translation of display sequents into labelled sequents).

Exploiting the work of Goré *et al.* [10] who present the display calculus for the basic tense logic Kt as a nested sequent with two types of nesting constructors, we show the equivalence of the display calculus to a calculus on labelled directed graphs whose underlying undirected graph is a tree. These structures –*labelled UT graphs*–are a natural generalization of the labelled trees shown in [11] to correspond to nested sequents [3,13].

In particular, we give a bi-directional embedding between proofs in the display calculus and the labelled UT graph calculus. The latter are then mapped into Negri's [17] labelled sequent proofs. In the reverse direction, we then consider specifically Negri's labelled calculus for Kt and show that every derivation there is a derivation in the labelled UT graph calculus.

[1] More specifically, this is true of a display calculus for a logic such that every structural connective can be interpreted as a connective of the logic.

[2] Extending to primitive tense axioms [14] is straightforward though more syntactically involved.

2 Display and Labelled Calculi for Tense Logics

The tense logic Kt extends the normal modal logic K with the tense connectives $\blacklozenge$ and $\blacksquare$ and the following axioms and inference rule (see, e.g. [2,4]):

$$\blacksquare(p \to q) \to (\blacksquare p \to \blacksquare q) \qquad \blacklozenge p \leftrightarrow \neg\blacksquare\neg p \qquad \frac{A}{\blacksquare A}\,(\mathsf{nec})$$

$$p \to \Box\blacklozenge p \qquad p \to \blacksquare \Diamond p$$

An intuitive interpretation of $\Box A$ is the statement "it will always be the case that A" (i.e. it is necessarily the case that in the future A). Then $\blacksquare A$ can be interpreted as "it has always been the case that A" (it is necessarily the case that in the past A). Then $\Diamond A$ may be interpreted as "it is possible that in the future A", and $\blacklozenge A$ as "it is possible that in the past A". Of course, suitable other interpretations may be used as demanded by application.

We assume that our language consists of formulae in negation normal form, where all negation signs are pushed inward onto the propositional atoms. In particular, formulae are built from literals p and $\overline{p}$ using $\wedge$, $\vee$, $\Diamond$, $\Box$, $\blacklozenge$, and $\blacksquare$. Note that all results still hold for the full language where the $\neg$, $\to$, and $\leftrightarrow$ connectives are taken as primitive. Nevertheless, we restrict ourselves to negation normal form for matters of convenience.

The logics we consider in this paper are extensions of Kt with the Scott-Lemmon axioms $\Diamond^h\Box^i p \to \Box^j \Diamond^k p$ (or equivalently, $\blacklozenge^h \Diamond^j p \to \Diamond^i \blacklozenge^k p$), for $h, j, i, k \geq 0$. In negation normal form and in the absence of implication, the axioms become $\Box^h \Diamond^i \overline{p} \vee \Box^j \Diamond^k p$ (equivalently, $\blacksquare^h \Box^j \overline{p} \vee \Diamond^i \blacklozenge^k p$). We have limited ourselves here to the Scott-Lemmon axioms in order to simplify the notation and exposition, and also because this class of axioms is well-known within the modal logic community. Nevertheless, it is worth observing that our results extend in a natural way beyond the Scott-Lemmon axioms to Kracht's [14] *primitive tense axioms* (or, equivalently, $\mathcal{I}_2$ [5] or analytic inductive [12] axioms).

2.1 Display Calculi for Tense Logics

Introduced under the name Display Logic, Belnap's Display Calculus [1] generalises Gentzen's sequent calculus by supplementing the structural connective (comma) with new structural connectives. The beauty of the display calculus lies in a general cut-elimination theorem for all calculi obeying eight easily verifiable syntactic conditions [1,22]; this makes the display calculus a good candidate for capturing large classes of logics in a unified way, irrespective of their semantics or connectives.

We will present Goré *et al.*'s [10] display calculus SKT for Kt. This calculus can be seen as a one-sided version of Kracht's [14] display calculus for Kt, and also as a variant of Kashima's calculus [13]. The sequents of SKT are generated by the following grammar: $X := A | X, X | \circ\{X\} | \bullet\{X\}$.

Definition 1 (The Calculus SKT[10])

$$\frac{}{\Gamma, p, \overline{p}}\ (\mathit{id}) \qquad \frac{\Gamma, A, B}{\Gamma, A \vee B}\ (\vee) \qquad \frac{\Gamma, A \qquad \Gamma, B}{\Gamma, A \wedge B}\ (\wedge)$$

$$\frac{\Gamma, \Delta, \Delta}{\Gamma, \Delta}\ (\mathit{ctr}) \qquad \frac{\Gamma}{\Gamma, \Delta}\ (\mathit{wk}) \qquad \frac{\Gamma, \circ\{\Delta\}}{\bullet\{\Gamma\}, \Delta}\ (\mathit{rf}) \qquad \frac{\Gamma, \bullet\{\Delta\}}{\circ\{\Gamma\}, \Delta}\ (\mathit{rp})$$

$$\frac{\Gamma, \bullet\{A\}}{\Gamma, \blacksquare A}\ (\blacksquare) \qquad \frac{\Gamma, \circ\{A\}}{\Gamma, \Box A}\ (\Box) \qquad \frac{\Gamma, \bullet\{\Delta, A\}, \blacklozenge A}{\Gamma, \bullet\{\Delta\}, \blacklozenge A}\ (\blacklozenge) \qquad \frac{\Gamma, \circ\{\Delta, A\}, \Diamond A}{\Gamma, \circ\{\Delta\}, \Diamond A}\ (\Diamond)$$

SKT is referred to as a shallow nested sequent calculus because (i) the $\circ\{\ \}$ and $\bullet\{\ \}$ provide (two types of) nesting and (ii) all the rules are shallow in the sense that they operate at the *root* of the sequent (when the sequent is viewed in terms of its grammar tree). Although the rules in SKT are shallow, the two rules (rf) and (rp) can be used to bring nested formulae to the root.

Definition 2 (display property). *A display calculus has the* display property *if it contains a set of rules (the '*display rules*') such that for any sequent X containing an occurrence of Y, there exists Z such that Y, Z is derivable from X using the display rules.*

The display property states that any substructure in X can be brought to the 'top level' using the display rules. By inspection, SKT has the display property when $\{(\mathsf{rp}), (\mathsf{rf})\}$ is chosen to be the set of display rules. Incidentally, the display property is a crucial component in the proof of the general cut-elimination theorem. The interpretation $\mathcal{I}$ of a display sequent as a tense formula is defined as follows.

$$\mathcal{I}(A) = A \text{ for every formula } A \qquad \mathcal{I}(\circ X) = \Box\mathcal{I}(X)$$
$$\mathcal{I}(X, Y) = \mathcal{I}(X) \vee \mathcal{I}(Y) \qquad \mathcal{I}(\bullet X) = \blacksquare\mathcal{I}(X)$$

A modular method of extending a base display calculus for Kt by a large class of axioms inclusive of the Scott-Lemmon axioms was introduced in [14] (see also [5]). Following [14], Goré *et al.* [10] present the rule $d(h, i, j, k)$ corresponding to the Scott-Lemmon axiom $\blacksquare^h \Box^j \overline{p} \vee \Diamond^i \blacklozenge^k p$.

$$\frac{\Gamma, \circ^i\{\bullet^k\{\Delta\}\}}{\Gamma, \bullet^h\{\circ^j\{\Delta\}\}}\ d(h, i, j, k)$$

Theorem 1 ([10,14]). *Let S be any finite set of Scott-Lemmon axioms. $A \in$ Kt+S iff A is derivable in SKT+S', where $S' = \{d(h, i, j, k) | \blacksquare^h \Box^j \overline{p} \vee \Diamond^i \blacklozenge^k p \in S\}$.*

2.2 Labelled Calculi for Tense Logics

Labelled sequents [8,16] generalise Gentzen sequents by the prefixing of *state variables* to formulae occurring in the sequent and by making the relational semantics explicit in the syntax. A labelled sequent has the form $\mathcal{R}, \Gamma$ where the *relation mset* (multiset) $\mathcal{R}$ consists of terms of the form Rxy. Meanwhile Γ is a multiset of labelled formulae (e.g. $x : A \rightarrow B$, $y : p$). A labelled sequent can be viewed as a directed graph (defined using the set $\mathcal{R}$) with formulae decorating each node [19,20].

Negri [17] has presented a method for generating cut-free and contraction-free labelled sequent calculi for the large family of modal logics whose Kripke semantics are defined by geometric (first-order) formulae. The proof of cut-elimination is general in the sense that it applies uniformly to every modal logic defined by geometric formulae. This result has been extended to labelled sequent calculi for intermediate and other non-classical logics [6] and indeed to arbitrary first-order formulae [7]. See also Viganò [21] where non-classical logics with semantics defined by Horn formulae are investigated using cut-free labelled calculi introduced therein.

We begin by extending in the natural way the usual labelled sequent calculus for K to a labelled sequent calculus for Kt.

Definition 3 (The labelled sequent calculus G3Kt [17])

$$\frac{}{\mathcal{R}, x : p, x : \overline{p}, \Gamma}\ (\mathit{id})$$

$$\frac{\mathcal{R}, x : A, x : B, \Gamma}{\mathcal{R}, x : A \lor B, \Gamma}\ (\lor) \qquad \frac{\mathcal{R}, x : A, \Gamma \qquad \mathcal{R}, x : B, \Gamma}{\mathcal{R}, x : A \land B, \Gamma}\ (\land)$$

$$\frac{\mathcal{R}, Ryx, y : A, \Gamma}{\mathcal{R}, x : \blacksquare A, \Gamma}\ (\blacksquare)^* \qquad \frac{\mathcal{R}, Rxy, y : A, \Gamma}{\mathcal{R}, x : \Box A, \Gamma}\ (\Box)^*$$

$$\frac{\mathcal{R}, Ryx, y : A, x : \blacklozenge A, \Gamma}{\mathcal{R}, Ryx, x : \blacklozenge A, \Gamma}\ (\blacklozenge) \qquad \frac{\mathcal{R}, Rxy, y : A, x : \Diamond A, \Gamma}{\mathcal{R}, Rxy, x : \Diamond A, \Gamma}\ (\Diamond)$$

The $(\Box)$ and $(\blacksquare)$ rules have a side condition: $(*)$ the variable y does not occur in the conclusion. When a variable is not allowed to occur in the conclusion of an inference, we refer to it as an *eigenvariable.*

Following the method in [17], the rule $l(h, i, j, k)$ corresponding to the Scott-Lemmon axiom $\blacksquare^h \Box^j \bar{p} \lor \Diamond^i \overset{k}{\blacklozenge} p$ is given below. We use the notation $R^n xz$ to represent a relational sequence Rxy_1, Ry_1y_2, ..., $Ry_{n-1}z$ of length n.

$$\frac{\mathcal{R}, R^i vx, R^k ux, R^h wv, R^j wu, v : \Delta, u : \Delta', \Gamma}{\mathcal{R}, R^h wv, R^j wu, v : \Delta, u : \Delta', \Gamma}\ l(h, i, j, k)^*$$

$(*)$ All variables occurring in the relational atoms $R^i vx, R^k ux$ with the exception of v and u are eigenvariables.

Remark 1. In the rule above, some care is needed in the boundary case when some of the parameters h, i, j, and k are zero. The table below specifies the instances of the rule depending on whether the parameter is greater than zero (marked with $>$), or equal to zero (marked with 0):

h	j	i	k	Premise	Conclusion
$>$	$>$	$>$	$>$	$\mathcal{R}, R^i vx, R^k ux, R^h wv, R^j wu, v : \Delta, u : \Delta', \Gamma$	$\mathcal{R}, R^h wv, R^j wu, v : \Delta, u : \Delta', \Gamma$
0	$>$	$>$	$>$	$\mathcal{R}, R^i vx, R^k ux, R^j vu, v : \Delta, u : \Delta', \Gamma$	$\mathcal{R}, R^j vu, v : \Delta, u : \Delta', \Gamma$
0	$>$	$>$	0	$\mathcal{R}, R^i vu, R^j wu, v : \Delta, u : \Delta', \Gamma$	$\mathcal{R}, R^j vu, v : \Delta, u : \Delta', \Gamma$
$>$	0	0	$>$	$\mathcal{R}, R^k uv, R^h uv, v : \Delta, u : \Delta', \Gamma$	$\mathcal{R}, R^h uv, v : \Delta, u : \Delta', \Gamma$
0	0	$>$	$>$	$\mathcal{R}, R^i vx, R^k vx, v : \Delta, v : \Delta', \Gamma$	$\mathcal{R}, v : \Delta, v : \Delta', \Gamma$
0	0	$>$	0	$\mathcal{R}, R^i vv, v : \Delta, v : \Delta', \Gamma$	$\mathcal{R}, v : \Delta, v : \Delta', \Gamma$
$>$	$>$	$>$	0	$\mathcal{R}, R^i vu, R^h wv, R^j wu, v : \Delta, u : \Delta', \Gamma$	$\mathcal{R}, R^h wv, R^j wu, v : \Delta, u : \Delta', \Gamma$
0	0	0	0	$\mathcal{R}, v : \Delta, u : \Delta', \Gamma$	$\mathcal{R}, v : \Delta, u : \Delta', \Gamma$

Although there are sixteen cases to consider, we only give eight of these as the others are similar. For some entries in the table, the equality symbol that arises ($R^0 uv$ is taken to be $u = v$) has been eliminated by suitable argumentation. This argumentation can be formalised using the equality rules specified by Negri [17]. In particular, when $i = k = 0$ and $h > 0, j > 0$ the rule obtained in this way has the following form.

$$\frac{\mathcal{R}, R^h wv, R^j wv, v : \Delta, v : \Delta', \Gamma}{\mathcal{R}, R^h wv, R^j wu, v : \Delta, u : \Delta', \Gamma}\ l(h,i,j,k)^*$$

Negri [17] does not explicitly consider structural rules of this form (observe how $v : \Delta'$ in the premise becomes $u : \Delta'$ in the conclusion). The results in this paper apply to such rules as well, by extending Negri's arguments in order to justify the elimination of the equality symbol.

The following contraction and weakening rules are admissible [17] in $\mathsf{G3Kt} + l(h,i,j,k)$.

$$\frac{\mathcal{R}, \mathcal{Q}, \mathcal{Q}, \Delta, \Delta, \Gamma}{\mathcal{R}, \mathcal{Q}, \Delta, \Gamma}\ (\mathsf{ctr}) \qquad\qquad \frac{\mathcal{R}, \Gamma}{\mathcal{R}, \mathcal{Q}, \Gamma, \Delta}\ (\mathsf{wk})$$

Theorem 2 ([17]). *Let S be any finite set of Scott-Lemmon axioms. $A \in \mathsf{Kt}{+}S$ iff $x : A$ is derivable in $\mathsf{SKT} + S'$, where $S' = \{l(h,i,j,k) | \blacksquare^h \Box^j \bar{p} \vee \Diamond^i \blacklozenge^k p \in S\}$.*

3 Interpreting a Display Sequent as a Labelled UT

In this section we show how to translate (back and forth) a display sequent into a labelled directed graph whose underlying undirected graph is a tree.

We write $V = V_1 \sqcup V_2$ to mean that $V = V_1 \cup V_2$ and $V_1 \cap V_2 = \emptyset$. The multiset union of multisets M_1 and M_2 is denoted $M_1 \uplus M_2$. A *labelling function* L is a map from a set V to a multiset of tense formulae. For labelling functions L_1 and L_2 on the set V_1 and V_2 respectively, let $L_1 \cup L_2$ be the labelling function on $V_1 \cup V_2$ defined as follows:

$$L_1 \cup L_2(x) = \begin{cases} L_1(x) & x \in V_1, x \notin V_2 \\ L_2(x) & x \notin V_1, x \in V_2 \\ L_1(x) \uplus L_2(x) & x \in V_1, x \in V_2 \end{cases}$$

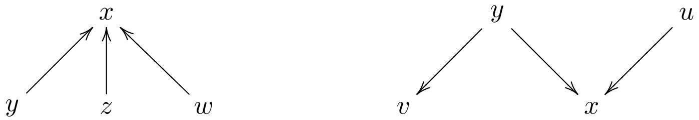
A *labelled graph* (V, E, L) is a directed graph (V, E) $(V \neq \emptyset)$ equipped with a labelling function L on V.

Definition 4 (Labelled graph isomorphism). *We say that two labelled graphs $u_1 = (V_1, E_1, L_1)$ and $u_2 = (V_2, E_2, L_2)$ are isomorphic (written $u_1 \cong u_2$) if and only if there is an isomorphism $f : V_1 \to V_2$ such that:*

(i) for every $x, y \in V_1$, $(x, y) \in E_1$ iff $(fx, fy) \in E_1$
(ii) for every $x \in V$, $L(x) = L(fx)$.

Definition 5 (Labelled UT). *A labelled graph whose underlying (undirected) graph is a tree is termed a UT (underlying tree).*

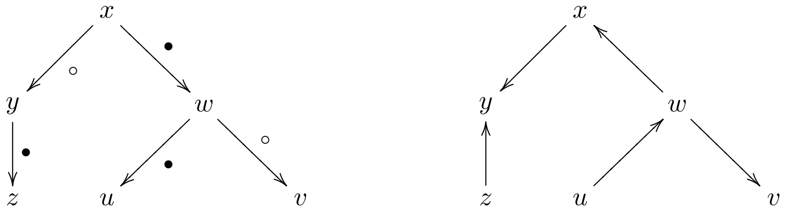
Example 1. Assuming that the nodes are decorated with multisets of formulae, the following two graphs represent labelled UTs:



Interpreting a Display Sequent Γ as a labelled UT. Every display sequent has a natural interpretation as a labelled tree with two types of directed edges: $\overset{\circ}{\to}$ and $\overset{\bullet}{\to}$. If we interpret every directed edge $\alpha \overset{\bullet}{\to} \beta$ as the directed edge $\beta \overset{\circ}{\leftarrow} \alpha$, we can then interpret every display sequent as a connected labelled graph with a *single* type of directed edge (so we can drop the $\circ$ symbol altogether). Moreover, it is easy to see that its underlying graph (i.e. the undirected graph obtained obtained treating all edges as undirected) has no cycles.

Remark 2. Every display sequent Γ can be interpreted naturally as a UT.

Example 2. First interpret the display sequent $A, \circ\{B, \bullet\{\}\}, \bullet\{D, E, \bullet\{F\}, \circ\{G\}\}$ as the labelled tree with two types of directed edges, below left. Next, convert this labelled tree to a labelled graph (with a single type of directed edge) by reading each $\alpha \overset{\bullet}{\to} \beta$ as $\alpha \leftarrow \beta$ (below right).



$L(x) = \{A\}$	$L(y) = \{B\}$	$L(z) = \emptyset$
$L(w) = \{D, E\}$	$L(u) = \{F\}$	$L(v) = \{G\}$

For concreteness let us formally define the map du from a display sequent to a UT. Let $\mathbb{N}^{<\mathbb{N}}$ denote the set of finite sequences on $\mathbb{N}$.

Given $(x) \in \mathbb{N}^{<\mathbb{N}}$ and a display sequent Γ, consider the following recursive definition for $du_{(x)}(\Gamma)$ on the depth of Γ:

1. Base case. $\Gamma = A_1, \ldots, A_M$. A pictorial representation is given below right.

$$du_{(x)}(A_1, \ldots, A_M) = (\{(x)\}, \emptyset, x \mapsto \{A_1, \ldots, A_n\}) \qquad \overset{(x)}{\boxed{A_1, \ldots, A_M}}$$

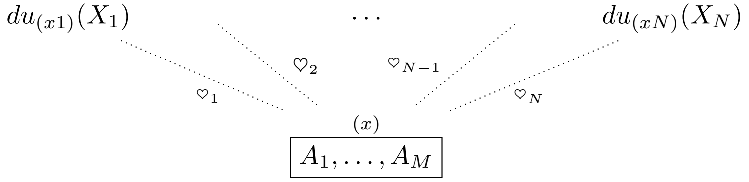
2. Inductive case. $\Gamma = A_1, \ldots, A_M, \heartsuit_1\{X_1\}, \ldots, \heartsuit_N\{X_N\}$ where $\heartsuit_j \in \{\circ, \bullet\}$. Since each $\heartsuit_j\{X_j\}$ has strictly smaller depth than Γ, the following are well-defined:

$$du_{(xj)}(\heartsuit_j\{X_j\}) = (V_j, E_j, L_j) \text{ for } 1 \leq j \leq N$$

Define $du_{(x)}(\Gamma) = (V, E, L)$ such that

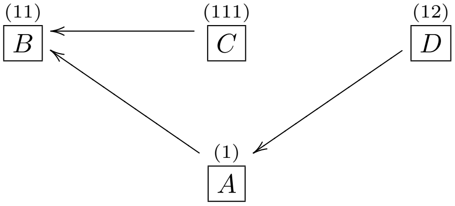
$$V = \{(x)\} \cup V_1 \cup \ldots \cup V_N$$
$$E = \{((x), (xj)) \mid \heartsuit_j = \circ\} \cup \{((xj), (x)) \mid \heartsuit_j = \bullet\} \cup E_1 \cup \ldots \cup E_N$$
$$L = \{(x) \mapsto \{A_1, \ldots, A_M\}\} \cup L_1 \cup \ldots \cup L_N$$

A pictorial representation is given below. The orientation of the arrows is determined by $\heartsuit_j$. If $\heartsuit_j = \circ$ then the arrow directs away from (x); if $\heartsuit_j = \bullet$ then the arrow directs towards (x)



Remark 3. Every comma occurring in the display sequent Γ is associated (though not necessarily a one-to-one association) with a vertex in $du(\Gamma)$.

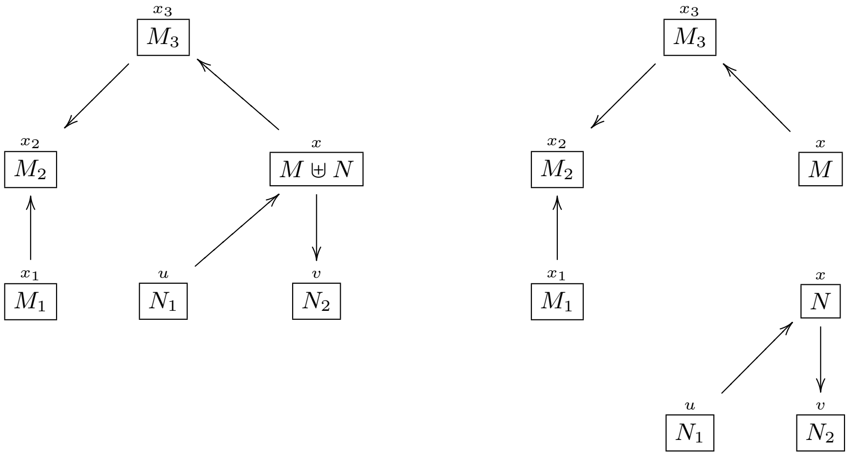
Example 3. Given $\Gamma = A, \circ\{B, \bullet\{C\}\}, \bullet\{D\}$, the UT $du_{(1)}(\Gamma) = (V, E, L)$ is computed below:



Note that in practice we use the more familiar symbols x, y, z, ... to denote labels. The numerical labels are used here for technical convenience.

Definition 6 ($u[v]$ notation). *We write $u[v]$ to mean the labelled graph containing labelled subgraphs $u[\,]$ and v which have a single vertex x in common such that the label of x in $u[v]$ is the union of $L(x)$ from $u[\,]$ and v.*

Example 4. Consider the labelled graph $u[v]$—x is the common vertex between $u[]$ and v—shown below right. Then either of the two labelled graphs below right may be v, and the other will be $u[]$.



If $u[v] = (V, E, L)$, then there exist partitions $V = V_1 \sqcup \{x\} \sqcup V_2$, $E = E_1 \sqcup E_2$, and L_1 and L_2 such that $L = L_1 \cup L_2$, where $u[\,] = (V_1 \sqcup \{x\}, E_1, L_1)$ and $v = (V_2 \sqcup \{x\}, E_2, L_2)$. In particular, $L(x) = L_1(x) \uplus L_2(x)$. Note that when $u[v]$ is a labelled UT, then $u[\,]$ and v must necessarily be labelled UTs.

We have seen that every display sequent defines (up to isomorphism) a labelled UT. With a slight abuse of notation, we will use the display sequent notation to denote a labelled UT. For example, we will write $u[X]$ to mean the labelled graph such that the labelled graph $u[\,]$ and the labelled UT $du(X)$ are subgraphs with a single common vertex. The context will make it clear if we are referring to a display sequent or a labelled UT.

The translation from a display sequent to a labelled UT extends naturally to a translation from a display sequent rule to a labelled UT rule. This leads us to the definition of the following calculus.

Definition 7 (UT calculus). *Every sequent in this calculus is a labelled UT.*

$$\frac{}{u[p, \overline{p}]}\,(\mathit{id})_u \qquad \frac{u[A] \qquad u[B]}{u[A \wedge B]}\,(\wedge)_u \qquad \frac{u[A, B]}{u[A \vee B]}\,(\vee)_u$$

$$\frac{A, \circ\{X\}}{\blacksquare A, X}\,(\blacksquare)_u \quad \frac{u[\circ\{\Delta, A\}, \Diamond A]}{u[\circ\{\Delta\}, \Diamond A]}\,(\Diamond)_u \quad \frac{u[\circ\{\Delta, \blacklozenge A\}, A]}{u[\circ\{\Delta, \blacklozenge A\}]}\,(\blacklozenge)_u$$

$$\frac{u[\circ\{A\}]}{u[\Box A]}\,(\Box)_u \qquad \frac{u[\Gamma]}{u[\Gamma, \Delta]}\,(\mathit{wk})_u \qquad \frac{u[\Delta, \Delta]}{u[\Delta]}\,(\mathit{ctr})_u$$

For convenience, we drop the subscript (x) and write du for $du_{(x)}$.

Recall that $\mathsf{SKT} + d(h,i,j,k)$ (see below left) is a calculus for the extension of Kt with the Scott-Lemmon axiom $\blacksquare^h \Box^j \bar{p} \vee \Diamond^i \blacklozenge^k p$. We define the UT rule $u(h,i,j,k)$ as below right.

$$\frac{\Gamma, \circ^i\{\bullet^k\{\Delta\}\}}{\Gamma, \bullet^h\{\circ^j\{\Delta\}\}}\; d(h,i,j,k) \qquad\qquad \frac{u[\circ^i\{\bullet^k\{\Delta\}\}]}{u[\bullet^h\{\circ^j\{\Delta\}\}]}\; u(h,i,j,k)$$

Since display sequents may be interpreted as trees with two types of edges ($\circ$-edges and $\bullet$-edges), they possess a root node, whereas UTs do not possess a root in general. Nevertheless, the underlying tree structure of a UT permits us to view any node as the root, and the lemma below ensures that we obtain deductively equivalent labelled UTs via the residuation rules regardless of the node where we begin the translation.

Lemma 1. *For every Γ and Δ: $du(\Gamma, \circ\{\Delta\}) \cong du(\bullet\{\Gamma\}, \Delta)$*

Proof. Let $(V, E, L) = du(\Gamma, \circ\{\Delta\})$. Then there exists $x, y \in V$ and $(x, y) \in E$ such that $V = V_1 \sqcup \{x\} \sqcup V_2 \sqcup \{y\}$ and $E = E_1 \sqcup E_2 \sqcup \{(x, y)\}$ and $du(\Gamma) = (V_1 \sqcup \{x\}, E_1, L|_{V_1 \sqcup \{x\}})$ and $du(\Delta) = (V_2 \sqcup \{y\}, E_2, L|_{V_2 \sqcup \{y\}})$. In particular, observe that comma displayed in $\Gamma, \circ\{\Delta\}$ corresponds to x and the nesting where Δ occurs corresponds to y.

Now consider $du(\bullet\{\Gamma\}, \Delta) = (V', E', L')$. There exists $u, v \in V'$ and $(u, v) \in E$ such that $V' = V'_1 \sqcup \{u\} \sqcup V'_2 \sqcup \{y\}$ and $E' = E'_1 \sqcup E'_2 \sqcup \{(u, v)\}$ and $du(\Gamma) = (V'_1 \sqcup \{u\}, E'_1, L'|_{V'_1 \sqcup \{u\}})$ and $du(\Delta) = (V'_2 \sqcup \{v\}, E'_2, L'|_{V'_2 \sqcup \{v\}})$. In particular, observe that comma displayed in $\bullet\{\Gamma\}, \Delta$ corresponds to v and the nesting where Γ occurs corresponds to u.

It follows that there are isomorphisms witnessing each of the following such that x maps to u (first line) and y maps to v (second line).

$$\begin{aligned} du(\Gamma) = (V_1 \sqcup \{x\}, E_1, L|_{V_1 \sqcup \{x\}}) &\cong (V'_1 \sqcup \{u\}, E'_1, L'|_{V'_1 \sqcup \{u\}}) = du(\Gamma) \\ du(\Delta) = (V_2 \sqcup \{y\}, E_2, L|_{V_2 \sqcup \{y\}}) &\cong (V'_2 \sqcup \{v\}, E'_2, L'|_{V'_2 \sqcup \{v\}}) = du(\Delta) \end{aligned}$$

Taking the graph union of these disjoint graphs:

$$\begin{aligned} &(V_1 \sqcup \{x\} \sqcup V_2 \sqcup \{y\}, E_1 \sqcup E_2, L|_{V_1 \sqcup \{x\}} \cup L|_{V_2 \sqcup \{y\}}) \cong \\ &\qquad\qquad (V'_1 \sqcup \{u\} \sqcup V'_2 \sqcup \{y\}, E'_1 \sqcup E'_2, L'|_{V'_1 \sqcup \{u\}} \cup L'|_{V'_2 \sqcup \{v\}}) \end{aligned}$$

Adding the edge (x, y) on the left and (u, v) on the right, we get $(V, E, L) \cong (V', E', L')$.

Interpreting a Labelled UT as a Display Sequent. Given a UT $u = \langle V, E, L \rangle$ we first pick a vertex $x \in V$ to compute the display sequent $ud_x(u)$. If $E = \emptyset$, then $ud(u) = L(x)$ is the desired display sequent. Otherwise, for all n forward looking edges $(x, y_i) \in E$ (with $1 \leq i \leq n$) where y_i is the common label of $u = u[v_i]$ and v_i, and for all k backward looking edges $(z_j, x) \in E$ (with

$1 \leq j \leq k$) where z_j is the common label of $u = u[w_j]$ and w_j, we define the image of $ud_x(u)$ as the display sequent

$$L(x), \circ\{ud_{y_1}(v_1)\}, \ldots, \circ\{ud_{y_n}(v_n)\}, \bullet\{ud_{z_1}(w_1)\}, \ldots, \bullet\{ud_{z_k}(w_k)\}$$

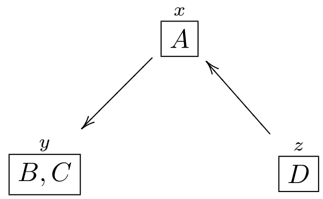
Since the UTs $v_1, \ldots, v_n, w_1, \ldots, w_k$ are smaller than u, the recursive definition of ud is well-founded.

Lemma 2. *For any UT $u = \langle V, E, L\rangle$, and for any vertices $x, y \in V$, the display sequent $ud_x(u)$ is derivable from $ud_y(u)$ via the residuation rules* (rf) *and* (rp).

Proof. Follows by Lemma 1.

When translating a labelled UT we must choose a vertex as the starting point of our translation. This lemma states that all display sequents obtained from choosing a different vertex are mutually derivable from one another. In fact, all such display sequents are display equivalent, meaning they are derivable from each other by use of the residuation rules (rp) and (rf) only. To clarify the translation procedure, we provide an example below of the various display sequents obtained from translating at a different vertex initially.

Example 5. Suppose we are given the labelled UT $u = \langle V, E, L\rangle$ where $V = \{x, y, z\}$, $E = \{(x, y), (z, x)\}$, $L(x) = \{A\}$, $L(y) = \{B, C\}$, and $L(z) = \{D\}$. A pictorial representation of the labelled UT u is given on the left with the corresponding display sequent translations on the right:



$$ud_x(u) = A, \circ\{B, C\}, \bullet\{D\}$$
$$ud_y(u) = B, C, \bullet\{A, \bullet\{D\}\}$$
$$ud_z(u) = D, \circ\{A, \circ\{B, C\}\}$$

When providing the construction of an effective translation between display calculus proofs and UT calculus proofs, we make use of the notation Γ, Δ for display sequents and $u[v]$ for corresponding labelled UTs (under the translation). The following lemma ensures that the pieces of the display sequent Γ and Δ, and the pieces of the labelled UT $u[]$ and v, correctly map to each other under our translation functions.

Lemma 3

(i) For every Γ and Δ, $du(\Gamma, \Delta)$ is the UT $u[v]$, where v is the UT $du(\Delta)$ and $u[\,]$ is the UT $du(\Gamma)$.

(ii) For every UT $u[v]$, $ud(u[v])$ is the display sequent Γ, Δ (up to display equivalence) where $\Gamma = ud(u[\,])$ and $\Delta = ud(v)$.

Proof. By construction of du and ud.

Theorem 3 (Translating derivations: $\mathsf{SKT} + S$ and UT calculus+S'). *Let S be any finite set of $d(h,i,j,k)$ rules and S' be the set $\{u(h,i,j,k) | d(h,i,j,k) \in S\}$. Then:*

(i) Let δ be a derivation of Γ in $\mathsf{SKT}+S$. Then there is an effective translation of δ to a derivation δ' of $du(\Gamma)$ in the UT calculus with S'.
(ii) Let δ be a derivation of the labelled UT u in the UT calculus with S'. Then there is an effective translation of δ to a derivation of $ud(g)$ in $\mathsf{SKT}+S$.

Proof. (i) Induction on the height of δ.

Base case. $du(\Gamma, p, \bar{p})$ is a UT of the form $u[p, \bar{p}]$ (Lemma 3(i)) and is hence an initial sequent in the UT calculus.

Inductive case. It suffices to simulate each rule instance of SKT in the UT calculus. Every rule in SKT other than (rf), (rp), $(\blacksquare)$ and $(\blacklozenge)$ has the form below left for suitable Y_1 and Y_0; moreover, there is a corresponding rule in the UT calculus as shown below right.

$$\frac{\Gamma, Y_1}{\Gamma, Y_0}\,(\mathsf{r}) \qquad \frac{u[\Gamma, Y_1]}{u[\Gamma, Y_0]}\,(\mathsf{r})_u$$

The induction hypothesis gives us a derivation of $du(\Gamma, Y_1) = u[\Gamma, Y_1]$. Applying $(\mathsf{r})_u$ we get $u[\Gamma, Y_0] = du(\Gamma, Y_0)$ as required.

We consider the remaining rules below.

$$\frac{\Gamma, \circ\{\Delta\}}{\bullet\{\Gamma\}, \Delta}\,(\mathsf{rf}) \qquad \begin{array}{c} du_x(\Gamma, \circ\{\Delta\}) \\ \hdashline \cong du_x(\bullet\{\Gamma\}, \Delta) \end{array} \text{ Lem. 1}$$

$$\frac{\Gamma, \bullet\{\Delta\}}{\circ\{\Gamma\}, \Delta}\,(\mathsf{rp}) \qquad \begin{array}{c} du_x(\Gamma, \bullet\{\Delta\}) \\ \hdashline \cong du_x(\circ\{\Gamma\}, \Delta) \end{array} \text{ Lem. 1}$$

$$\frac{\Gamma, \bullet\{A\}}{\Gamma, \blacksquare A}\,(\blacksquare) \qquad \begin{array}{c} du_x(\Gamma, \bullet\{A\}) \\ \hdashline \dfrac{\circ\{\Gamma\}, A}{\Gamma, \blacksquare A}\,(\blacksquare) \end{array}$$

$$\frac{\Gamma, \bullet\{\Delta, A\}, \blacklozenge A}{\Gamma, \bullet\{\Delta\}, \blacklozenge A}\,(\blacklozenge) \qquad \begin{array}{c} du_x(\Gamma, \bullet\{\Delta, A\}, \blacklozenge A) \\ \hdashline \dfrac{\Delta, A, \circ\{\Gamma, \blacklozenge A\}}{\Delta, \circ\{\Gamma, \blacklozenge A\}}\,(\blacklozenge) \\ \hdashline du(\Gamma, \bullet\{\Delta\}, \blacklozenge A) \end{array}$$

(ii) Induction on the height of δ. The argument is similar to the above case and uses Lemma 3(ii).

4 From Labelled UTs to Labelled Sequents

We identify a subclass of labelled sequents which we call $\mathsf{G3Kt}(UT)$ *sequents*, and prove that they correspond to labelled UT graphs. Due to the relations of the latter with the display calculi shown in the previous section, it follows that every derivation in the $\mathsf{SKT} + u(h,i,j,k)$ calculus corresponds to a derivation in the labelled calculus restricted to $\mathsf{G3Kt}(UT)$ sequents.

Transforming a labelled UT $u = (V,E,L)$ into a labelled sequent $R, -$. Define $\mathcal{R} = \{Rxy | (x, y) \in E\}$ and

$$\Gamma = \biguplus_{x \in V, L(x) \neq \emptyset} x : L(x)$$

where $x : L(x)$ represents the multiset $L(x)$ with each formula prepended with a label x.

Example 6. The UT $u = \langle V, E, L\rangle$ where $V = \{x, y, z\}$, $E = \{(x, y), (z, x)\}$, $L(x) = \{A\}$, $L(y) = \{B\}$, and $L(z) = \{C\}$ corresponds to the labelled sequent $Rxy, Rzx, x : A, y : B, z : C$.

Transforming a labelled sequent $\mathcal{R}, \Gamma$ into a labelled graph (V,E,L). Let V be the set of all labels occurring in $\mathcal{R}, \Gamma$. Define

$$E = \{(x, y) | Rxy \in \mathcal{R}\} \quad L(x) = \{\text{multiset of formulae with label } x \text{ in } \Gamma\}$$

Example 7. The labelled sequent $Rxy, Ryz, Rux, x : A, z : B, z : C, u : D$ becomes the UT $u = \langle V, E, L\rangle$ where $V = \{x, y, z, u\}$, $E = \{(x, y), (y, z), (u, x)\}$, $L(x) = \{A\}$, $L(y) = \emptyset$, $L(z) = \{B, C\}$ and $L(u) = \{D\}$.

The reader will observe that the translations are obtained rather directly. This is because the main difference between a labelled graph and a labelled sequent is notation. The main step of the translation was already established in the previous section. Our interest in this work is the image of a display sequent in the labelled calculus. This motivates the following definitions.

Definition 8 (G3Kt(UT) sequent). *A labelled sequent whose image (under the above translation) is a labelled UT is called a G3Kt(UT)* sequent.

Definition 9 (G3Kt(UT) calculus). *Define the calculus G3Kt(UT) to be the labelled calculus restricted to G3Kt(UT) sequents and with weakening and contraction defined as follows:*

$$\frac{\mathcal{R}, \Gamma}{\mathcal{R}, \mathcal{Q}, \Delta, \Gamma} \, (\mathsf{wk})^*_{ul} \qquad \frac{\mathcal{R}, \mathcal{Q}, \hat{\mathcal{Q}}, \Delta, \hat{\Delta}, \Gamma}{\mathcal{R}, \mathcal{Q}, \Delta, \Gamma} \, (\mathsf{ctr})^*_{ul}$$

Weakening has the side condition that the conclusion must be a G3Kt(UT)-sequent. Contraction possesses side conditions that ensure it behaves just as the $(\mathsf{ctr})_u$ rule:

1. *The labelled graph of $\hat{\mathcal{Q}}, \hat{\Delta}$ must be isomorphic to the labelled graph of $\mathcal{Q}, \Delta$.*
2. *The conclusion must be a G3Kt(UT)-sequent.*
3. *Both $\mathcal{Q}, \Delta$ and $\hat{\mathcal{Q}}, \hat{\Delta}$ form labelled UTs that share a root, and all other variables in $\hat{\mathcal{Q}}, \hat{\Delta}$ do not appear in the conclusion of the inference, i.e. they are eigenvariables.*

We use the notation $(\mathsf{r})_{ul}$ *to indicate the remaining inference rules of* G3Kt(UT).

For $h, i, j, k \in \mathbb{N}$, define $ul(h, i, j, k)$ as follows:

$$\frac{\mathcal{R}, R^i vx, R^k ux, v : \Delta, u : \Delta', \Gamma}{\mathcal{R}, R^h wv, R^j wu, v : \Delta, u : \Delta', \Gamma}\ ul(h, i, j, k)^*$$

The asterisk indicates the following side conditions: (i) all variables occurring in $R^i vx, R^k ux$ with the exception of v and u are eigenvariables and (ii) all variables occurring in $R^h wv, R^j wu$ with the exception of v and u are fresh.

Remark 4. Similar to the presentation of the $l(h, i, j, k)$ rules (cf. Remark 1), we provide the table below showing the different instances of the rule depending on the values of the parameters h, i, j, and k. The reduction in cases is due to the fact that we allow the $ul(h, i, j, k)$ rules to relabel formulae from premise to conclusion–an action which is not allowed for the $l(h, i, j, k)$ rules.

i	k	Premise
>	>	$\mathcal{R}, R^i vx, R^k ux, v : \Delta, u : \Delta', \Gamma$
0	>	$\mathcal{R}, R^k uv, v : \Delta, u : \Delta', \Gamma$
>	0	$\mathcal{R}, R^i vu, v : \Delta, u : \Delta', \Gamma$
0	0	$\mathcal{R}, v : \Delta, v : \Delta', \Gamma$

h	j	Conclusion
>	>	$\mathcal{R}, R^h wv, R^j wu, v : \Delta, u : \Delta', \Gamma$
0	>	$\mathcal{R}, R^j wu, w : \Delta, u : \Delta', \Gamma$
>	0	$\mathcal{R}, R^h wv, v : \Delta, w : \Delta', \Gamma$
0	0	$\mathcal{R}, w : \Delta, w : \Delta', \Gamma$

To see that the G3Kt(UT) + $ul(h, i, j, k)$ calculus is well-defined, it suffices to observe that the conclusion of every G3Kt rule is a G3Kt(UT) sequent given that the premise(s) is (are) G3Kt(UT) sequents.

Lemma 4. *If the premise of a* G3Kt(UT) + $ul(h, i, j, k)$ *inference is a* G3Kt(UT)*-sequent, then the conclusion is an* G3Kt(UT)*-sequent.*

Proof. We argue the result for the $(\mathsf{wk})_{ul}$, $(\mathsf{ctr})_{ul}$, $(\blacksquare)_{ul}$, and $ul(h, i, j, k)$ rules since all other cases are similar or trivial.

Case 1 and 2. These cases follow from the side conditions on the $(\mathsf{wk})_{ul}$ and $(\mathsf{ctr})_{ul}$ rules, which only allow application of the rule when the result is a G3Kt(UT) sequent.

Case 3. Assume that $\mathcal{R}, Ryx, y : A, \Gamma$ is a G3Kt(UT)-sequent and that $u = \langle V, E, L \rangle$ is the corresponding UT. Since y is an eigenvariable, the conclusion $\mathcal{R}, x : \blacksquare A, \Gamma$ gives a labelled graph $u' = \langle V', E', L' \rangle$ where $V' = V - \{y\}$, $E' = E - \{(y, x)\}$, $L'(y)$ is undefined, $L'(x)$ is equal to $L(x)$ extended with $x \mapsto \{\blacksquare A\}$, and L' is equal to L for all other labels in V'.

Case 4. We prove the claim for when $h, i, j, k > 0$ since other cases are similar. Assume that the premise $\mathcal{R}, R^i xy, R^k zy, \Gamma$ is a G3Kt(UT)-sequent with all variables y_m strictly between x and z eigenvariables. Observe that in $u = \langle V, E, L \rangle$ there is a path of length $i + k$ from the node x to z where the first i edges are forward looking, and the last k edges are backwards looking. Observe that the UT $u' = \langle V', E', L' \rangle$ of the conclusion $\mathcal{R}, R^h wx, R^j wz, \Gamma$ will contain a path of length $h + j$ from the node x to z where the first h edges are backwards looking, and the last j edges are forwards looking. Due to the eigenvariable condition on all nodes y_m strictly between x and z, it cannot be the case that an edge given by $\mathcal{R}$ contains a label y_m, and it must be the case that $L(y_m) = \emptyset$ (thus ensuring u' is connected). Also, all new nodes along the $h+j$-path strictly between x and z will be fresh (thus ensuring u' is free of cycles). Hence, u' will be a UT.

Lemma 5 (Translating derivations: $\mathsf{G3Kt}(UT)+S$ and UT calculus$+S'$). *Let S be any finite set of $ul(h,i,j,k)$ rules and $S' = \{u(h,i,j,k) | ul(h,i,j,k) \in S\}$. Then*

(i) Let δ be a derivation of $x : A$ in $\mathsf{G3Kt}(UT) + S$. Then there is an effective translation of δ to a derivation δ' of A in the UT calculus$+S'$.
(ii) Let δ be a derivation of A in the UT calculus$+S'$. Then there is an effective translation of δ to a derivation δ' of $x : A$ in $\mathsf{G3Kt}(UT) + S$.

Proof. Follows from the observation that the translation of every rule instance in $\mathsf{G3Kt}(UT) + S$ is a rule instance in the UT calculus$+S'$ and *vice versa.*

Combining the previous results we obtain:

Theorem 4 (Translating derivations: $\mathsf{SKT} + S$ and $\mathsf{G3Kt}(UT) + S'$). *Let S be any finite set of $d(h,i,j,k)$ rules and $S' = \{ul(h,i,j,k)|d(h,i,j,k) \in S\}$. Then*

1. *Let δ be a derivation of A in $\mathsf{SKT} + S$. Then there is an effective translation of δ to a derivation δ' of $x : A$ in $\mathsf{G3Kt}(UT) + S'$.*
2. *Let δ be a derivation of $x : A$ in $\mathsf{G3Kt}(UT) + S'$. Then there is an effective translation δ to a derivation δ' of A in $\mathsf{SKT} + S$.*

Proof. Immediate from Theorem 3 and Lemma 5.

Example 8. Below we translate a derivation of $\blacksquare\Box\bar{p} \vee \Diamond\blacklozenge p$ in $\mathsf{SKT} + d(1,1,1,1)$ to a derivation in $\mathsf{G3Kt}(UT) + ul(1,1,1,1)$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\blacklozenge p, \bullet\{p,\bar{p}\}, \bullet\{\Diamond\blacklozenge p\}}{\blacklozenge p, \bullet\{\bar{p}\}, \bullet\{\Diamond\blacklozenge p\}}\,(\blacklozenge)}{\circ\{\blacklozenge p, \bullet\{\bar{p}\}\}, \Diamond\blacklozenge p}\,(\text{rp})}{\circ\{\bullet\{\bar{p}\}\}, \Diamond\blacklozenge p}\,(\Diamond)}{\bullet\{\circ\{\bar{p}\}\}, \Diamond\blacklozenge p}\,d(1,1,1,1)}{\circ\{\bar{p}\}, \circ\{\Diamond\blacklozenge p\}}\,(\text{rp})}{\Box\bar{p}, \circ\{\Diamond\blacklozenge p\}}\,(\Box)}{\bullet\{\Box\bar{p}\}, \Diamond\blacklozenge p}\,(\text{rf})}{\blacksquare\Box\bar{p}, \Diamond\blacklozenge p}\,(\blacksquare)}{\blacksquare\Box\bar{p} \vee \Diamond\blacklozenge p}\,(\vee)$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{Rzu, Rxu, u : \blacklozenge p, z : p, z : \bar{p}, x : \Diamond\blacklozenge p}{Rzu, Rxu, u : \blacklozenge p, z : \bar{p}, x : \Diamond\blacklozenge p}\,(\blacklozenge)_{ul}}{Rzu, Rxu, z : \bar{p}, x : \Diamond\blacklozenge p}\,(\Diamond)_{ul}}{Ryz, Ryx, z : \bar{p}, x : \Diamond\blacklozenge p}\,ul(1,1,1,1)}{Ryx, y : \Box\bar{p}, x : \Diamond\blacklozenge p}\,(\Box)_{ul}}{x : \blacksquare\Box\bar{p}, x : \Diamond\blacklozenge p}\,(\blacksquare)_{ul}}{x : \blacksquare\Box\bar{p} \vee \Diamond\blacklozenge p}\,(\vee)_{ul}$$

5 Labelled UTs vs Labelled Sequents

In the previous sections, we observed how to embed the display calculus $\mathsf{SKT}+S$ (for a finite set S of $d(h,i,j,k)$ rules) in the labelled calculus formalism, in particular, as a proper fragment, which we called $\mathsf{G3Kt}(UT)+S'$ ($S'=\{ul(h,i,j,k)|d(h,i,j,k)\in S\}$). Indeed, an $\mathsf{G3Kt}(UT)$-sequent is a severe restriction of a labelled sequent since the underlying graph in the former is restricted to a tree. For example, $Rxy, Ryx, x : A, z : B$ is a labelled sequent in the traditional sense, but fails to be a $\mathsf{G3Kt}(UT)$-sequent since it contains the relational cycle Rxy, Ryx and is not connected due to $z : B$. As a result we have two distinct labelled calculi for Scott-Lemmon extensions of Kt. In this section we investigate the natural question that arises: what is the relationship between these calculi? As seen below, the labelled calculus simulates $\mathsf{G3Kt}(UT)+S'$, despite the slightly different rules (i.e. $ul(h,i,j,k)$) used by the latter to capture the Scott-Lemmon axioms. The next question is therefore whether the converse also holds, that is, whether the two calculi can represent the same proofs. In the case of the normal minimal tense logic Kt the answer is affirmative.

From $\mathsf{G3Kt}(UT)+ul(h,i,j,k)$ to $\mathsf{G3Kt}+l(h,i,j,k)$. As observed in Remark 1, the structural rules corresponding to $i=k=0$ and $h>0, j>0$ do not match the form of the rules given in [17] (and hence the results in [17] need to be suitably extended to apply to this case). Although an effective translation from $\mathsf{G3Kt}(UT)+ul(h,i,j,k)$ to $\mathsf{G3Kt}+l(h,i,j,k)$ can be defined for this case, in the following we restrict ourselves to $i>0$ or $k>0$.

Lemma 6 (*[17], Sect. 4*)**.** *The calculus $\mathsf{G3Kt}+l(h,i,j,k)$ admits height preserving substitution of variables.*

In the following, the requirement that $i>0$ or $k>0$ may be dropped by a slight extension of Negri's arguments (see after Remark 1).

Theorem 5. *Let δ be a derivation of $x : A$ in $\mathsf{G3Kt}(UT)+ul(h,i,j,k)$, with $i>0$ or $k>0$. Then there is an effective translation of δ to a derivation δ' of $x : A$ in $\mathsf{G3Kt}+l(h,i,j,k)$.*

Proof. We prove the result by induction on the height of the derivation δ.

Base case. It is easy to see that initial sequents of $\mathsf{G3Kt}(UT)$ are initial sequents of $\mathsf{G3Kt}$.

Inductive step. We show the inductive step for four instances $ul(h,i,j,k)$, $ul(h,i,0,k)$, $ul(0,i,0,k)$, and $ul(0,i,0,0)$ $(h,i,j,k>0)$. We also show the inductive step for the $(\mathsf{ctr})_{ul}$ rule. The translation of the other rules is trivial.

$$\frac{\mathcal{R}, R^i vx, R^k ux, v : \Delta, u : \Delta', \Gamma}{\mathcal{R}, R^h wv, R^j wu, v : \Delta, u : \Delta', \Gamma} \qquad \dfrac{\dfrac{\mathcal{R}, R^i vx, R^k ux, v : \Delta, u : \Delta', \Gamma}{\mathcal{R}, R^h wv, R^j wu, R^i vx, R^k ux, v : \Delta, u : \Delta', \Gamma}\,(\mathsf{wk})}{\mathcal{R}, R^h wv, R^j wu, v : \Delta, u : \Delta', \Gamma}\,l(h,i,j,k)$$

$$\dfrac{\mathcal{R}, R^i vx, R^k ux, v : \Delta, u : \Delta', \Gamma}{\mathcal{R}, R^h wv, v : \Delta, w : \Delta', \Gamma} \qquad \dfrac{\dfrac{\mathcal{R}, R^i vx, R^k ux, v : \Delta, u : \Delta', \Gamma}{\mathcal{R}, R^h uv, R^i vx, R^k ux, v : \Delta, u : \Delta', \Gamma}\,(\mathsf{wk})}{\mathcal{R}, R^h uv, v : \Delta, u : \Delta', \Gamma}\,l(h,i,0,k)$$

$$\dfrac{\mathcal{R}, R^i vx, R^k ux, v : \Delta, u : \Delta', \Gamma}{\mathcal{R}, v : \Delta, v : \Delta', \Gamma} \qquad \dfrac{\dfrac{\mathcal{R}, R^i vx, R^k ux, v : \Delta, u : \Delta', \Gamma}{\mathcal{R}, R^i vx, R^k vx, v : \Delta, v : \Delta', \Gamma}\,\text{lem. 6}}{\mathcal{R}, v : \Delta, v : \Delta', \Gamma}\,l(0,i,0,k)$$

$$\dfrac{\mathcal{R}, R^i vu, v : \Delta, u : \Delta', \Gamma}{\mathcal{R}, v : \Delta, v : \Delta', \Gamma} \qquad \dfrac{\dfrac{\mathcal{R}, R^i vu, v : \Delta, u : \Delta', \Gamma}{\mathcal{R}, R^i vv, v : \Delta, v : \Delta', \Gamma}\,\text{lem. 6}}{\mathcal{R}, v : \Delta, v : \Delta', \Gamma}\,l(0,i,0,0)$$

$$\dfrac{\mathcal{R}, \mathcal{Q}, \hat{\mathcal{Q}}, \Delta, \hat{\Delta}, \Gamma}{\mathcal{R}, \mathcal{Q}, \Delta, \Gamma} \qquad \dfrac{\dfrac{\mathcal{R}, \mathcal{Q}, \hat{\mathcal{Q}}, \Delta, \hat{\Delta}, \Gamma}{\mathcal{R}, \mathcal{Q}, \mathcal{Q}, \Delta, \Delta, \Gamma}\,\text{lem. 6}}{\mathcal{R}, \mathcal{Q}, \Delta, \Gamma}\,(\mathsf{ctr})$$

Example 9. The derivation of $x : \blacksquare\Box\bar{p} \vee \Diamond\blacklozenge p$ in $\mathsf{G3Kt}(UT) + ul(1,1,1,1)$ (see Example 8) can be transformed into a derivation in $\mathsf{G3Kt}+l(1,1,1,1)$ as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{Rzu, Rxu, u : \blacklozenge p, z : p, z : \bar{p}, x : \Diamond\blacklozenge p}{Rzu, Rxu, u : \blacklozenge p, z : \bar{p}, x : \Diamond\blacklozenge p}\,(\blacklozenge)}{Rzu, Rxu, z : \bar{p}, x : \Diamond\blacklozenge p}\,(\Diamond)}{Rzu, Rxu, Ryz, Rzx, z : \bar{p}, x : \Diamond\blacklozenge p}\,(\text{w})}{Ryz, Ryx, z : \bar{p}, x : \Diamond\blacklozenge p}\,l(1,1,1,1)}{Ryx, y : \Box\bar{p}, x : \Diamond\blacklozenge p}\,(\Box)}{x : \blacksquare\Box\bar{p}, x : \Diamond\blacklozenge p}\,(\blacksquare)}{x : \blacksquare\Box\bar{p} \vee \Diamond\blacklozenge p}\,(\vee)$$

From $\mathsf{G3Kt}$ + $l(h, i, j, k)$ to $\mathsf{G3Kt}(UT)$+$ul(h, i, j, k)$. Consider now the converse direction. Let S be a finite set of Scott-Lemmon axioms and define

$$S_{ul} = \{ul(h,i,j,k) \mid \blacksquare^h \Box^j \bar{p} \vee \Diamond^i \blacklozenge^k p \in S\}$$
$$S_l = \{l(h,i,j,k) \mid \blacksquare^h \Box^j \bar{p} \vee \Diamond^i \blacklozenge^k p \in S\}$$

Given a derivation δ in $\mathsf{G3Kt} + S_l$, in general δ will not be a derivation in $\mathsf{G3Kt}(UT) + S_{ul}$ because some sequents in δ (possibly even the endsequent) may not be a $\mathsf{G3Kt}(UT)$-sequent. A more meaningful question is: given a derivation of $x : A$ in $\mathsf{G3Kt} + S_l$, is there a derivation of $x : A$ in $\mathsf{G3Kt}(UT) + S_{ul}$ that is *effectively related to* δ? The constraint that the new derivation is "effectively related" is crucial, for otherwise one could trivially relate δ with the derivation δ' obtained from the following equivalence:

$$\vdash^{\delta}_{\mathsf{G3Kt}+S_l} x : A \text{ iff } A \in \mathsf{Kt} + \blacksquare^h \Box^j \bar{p} \vee \Diamond^i \blacklozenge^k p \text{ iff } \exists \delta'. \vdash^{\delta'}_{\mathsf{G3Kt}(UT)+S_{ul}} x : A$$

Although the phrase 'effectively related' has not been explicitly defined, what we envisage is a local (i.e. rule by rule) transformation on δ, which is sensitive to its structure, that ultimately yields a $\mathsf{G3Kt}(UT) + S_{ul}$ derivation of $x : A$. Notice that the $\mathsf{G3Kt}(UT) + S_{ul}$ derivation obtained via the above argument is not sensitive to the input in the sense that any two $\mathsf{G3Kt}+S_l$ derivations of $x : A$ would be mapped to the same $\mathsf{G3Kt}(UT) + S_{ul}$ derivation.

In the boundary case for Kt when $S = S_l = S_{ul} = \emptyset$ we have the following result, which also establishes that $\mathsf{G3Kt}$ is an internal calculus with respect to derivations that end with a single formula.

Proposition 1. *Every labelled derivation in G3Kt of $x : A$ is also a derivation in G3Kt(UT).*

Proof. We argue by contradiction. Let δ be a derivation of $x : A$ in $\mathsf{G3Kt}$ and suppose there is a labelled sequent $\mathcal{R}, \Gamma$ in δ that is not a $\mathsf{G3Kt}(UT)$-sequent. This means that the underlying graph of $\mathcal{R}$ is not a tree. If $\mathcal{R}$ is not connected, then by inspection of the rules of $\mathsf{G3Kt}$, the underlying graph of every sequent below it (and hence $x : A$) would not be connected and this is a contradiction. On the other hand, if $\mathcal{R}$ is connected and its underlying graph is not a tree, then the underlying graph must contain a cycle. This follows from the fact that $\mathcal{R}$ is assumed connected, and the fact that any acyclic connected graph forms a tree. This means that there exist x, y, w such that $\{Rxw, Ryw\} \subseteq \mathcal{R}$. By inspection of the rules of $\mathsf{G3Kt}$, every sequent below $\mathcal{R}, \Gamma$ will contain this cycle contradicting the assumption that $x : A$ is the end sequent.

This argument does not work for *extensions of* $\mathsf{G3Kt}$ because the additional structural rules may be capable of removing cycles in the following sense: the underlying (i.e. undirected) graph of the premise might have a cycle yet the underlying graph of the conclusion might not (this was not the case for any rule in $\mathsf{G3Kt}$). Indeed, consider the rule for transitivity:

$$\frac{\mathcal{R}, Rxy, Ryz, Rxz, \Gamma}{\mathcal{R}, Ryz, Rxz, \Gamma}\ (\mathsf{Trans})$$

In a rule instance of (Trans), the underlying graph of the premise necessarily contains a cycle. However, it need not be the case that the underlying graph of the conclusion contains a cycle. As a consequence, a labelled derivation of $x : A$ in $\mathsf{G3Kt} + (\mathsf{Trans})$ may contain sequents whose underlying graph is not a tree. Such a derivation cannot be a derivation in any UT calculus.

It is tempting to replace (Trans) with its non-invertible form in order to remove the cycle. However the (ctr) rule seems not to be admissible in $\mathsf{G3Kt} + (\mathsf{Trans}')$ which means that it needs to be included to ensure completeness. The simulation of the above rule instance in $\mathsf{G3Kt}+$ (ctr) + (Trans') below right indicates that we have merely shifted the problem to a new setting.

$$\frac{\mathcal{R}, Rxz, \Gamma}{\mathcal{R}, Rxy, Ryz, \Gamma}\ (\mathsf{Trans}') \qquad \frac{\dfrac{\mathcal{R}, Rxy, Ryz, Rxz, \Gamma}{\mathcal{R}, Rxy, Ryz, Rxy, Ryz, \Gamma}\ (\mathsf{Trans}')}{\mathcal{R}, Rxy, Ryz, \Gamma}\ (\mathsf{ctr})$$

In summary: embedding the display calculus into the labelled calculus has yielded two seemingly distinct labelled calculi for the tense logics:

$\mathsf{G3Kt} + l(h,i,j,k)$ and $\mathsf{G3Kt}(UT) + ul(h,i,j,k)$. Investigating the (im)possibility of a pointwise translation from the derivations in the former to the latter is an interesting problem which we defer to future work.

Acknowledgments. Work supported by the FWF projects: START Y544-N23 and I 2982.

References

1. Belnap Jr., N.D.: Display logic. J. Philos. Logic **11**(4), 375–417 (1982)
2. Blackburn, P., de Rijke, M., Venema, Y.: Modal Logic. Cambridge Tracts in Theoretical Computer Science, vol. 53. Cambridge University Press, Cambridge (2001)
3. Brünnler. K.: Deep sequent systems for modal logic. In: Advances in Modal Logic, vol. 6, pp. 107–119. College Publications, London (2006)
4. Chagrov, A., Zakharyashchev, M.: Modal companions of intermediate propositional logics. Stud. Logica. **51**(1), 49–82 (1992)
5. Ciabattoni, A., Ramanayake, R.: Power and limits of structural display rules. ACM Trans. Comput. Logic **17**(3), 1–39 (2016)
6. Dyckhoff, R., Negri, S.: Proof analysis in intermediate logics. Arch. Math. Log. **51**(1–2), 71–92 (2012)
7. Dyckhoff, R., Negri, S.: Geometrization of first-order logic. Bull. Symbolic Logic **21**, 123–163 (2015)
8. Fitting, M.: Proof Methods for Modal and Intuitionistic Logics. Synthese Library, vol. 169. D. Reidel Publishing Co., Dordrecht (1983)
9. Fitting, M.: Prefixed tableaus and nested sequents. Ann. Pure Appl. Logic **163**(3), 291–313 (2012)
10. Goré, R., Postniece, L., Tiu, A.: On the correspondence between display postulates and deep inference in nested sequent calculi for tense logics. Log. Methods Comput. Sci. **7**(2), 1–38 (2011). (2:8)
11. Goré, R., Ramanayake, R.: Labelled tree sequents, tree hypersequents and nested (deep) sequents. In: Advances in Modal Logic, vol. 9. College Publications, London (2012)
12. Greco, G., Ma, M., Palmigiano, A., Tzimoulis, A., Zhao, Z.: Unified correspondence as a proof-theoretic tool. J. Logic Comput. (2016, to appear). https://doi.org/10.1093/logcom/exw022
13. Kashima, R.: Cut-free sequent calculi for some tense logics. Stud. Logica. **53**(1), 119–135 (1994)
14. Kracht, M.: Power and weakness of the modal display calculus. In: Proof Theory of Modal Logic (Hamburg, 1993) Applied Logic Series, vol. 2, pp. 93–121. Kluwer Academic Publishers, Dordrecht (1996)
15. Lemmon, E.J., Scott, D.S.: The 'Lemmon Notes': An Introduction to Modal Logic. Blackwell, Oxford (1977)
16. Mints, G.: Indexed systems of sequents and cut-elimination. J. Philos. Logic **26**(6), 671–696 (1997)
17. Negri, S.: Proof analysis in modal logic. J. Philos. Logic **34**(5–6), 507–544 (2005)
18. Ramanayake, R.: Inducing syntactic cut-elimination for indexed nested sequents. In: Proceedings of IJCAR, pp. 416–432 (2016)
19. Restall, G.: Comparing modal sequent systems. http://consequently.org/papers/comparingmodal.pdf

20. Restall, G., Poggiolesi, F.: Interpreting and applying proof theory for modal logic. In: Restall, G., Russell, G. (eds.) New Waves in Philosophical Logic, pp. 39–62 (2012)
21. Viganò, L.: Labelled Non-Classical Logics. Kluwer Academic Publishers, Dordrecht (2000). With a foreword by Dov M. Gabbay
22. Wansing, H.: Displaying Modal Logic. Trends in Logic-Studia Logica Library, vol. 3. Kluwer Academic Publishers, Dordrecht (1998)

Notions of Cauchyness and Metastability

Hannes Diener[1(✉)] and Robert Lubarsky[2]

[1] University of Canterbury, Christchurch, New Zealand
hannes.diener@canterbury.ac.nz
[2] Florida Atlantic University, Boca Raton, FL 33431, USA
Robert.Lubarsky@alum.mit.edu

Abstract. We show that several weakenings of the Cauchy condition are all equivalent under the assumption of countable choice, and investigate to what extent choice is necessary. We also show that the syntactically reminiscent notion of metastability allows similar variations, but is empty in terms of its constructive content.

Keywords: Cauchy condition · Metastability · Axiom of choice
Constructive analysis

1 Almost Cauchyness

Apart from the last section, we work in Bishop style constructive mathematics [4]—that is mathematics using intuitionistic instead of classical logic and some appropriate set-theoretic or type theoretic foundation [1]. Unlike Bishop, however, we do not freely use the axiom of countable/dependent choice, but explicitly state every such use.

In [3] a weakened form of the usual Cauchy condition is considered. There a sequence $(x_n)_{n\geqslant 1}$ in a metric space (X, d) is called *almost Cauchy*, if for any strictly increasing $f, g : \mathbb{N} \to \mathbb{N}$

$$d(x_{f(n)}, x_{g(n)}) \to 0$$

as $n \to \infty$. (This property will be named C2 below). Unsurprisingly, and as indicated by its name, every Cauchy sequence is almost Cauchy. In the same paper mentioned above it is also shown that Ishihara's principle BD-N suffices to show the converse: that every almost Cauchy sequence is Cauchy. Thus the two conditions are equivalent not only in classical mathematics (CLASS), but also in Brouwer's intuitionism (INT) and Russian recursive mathematics á la Markov (RUSS) as in all these models BD-N holds. In fact, it was only recently that it has been shown that there are models[1] in which this principle fails [7,10]. In this

[1] As BISH is not formalised in the same spirit as normal, everyday, mathematics is formalised, we use the phrase "model of" here somewhat loosely. Of course there are strict formalisations of BISH and the structures falsifying BD-N are models of such formalisations.

S. Artemov and A. Nerode (Eds.): LFCS 2018, LNCS 10703, pp. 140–153, 2018.
https://doi.org/10.1007/978-3-319-72056-2_9

paper we will link the notion of almost Cauchyness to various other weakenings proposed by Fred Richman and investigate similarities and differences to the notion of metastability which was proposed by Terence Tao.

Without further ado we will start the mathematical part of the paper with the following convention: For two natural numbers n, m the interval $[n, m]$ will denote all natural numbers between n and m; notice that this notation does not necessitate $n \leqslant m$.

Proposition 1. *Consider the following conditions for a sequence $(x_n)_{n\geqslant 1}$ in a metric space (X, d), where each condition should be read as prefaced by "for every $\epsilon > 0$ and for all strictly increasing $f, g : \mathbb{N} \to \mathbb{N}$ there exists $N \in \mathbb{N}$ such that for all $n \geqslant N$"*

C1 $d(x_n, x_{g(n)}) < \varepsilon$
C1′ $\forall i, j \in [n, g(n)] : d(x_i, x_j) < \varepsilon$
C2 $d(x_{f(n)}, x_{g(n)}) < \varepsilon$
C2′ $\forall i, j \in [f(n), g(n)] : d(x_i, x_j) < \varepsilon$
C3 $d(x_{g(n)}, x_{g(n+1)}) < \varepsilon$
C3′ $\forall i, j \in [g(n), g(n+1)] : d(x_i, x_j) < \varepsilon$

The following implications hold.

$$\begin{tikzcd}
C1' \arrow[r, leftrightarrow] \arrow[d] & C2' \arrow[r, leftrightarrow] \arrow[d] & C3' \\
C1 \arrow[r, leftrightarrow] \arrow[d] & C2 & \\
C3 & &
\end{tikzcd}$$

Furthermore all *conditions are equivalent using countable choice.*

Proof. C1 implying C2 is a simple consequence of the triangle inequality—nevertheless, this small point is of importance in the later discussion. It is also trivial to see that Ci′ implies Ci for $i = 1, 2, 3$. With $f = \mathrm{id}$ one can also see that C1 is a special case of C2 and the same holds for C1′ and C2′. Similarly one can see that C2 implies C3 and C2′ implies C3′. Since $[f(n), g(n)] \subset [n, \max\{g(n), f(n)\}]$ for strictly increasing f and g C1′ implies C2′

To see that C3′ implies C1′ consider the intervals

$$G_n = [g^n(0), g^{n+2}(0)] \ .$$

We claim that for every n there is a k such that

$$[n, g(n)] \subset G_k \ . \tag{1}$$

To see this let $n \in \mathbb{N}$ be arbitrary. Since g is strictly increasing we can easily show by induction that $g^n(0) \geqslant n$. Therefore there exists $k \leqslant n$ such that

$$g^k(0) \leqslant n \leqslant g^{k+1}(0) \ .$$

Applying g to the second of these two inequalities we also get $g(n) \leqslant g^{k+2}(0)$, and thus $[n, g(n)] \subset G_k$.

Now consider the functions f and h defined by $f(n) = g^{2n}(0)$ and $h(n) = g^{2n+1}(0)$. By C3′ eventually

$$\forall i, j \in [f(n), f(n+1)] : d(x_i, x_j) < \varepsilon$$

and

$$\forall i, j \in [h(n), h(n+1)] : d(x_i, x_j) < \varepsilon \ .$$

Since for even n, say $n = 2k$, we have $G_n = [f(k), f(k+1)]$ and for odd n, say $n = 2\ell + 1$ we have that $G_n = [h(\ell), h(\ell + 1)]$, we can conclude that eventually

$$\forall i, j \in G_n : d(x_i, x_j) < \varepsilon$$

and thus have shown C1′.

For the rest of the proof we will assume countable choice and prove that C3 implies C3′, which in turn by transitivity will show that all conditions are equivalent. To this end let g be an arbitrary increasing function and $\varepsilon > 0$. For each n choose natural numbers i_n and j_n and a binary flagging sequence λ_n such that $g(n) \leqslant i_n < j_n \leqslant g(n+1)$ and

$$\begin{aligned} \lambda_n = 0 &\implies \forall i, j \in [g(n), g(n+1)] : d(x_i, x_j) < \varepsilon \ , \\ \lambda_n = 1 &\implies d(x_{i_n}, x_{j_n}) > \frac{\varepsilon}{2} \ . \end{aligned}$$

Notice that it might happen that $i_{n+1} = j_n$, but at least always $j_n < i_{n+2}$. Therefore, to get strictly increasing functions, we need to work with two functions f and g defined by $f(2n) = i_{2n}$, $f(2n+1) = j_{2n}$, $g(2n) = i_{2n+1}$, and $g(2n+1) = j_{2n+1}$. By C3 there is N and M such that for all $n \geqslant N$ $d(x_{f(n)}, x_{f(n+1)}) < \frac{\varepsilon}{2}$ and for all $n \geqslant M$ $d(x_{g(n)}, x_{g(n+1)}) < \frac{\varepsilon}{2}$. We claim that there cannot be $n \geqslant \max\{N, M\}$ such that $\lambda_n = 1$. For if there were such an even n we would have the contradiction

$$\frac{\varepsilon}{2} < d(x_{i_n}, x_{j_n}) = d(x_{f(n)}, x_{f(n+1)}) \ .$$

We can treat the odd case in a similar fashion, and therefore $\lambda_n = 0$ for all $n \geqslant \max\{N, M\}$, which is saying that C3′ holds.

We say that a sequence is *almost Cauchy* if it satisfies C2′ (and therefore any of the above properties).[2] Naturally, we are going to consider the following statement

(aCC) Every almost Cauchy sequence in a metric space is Cauchy.

[2] Notice that in [3] “almost Cauchy” is defined as satisfying C2. Since in that work the authors assume countable choice this is not a conflicting definition. In the absence of choice it seems, to us, most natural to use the strongest notion.

As mentioned above, it is shown in [3] that BD-N implies that every almost Cauchy sequence is Cauchy, and therefore

$$\text{BD-N} \implies \text{aCC} .$$

In the same paper it is also shown that if one drops the triangle inequality and works with so called semi-metric spaces, then this is in fact an equivalence. However, the Berger, Bridges, and Palmgren proof crucially needs semi-metric spaces instead of metric spaces, since in [8] it is shown that the statement that every almost Cauchy sequence is Cauchy implies BD-N is not provable within BISH. This is done by giving a topological model T. Notice that T also proves countable choice, so this result does not change, even if we switch to any of the other conditions of Proposition 1.

Next we can show that, conveniently, it is enough to consider the monotone, real case.

Proposition 2. *(countable choice) If every decreasing/increasing sequence of reals that satisfies the almost Cauchy condition is Cauchy, then aCC holds.*

Proof. Let x_n be a sequence satisfying C2′. First note that for every m the sequence defined by

$$r_n^{(m)} = \max_{i,j \in [m,m+n]} \{d(x_i, x_j)\}$$

is increasing. We want to show that it also satisfies the almost Cauchy condition. To this end let f and g be strictly increasing and $\varepsilon > 0$. We know, by C2′, that there is N such that for all $n \geqslant N$ and $i, j \in [m + f(n), m + g(n)]$

$$d(x_i, x_j) < {}^{\varepsilon}\!/_2 \; . \tag{2}$$

Now consider $k, \ell \in [f(n), g(n)]$. W.l.o.g. $\ell < k$. Then

$$\left| r_k^{(m)} - r_\ell^{(m)} \right| = \max_{i,j \in [m,m+k]} \{d(x_i, x_j)\} - \max_{i,j \in [m,m+\ell]} \{d(x_i, x_j)\} \; .$$

We will show that this distance is less than ε. First, by the definition of the maximum[3] we can choose $p, q \in [m, m + k]$ such that

$$d(x_p, x_q) > \max_{i,j \in [m,m+k]} \{d(x_i, x_j)\} - {}^{\varepsilon}\!/_2 \; .$$

We may assume that $p \leqslant q$. We have to distinguish three cases depending whether p and q are both to the left of $m + \ell$ or on both sides or to the right of it:

[3] Notice that in general, even given just two numbers x, y we cannot decide whether $x = \max\{x, y\}$ or $y = \max\{x, y\}$, since that would imply the non-constructive *lesser limited principle of omniscience.* We can, however, given real numbers $x_1, \ldots, x_n$ and $\varepsilon > 0$ find i such that $x_i > \max\{x_1, \ldots, x_n\} - \varepsilon$.

- If $q \leqslant m+\ell$, then

  $$\max_{i,j\in[m,m+\ell]}\{d(x_i,x_j)\} \geqslant d(x_p,x_q)$$

  and therefore

  $$\begin{aligned}\max_{i,j\in[m,m+k]}\{d(x_i,x_j)\} - \max_{i,j\in[m,m+\ell]}\{d(x_i,x_j)\}&\\ < d(x_p,x_q) + {\varepsilon}/{2} - \max_{i,j\in[m,m+\ell]}\{d(x_i,x_j)\}&\\ \leqslant d(x_p,x_q) + {\varepsilon}/{2} - d(x_p,x_q) < \varepsilon&\end{aligned}$$

- If $p \leqslant m+\ell < q$, then

  $$\max_{i,j\in[m,m+\ell]}\{d(x_i,x_j)\} \geqslant d(x_p,x_{m+\ell})\ .$$

  Furthermore if $m+\ell \leqslant q$, then $q \in [m+\ell, m+k] \subset [m+f(n), m+g(n)]$ and therefore $d(x_{m+\ell},x_q) < {\varepsilon}/{2}$ by Eq. 2. Together

  $$\begin{aligned}\max_{i,j\in[m,m+k]}\{d(x_i,x_j)\} - \max_{i,j\in[m,m+\ell]}\{d(x_i,x_j)\}&\\ < d(x_p,x_q) + {\varepsilon}/{2} - \max_{i,j\in[m,m+\ell]}\{d(x_i,x_j)\}&\\ \leqslant d(x_p,x_{m+\ell}) + d(x_{m+\ell},x_q) + {\varepsilon}/{2} - \max_{i,j\in[m,m+\ell]}\{d(x_i,x_j)\}&\\ \leqslant d(x_p,x_{m+\ell}) + d(x_{m+\ell},x_q) + {\varepsilon}/{2} - d(x_p,x_{m+\ell})&\\ < {\varepsilon}/{2} + {\varepsilon}/{2} = \varepsilon&\end{aligned}$$

- If $m+\ell \leqslant p$, then $p,q \in [m+\ell, m+k] \subset [m+f(n), m+g(n)]$ and therefore $d(x_p,x_q) < {\varepsilon}/{2}$ by Eq. 2. Thus we have

  $$\max_{i,j\in[m,m+k]}\{d(x_i,x_j)\} < d(x_p,x_q) + {\varepsilon}/{2} < {\varepsilon}/{2} + {\varepsilon}/{2} = \varepsilon\ .$$

  And in particular

  $$\max_{i,j\in[m,m+k]}\{d(x_i,x_j)\} - \max_{i,j\in[m,m+\ell]}\{d(x_i,x_j)\} \leqslant \max_{i,j\in[m,m+k]}\{d(x_i,x_j)\} < \varepsilon$$

That is in all cases for $n \geqslant N$ and $k,\ell \in [f(n), g(n)]$

$$\left|r_k^{(m)} - r_\ell^{(m)}\right| < \varepsilon\ ,$$

which means that the sequence $\left(r_n^{(m)}\right)_{n\geqslant 1}$ satisfies the almost Cauchy condition. Thus, by our assumption, it is Cauchy and converges to a limit, say y_m.

Since by definition $r_{n+1}^{(m)} \geqslant r_n^{(m+1)}$ we also have that in the limit $y_m \geqslant y_{m+1}$ ([4, Proposition 2.3.4.f]), so $(y_m)_{m\geqslant 1}$ is decreasing. We want to show that it also satisfies the almost Cauchy condition C2′. So let f and g be strictly increasing and $\varepsilon > 0$. Since $(r_k^{(i)})_{k\geqslant 1}$ converges to y_i for every n we can use countable choice to fix a function $h : \mathbb{N} \to \mathbb{N}$ such that

$$\forall i \in [f(n), g(n)] : |y_i - r_k^{(i)}| < {}^{\varepsilon}\!/_4$$

for all $k \geqslant h(n)$. Since x_n satisfies C2′ there exists N such that for all $n \geqslant N$ we have

$$\forall i', j' \in [\min\{f(n), g(n)\}, \max\{f(n), g(n)\} + h(n)] : d(x_{i'}, x_{j'}) < {}^{\varepsilon}\!/_4 \ .$$

Then, in particular, for all $i, j \in [f(n), g(n)]$ we have that

$$\begin{aligned} &\left| \max_{\ell,\ell'\in[i,i+h(n)]} d(x_\ell, x_{\ell'}) - \max_{p,p'\in[j,j+h(n)]} d(x_p, x_{p'}) \right| \\ &\qquad \leqslant \left| \max_{\ell,\ell'\in[i,i+h(n)]} d(x_\ell, x_{\ell'}) \right| + \left| \max_{p,p'\in[j,j+h(n)]} d(x_p, x_{p'}) \right| \\ &\qquad\qquad \leqslant {}^{\varepsilon}\!/_4 + {}^{\varepsilon}\!/_4 = {}^{\varepsilon}\!/_2 \ , \end{aligned}$$

since

$$[i, i + h(n)] \subset [\min\{f(n), g(n)\}, \max\{f(n), g(n)\} + h(n)]$$

and

$$[j, j + h(n)] \subset [\min\{f(n), g(n)\}, \max\{f(n), g(n)\} + h(n)] \ .$$

Combining all of this we get that for all $n \geqslant N$ and $i, j \in [f(n), g(n)]$

$$\begin{aligned} |y_i - y_j| &\leqslant |y_i - r_{h(n)}^{(i)}| + |y_j - r_{h(n)}^{(j)}| + |r_{h(n)}^{(i)} - r_{h(n)}^{(j)}| \\ &\leqslant {}^{\varepsilon}\!/_4 + {}^{\varepsilon}\!/_4 + {}^{\varepsilon}\!/_2 \ . \end{aligned}$$

So $(y_m)_{m\geqslant 1}$ is a Cauchy sequence converging to a limit $z \geqslant 0$. We want to show that $z = 0$. So assume[4] $z > 0$. That means that $y_m \geqslant z > 0$ for all $m \in \mathbb{N}$, since y_m is decreasing. Therefore, using countable choice, we can fix $g : \mathbb{N} \to \mathbb{N}$ such that

$$r_{g(m)}^{(m)} > {}^{z}\!/_2 \ .$$

But if we apply property C2′ of $(x_n)_{n\geqslant 1}$ to $f = \mathrm{id}$, $\mathrm{id} + g$, and $\varepsilon = {}^{z}\!/_2$ we get that for $i, j \in [m, m + g(m)]$

$$d(x_i, x_j) < {}^{z}\!/_2$$

eventually, and therefore

$$r_{g(m)}^{(m)} = \max_{i,j\in[m,m+g(m)]} \{d(x_i, x_j)\} < {}^{z}\!/_2$$

eventually. This is a contradiction and thus $z = 0$.

So $z = 0$ and since $d(x_i, x_j) \leqslant y_m$ for all $i, j \geqslant m$, we have shown that $(x_n)_{n\geqslant 1}$ is Cauchy.

[4] We remind the reader that even constructively equality is stable.

2 Metastability

In a program suggested by Terence Tao [13], it is proposed to recover the "finite" (constructive) content of theorems by replacing them with logically (using classical logic) equivalent ones that can be proven by finite methods. Since often there is no way to establish the Cauchy condition it is suggested to be replaced with the following notion of metastability. A sequence $(x_n)_{n\geqslant 1}$ in a metric space (X, d) is called *metastable* iff

$$\forall \epsilon > 0, f : \exists m : \forall i, j \in [m, f(m)] : d(x_i, x_j) < \varepsilon \ .$$

Notice that this is almost the same definition as C1$'$, and, in fact, one can easily show that an almost Cauchy sequence is metastable. However—as we will see—metastability contains almost no constructive content.

As noted in [2] every non-decreasing sequence of reals bounded by $B \in \mathbb{R}$ is metastable since it is impossible that $d(x_m, x_{f(m)}) > \frac{\varepsilon}{2}$ for all $1 \leqslant m \leqslant \frac{2B}{\varepsilon}$. How about the converse: is every non-decreasing metastable sequence bounded? There is no hope in finding a constructive proof since we will see that it is equivalent to the non-constructive *limited principle of omniscience*

(LPO) For every binary sequence $(a_n)_{n\geqslant 1}$ we can decide whether

$$\forall n \in \mathbb{N} : a_n = 0 \vee \exists n \in \mathbb{N} : a_n = 1.$$

Under the assumption of countable choice LPO is equivalent to deciding for all real numbers whether $x < 0 \vee x = 0 \vee 0 < x$. Countable choice is needed to given a real number x construct a sequence of rationals converging to x. LPO is also equivalent to even stronger statements:

Proposition 3. *(countable choice) LPO is equivalent to either of the following*

1. *The Bolzano Weierstraß theorem: every sequence of reals in* $[0, 1]$ *has a convergent subsequence.*
2. *For every binary sequence* $(a_n)_{n\geqslant 1}$

$$\exists N : \forall n \geqslant N : a_n = 0 \vee \exists k_n \in \mathbb{N}^{\mathbb{N}} : a_{k_n} = 1.$$

Proof. The equivalence of LPO with the Bolzano Weierstraß theorem can be found in [11].

2 obviously implies LPO. Conversely we can show 2 by applying LPO countably many times: using LPO (and unique choice) construct a binary sequence b_n such that

$$b_k = 0 \implies \exists n \geqslant k : a_n = 1$$
$$b_k = 1 \implies \forall n \geqslant k : a_n = 0$$

Now, using LPO again, either $\exists N : b_N = 1$ or $\forall k : b_k = 0$. In the first case $\forall n \geqslant N : a_n = 0$. In the second case we can use unique choice[5] to find $k_n \in \mathbb{N}^{\mathbb{N}}$ such that $\forall n \in \mathbb{N} : a_{k_n} = 1$.

[5] To use unique choice we need to always pick the *smallest* $k_{n+1} > k_n$ such that $a_{k_{n+1}} = 1$.

Proposition 4. *(countable choice) LPO is equivalent to the statement that every metastable, non-decreasing sequence of rationals is bounded.*

Proof. Assume $(x_n)_{n\geqslant 1}$ is non-decreasing and metastable. For every k we can fix, using LPO countably many times, a binary sequence $(\lambda^{(k)})_{n\geqslant 1}$ such that

$$\lambda_n^{(k)} = 0 \implies x_n \leqslant k$$
$$\lambda_n^{(k)} = 1 \implies x_n > k \ .$$

Then for every k, using LPO on $(\lambda_n^{(k)})_{n\geqslant 1}$, we can decide whether k is an upper bound of $(x_n)_{n\geqslant 1}$ or not. So we can fix another binary sequence η_k such that

$$\eta_k = 1 \implies k \text{ is an upper bound}$$
$$\eta_k = 0 \implies \exists \ell : x_\ell > k \ .$$

Using LPO yet again, we can thus either find an upper bound or, using dependent choice, we can fix a function $f : \mathbb{N} \to \mathbb{N}$ such that $x_{f(n+1)} > x_{f(n)} + 1$ for all $n \in \mathbb{N}$. Since x_n is non-decreasing f is increasing. Furthermore

$$d(x_{f(n+1)}, x_{f(n)}) > 1 \ ;$$

a contradiction to the metastability. Hence $(x_n)_{n\geqslant 1}$ is bounded.

Conversely, let $(a_n)_{n\geqslant 1}$ be a binary sequence that has, w.l.o.g., at most one 1. Now consider

$$x_n = \sum_{i=1}^{n} i a_i \ . \tag{3}$$

It is easy to see that x_n is metastable: if $f : \mathbb{N} \to \mathbb{N}$ is increasing, then either $a_i = 0$ for all $i \in [1, f(1)]$ or $a_i = 0$ for all $i \in [f(2), f(f(2))]$. In both cases x_i is constant on an interval of the form $[m, f(m)]$.

Now if x_n is bounded, there is $N \in \mathbb{N}$ with $x_n < N$. If there was $i > N$ with $a_i = 1$, then $x_i = i > N$ which is a contradiction. Hence $a_i = 0$ for all $i > N$, that is we only need to check finitely many entries to see if $(a_n)_{n\geqslant 1}$ consists of 0s or whether there is a term equalling 1.

Since the construction of the sequence in the proof above (see Eq. 3) relies on the terms being potentially very large one might still hope that there is maybe a chance that every *bounded*, metastable sequence converges. However, also this statement is equivalent to LPO.

Proposition 5. *(countable choice) LPO is equivalent to the statement that every bounded, metastable sequence of rationals converges.*

Proof. Assume that LPO holds and that $(x_n)_{n\geqslant 1}$ is a bounded and metastable sequence of rationals. Since LPO implies the Bolzano Weierstraß theorem (see Proposition 4) there exists $x \in \mathbb{R}$ and $k_n \in \mathbb{N}^{\mathbb{N}}$ such that x_{k_n} converges to x. Now let $\varepsilon > 0$ be arbitrary. For every $n \in \mathbb{N}$ we can use LPO to decide whether

$$|x - x_n| < \varepsilon \vee |x - x_n| \geqslant \varepsilon \ .$$

So, using (unique) countable choice we can fix a binary sequence $(\lambda_n)_{n\geqslant 1}$ such that

$$\begin{aligned} \lambda_n = 0 &\implies |x - x_n| < \varepsilon \\ \lambda_n = 1 &\implies |x - x_n| \geqslant \varepsilon \ . \end{aligned}$$

By Proposition 4 either there exists N such that $\lambda_n = 0$ for all $n \geqslant N$ or there exists a strictly increasing $\ell_n \in \mathbb{N}^{\mathbb{N}}$ such that $\lambda_{\ell_n} = 1$ for all $n \in \mathbb{N}$. We will show that the second alternative is ruled out by the metastability: fix M such that $|x_{k_n} - x| < \frac{\varepsilon}{2}$ for $n \geqslant M$ and hence

$$|x_{k_n} - x_{\ell_n}| \geqslant \frac{\varepsilon}{2} \text{ for } n \geqslant M \ . \tag{4}$$

Now define $f : \mathbb{N} \to \mathbb{N}$ by $f(n) = \max\{k_{n+M}, \ell_{n+M}\}$. Then f is increasing. Since $(x_n)_{n\geqslant 1}$ is metastable there exists m such that for all $i, j \in [m, f(m)]$ we have $|x_i - x_j| < \frac{\varepsilon}{2}$. Since $k_{m+M}, \ell_{m+M} \in [m + M, f(m)]$ we get the desired contradiction to 4.

Conversely let $(a_n)_{n\geqslant 1}$ be a binary sequence with at most one term equalling 1. We will show that $(a_n)_{n\geqslant 1}$ is metastable. So let $f : \mathbb{N} \to \mathbb{N}$ an increasing function. Now either there exists $i \in [1, f(1)]$ such that $a_i = 1$ or for all $i \in [1, f(1)]$ we have $a_i = 0$. In the first case, since $(a_n)_{n\geqslant 1}$ has at most one 1, for all $i \in [f(1) + 1, f(f(1) + 1)]$ we have $a_i = 0$. In both cases there exists m such that, regardless of $\varepsilon > 0$, we have

$$\forall i, j \in [m, f(m)] : |a_i - a_j| = 0 < \varepsilon \ ;$$

that is $(a_n)_{n\geqslant 1}$ is metastable. Now if this sequence converges it must converge to 0. So there exists $N \in \mathbb{N}$ such that for all $n \geqslant N$ we have $a_N = 0$. So we only need to check finitely many indices $n \in \mathbb{N}$ for $a_n = 1$, and hence LPO holds.

It is only natural to ask how variants of metastability along the lines considered in the first section interact. To this end let us consider the properties $MS^{(\prime)}1 - 3$ of a sequence, which are the same as $C^{(\prime)}1 - 3$ in Proposition 1, only that they are read as being prefaced by "for every $\epsilon > 0$ and for all strictly increasing $f, g : \mathbb{N} \to \mathbb{N}$ there exists $n \in \mathbb{N}$." With this notation metastability, as defined above, is $MS1'$. Not surprisingly we can reuse large parts of the proof of Proposition 1 in the next one.

Proposition 6. *The following implications hold among conditions $MS^{(\prime)}1 - 3$ for a sequence in a metric space.*

$$\begin{tikzcd} MS1' \arrow[r, leftrightarrow] & MS2' \arrow[r, leftrightarrow] \arrow[d] & MS3' \\ & MS2 \arrow[dl] \arrow[dr] & \\ MS1 & & MS3 \end{tikzcd}$$

The proof is identical to the one of Proposition 1, apart from the proof that $MS1$ implies $MS2$ which does not translate and which leaves us therefore with fewer implications.

3 Choice Is Necessary

In recent years, there has been an increasing sensitivity to the use of countable choice in constructive mathematics. In Bishop's words, "meaningful distinctions deserve to be preserved" and some researchers have argued [12] that the distinctions which are removed by the use of countable choice are, indeed, meaningful.

So the elephant-in-the-room question raised by Sect. 1 is, whether the use of countable choice in Proposition 1 was really necessary. We will show in the next proposition that this is the case—at least to prove the equivalence between the weakest (C3) and the strongest (C3′) notion. This means that in the absence of choice the middle (C1) must be inequivalent to at least one of the other two levels, but it is not clear to which, and whether it is to both of them.

Theorem 1. *C3 does not imply C3′.*

Proof. We will first sketch the basic idea. The counter-example will involve a sequence x_n of reals. We will also make use of particular natural numbers $a < b$, with several counter-examples i, j coming from the interval $[a, b]$. These a and b will be non-standard, and also a non-standard distance apart. The sequence x_n will be 0 outside the interval $[a, b]$; within that interval, the sequence will increase by $2/(b - a)$ each step for the first half of that interval, up to a value of 1, and then decrease that same amount for each step in the second half, back down to 0. How does this help satisfy C3? For g's which take on no values in $[a, b]$, there is nothing to do, as then $d(x_{g(n)}, x_{g(n+1)})$ is always 0. For other g's, in the end we will see that we need concern ourselves only with standard ϵ. For values within $[a, b]$, whenever $g(n + 1) - g(n)$ is standard, $d(x_{g(n)}, x_{g(n+1)})$ is infinitesimal and hence less than ϵ. Of course, from x_n one can easily define a and b, and so their midpoint $(a + b)/2$, which would ruin C3. To avoid this, we will need to fuzz x_n up, so that the earlier cases mentioned are the only ones that happen.

In order to accomplish all of this we will need to exercise some care in the choice not only of a and b but also in the model in which they are embedded. It is easiest to work with an ultrapower of the universe V. (For the model theory about to be used, see standard references, such as [5, Sect. 4.3].) Where $\mathfrak{c}$ is (the size of) the continuum, take an ultrapower M using a $\mathfrak{c}$-regular ultrafilter. Then M is $\mathfrak{c}^+$ saturated over V ([5, Cororllary 4.3.14]). In the following, we will identify a set in V with its image in M. In particular, g refers both to a function (from $\mathbb{N}$ to $\mathbb{N}$) in V and to its image in M.

The point of the saturation is that the model realizes any type of size $\mathfrak{c}$. The type of interest to us is in a triple a, b, and k of (symbols standing for) natural numbers. Start by including the formulas $b > a, b-a > 0, b-a > 1, \ldots$, as well as $2^k \leq a$ and $b \leq 2^{k+1}$. This much is easily seen to be consistent, by compactness.

Toward realizing the first option listed above, consider an axiom which says "g takes on no values in (a, b)"; more formally,

$$\phi_g = \exists n : g(n) \leq a \wedge b \leq g(n + 1) \ ,$$

for some $g : \mathbb{N} \to \mathbb{N}$ in V.

Of course, ϕ_g might not be consistent (with the rest of the type); consider for example the identity function. For those g's, we will go toward the second option from above. For any $g \in V$, and standard natural number β (for "bound"), let $\psi_{g,\beta}$ be

$$\forall k : (\text{if } g(k) \text{ or } g(k+1) \text{ is in the interval } (a,b), \text{ then } g(k+1) - g(k) < \beta) \ .$$

Notice that there are only $\mathfrak{c}$-many formulas of the form ϕ_g and $\psi_{g,\beta}$. Let the type Ty be a maximal consistent extension of the starting formulas by ϕ_g's and $\psi_{g,\beta}$'s. By the $\mathfrak{c}^+$-saturation of M, Ty is realized in M. We would like to show that, for all increasing $h \in V$, either $\phi_h \in Ty$ or, for some β, $\psi_{h,\beta} \in Ty$.

If h takes on no values in (a,b), then ϕ_h is true, and hence consistent with Ty, and so by maximality is in Ty. Else consider the non-empty set

$$I_h = \{h(k+1) - h(k) \,|\, h(k) \text{ or } h(k+1) \text{ is in } (a,b)\} \, .$$

If every member of I_h were standard, then, since I_h is definable in M, it has a standard bound, say β. Immediately, $\psi_{h,\beta}$ is true, and so is consistent with Ty, and therefore is in Ty. The other possibility is that I_h contains a non-standard element. There are several cases here.

The simplest case is that, for some k with $h(k+1) - h(k)$ non-standard, $a \leq h(k)$ and $h(k+1) \leq b$. In that case, a and b could be re-interpreted to be $h(k)$ and $h(k+1)$ respectively. That would still satisfy Ty, and make ϕ_h true, and again we would be done by maximality. If that does not happen, then, whenever k generates a non-standard element of I_h, either $h(k) < a$ or $h(k+1) > b$ (so I_h contains at most 2 non-standard elements). We will show what to do when both of those possibilities occur (for different k's, of course). This will call for a two-step procedure. If only one of those possibilities occurs, then only one of those steps need be done.

Toward this end, we have $h(k) < a$, and also $a < h(k+1) < b$, else I_h would be empty. The first sub-case is that $h(k+1) - a$ is non-standard. Then, similarly to the above, we could re-interpret b as $h(k+1)$ (and leave a fixed), and get ϕ_h consistent with Ty. The other sub-case is that $h(k+1) - a$ is standard. Then we could interpret a as $h(k+1)$ (and leave b fixed). Re-interpreting Ty with this new choice of a, the new I_h has size 1. Now one considers the other choice of k, with $h(k) < b < h(k+1)$, and argues similarly.

So we conclude that Ty is complete in this sense. Returning to the model construction, we are going to work over a two-node Kripke model. To the bottom node associate V, and the top M. Take the full model F over that structure. (For the definition of a full model, see [6].) Consider the sequence x_n described at the beginning of this work: x_n is 0 outside of $[a,b]$, increases starting at a by $2/(b-a)$ at each step up to a value of 1 at the midpoint, then decreases by the same amount back to 0 at b. It would be no trouble to show C3 for this sequence for $g \in V$. The problem is, the full model contains a lot more functions g than just those from V. In particular, from x_n, a and b are easily definable, which would kill C3 holding. So we need to hide things better. This can be done by working within a topological model (built over the full model), and then taking a sub-model of it.

Working in F, let the space T consist of all sequences y_n such that $|x_n - y_n| < 1/n$ (starting the indexing from 1, obviously), except for n in the interval $[a, b]$, in which $|x_n - y_n| < 2/(b-a)$. A basic open set is given by restricting each component y_n to an open interval. Let G_n be the generic. In passing, we mention that, by standard arguments, any $g : \mathbb{N} \to \mathbb{N}$ in the topological model is in its ground model, which in this case is the full model F. We need more than that: we need a model in which any such g, at least at $\bot$, is in V.

To this end, we build essentially $L[G]$. At $\top$, this would be unambiguous. That is, at $\top$ we have a topological model over M, which models IZF_{Ref}, the version with Reflection. It was shown in [9] that such a theory can define its version of L and show it to be a model of IZF_{Ref}. It is not immediately clear, though, that this construction is consistent with what we need to do at $\bot$. So we describe the situation at $\bot$, and bring $\top$ along for the ride, and show what we need to for both.

The definition of $L_\alpha[G]$, inductively on α an ordinal of M, is straightforward, and is the same as in classical set theory. For $\alpha \in V$ an ordinal, its image in the full model, for which we temporarily use the notation α_f (f for "full"), works as follows: $\bot \Vdash$ "$x \in \alpha_f$" iff for some $\beta < \alpha$, $\bot \Vdash$ "$x = \beta_f$"; and $\top \Vdash$ "$x \in \alpha_f$" iff, in M, identifying α with its image under the elementary embedding into M, for some $\beta < \alpha$, $\bot \Vdash$ "$x = \beta_f$". Since α_f is in the full model, which is the ground model for the topological model, it is also in the topological model. So within the topological model, the set $L_{\alpha_f}[G]$ can be defined by induction. We do not know whether the topological model can separate ordinals of the form α_f from any others, or even whether the full model can do so, so the final step is done in V resp. M: at bottom, $L[G]$ is defined in V to be the union over the ordinals α of $L_{\alpha_f}[G]$, whereas at $\top$ that union is taken in M.

What remains to be shown? For one, IZF, which we postpone to the end. By the presence of G, it should be clear that C3$'$ fails, for $g(n) = 2^n$: even if the generic differs from x_n, it is only by an infinitesimal amount at each component. All that remains is that C3 holds for G. At $\top$, G is a Cauchy sequence, so that is taken care of. We need check C3 for G only at $\bot$.

At $\bot$, we need concern ourselves with only standard ϵ. For any such $\epsilon > 1/n$, n standard, let N be n. For $g \in V$, the whole set-up all along the way is to make C3 true for that g. So we will be done if we can show that any h is in V: if $\bot \Vdash h : \mathbb{N} \to \mathbb{N}$ then, for some $g \in V$, $\bot \Vdash h = g$.

By hypothesis, $\bot$ forces a standard value for h on each standard input. So that is the obvious choice for g: let g be such that $g(n)$ is the value forced by $\bot$ for $h(n)$. In our model, let X be $\{n \mid h(n) \neq g(n)\}$. This could have only non-standard elements. We will show that it is decidable: for any n, either $T \models n \in X$ or $T \models n \notin X$. Then we will show that any decidable set is either empty or has a standard member. That then suffices.

For decidability: any value for $h(n)$ has to be forced by T, by standard arguments, as follows. If not, let O be a maximal open set forcing a value for

$h(n)$. Pick a point on the boundary of O. What value could a neighborhood of it force? If there is no such neighborhood, then h is not total, so need not be considered. If it forces a different value than O does, then, by connectedness, consider the overlap: $h(n)$ is then no longer single-valued. So whatever value $h(n)$ has is forced by T. Now compare that to $g(n)$.

As for a non-empty decidable set X having standard members: If it has a member at all, consider the definition ϕ of X over $L_\alpha[G]$. With regard to the parameters in ϕ, one can unpack them by their definitions, ultimately reducing the parameters used to finitely many standard ordinals and G. We will in the course of this argument consider alternate interpretations of a and b. Of course, when doing so, there is no longer any reason to believe that C3 still holds, or that C3$'$ does not. This is of no matter for showing our current goal. The construction of T, and of $L[G]$, still makes sense, for any choice of $a < b$. Now consider the space T_1 based on the pair $a-1, b-1$. Notice that $T \cap T_1$ is a non-empty open subset of both T and T_1. So it forces the same facts about X that T does, and that T_1 does. So the interpretation of X stays the same when we shift a and b down by 1. Iterate this procedure until the lower number is some standard value, say $\overline{a}$, larger than all of the natural number parameters used in ϕ. Then hold $\overline{a}$ fixed, and reduce the upper number by one. By similar arguments, again X remains unchanged. Iterate until this upper number is standard, say $\overline{b}$. Call the space based on $\overline{a}$ and $\overline{b}$ U. In M, X is still interpreted the same way, so, in M, $U \models$ "X has a member." By elementarity, the same holds in V. Any such member there has to be standard: $V \models k \in X$. So X has a standard member.

Finally, we sketch briefly why IZF holds. For the axioms of Empty Set and Infinity, $\emptyset$ is definable over $L_0[G]$, and ω over $L_\omega[G]$. Pair and Union hold easily. Extensionality is valid because that is how equality is defined. $\in$-Induction holds, even though M is ill-founded, because $\in$-Induction holds in M. Reflection holds because it holds in V: If V_α is a Σ_n-elementary substructure of V, then the initial segment of $L[G]$ up to α is itself a Σ_n-elementary substructure of the whole thing. From this, Separation follows easily. For Power Set, given $x \in L[G]$, since the whole construction took place in V, the ordinals at which new subsets of x appear are bounded, and at the next level they can all be collected into one set.

References

1. Aczel, P., Rathjen, M.: Notes on constructive set theory. Technical report 40, Institut Mittag-Leffler. The Royal Swedish Academy of Sciences (2001)
2. Avigad, J., Dean, E.T., Rute, J.: A metastable dominated convergence theorem. J. Logic Anal. **4**(3), 1–19 (2012)
3. Berger, J., Bridges, D., Palmgren, E.: Double sequences, almost cauchyness and BD-N. Logic J. IGPL **20**(1), 349–354 (2012)
4. Bishop, E., Bridges, D.: Constructive Analysis. Springer, Heidelberg. https://doi.org/10.1007/978-3-642-61667-9 (1985)
5. Chang, C.C., Keisler, H.J.: Model Theory. Dover Books on Mathematics, 3rd edn. Dover Publications, New York (2013)

6. Hendtlass, M., Lubarsky, R.: Separating fragments of WLEM, LPO, and MP. J. Symbol. Logic **81**(4), 1315–1343 (2016)
7. Lietz, P.: From Constructive Mathematics to Computable Analysis via the Realizability Interpretation. PhD thesis, TU Darmstadt (2004)
8. Lubarsky, R.S., Diener, H.: Principles weaker than BD-N. J. Symbol. Logic **78**(3), 873–885 (2014)
9. Lubarsky, R.S.: Intuitionistic *L*. In: Crossley et al. (ed.) Logical Methods. In Honor of Anil Nerode's Sixtieth Birthday of Progress in Computer Science and Applied Logic, vol. 12, pp. 555–571. Birkhäuser Boston, Boston (1993)
10. Lubarsky, R.S.: On the failure of BD-N and BD, and an application to the anti-Specker property. J. Symbol. Logic **78**(1), 39–56 (2013)
11. Mandelkern, M.: Limited omniscience and the Bolzano-Weierstrass principle. Bull. London Math. Soc. **20**, 319–320 (1988)
12. Richman, F.: Constructive mathematics without choice. In: Schuster, P., Berger, U., Osswald, H. (eds.) Reuniting the Antipodes: Constructive and Nonstandard Views of the Continuum. Synthese Library. Kluwer Academic Publishers, Dordrecht (2001)
13. Tao, T.: Soft analysis, hard analysis, and the finite convergence principle, May 2007. http://terrytao.wordpress.com/2007/05/23/soft-analysis-hard-analysis-and-the-finite-convergence-principle/

A Gödel-Artemov-Style Analysis of Constructible Falsity

Thomas Macaulay Ferguson[1,2](✉)

[1] Cycorp, Austin, TX, USA
tferguson@gradcenter.cuny.edu
[2] Saul Kripke Center, CUNY Graduate Center, New York, NY, USA

Abstract. David Nelson's *logic of constructible falsity* N is a well-known conservative extension to intuitionistic logic Int. Heinrich Wansing has suggested that extending the provability interpretation of Int to such extensions requires that one enriches the single category of *formal proofs* assumed intuitionistically with further categories representing *formal refutations*. This paper adapts the framework of Sergei Artemov's *justification logic*—which has provided incredible insight into Int—to capture a proof/refutation interpretation of N. To represent distinct types of justification, we identify the distinct *agents* of Tatiana Yavorskaya-Sidon's two-agent logic of proofs $\mathsf{LP}^2_{\uparrow\uparrow}$ with categories of proof and refutation, permitting an embedding of N into $\mathsf{LP}^2_{\uparrow\uparrow}$. In conclusion, we describe how a Gödel-Artemov-style analysis can be given for Cecylia Rauszer's Heyting-Brouwer logic and show that Melvin Fitting's semantic realization proof can be extended to normal multimodal logics in general.

Keywords: Justification logic · Constructible falsity · Refutation Intuitionistic logic · Logic of proofs

1 Introduction

From formal, philosophical, and practical perspectives, intuitionistic logic (Int) has proven to be an extraordinarily natural and fruitful framework. Since its development, a number of systems related to intuitionistic logic have appeared, such as David Nelson's logics of constructible falsity of [14,16] and Cecylia Rauszer's Heyting-Brouwer logic HB of [19,20]. Each of these systems enrich the language of Int with novel connectives. The success of intuitionistic logic has often led to an expectation that intuitionistic interpretations will naturally extend to these conservative systems. In [25], Heinrich Wansing has suggested that many of these systems are best interpreted as logics in which both proof and refutation receive equal treatment.

In this paper, we lay the groundwork for an investigation into the interpretation of constructible falsity and Heyting-Brouwer logic (and their compatriots) by adapting the framework of Sergei Artemov's *logic of proofs* to Nelson's system. It is hoped that the naturalness of the present Gödel-Artemov-style interpretation of constructible falsity will serve as a platform from which to launch

S. Artemov and A. Nerode (Eds.): LFCS 2018, LNCS 10703, pp. 154–169, 2018.
https://doi.org/10.1007/978-3-319-72056-2_10

a deeper investigation into Heyting-Brouwer logic and other systems that complement intuitionistic notions of provability with refutation.

1.1 The Logic of Constructible Falsity

The primary target for the present analysis is Nelson's *logic of constructible falsity* N (frequently encountered as "N_3" or "CF"). N, introduced in [14], was motivated by Nelson's diagnosis of deficiencies in intuitionistic disproof. N enriches the familiar language of intuitionistic logic by adding to the intuitionistic negation an additional *strong* negation connective representing a *constructible* falsity. In [14], Nelson argues that the intuitionistic framework implicitly imposes asymmetries to the notions of truth and falsity that reveal that *disproof* in Int is non-constructive. For example, while a *proof* of a disjunction $\varphi \vee \psi$ requires either an explicit proof of φ or an explicit proof of ψ, a *disproof* of a conjunction $\varphi' \wedge \psi'$ *does not* require an explicit disproof of one of the conjuncts. Intuitionistically, the judgment that a simple contradiction $\varphi \wedge \frown\varphi$ is false (where $\frown$ represents intuitionistic negation)[1] requires no judgment concerning which conjunct in particular *contributes* to the contradiction's falsehood.

The aim of the present paper is to contribute to the understanding of N and, ultimately, related systems by shining a light on some of its subtle and implicit machinery. The route through which this contribution will be made is an extension of Sergei Artemov's program of *justification logic*, in which the techniques and operations of traditional modal logic are made explicit by introducing terms corresponding to an ontology of idealized justifications (*cf.* [4]).

One of the most notable successes of Artemov's program is its illumination of the inner machinery of intuitionistic logic, provided by recasting the familiar Gödel-McKinsey-Tarski translation of intuitionistic logic as an embedding of Int into Artemov's *logic of proofs* LP—the explicit counterpart to $\mathsf{S4}$. The close relationship between intuitionistic logic and its conservative extensions suggests that exegetical work on Int may lead to a similar clarification of interpretive matters in N.

The type of constructivity implicit in N requires that one must countenance not merely a category of *proofs* but must also embrace a distinct category of *disproofs*. This suggests that to recapture Artemov's success in the context of the system N requires an adaptation of justification logic that is versatile enough to distinguish proof from refutation. For example, successfully applying this framework would permit us to formally represent distinct and independent theses representing various ways in which proofs and refutations might be thought to interact with one another.[2]

[1] This paper follows the notational convention for negations of [24], in which four types of negation connective are studied. The notation for intuitionistic negation $\frown$ reflects its duality with the *conegation* $\smile$ of Heyting-Brouwer logic.

[2] Both Nelson's realizability semantics of [14] and Lopez-Escobar's BHK-style interpretation of [10] permit that a verifier of one formula may act as a falsifier of another formula. It is not necessary that the categories distinguish proofs and refutations *ontologically* so much as that they track the difference between the purpose or sense in which a justification is presented.

1.2 Artemov's Program of Justification Logic

Kurt Gödel took the first steps towards making the engine driving intuitionistic operations explicit in the abstract [9], in which it is observed that the theorems of intuitionistic propositional logic can be embedded into the modal logic $\mathsf{S4}$. Under Gödel's interpretation of the $\mathsf{S4}$ modality, "$\Box\varphi$" is read as "φ is provable." Alternative—and arguably more natural—versions of Gödel's 1933 translation are frequently adopted, as found in, *e.g.*, in [12], including the translation favored by Artemov himself.

With the introduction of Artemov's logic of proofs LP in [2], the logic of proofs and intuitionistic logic are succinctly represented in the following "foundational picture" drawn from [3], which acts as a portrait of what was revealed by the Gödel-McKinsey-Tarski-style translations found in [9,12]:

$$\mathsf{Int} \hookrightarrow \mathsf{S4} \hookrightarrow \mathsf{LP} \hookrightarrow \text{Classical Proofs}$$

Nelson's suggestion that N should admit provability-style interpretations analogous to Gödel's interpretation increases the plausibility of the merits of an analogous "foundational picture" for constructible falsity. This plausibility is reinforced by the fact that N can be faithfully embedded into modal logics that are very similar to $\mathsf{S4}$, a fact studied in detail in [17].

On its face, the philosophical discussions of constructible falsity in [14,15] are formulated in terms that allow us to readily export the fundamental concepts of intuitionistic logic and apply them to N. Nelson emphatically endorses the intuitionistic understanding of *proof* but insists that intuitionistic conceptions of proof must be supplemented by a constructive notion of falsity. The task of articulating a fundamental picture of N, then, might be best considered as finding a natural way of *extending* the corresponding picture for Int. By Richmond Thomason's result in [23], there exists a translation from N to the modal logic $\mathsf{S4}$ that successfully translates N within the $\mathsf{S4}$ consequence relation. In many ways, Thomason's translation is a salient and faithful representation of Nelson's position on constructible falsity. The translation, for example, captures Nelson's notions of proof and disproof by the distinct $\mathsf{S4}$ modalities of $\Box$ and $\Box\neg$, respectively, and provides a framework that is more expressive than the standard Gödel-McKinsey-Tarski translations. Seeking to reconcile Nelson's intuitions and Thomason's translation of N suggests the following foundational picture:

$$\mathsf{N} \hookrightarrow \mathsf{S4} \hookrightarrow ? \hookrightarrow \text{Classical Proofs and Disproofs}$$

Despite the apparent acceptability of the above picture, Thomason's translation faces some limitations. By Thomason's result, N can be mapped into Artemov's LP as well, which at first blush suggests that LP itself might fill in the lacuna. However, Nelson's work presupposes a non-trivial distinction between the categories of proof and refutation. Contrary to the spirit of Nelson's investigations, the formalism of LP proper requires a collapse of the categories of proof and disproof.

In light of this, in the following sections, I suggest that this distinction *is* respected by the additional expressivity of the bimodal logic $\mathsf{S4}^2_{\mathsf{Triv}}$—a bimodal logic with two interchangeable $\mathsf{S4}$ modalities $\Box_0$ and $\Box_1$. Furthermore, I will suggest that Tatiana Yavorskaya-Sidon's two-agent logic of proofs $\mathsf{LP}^2_{\uparrow\uparrow}$—introduced in [29]—provides a more satisfactory analysis of N. This leads to the following foundational picture of Nelson's logic of constructible falsity:

$$\mathsf{N} \hookrightarrow \mathsf{S4}^2 \hookrightarrow \mathsf{LP}^2_{\uparrow\uparrow} \hookrightarrow \text{Classical Proofs and Disproofs}$$

Now, we will begin to outline the formalisms required to provide the foundational picture of N.

2 Constructible Falsity and Two Modal Companions

Now, we introduce Nelson's logic in conjunction with two modal logics to which it may be faithfully translated. As N expands the language of intuitionistic logic, we will first discuss a number of formal languages.

2.1 The Basic System of Constructible Falsity N

We begin to define the richer languages by defining a set of atomic formulae:

Definition 1. **At** *is a denumerable set of atomic formulae* $\{p_0, p_1, ..., q_0, ...\}$.

From **At**, we construct the languages of classical logic (CL) and N.

Note that each of these systems includes a distinct, characteristic negation (or negation-like) operator. To prevent ambiguity, we will employ the notational convention for negations of [24]. That is, we employ "$\neg$" to denote classical negation while Nelson's *strong negation* will be denoted by "$\sim$." In the case of other connectives—disjunction, conjunction, implication, *etc.*—we allow context to determine which reading is appropriate. We then define the following formal languages:

$$\begin{aligned} \mathscr{L}_{\mathsf{CL}}&: \varphi ::= p|\neg\varphi|\varphi \wedge \varphi|\varphi \vee \varphi|\varphi \to \varphi \\ \mathscr{L}_{\mathsf{N}}&: \varphi ::= p|{\sim}\varphi|\varphi \wedge \varphi|\varphi \vee \varphi|\varphi \to \varphi \end{aligned}$$

It is worth mentioning that intuitionistic negation is definable in N, as any contradiction $\varphi \wedge {\sim}\varphi$ can serve as a definable *falsum* constant from which intuitionistic negation may be defined so that $\frown\varphi =_{df} \varphi \to (\varphi \wedge {\sim}\varphi)$.

We stick to Kripke semantics for the systems that follow.

Definition 2. *An n-ary* Kripke frame *is an $n+1$-tuple $\langle W, \leq_0, ..., \leq_{n-1}\rangle$ where each $\leq_i$ is a binary relation on the nonempty set W.*

In the sequel, we will sometimes use "$\geq$" to indicate the inverse of an accessibility relation $\leq$.

Definition 3. *An N-model $\mathfrak{M}$ is a 3-tuple $\langle W, \leq, v^+, v^-\rangle$ where $\langle W, \leq\rangle$ is a reflexive and transitive Kripke frame and v^+ and v^- are maps from* **At** *to $\wp(W)$ such that for all $p \in$* **At***:*

- *if* $w \in v^{+}(p)$ *and* $w \leq w'$*, then* $w' \in v^{+}(p)$
- *if* $w \in v^{-}(p)$ *and* $w \leq w'$*, then* $w' \in v^{-}(p)$
- $v^{+}(p) \cap v^{-}(p) = \varnothing$

Nelson's demand for independent notions of constructible truth and falsity leads to the formulation of *two* forcing relations in tandem:

- $\mathfrak{M}, w \Vdash^{+} p$ *iff* $w \in v^{+}(p)$ for $p \in \mathbf{At}$
- $\mathfrak{M}, w \Vdash^{+} {\sim}\varphi$ *iff* $\mathfrak{M}, w \Vdash^{-} \varphi$
- $\mathfrak{M}, w \Vdash^{+} \varphi \wedge \psi$ *iff* $\mathfrak{M}, w \Vdash^{+} \varphi$ and $\mathfrak{M}, w \Vdash^{+} \psi$
- $\mathfrak{M}, w \Vdash^{+} \varphi \vee \psi$ *iff* $\mathfrak{M}, w \Vdash^{+} \varphi$ or $\mathfrak{M}, w \Vdash^{+} \psi$
- $\mathfrak{M}, w \Vdash^{+} \varphi \rightarrow \psi$ *iff* for all w' s.t. $w \leq w'$, if $\mathfrak{M}, w' \Vdash^{+} \varphi$ then $\mathfrak{M}, w' \Vdash^{+} \psi$

- $\mathfrak{M}, w \Vdash^{-} p$ *iff* $w \in v^{-}(p)$ for $p \in \mathbf{At}$
- $\mathfrak{M}, w \Vdash^{-} {\sim}\varphi$ *iff* $\mathfrak{M}, w \Vdash^{+} \varphi$
- $\mathfrak{M}, w \Vdash^{-} \varphi \wedge \psi$ *iff* $\mathfrak{M}, w \Vdash^{-} \varphi$ or $\mathfrak{M}, w \Vdash^{-} \psi$
- $\mathfrak{M}, w \Vdash^{-} \varphi \vee \psi$ *iff* $\mathfrak{M}, w \Vdash^{-} \varphi$ and $\mathfrak{M}, w \Vdash^{-} \psi$
- $\mathfrak{M}, w \Vdash^{-} \varphi \rightarrow \psi$ *iff* $\mathfrak{M}, w \Vdash^{+} \varphi$ and $\mathfrak{M}, w \Vdash^{-} \psi$

Validity of an inference $\Gamma \vDash_{\mathsf{N}} \varphi$ is defined as the preservation of truth at each point in each model.[3]

Definition 4. *$\Gamma \vDash_{\mathsf{N}} \varphi$ holds if for every point w in every N-model $\mathfrak{M}$, whenever $\mathfrak{M}, w \Vdash^{+} \psi$ for every $\psi \in \Gamma$, also $\mathfrak{M}, w \Vdash^{+} \varphi$.*

As a Hilbert-style calculus, N may be considered as an axiomatic extension of positive intuitionistic propositional logic by adding the following schemes governing strong negation, where "$\leftrightarrow$" is defined in the standard fashion:

$\mathbf{N}_1$	${\sim}(\varphi \wedge \psi) \leftrightarrow ({\sim}\varphi \vee {\sim}\psi)$
$\mathbf{N}_2$	${\sim}(\varphi \vee \psi) \leftrightarrow ({\sim}\varphi \wedge {\sim}\psi)$
$\mathbf{N}_3$	${\sim}(\varphi \rightarrow \psi) \leftrightarrow (\varphi \wedge {\sim}\psi)$
$\mathbf{N}_4$	${\sim}{\sim}\varphi \leftrightarrow \varphi$
$\mathbf{N}_5$	${\sim}\varphi \rightarrow (\varphi \rightarrow \psi)$

This gives us the following definition of provability in N:

Definition 5. *$\Gamma \vdash_{\mathsf{N}} \varphi$ if there is a proof of φ from hypotheses Γ through the rules and axioms of positive intuitionistic logic supplemented by the axiom schema $\mathbf{N}_1$–$\mathbf{N}_5$.*

[3] As a referee has pointed out, Nelson's N is on its face very similar to classical logic and might be expected to be complete with respect to Boolean algebras. That this is not the case can be seen by observing that N is not closed under uniform substitution; while ${\sim}(\varphi \rightarrow \psi)$ is logically equivalent to $\varphi \wedge {\sim}\psi$, it is not the case that ${\sim}{\sim}(\varphi \rightarrow \psi)$ (*i.e.*, $\varphi \rightarrow \psi$) is equivalent to ${\sim}(\varphi \wedge {\sim}\psi)$ (*i.e.*, ${\sim}\varphi \vee \psi$).

N, like Int, is *constructive* in the sense that it enjoys the *disjunction property* according to which if a formula $\varphi \vee \psi$ is a theorem, then either φ or ψ is a theorem as well. N enhances the constructivity of Int by adding to this the *constructible falsity property* by which if $\sim(\varphi \wedge \psi)$ is a theorem, either $\sim\varphi$ or $\sim\psi$ is a theorem, that is, if a conjunction is false then one can pinpoint which conjunct fails.[4]

2.2 Bi-modal Logics $\mathsf{S4}^2$ and $\mathsf{S4}^2_{\mathsf{Triv}}$

An interest in considering multi-modal systems in the interpretation of N has been acknowledged. Hence, we will at this point consider *bimodal* logics, that is, expansions of classical logic CL that include two independent modal operators $\Box_0$ and $\Box_1$. For purposes that will become clear, we employ without loss of generality *notational variants* of the following systems by using a modal operator $\hexagon_1$ as a shorthand for $\Box_1\neg$. From these operators, we describe the language $\mathscr{L}^2_\Box$:

Definition 6. *The language* $\mathscr{L}^2_\Box$ *is defined with* $p \in \mathbf{At}$ *as follows:*

$$\varphi ::= p|\neg\varphi|\varphi \wedge \varphi|\varphi \vee \varphi|\varphi \to \varphi|\Box_0\varphi|\hexagon_1\varphi$$

The basic bimodal system for which all modal logics considered herein will be based is K^2, the bimodal logic for which the two modalities each behave as in the monomodal logic K:

K$_0$ $\Box_0(\varphi \to \psi) \to (\Box_0\varphi \to \Box_0\psi)$
K$_1$ $\hexagon_1(\varphi \wedge \neg\psi) \to (\hexagon_1\neg\varphi \to \hexagon_1\neg\psi)$

And rule of inference:

Necessitation From theoremhood of φ, infer $\Box_0\varphi$ and $\hexagon_1\neg\varphi$

In order to succinctly define extensions of K^2, we consider the following operation:

Definition 7. *Let* L *be an axiomatic system and let* $\mathbf{A}$ *be an axiom scheme or rule of inference. Then* $\mathsf{L} \oplus \mathbf{A}$ *is the axiomatic system determined by adding* $\mathbf{A}$ *to the axiom schemes of* L *and closing the union under the rules of* L.

Definition 8. *A* K^2*-model* $\mathfrak{M}$ *is a* 4*-tuple* $\langle W, \leq_0, \leq_1, v\rangle$ *such that* $\langle W, \leq_0, \leq_1\rangle$ *is a binary Kripke frame and* $v : \mathbf{At} \to \wp(W)$.

On each model $\mathfrak{M}$, a forcing relation is extended recursively as follows:

Definition 9. *The forcing relation is defined so that:*

- $\mathfrak{M}, w \Vdash p$ *if* $w \in v(p)$ *for* $p \in \mathbf{At}$
- $\mathfrak{M}, w \Vdash \neg\varphi$ *if* $\mathfrak{M}, w \nVdash \varphi$

[4] In the quantified case, the intuitionistic *existence property* is dualized by a property in N so that when a negated universal formula $\sim\forall x\varphi(x)$ is a theorem, there is a term t that witnesses this fact, *i.e.*, for which $\sim\varphi(t)$ is a theorem.

- $\mathfrak{M}, w \Vdash \varphi \wedge \psi$ *if* $\mathfrak{M}, w \Vdash \varphi$ *and* $\mathfrak{M}, w \Vdash \psi$
- $\mathfrak{M}, w \Vdash \varphi \vee \psi$ *if* $\mathfrak{M}, w \Vdash \varphi$ *or* $\mathfrak{M}, w \Vdash \psi$
- $\mathfrak{M}, w \Vdash \varphi \rightarrow \psi$ *if* $\mathfrak{M}, w \nVdash \varphi$ *or* $\mathfrak{M}, w \Vdash \psi$
- $\mathfrak{M}, w \Vdash \Box_0 \varphi$ *if for all* w' *such that* $w \leq_0 w'$, $\mathfrak{M}, w' \Vdash \varphi$
- $\mathfrak{M}, w \Vdash \hexagon_1 \varphi$ *if for all* w' *such that* $w \leq_1 w'$, $\mathfrak{M}, w' \nVdash \varphi$

Validity is defined as usual, that is, in a similar fashion to that of validity in N.

Just as the axiom scheme **K** has an instance for each modality, the axiom schemes **T** and **4** give rise to bimodal analogues:

$\mathbf{T}_0$	$\Box_0 \varphi \rightarrow \varphi$
$\mathbf{T}_1$	$\hexagon_1 \varphi \rightarrow \neg \varphi$
$\mathbf{4}_0$	$\Box_0 \varphi \rightarrow \Box_0 \Box_0 \varphi$
$\mathbf{4}_1$	$\hexagon_1 \varphi \rightarrow \hexagon_1 \neg \hexagon_1 \varphi$

We now introduce an intermediate system: the bimodal logic $\mathsf{S4}^2$.

Definition 10. $\mathsf{S4}^2 = \mathsf{K}^2 \oplus \mathbf{T}_0 \oplus \mathbf{T}_1 \oplus \mathbf{4}_0 \oplus \mathbf{4}_1$

Definition 11. *An* $\mathsf{S4}^2$*-model* $\mathfrak{M}$ *is a* K^2 *model for which* $\leq_0$ *and* $\leq_1$ *are reflexive and transitive.*

It is well-known that the Hilbert-style presentation of $\mathsf{S4}^2$ is sound and complete with respect to the above class of models. Our primary interest $\mathsf{S4}^2_{\mathsf{Triv}}$, a trivially bimodal extension of $\mathsf{S4}$ introduced by Yavorskaya-Sidon in [30]. The system is "trivial" in the sense that the modalities $\Box_0$ and $\Box_1$ (or $\Box_0\neg$ and $\hexagon_1$ in our notational variant) are intersubstitutable in any context. Axiomatically, $\mathsf{S4}^2_{\mathsf{Triv}}$ is defined as follows:

Definition 12. $\mathsf{S4}^2_{\mathsf{Triv}} = \mathsf{S4}^2 \oplus \Box_0 \neg \varphi \rightarrow \hexagon_1 \varphi \oplus \hexagon_1 \varphi \rightarrow \Box_0 \neg \varphi$

Model theoretically, $\mathsf{S4}^2_{\mathsf{Triv}}$ corresponds to the condition that $\leq_0 = \leq_1$. Now, let us briefly observe the "triviality" in more detail.

Definition 13. *The map* $\_^\dagger$ *maps formulae of the bimodal language of* $\mathsf{S4}^2_{\mathsf{Triv}}$ *to formulae of the monomodal language of* $\mathsf{S4}$ *replacing each instance of* $\Box_0$ *and* $\Box_1$ *with an instance of* $\Box$.

For φ in the language of $\mathsf{S4}^2_{\mathsf{Triv}}$, we may observe the following:

Observation 1. $\vdash_{\mathsf{S4}} \varphi^\dagger$ *iff* $\vdash_{\mathsf{S4}^2_{\mathsf{Triv}}} \varphi$

Proof. Left-to-right is trivial. Because there are two copies of $\mathsf{S4}$ contained in $\mathsf{S4}^2_{\mathsf{Triv}}$, one can replace each instance of $\Box$ with $\Box_0$ and yield an $\mathsf{S4}^2_{\mathsf{Triv}}$ theorem in the process. Right-to-left can be easily established by induction on length of proofs.

The feature described in Observation 1 provides a bridge between N and extensions of LP with sufficient machinery to formally distinguish proofs from refutations.

2.3 Two Faithful Translations of N

As was suggested in Sect. 1, the Gödel-McKinsey-Tarski translation of intuitionistic logic into $\mathsf{S4}$ forms the core of similar translations for N. In [23], Richmond Thomason provided a translation of N into $\mathsf{S4}$, acknowledging the influence of the Gödel-McKinsey-Tarski translation. Thomason's translation, of course, is limited by the use of a single modal operator, entailing that refutations must reduce to proofs. To more closely mirror Nelson's intuitions, we can increase the expressivity of Thomason's map by defining an enriched version $\_^{\ddagger}$ translating the language of N into the bimodal language of $\mathsf{S4}^2_{\mathsf{Triv}}$.

Definition 14. *The map* $\_^{\ddagger}$ *maps formulae of the language of* N *to the bimodal language of* $\mathsf{S4}^2_{\mathsf{Triv}}$ *by the following recursive definition:*

$$\begin{array}{ll}
(p)^{\ddagger} = \Box_0 p \textit{ for } p \in \mathbf{At} & (\sim p)^{\ddagger} = \hexagon_1 p \textit{ for } p \in \mathbf{At} \\
(\varphi \wedge \psi)^{\ddagger} = \varphi^{\ddagger} \wedge \psi^{\ddagger} & (\sim(\varphi \wedge \psi))^{\ddagger} = (\sim\varphi)^{\ddagger} \vee (\sim\psi)^{\ddagger} \\
(\varphi \vee \psi)^{\ddagger} = \varphi^{\ddagger} \vee \psi^{\ddagger} & (\sim(\varphi \vee \psi))^{\ddagger} = (\sim\varphi)^{\ddagger} \wedge (\sim\psi)^{\ddagger} \\
(\varphi \rightarrow \psi)^{\ddagger} = \Box_0(\varphi^{\ddagger} \rightarrow \psi^{\ddagger}) & (\sim(\varphi \rightarrow \psi))^{\ddagger} = (\varphi^{\ddagger}) \wedge (\sim\psi)^{\ddagger} \\
\multicolumn{2}{c}{(\sim\sim\varphi)^{\ddagger} = \varphi^{\ddagger}}
\end{array}$$

Note that while Nelson's refutability semantics for N treats P-realizers and N-realizers—proofs and refutations—as distinct categories of objects, Thomason's translation treats Nelsonian refutability as a defined notion, that is, as the provability of a negation. Thomason's translation is therefore not fine-grained enough to fully capture Nelson's intuitions. More adequate is a translation into the language $\mathscr{L}^2_{\Box}$, in which distinct, Gödelian readings can be given to $\Box_0$ and $\hexagon_1$ that syntactically distinguish between proofs and refutations.

Given the translations $\_^{\dagger}$ and $\_^{\ddagger}$, we are capable of providing a more fine-grained representation of one of Thomason's principal results from [23], where φ is a formula in the language of N:

Theorem 1 (Thomason). $\vdash_{\mathsf{N}} \varphi$ *iff* $\vdash_{\mathsf{S4}} (\varphi^{\ddagger})^{\dagger}$

As a corollary, we may infer the following for formulae φ in the language of N:

Observation 2. $\vdash_{\mathsf{N}} \varphi$ *iff* $\vdash_{\mathsf{S4}^2_{\mathsf{Triv}}} \varphi^{\ddagger}$

Following Nelson's picture of N as a logic that tacitly distinguishes proofs from refutations, we can look to two-agent epistemic logics to uncover the implicit machinery of proofs *and refutations* in the system $\mathsf{S4}^2_{\mathsf{Triv}}$.

3 Two-Agent Logics of Proof and N

Myriad accounts of multiple-agent epistemic logic have been introduced and analyzed, that is, systems in which an agent may have knowledge concerning *other* agents' knowledge. The system most important to the present task, however, is the two-agent generalization of Artemov's LP, in which the *justifications* of one agent are influenced by the justifications of the other agent. In this section, we will pay particular attention to Tatiana Yavorskaya-Sidon's two-agent logic of proofs LP^2, described in [29,30], although we will employ some of the conventions of Antonis Achilleos' study of LP^2 and related systems from [1].

The intuitions underlying Yavorskaya-Sidon's work distinguish the justifications of each agent from those of any other, leading to an agent-centric distinction between *categories* of justification. While Yavorskaya-Sidon interprets the typing of justification terms as an indication of *the particular agent* who possesses some justification or other, the agents are abstract enough to stand in for the categories of justification we have been considering. When we come to examine explicit interpretations of $\mathsf{S4}^2_{\mathsf{Triv}}$, we will inherit a great degree of flexibility with respect to the basis upon which these types are distinguished.

3.1 Two-Agent LP with Conversions

Yavorskaya-Sidon initially offers the basic system LP^2 before defining extensions determined by different stipulations about the interactions between different agents' justifications. We presume that each agent may possess a distinct type of justification, represented by the members of the sets $\mathbf{P}_0$ and $\mathbf{P}_1$.

Definition 15. *$\mathbf{P}_0$ and $\mathbf{P}_1$ are two sets of justification terms defined so that there are two disjoint sets of variables $\mathbf{V}_i = \{x_i^0, x_i^1, ..., y_i^0, ...\}$ and two disjoint sets of constants $\mathbf{C}_i = \{c_i^0, c_i^1, ..., d_i^0, ...\}$. Each $\mathbf{P}_i$ is recursively constructed in Backus-Naur form with $x_i^j \in \mathbf{V}_i$ and $c_i^k \in \mathbf{C}_i$:*

$$t ::= x_i^j | c_i^k | t + t | t \cdot t | !t$$

An atomic proof constant in $\mathbf{C}_i$ is interpreted as a simple proof, while the above operations on these terms construct iteratively more complex proofs. A term $s+t$—the *sum* of proofs s and t—is a proof that proves precisely those statements either proven by s or proven by t. A term $s \cdot t$—the *application* of the proof s to the proof t—corresponds to a proof such that if s proves a conditional statement the antecedent of which is proven by t, then $s \cdot t$ proves the consequent. Finally, the term $!t$ represents a *proof checker* that, when applied to a purported proof t of a formula, can verify whether t indeed proves that formula.

Now, we define a language including these justification terms, again appealing to a notational variant of the one used in [30].

Definition 16. *The language $\mathscr{L}_2$ is recursively defined in Backus-Naur form with $p \in \mathbf{At}$, $t_0 \in \mathbf{P}_0$, and $t_1 \in \mathbf{P}_1$:*

$$\varphi ::= p | \neg\varphi | \varphi \wedge \varphi | \varphi \vee \varphi | \varphi \to \varphi | \llbracket t_0 \rrbracket_0 \varphi | \llparenthesis t_1 \rrparenthesis_1 \varphi$$

The intended reading of $\llbracket t \rrbracket_0 \varphi$ is that t is a *proof* of φ while $\llparenthesis t \rrparenthesis_1$ represents that t is a *refutation* of φ.

An important notion in justification logic is that of a *constant specification.* In the context of many-agent justification logic, a constant specification is a set of formulae reflecting the extent to which the agents possess justifications for *axioms.*

Definition 17. *A* constant specification CS *is a set (possibly empty) of formulae of* $\mathscr{L}_2$ *of the form:*

$$\sigma_0\sigma_1...\sigma_{n-1}\varphi$$

where φ *is an instance of an axiom and each* σ_i *is an instance of either* $\llbracket t_i \rrbracket_0$ *or* $\llparenthesis t_i \rrparenthesis_1\neg$. *We assume that a constant specification* CS *enjoys the closure property such that whenever* $\llbracket s \rrbracket_0\psi \in CS$ *or* $\llparenthesis t \rrparenthesis_1\neg\psi \in CS$, *also* $\psi \in CS$. *We call the maximal constant specification* TCS *for* total *constant specification.*

Definition 18. *We say that a constant specification* CS *is* axiomatically appropriate *if for every formula* ψ *such that* ψ *is either*

- *a substitution instance of an axiom scheme* $\mathbf{A}$, *or*
- *a member of* CS

there exist $c, c' \in \mathbf{C}_i$ *such that* $\llbracket c \rrbracket_0\psi \in CS$ *and* $\llparenthesis c' \rrparenthesis_1\neg\psi \in CS$.[5]

Now, to define LP^2 and its extensions from a proof-theoretic perspective, we consider a number of axiom schemes.

For each $i \in \{0, 1\}$ and $s, t \in \mathbf{P}_i$, we include the following axiom schema:

Application$_0$	$\llbracket s \rrbracket_0(\varphi \to \psi) \to (\llbracket t \rrbracket_0\varphi \to \llbracket s \cdot t \rrbracket_0\psi)$
Application$_1$	$\llparenthesis s \rrparenthesis_1(\varphi \wedge \neg\psi) \to (\llparenthesis t \rrparenthesis_1\neg\varphi \to \llparenthesis s \cdot t \rrparenthesis_1\neg\psi)$
Sum$_0$	$\llbracket s \rrbracket_0\varphi \to \llbracket s + t \rrbracket_0\varphi$ and $\llbracket t \rrbracket_0\varphi \to \llbracket s + t \rrbracket_i\varphi$
Sum$_1$	$\llparenthesis s \rrparenthesis_1\varphi \to \llparenthesis s + t \rrparenthesis_1\varphi$ and $\llparenthesis t \rrparenthesis_1\varphi \to \llparenthesis s + t \rrparenthesis_1\varphi$
Proof Checker$_0$	$\llbracket t \rrbracket_0\varphi \to \llbracket !t \rrbracket_0\llbracket t \rrbracket_0\varphi$
Proof Checker$_1$	$\llparenthesis t \rrparenthesis_1\varphi \to \llparenthesis !t \rrparenthesis_1\neg\llparenthesis t \rrparenthesis_1\varphi$
Reflection$_i$	$\llbracket t \rrbracket_0\varphi \to \varphi$
Reflection$_i$	$\llparenthesis t \rrparenthesis_1\varphi \to \neg\varphi$

This permits us to define the two-agent logic of proofs as follows.

Definition 19. *With respect to a constant specification* CS, *we define the* two-agent logic of proofs LP^2_{CS}

$$\mathsf{PC} \oplus \left(\bigoplus_{i \in \{0,1\}} (\mathbf{Application}_i \oplus \mathbf{Sum}_i \oplus \mathbf{Proof\ Checker}_i \oplus \mathbf{Reflection}_i)\right) \oplus CS$$

We employ the convention of referring to LP^2_{TCS} *as* LP^2.

In LP^2, the only particular interactions between agents are those codified by a constant specification. For example, in the case of TCS, the interaction amounts to common knowledge of axioms. To capture more elaborate interactions between the two agents of LP^2, Yavorskaya-Sidon introduces a number of unary operations on proof terms including $\uparrow^0_1$ and $\uparrow^1_0$ representing the conversion between agents' justifications. With these operations, Yavorskaya-Sidon proposes extensions of LP^2 by enriching its theory by the following axioms:

[5] The property of being axiomatically appropriate is sometimes described as "fullness" in, *e.g.*, [5] or [21].

0-1 Conversion	$\llbracket t \rrbracket_0 \neg\varphi \rightarrow \llparenthesis \uparrow_0^1 t \rrparenthesis_1 \varphi$
1-0 Conversion	$\llparenthesis t \rrparenthesis_1 \varphi \rightarrow \llbracket \uparrow_1^0 t \rrbracket_0 \neg\varphi$

As an illustration, note that it is trivial to convert a classical proof of φ into a refutation of $\neg\varphi$. If we read the operator $\llbracket t \rrbracket_0 \varphi$ as asserting that t is a proof of φ, then this example is captured by **0-1 Conversion** to infer $\llparenthesis \uparrow_0^1 t \rrparenthesis_1 \neg\varphi$, *i.e.*, that $\uparrow_0^1 t$ modifies proof t to produce a refutation of $\neg\varphi$.

From these axioms, Yavorskaya-Sidon defines the two-agent logic of proofs $\mathsf{LP}^2_{\uparrow\uparrow}$:

Definition 20. *With respect to a constant specification CS, the logic $\mathsf{LP}^2_{CS\uparrow\uparrow}$ is defined so that:*

$$\mathsf{LP}^2_{CS\uparrow\uparrow} = \mathsf{LP}^2_{CS} \oplus \textbf{0-1 Conversion} \oplus \textbf{1-0 Conversion}$$

The semantical approach to justification logics that we will adopt follows the approach initiated by Alexey Mkrtychev in [13] and exhaustively developed by Melvin Fitting in a number of papers, *e.g*, [5].

Definition 21. *An LP^2* Fitting model $\mathfrak{M}$ *is a 6-tuple $\langle W, \leq_0, \leq_1, \mathcal{E}_0, \mathcal{E}_1, v\rangle$ such that*

- *W is a nonempty set of points*
- *$\leq_0$ and $\leq_1$ are partial orders on W*
- *$\mathcal{E}_i$ is a function from $\mathbf{P}_i \times \mathscr{L}_2$ to $\wp(W)$*
- *v is a map from $\mathbf{At}$ to $\wp(W)$*

where $\mathfrak{M}$ enjoys the following properties for each $i \in \{0, 1\}$ and $s, t \in \mathbf{P}_i$:

Monotonicity	*If $w \in \mathcal{E}_i(t, \varphi)$ and $w \leq_i w'$, then $w' \in \mathcal{E}_i(t, \varphi)$*
Application	$\mathcal{E}_i(s, \varphi \rightarrow \psi) \cap \mathcal{E}_i(t, \psi) \subseteq \mathcal{E}_i(s \cdot t, \psi)$
Sum	$\mathcal{E}_i(s, \varphi) \cup \mathcal{E}_i(t, \varphi) \subseteq \mathcal{E}_i(s + t, \varphi)$
Proof Checker	$\mathcal{E}_i(t, \varphi) \subseteq \mathcal{E}_i(!t, \llbracket t \rrbracket_i \varphi)$

In addition to the familiar notion of truth at a point, the Fitting model also is equipped with *evidence functions* $\mathcal{E}_0$ and $\mathcal{E}_1$. The intuitive reading of the fact that $w \in \mathcal{E}_i(t, \varphi)$ where $w \in W$ and $t \in \mathbf{P}_i$ is that t counts as adequate evidence for φ for agent i at state w. Note that in general truth of φ does not entail the existence of evidence for φ, nor does the existence of evidence for φ entail the truth of the formula.

Definition 22. *A model $\mathfrak{M}$* meets *a constant specification CS if for all $\llbracket t \rrbracket_i \varphi \in CS$ or $\llparenthesis t \rrparenthesis_i \neg\varphi \in CS$,*

$$W = \mathcal{E}_i(t, \varphi).$$

We then provide an account of a forcing relation in the model by adapting the particulars of Definition 9:

Definition 23. *In a model $\mathfrak{M} = \langle W, \leq_0, \leq_1, \mathcal{E}_0, \mathcal{E}_1, v\rangle$, the relation $\Vdash$ is defined by replacing the conditions for $\Box_0$ and $\bigcirc_1$ from Definition 9 with:*

- $\mathfrak{M}, w \Vdash \llbracket t \rrbracket_0 \varphi$ *if* $\begin{cases} \textit{for all } w' \textit{ such that } w \leq_0 w',\ w' \Vdash \varphi, \textit{ and} \\ w \in \mathcal{E}_i(t, \varphi) \end{cases}$
- $\mathfrak{M}, w \Vdash \llparenthesis t \rrparenthesis_1 \varphi$ *if* $\begin{cases} \textit{for all } w' \textit{ such that } w \leq_i w',\ w' \nVdash \varphi, \textit{ and} \\ w \in \mathcal{E}_i(t, \neg\varphi) \end{cases}$

Hence, for a formula $\llbracket t \rrbracket_0 \varphi$ to be true at a point w is not only that t is counted as adequate evidence for φ but also that φ is necessary along the relation $\leq_0$.

3.2 Interpreting N in $\mathsf{LP}^2_{\uparrow\ \uparrow}$

An important notion in justification logics is that of a *forgetful projection*, defined as an injective map from the language of a justification logic to that of its non-explicit, monomodal counterpart in which instances of justification terms are uniformly replaced by instances of the necessity operator $\Box$. For example, in the single-agent LP, the forgetful projection of the formula $\llbracket t \rrbracket (p \wedge \neg \llbracket s \rrbracket q)$ is $\Box(p \wedge \neg \Box q)$. In the field of justification logics, the notion of a forgetful projection is important because it provides a framework in which the correctness of an explicit justification logic with respect to its non-explicit counterpart can be judged.

As $\mathsf{S4}^2_{\mathsf{Triv}}$ is bimodal, the notion of forgetful projection must be generalized to the case of bimodal logics.

Definition 24. *The* forgetful projection φ° *of a formula* $\varphi \in \mathscr{L}_2$ *is determined by the map* $\_^\circ : \mathscr{L}_2 \to \mathscr{L}^2_\Box$ *defined by the following clauses:*

- $(p)^\circ = p$ *for* $p \in \mathbf{At}$
- $(\neg\varphi)^\circ = \neg(\varphi^\circ)$
- $(\varphi * \psi)^\circ = (\varphi^\circ) * (\psi^\circ)$ *for* $* \in \{\wedge, \vee, \to\}$
- $(\llbracket t \rrbracket_0 \varphi)^\circ = \Box_0(\varphi^\circ)$ *for all* $t \in \mathbf{P}_0$
- $(\llparenthesis t \rrparenthesis_1 \varphi)^\circ = \bigcirc_1(\varphi^\circ)$ *for all* $t \in \mathbf{P}_1$

In general, for a set of formulae Γ*, we will let* Γ° *denote the set* $\{\varphi^\circ \mid \varphi \in \Gamma\}$*.*

This effectively projects the rich language of LP^2 onto the less expressive bimodal language $\mathscr{L}^2_\Box$. Then the Realization Theorem correlating $\mathsf{S4}^2_{\mathsf{Triv}}$ and $\mathsf{LP}^2_{\uparrow\uparrow}$ in [30] gives us the following:

Theorem 2. (Yavorskaya-Sidon) $\vdash_{\mathsf{S4}^2_{\mathsf{Triv}}} \varphi$ *iff there is a formula* $\psi \in \mathscr{L}_{\mathsf{LP}^2_{\uparrow\uparrow}}$ *such that* $\psi^\circ = \varphi$ *and* $\vdash_{\mathsf{LP}^2_{\uparrow\uparrow}} \psi$*.*

Taken together, Observation 2 and Theorem 2 suggest that whenever one can prove a formula φ in N, one can constructively determine and represent its proof- and refutation-theoretic content in a corresponding $\mathsf{LP}^2_{\uparrow\uparrow}$ theorem.

This observation suggests that $\mathsf{LP}^2_{\uparrow\uparrow}$ supplies an appropriate account of proofs and disproofs in Nelson's N. Returning to Artemov's foundational picture of Int from Sect. 1, the foregoing theorems lead to the following Gödel-Artemov-style "foundational picture" of the logic of constructible falsity:

$$\mathsf{N} \hookrightarrow \mathsf{S4}^2_{\mathsf{Triv}} \hookrightarrow \mathsf{LP}^2_{\uparrow\uparrow} \hookrightarrow \text{Proofs and Disproofs}$$

Nelson's [14] ensures that the introduction of the formal system N is accompanied by an extraordinarily salient critique of the restrictions implicit in the intuitionistic account of constructivity. In hindsight, the clarity of the philosophical foundation codified by N—and its reliance on the same concepts employed in the Gödel-Artemov-style interpretation of Int—leads to a sense of inevitability with respect to the above picture.

4 Preliminary Remarks on Heyting-Brouwer Logic

I believe that the naturalness of the above "foundational picture" confirms that Nelson's distinctions were timely and serves to reinforce his critique of intuitionistic constructivity. Moreover, the ease by which Artemov's program extends to N demonstrates the utility of employing distinct modalities to model notions of falsification alongside provability.

There are quite a few neighbors of intuitionistic logic with embeddings into modal logics including a falsification-type interpretation of one or more modal operators. Fitting, for example, has given in [8] an evidence-based interpretation of the paraconsistent variant of N via an embedding. Yaroslav Shramko's [22] suggests a straightforward falsificationist interpretation of dual-intuitionistic logic, and Wansing's 2Int described in [27] involves a proof-and-refutation interpretation that is distinct from Nelson's. The story in these cases, however, is typically not as straightforward as Nelson's and Artemov's approach to these systems, and a hope for the Gödel-Artemov-style interpretation of N is that it might serve to help clarify the interpretations of these systems. I'd like to conclude by providing some preliminary remarks on the application of the foregoing work to the case of Rauszer's Heyting-Brouwer logic HB.

Just as Nelson suggests two categories of proof and refutation, Heinrich Wansing's [26] suggests that HB can be interpreted in terms of proofs and their *duals*, where a dual proof is a "canonical reduction to non-truth." Hence, it is *prima facie* plausible that the picture corresponding to HB can be filled in by examining a formal analogue to LP with elements corresponding not only to proofs but to dual proofs as well.

In [11,28], it has been demonstrated that a Gödel-McKinsey-Tarski-style translation of HB embeds the logic into the bimodal tense logic $\mathsf{K}_t\mathsf{T4}$. Given an adequate explicit counterpart to $\mathsf{K}_t\mathsf{T4}$—a system that by convention might be called $\mathsf{J}_t\mathsf{T4}$—the corresponding foundational picture might be represented as:

$$\mathsf{HB} \hookrightarrow \mathsf{K}_t\mathsf{T4} \hookrightarrow \mathsf{J}_t\mathsf{T4} \hookrightarrow \text{Proofs and Dual Proofs}$$

There are a number of extraordinarily interesting features that such a foundational picture seems to bring to light. For example, there is a clear analogy between the characteristic *interaction axioms* of $\mathsf{K}_t\mathsf{T4}$ governing the interaction between the tense operators and the negative introspection axiom **5**. Because HB is non-constructive—it proves a version of excluded middle with respect to its *conegation* connective—an adequate characterization of $\mathsf{J}_t\mathsf{T4}$ aligns the failure of constructivity and the interaction between proofs and dual proofs. Furthermore, there have been a number of issues concerning cut eliminability in HB—[18] shows, for example, that Rauszer's proof of cut-elimination is faulty—that might be clarified by making explicit the implicit interaction between proof and refutation in HB.

Although space considerations prevent describing a full characterization of such a $\mathsf{J}_t\mathsf{T4}$, we will close the paper with a useful and general demonstration that such a system in fact exists by extending the semantic proof of Realization introduced by Fitting in [6] or [7] to the multimodal case. Between these papers, Fitting proves that if KL is any normal monomodal logic with a corresponding justification logic JL, then given very modest conditions on the canonical model of JL, Realization holds.

First, let us examine an important property called *Internalization*. This is the property that for an axiomatically appropriate constant specification CS, a logic JL_{CS} is able to "internalize" its own notion of proof so that for every JL_{CS} theorem φ, one can construct a term t such that $[\![t]\!]_0\varphi$ is also a theorem. Then:

Theorem 3. **(Fitting)** *Let* KL *be a normal monomodal logic characterized by a class of Kripke frames* $\mathscr{F}$ *and let* JL *be an axiomatizable justification logic enjoying Internalization whose canonical frame (*i.e.*, the frame of the canonical model* $\mathfrak{M}_{\mathsf{JL}}$*) is a member of* $\mathscr{F}$*. Then every theorem of* KL *has a provable realization in a logic* JL_{CS} *for some constant specification* CS*.*

Much of this generality stems from Fitting's observation that the semantic proof of Realization for LP in [5] makes no essential use of the particulars of LP. In [7], in which the proof of [5] is corrected and refined, the proof of Realization only covers the monomodal case but even a cursory inspection of its arguments reveals that the proof extends to normal multimodal logics and their justification counterparts as well. Formally, the adaptation of the proof requires only changes to the bookkeeping conventions of [6,7].

In the monomodal case, Fitting tracks instances of "$\Box$" in modal formulae by decorating instances of the operator with distinct natural number subscripts, *e.g.*, an "annotated" version of the formula $\Box p \to (q \vee \Box\Box p)$ is $\Box^3 p \to (q \vee \Box^4\Box^7 p)$.[6] When "$\Box^i$" appears positively in an annotated formula φ, Fitting associates the operator with a distinct justification variable x^i to record the correspondence throughout the realization process. In the present paper, we have explicitly typed the justification terms, so a convention is required to associate instances of the operators $\Box_0$ and $\Box_1$ with the appropriate type of variable.

[6] In the monomodal case, Fitting indicates particular instances of "$\Box$" by decorating them with subscripts rather than superscripts. In the multimodal case, this notation would contradict our convention of distinguishing between distinct modal operators by subscripts.

We are thus able to provide the following generalization by describing a procedure to adapt Fitting's semantic proof to the multimodal case:

Observation 3. *Let* KL^n *be a normal multimodal logic with modalities* $\Box_0, ...,$ $\Box_{n-1}$ *that is characterized by a class of Kripke frames* $\mathscr{F}$ *and let* JL^n *be an axiomatizable, n-agent justification logic enjoying Internalization whose canonical frame (*i.e.*, the frame of the canonical model* $\mathfrak{M}_{\mathsf{JL}^n}$*) is a member of* $\mathscr{F}$*. Then every theorem of* KL^n *has a provable realization in a logic* JL^n_{CS} *for some constant specification* CS*.*

Proof. In his proof, Fitting fixes an enumeration of justification variables $x^0, x^1, ...$ and associates a variable x^i with the decorated operator $\Box^i$ for each $i \in \omega$. In response, we explicitly type our variables $\{x^0_0, x^0_1, ..., x^1_0, x^1_1, ...\}$ and for each $i < n$ associate variables x^j_i with the annotated operator $\Box^j_i$, that is, the instance of "$\Box_i$" annotated by the index j. Retaining a distinction between types of variables also demands that whenever Fitting appeals to a property such as Internalization, care is taken to prefix a provable formula with a justification term of the appropriate type. Likewise, whenever [7] proves a lemma by induction on complexity of formulae, clauses for formulae $\Box\varphi$ must be reproduced n times for formulae $\Box_i\varphi$, with minor adjustments in notation.

The clarity and generality of Fitting's proof of Realization in [7] ensures that these bookkeeping matters are very straightforward and, by the foregoing considerations, guarantee that the reach of Fitting's result extends to normal multimodal logics as well. The definition of the system $\mathsf{J}_t\mathsf{T4}$ and the proofs that the system has Strong Internalization and that its frames are among those which characterize $\mathsf{K}_t\mathsf{T4}$ are left for a future investigation into the Gödel-Artemov-style analysis of Heyting-Brouwer logic and other related systems.

References

1. Achilleos, A.: On the complexity of two-agent justification logic. In: Bulling, N., van der Torre, L., Villata, S., Jamroga, W., Vasconcelos, W. (eds.) CLIMA 2014. LNCS (LNAI), vol. 8624, pp. 1–18. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-09764-0_1
2. Artemov, S.: Operational modal logic. Technical report MSI 95–29, Cornell University (1995)
3. Artemov, S.: On two models of provability. In: Gabbay, D., Zakharyaschev, M., Goncharov, S.S. (eds.) Mathematical Problems from Applied Logics II, pp. 1–52. Springer, New York (2007). https://doi.org/10.1007/978-0-387-69245-6_1
4. Artemov, S.: The ontology of justifications in the logical setting. Stud. Logica. **100**(1–2), 17–30 (2012)
5. Fitting, M.: The logic of proofs, semantically. Ann. Pure Appl. Logic **132**(1), 1–25 (2005)
6. Fitting, M.: Realization implemented. Technical report TR-2013005, City University of New York (2013)
7. Fitting, M.: Justification logics and realization. Technical report TR-2014004, City University of New York (2014)

8. Fitting, M.: Paraconsistent logic, evidence, and justification. Stud. Logica, 1–18 (2017, to appear)
9. Gödel, K.: Eine Interpretation des intuitionistischen Aussagenkalküls. In: Ergebnisse eines mathematischen Kolloquiums, vol. 4, pp. 39–40 (1933)
10. López-Escobar, E.G.K.: Refutability and elementary number theory. Indagationes Math. **75**(4), 362–374 (1972)
11. Łukowski, P.: Modal interpretation of Heyting-Brouwer logic. Bull. Sect. Logic **25**(2), 80–83 (1996)
12. McKinsey, J., Tarski, A.: Some theorems about the sentential calculi of Lewis and Heyting. J. Symbolic Logic **13**(1), 1–15 (1948)
13. Mkrtychev, A.: Models for the logic of proofs. In: Adian, S., Nerode, A. (eds.) LFCS 1997. LNCS, vol. 1234, pp. 266–275. Springer, Heidelberg (1997). https://doi.org/10.1007/3-540-63045-7_27
14. Nelson, D.: Constructible falsity. J. Symbolic Logic **14**(1), 16–26 (1949)
15. Nelson, D.: Negation and separation of concepts in constructive systems. In: Heyting, A. (ed.) Constructivity in Mathematics, pp. 208–225. North-Holland, Amsterdam (1959)
16. Nelson, D., Almukdad, A.: Constructible falsity and inexact predicates. J. Symbolic Logic **49**(1), 231–233 (1984)
17. Odintsov, S.: Constructive Negations and Paraconsistency. Springer, Dordrecht (2008). https://doi.org/10.1007/978-1-4020-6867-6
18. Pinto, L., Uustalu, T.: Proof search and counter-model construction for bi-intuitionistic propositional logic with labelled sequents. In: Giese, M., Waaler, A. (eds.) TABLEAUX 2009. LNCS (LNAI), vol. 5607, pp. 295–309. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02716-1_22
19. Rauszer, C.: A formalization of the propositional calculus of H-B logic. Stud. Logica. **33**(1), 23–34 (1974)
20. Rauszer, C.: Semi-Boolean algebras and their application to intuitionistic logic with dual operations. Fundamenta Math. **83**(1), 219–249 (1974)
21. Rubtsova, N.: Evidence reconstruction of epistemic modal logic S5. In: Grigoriev, D., Harrison, J., Hirsch, E.A. (eds.) CSR 2006. LNCS, vol. 3967, pp. 313–321. Springer, Heidelberg (2006). https://doi.org/10.1007/11753728_32
22. Shramko, Y.: A modal translation for dual-intuitionistic logic. Rev. Symbolic Logic **9**(2), 251–265 (2016)
23. Thomason, R.H.: A semantical study of constructible falsity. Zeitschrift für Mathematische Logik und Grundlagen der Mathematik **15**(16–18), 247–257 (1969)
24. Wansing, H.: Sequent systems for modal logics. In: Gabbay, D.M., Guenthner, F. (eds.) Handbook of Philosophical Logic, vol. 8, pp. 61–145. Kluwer, Boston (2002)
25. Wansing, H.: Constructive negation, implication, and co-implication. J. Appl. Non-class. Logics **18**(2–3), 341–364 (2008)
26. Wansing, H.: Proofs, disproofs, and their duals. In: Areces, C., Goldblatt, R. (eds.) Advances in Modal Logic, vol. 7, pp. 483–505. College Publications, London (2008)
27. Wansing, H.: Falsification, natural deduction and bi-intuitionistic logic. J. Logic Comput. **26**(1), 425–450 (2016)
28. Wolter, F.: On logics with coimplication. J. Philos. Logic **27**(4), 353–387 (1998)
29. Yavorskaya (Sidon), T.: Multi-agent explicit knowledge. In: Grigoriev, D., Harrison, J., Hirsch, E.A. (eds.) CSR 2006. LNCS, vol. 3967, pp. 369–380. Springer, Heidelberg (2006). https://doi.org/10.1007/11753728_38
30. Yavorskaya (Sidon), T.: Interacting explicit evidence systems. Theory Comput. Syst. **43**(2), 272–293 (2008)

Probabilistic Reasoning About Simply Typed Lambda Terms

Silvia Ghilezan[1,2], Jelena Ivetić[1], Simona Kašterović[1], Zoran Ognjanović[2], and Nenad Savić[3(✉)]

[1] Faculty of Technical Sciences, University of Novi Sad, Novi Sad, Serbia
{gsilvia,jelenaivetic,simona.k}@uns.ac.rs
[2] Mathematical Institute SANU, Belgrade, Serbia
zorano@mi.sanu.ac.rs
[3] Institute of Computer Science, University of Bern, Bern, Switzerland
savic@inf.unibe.ch

Abstract. Reasoning with uncertainty has gained an important role in computer science, artificial intelligence and cognitive science. These applications urge for development of formal models which capture reasoning of probabilistic features. We propose a *formal model* for reasoning about probabilities of simply typed lambda terms. We present its syntax, Kripke-style semantics and axiomatic system. The main results are the corresponding soundness and strong completeness, which rely on two key facts: the completeness of simple type assignment and the existence of a maximal consistent extension of a consistent set.

Keywords: Simply typed lambda calculus · Probabilistic logic Soundness · Strong completeness

1 Introduction

In the last three decades several formal tools have been developed for reasoning about uncertain knowledge. One of these approaches concerns formalization in terms of probabilistic logics. Although the idea of probabilistic logic can be traced back to Leibnitz, Lambert and Boole, the modern development was started by Nils Nilsson, who tried to provide a logical framework for uncertain reasoning [22]. After Nilsson, a number of researchers proposed formal systems for probabilistic reasoning, for example [9,10]. The general lack of compactness for probabilistic logics causes that one of the main proof-theoretical problems in this framework is to provide a strongly complete axiomatic system. Several infinitary logics have been introduce to deal with that issue, a detailed overview can be found in [16,24,25]. Note that the term "infinitary" concerns the meta language only, i.e. the object language is countable, and formulas are finite, while only proofs are allowed to be infinite. It turns out that this approach can be combined, for example, with temporal [7,23] and intuitionistic reasoning [20]. So, building on our previous experience (e.g. see [17,24,26]), we describe a class

S. Artemov and A. Nerode (Eds.): LFCS 2018, LNCS 10703, pp. 170–189, 2018.
https://doi.org/10.1007/978-3-319-72056-2_11

of so called measurable models for a probabilistic extension of the simply typed lambda terms and give a sound and strongly complete (every consistent set of formulas is satisfiable) infinitary axiomatization.

The λ-calculus, proposed by Church in the early 1930s, is a simple formal system capable of expressing all effectively computable functions, and it is equivalent to Turing machines. Typed λ-calculus is a restricted system, where application is controlled by objects (types) assigned to λ-terms. Already Church introduced the system with simple (functional) types that turned out to represent the computational interpretation of intuitionistic natural deduction as stated by the well-known Curry-Howard correspondence. A variety of type systems (such as intersection types, dependent types, polymorphic types, etc.) were proposed in the last decades, finding application in programming languages for certified compilers, automated theorem provers and proof assistants, software verification, computational linguistics, among others. For more details we refer the reader to [3,4,11]. Soundness and completeness of the simple type assignment has been proved with respect to semantics developed in [13,14].

Reasoning with uncertainty has gained an important role in computer science, artificial intelligence and cognitive science. These applications urge for development of formal models which capture reasoning of probabilistic features [12]. This is our motivation for developing a new formal model for reasoning about simply typed lambda terms.

Contributions and Main Results. We introduce in this paper a formal model $\mathsf{P}\Lambda_{\rightarrow}$ for reasoning about probabilities of simply typed lambda terms which is a combination of lambda calculus and probabilistic logic. We propose its syntax, Kripke-style semantics and an infinitary axiomatization.

We first endow the language of typed lambda calculus with a probabilistic operator $P_{\geq s}$ and obtain formulas of the form

$$P_{\geq s} M : \sigma$$

to express that the probability that the lambda term M is of type σ is equal to or greater than s. More generally, formulas are of the form $P_{\geq s}\alpha$, where α is a typed lambda statement $M : \sigma$ or its Boolean combination. We then propose a semantics of $\mathsf{P}\Lambda_{\rightarrow}$ based on a set of possible worlds, where each possible world is a lambda model. The set of possible worlds is equipped with a probability measure μ. The set $[\alpha]$ is the set of possible worlds that satisfy the formula α. Then the probability of α is obtained as $\mu([\alpha])$. Further, we give an infinitary axiomatization of $\mathsf{P}\Lambda_{\rightarrow}$ and prove the deduction theorem.

The main results are the soundness and strong completeness of $\mathsf{P}\Lambda_{\rightarrow}$ with respect to the proposed model, where strong completeness means that every consistent set of formulas is satisfiable. The construction of the canonical model is crucial for the proof and relays on two key facts. The first one is that the simple type assignment is complete with respect to the simple semantics and the second one is the property that every consistent set can be extended to the maximal consistent set.

Related Work. In the last decade, several probabilistic extensions of the λ-calculus have been introduced and investigated. They are concerned with introducing non-determinism and probabilities into the syntax and operational semantics of the λ-calculus in order to formalize computation in the presence of uncertainty rather than with providing a framework that would enable probabilistic reasoning about typed terms and type assignments.

Audebaud and Paulin-Mohring in [2] gave the axiomatic rules for the estimation of the probability that programs satisfy some given properties. Furthermore, Dal Lago and Zorzi in [19] considered a non deterministic extension of lambda calculus, defined small-step and big-step semantics, and proved that the calculus is sound and complete with respect to computable probability distributions, whereas Bizjak and Birkedal in [5] constructed a step-indexed logical relations for a probabilistic extension of a certain higher-order programming language and showed that the relation is sound and complete with respect to the contextual preorder. Ehrhard et al. in [8] study the probabilistic coherent spaces as a denotational semantics and show soundness of a probabilistic extension of the untyped λ-calculus, which is a quantitative refinement to the soundness of the untyped λ-calculus with respect to the Scott's model in a probabilistic setting.

A slightly similar approach to ours, that provides a framework for probabilistic reasoning about typed terms, was treated by Cooper et al. in [6], where the authors proposed a probabilistic type theory in order to formalize computation with statements of the form "a given type is assigned to a given situation with probability p". However, the developed theory was used for analyzing semantic learning of natural languages in the domain of computational linguistics, and no soundness or completeness issues were discussed.

We provide a formal model for probabilistic reasoning about simply typed lambda terms. Our formal model is developed along the lines of the method that was used in [17,18] to obtain a formal model for probabilistic justification logic. However, the logic of uncertain justification already existed ([21]), so the authors have compared in [17] their logic with Milnikel's logic proposed in [21], whereas, to the best of our knowledge, the formal model that we propose is the first one.

Outline of the Paper. Section 2 revisits basic notions of lambda calculus, simple type assignment and simple semantics. In Sect. 3 we present the syntax and Kripke-style semantics of our probabilistic formal model for reasoning about simply typed lambda terms. The axiomatic system, together with the soundness theorem, is given in Sect. 4. The completeness of the proposed probabilistic formal model is proved in Sect. 5.

2 Simple Type Assignment $\boldsymbol{\Lambda}_{\rightarrow}$

In this section, we recall some basic notions of lambda calculus ([3]), simple types ([4,14]), lambda models ([3,14,15]) and revisit the soundness and completeness result for the simple type assignment proved in [13].

2.1 Lambda Terms and Types

We recall now some basic notions of the simply typed lambda calculus.

Let $\mathtt{V}_\Lambda = \{x, y, z, \ldots, x_1, \ldots\}$ be a countable set of λ-term variables. *Terms (λ-terms)* are generated by the following grammar:

$$M ::= x \mid \lambda x.M \mid MM.$$

The set of all terms is denoted by Λ and is ranged over by $M, N, \ldots, M_1, \ldots$. The operator λx is a binder and the set of *free variables* of a term M is defined as usual. The α-conversion, the renaming of bound variables, enables to implement Barendregt's convention that bound variables are distinct from free variables.

The β-reduction is a rewriting rule $(\lambda x.M)N \to_\beta M[N/x]$. The definition and main properties of β-reduction (and $\beta\eta$-reduction) can be found in [3,14]. The lambda term M is β-equal, (β-convertible), to the lambda term N (notion $M =_\beta N$) if and only if there is a sequence $M \equiv N_0, N_1, \ldots, N_n \equiv N$, where $N_i \to_\beta N_{i+1}$ or $N_{i+1} \to_\beta N_i$ for all $i \in \{0, 1, \ldots, n\}$.

Let $\mathtt{V}_{\mathtt{Type}} = \{a, b, c, \ldots\}$ be a denumerable set of propositional variables. *Types (simple types)* are generated by the following grammar:

$$\sigma ::= a \mid \sigma \to \sigma.$$

The set of all types is denoted by $\mathtt{Type}$ and is ranged over by $\sigma, \tau, \ldots, \sigma_1, \ldots$.

A *lambda statement* is an expression of the form $M : \sigma$, where $M \in \Lambda$ and $\sigma \in \mathtt{Type}$. Moreover, $x : \sigma$ is a *basic statement.* A *basis (context)* is a set of basic statements with distinct term variables (can be infinite).

Definition 1. *The simple type assignment, $\Lambda_\to$, is defined ([13]) as follows:*

$$\frac{M : \sigma \to \tau \qquad N : \sigma}{MN : \tau}\ (\to_E)$$

$$\frac{\begin{array}{c}[x : \sigma] \\ \vdots \\ M : \tau\end{array}}{\lambda x.M : \sigma \to \tau}\ (\to_I)$$

$$\frac{M : \sigma \qquad M =_\beta N}{N : \sigma}\ (\mathrm{eq})$$

If $M : \sigma$ is derivable by the given rules from a basis Γ, it is denoted by $\Gamma \vdash M : \sigma$. In the sequel we work with this simple type assignment *à la* Curry. There is an equivalent simple type assignment *à la* Church, which is out of our scope.

2.2 Lambda Models

We assume that the reader is familiar with the notion of the lambda model and the *interpretation of terms* in it. Basic notions and definitions can be found in [3,13–15].

The term model $\mathcal{M} = \langle D, \cdot, [\![\ ]\!] \rangle$ is defined as follows:

Definition 2

(i) *The domain of a term model is a set of all convertibility-classes of terms. For $M \in \Lambda$, the convertibility-class represented by M will be denoted by $[M]$, i.e., $[M] = \{N : N =_\beta M\}$.*

(ii) *If $\rho : \mathtt{V}_\Lambda \to D$ is the valuation of term variables in D, then $[\![M]\!]_\rho \in D$ is the interpretation of $M \in \Lambda$ in $\mathcal{M}$ via ρ.*

(iii) *The map $\cdot$ is defined by*

$$[M] \cdot [N] = [MN],$$

and $[\![\ ]\!]_\rho$ is defined by

$$[\![M]\!]_\rho = [M[N_1, \dots, N_n/x_1, \dots, x_n]],$$

where $x_1, \dots, x_n$ are the free variables of M, and $\rho(x_i) = [N_i]$ and $[\cdots/\cdots]$ is simultaneous substitution.

(iv) *Let $\xi : \mathtt{V}_{\mathtt{Type}} \to \mathcal{P}(D)$ be a valuation of type variables. The interpretation of $\sigma \in \mathtt{Type}$ in $\mathcal{M}$ via ξ, denoted by $[\![\sigma]\!]_\xi \in \mathcal{P}(D)$, is defined:*
- $[\![a]\!]_\xi = \xi(a)$;
- $[\![\sigma \to \tau]\!]_\xi = \{d \in D \mid \forall e \in [\![\sigma]\!]_\xi,\ d \cdot e \in [\![\tau]\!]_\xi\}$.

(v)
- $\mathcal{M}, \rho, \xi \models M : \sigma$ *iff* $[\![M]\!]_\rho \in [\![\sigma]\!]_\xi$;
- $\mathcal{M}, \rho, \xi \models \Gamma$ *iff* $\mathcal{M}, \rho, \xi \models x : \sigma$ *for all* $x : \sigma \in \Gamma$;
- $\Gamma \models M : \sigma$ *iff* $(\forall \mathcal{M}, \rho, \xi \models \Gamma)\ \mathcal{M}, \rho, \xi \models M : \sigma$.

The soundness and completeness of type assignment is proved in [13] with this notion of lambda model. The above semantics is called *the simple semantics.* The following results are the key for proving strong completeness for the logic we propose in this paper.

Theorem 1 (Soundness). $\Gamma \vdash M : \sigma \Rightarrow \Gamma \models M : \sigma$.

Theorem 2 (Completeness). $\Gamma \models M : \sigma \Rightarrow \Gamma \vdash M : \sigma$.

3 Probabilistic Logical System for Simply Typed Lambda Terms $\mathsf{P}\Lambda_\to$

The probabilistic logical system for typed lambda terms, $\mathsf{P}\Lambda_\to$, is a probabilistic logic over the simple type assignment $\Lambda_\to$. In this section, we introduce the syntax and semantics of $\mathsf{P}\Lambda_\to$.

3.1 Syntax of $\mathsf{P}\Lambda_{\rightarrow}$

Let S be the set of rational numbers from $[0,1]$, i.e., $\mathsf{S} = [0,1] \cap \mathbb{Q}$. The *alphabet* of the logic $\mathsf{P}\Lambda_{\rightarrow}$ consists of

- all symbols needed to define simply typed lambda terms, given in Sect. 2.1,
- the classical propositional connectives $\neg$ and $\wedge$,
- the list of probability operators $P_{\geq s}$, for every $s \in \mathsf{S}$.

Other propositional connectives $\Rightarrow$, $\vee$, $\Leftrightarrow$ are defined as usual.

Basic Formulas. All lambda statements of the form $M : \sigma$, where $M \in \Lambda$ and $\sigma \in \mathtt{Type}$, or statements of the same form connected with Boolean connectives, will be called *basic formulas*. Basic formulas are generated by the following grammar:

$$\mathsf{For_B} \quad \alpha ::= M : \sigma \mid \alpha \wedge \alpha \mid \neg\alpha.$$

The set of all basic formulas is denoted by $\mathsf{For_B}$ and will be ranged over by $\alpha, \beta, \ldots$, possibly indexed.

Probabilistic Formulas. If $\alpha \in \mathsf{For_B}$ and $s \in \mathsf{S}$, then a *basic probabilistic formula* is any formula of the form $P_{\geq s}\alpha$. The set of all probabilistic formulas, denoted by $\mathsf{For_P}$, is the smallest set containing all basic probabilistic formulas which is closed under Boolean connectives.

Probabilistic formulas are generated by the following grammar:

$$\mathsf{For_P} \quad \phi ::= P_{\geq s}\alpha \mid \phi \wedge \phi \mid \neg\phi.$$

The set $\mathsf{For_P}$ will be ranged over by ϕ, $\psi, \ldots$, possibly with subscripts.

Formulas of $\mathsf{P}\Lambda_{\rightarrow}$. The language of $\mathsf{P}\Lambda_{\rightarrow}$ consists of both basic formulas and probabilistic formulas

$$\mathsf{For}_{\mathsf{P}\Lambda_{\rightarrow}} = \mathsf{For_B} \cup \mathsf{For_P}.$$

The set of formulas $\mathsf{For}_{\mathsf{P}\Lambda_{\rightarrow}}$ will be ranged over by $\mathfrak{A}, \mathfrak{A}_1, \mathfrak{A}_2, \ldots$.

We use the following abbreviations to introduce other inequalities:

$$\begin{aligned} &P_{<s}\alpha \text{ stands for } \neg P_{\geq s}\alpha,\\ &P_{\leq s}\alpha \text{ stands for } P_{\geq 1-s}\neg\alpha,\\ &P_{>s}\alpha \text{ stands for } \neg P_{\leq s}\alpha,\\ &P_{=s}\alpha \text{ stands for } P_{\geq s}\alpha \wedge \neg P_{>s}\alpha. \end{aligned}$$

We also denote both $\alpha \wedge \neg\alpha$ and $\phi \wedge \neg\phi$ by $\bot$ (and dually for $\top$).

Note that neither mixing of basic formulas and probabilistic formulas, nor nested probability operators is allowed.

For example, the following two expressions are *not* (well defined) formulas of the logic $\mathsf{P}\Lambda_{\rightarrow}$:

$$\alpha \wedge P_{\geq \frac{1}{2}}\beta, \qquad\qquad P_{\geq \frac{1}{3}}P_{\geq \frac{1}{2}}\alpha.$$

The former is not well defined since it is a Boolean combination of a basic formula and probabilistic formula, whereas the latter is not well defined $\mathsf{P}\Lambda_{\rightarrow}$ formula because it contains nested probability operators.

3.2 Semantics of $\mathsf{P\Lambda}_{\rightarrow}$

The semantics for $\mathsf{P\Lambda}_{\rightarrow}$ is a Kripke-style semantics based on the possible-world approach.

Definition 3 **($\mathsf{P\Lambda}_{\rightarrow}$-structure).** *A $\mathsf{P\Lambda}_{\rightarrow}$-structure is a tuple $\mathcal{M} = \langle W, \rho, \xi, H, \mu\rangle$, where:*

(i) *W is a nonempty set of worlds, where each world is one term model, i.e., for every $w \in W$, $w = \langle \mathcal{L}(w), \cdot_w, [\![\ ]\!]_w\rangle$;*
(ii) $\rho : \mathsf{V}_{\Lambda} \times \{w\} \longrightarrow \mathcal{L}(w)$, $w \in W$;
(iii) $\xi : \mathsf{V}_{\mathsf{Type}} \times \{w\} \longrightarrow \mathcal{P}(\mathcal{L}(w))$, $w \in W$;
(iv) *H is an algebra of subsets of W, i.e. $H \subseteq \mathcal{P}(W)$ such that*
  - $W \in H$,
  - *if $U, V \in H$, then $W \setminus U \in H$ and $U \cup V \in H$;*

(v) *μ is a finitely additive probability measure defined on H, i.e.,*
  - $\mu(W) = 1$,
  - *if $U \cap V = \emptyset$, then $\mu(U \cup V) = \mu(U) + \mu(V)$, for all $U, V \in H$.*

The elements of H are called *measurable worlds*. We will write $\rho_w(x)$, instead of $\rho(x, w)$ and similarly for ξ.

We say that a lambda statement $M : \sigma$ holds in a world w, notation $w \models M : \sigma$, if and only if

$$[\![M]\!]^w_\rho \in [\![\sigma]\!]^w_\xi,$$

where $[\![M]\!]^w_\rho$ is the interpretation of a term M in a world w via ρ, and $[\![\sigma]\!]^w_\xi$ is the interpretation of a type σ in a world w via ξ. Also, we define that
$w \models M : \sigma \wedge N : \tau$ iff $w \models M : \sigma$ and $w \models N : \tau$,
$w \models \neg(M : \sigma)$ iff $w \not\models M : \sigma$.

For a given $\alpha \in \mathsf{For}_{\mathsf{B}}$ and $\mathsf{P\Lambda}_{\rightarrow}$-structure $\mathcal{M}$, let

$$[\alpha]_{\mathcal{M}} = \{w \in W \mid w \models \alpha\}.$$

We will omit the subscript $\mathcal{M}$ when there is no ambiguity from the context.

Definition 4 (Measurable structure). *A structure $\mathcal{M}$ is measurable if $[\alpha]_{\mathcal{M}} \in H$ for every $\alpha \in \mathsf{For}_{\mathsf{B}}$. The class of all measurable structures of the logic $\mathsf{P\Lambda}_{\rightarrow}$ will be denoted by $\mathsf{P\Lambda}^{\mathsf{Meas}}_{\rightarrow}$.*

Definition 5 (Satisfiability relation). *The satisfiability relation $\models \subseteq \mathsf{P\Lambda}^{\mathsf{Meas}}_{\rightarrow} \times \mathsf{For}_{\mathsf{P\Lambda}_{\rightarrow}}$ is defined in the following way:*

- *$\mathcal{M} \models M : \sigma$ iff $w \models M : \sigma$, for all $w \in W$;*
- *$\mathcal{M} \models P_{\geq s}\alpha$ iff $\mu([\alpha]) \geq s$;*
- *$\mathcal{M} \models \neg\mathfrak{A}$ iff it is not the case that $\mathcal{M} \models \mathfrak{A}$;*
- *$\mathcal{M} \models \mathfrak{A}_1 \wedge \mathfrak{A}_2$ iff $\mathcal{M} \models \mathfrak{A}_1$ and $\mathcal{M} \models \mathfrak{A}_2$.*

Definition 6 (Formula satisfiability). *Let* $\mathfrak{A} \in \mathsf{For}_{\mathsf{P}\Lambda_\to}$ *be a formula and* $F \subseteq \mathsf{For}_{\mathsf{P}\Lambda_\to}$

- $\mathfrak{A}$ *is satisfiable if there is an* $\mathsf{P}\Lambda_\to^{\mathsf{Meas}}$*-model* $\mathcal{M}$ *such that* $\mathcal{M} \models \mathfrak{A}$*;*
- $\mathfrak{A}$ *is valid if for every* $\mathsf{P}\Lambda_\to^{\mathsf{Meas}}$*-model* $\mathcal{M}$*,* $\mathcal{M} \models \mathfrak{A}$*;*
- *A set of formulas* F *is satisfiable if there is a* $\mathsf{P}\Lambda_\to^{\mathsf{Meas}}$*-model* $\mathcal{M}$ *such that* $\mathcal{M} \models \mathfrak{A}$ *for every* $\mathfrak{A} \in F$*.*

We now give a couple of simple examples in order to clarify the above notions.

Example 1. Consider the following model with three worlds, i.e., let $\mathcal{M} = \langle W, \rho, \xi, H, \mu \rangle$, where:

- $W = \{w_1, w_2, w_3\}$,
- $H = \mathcal{P}(W)$,
- $\mu(\{w_j\}) = \frac{1}{3}$, $j = 1, 2, 3$,

and ρ and ξ are defined such that

$w_1 \models (x : \sigma \to \tau) \wedge (y : \sigma)$,
$w_2 \models (x : \sigma_1 \to \tau) \wedge (y : \sigma_1)$,
$w_3 \models (x : \sigma_2 \to \tau) \wedge (y : \sigma_2)$ (Fig. 1).

It is obvious that $\mathcal{M} \models P_{=\frac{1}{3}}(x : \sigma \to \tau)$, $\mathcal{M} \models P_{=\frac{1}{3}}(y : \sigma)$ and $\mathcal{M} \models P_{=1}(xy : \tau)$.

Note that this example shows that in the case of an application of two terms, the probability of an application can not be smaller than the probability of the conjunction of its components, but it can be any number greater than or equal to it (and less or equal to 1).

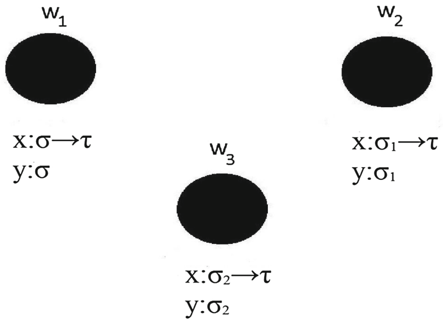


Fig. 1. An illustration of the Example 1.

The previous example showed that an application always has a probability bigger than or equal to the conjunction of its components. On the other hand, each conjunct can have a probability greater than the application as we will show in the following example.

Example 2. Let $\mathcal{M} = \langle W, \rho, \xi, H, \mu \rangle$, where:

- $W = \{w_1, w_2, w_3\}$,
- $H = \mathcal{P}(W)$,
- $\mu(\{w_j\}) = \frac{1}{3}$, $j = 1, 2, 3$,

and ρ and ξ are defined such that

$$\begin{aligned} w_1 &\models (x : \sigma \to \tau) \wedge (y : \sigma), \\ w_2 &\models (x : \sigma \to \tau) \wedge (y : \sigma_1), \\ w_3 &\models (x : \sigma_2 \to \tau) \wedge (y : \sigma). \end{aligned}$$

Now, it is clear that $\mathcal{M} \models P_{=\frac{1}{3}}(xy : \tau)$, while $\mathcal{M} \models P_{=\frac{2}{3}}(x : \sigma \to \tau)$ and $\mathcal{M} \models P_{=\frac{2}{3}}(y : \sigma)$.

The next example shows that we must provide an infinitary axiomatization in order to obtain strong completeness for our formal model.

Example 3. Consider the set

$$F = \{\neg P_{=0}\alpha\} \cup \{P_{<\frac{1}{n}}\alpha \mid n \quad is \quad a \quad positive \quad integer\}.$$

Every finite subset of F is clearly $\mathsf{P}\Lambda_{\to}^{\mathsf{Meas}}$-satisfiable, but the set F itself is not, since there is no real number greater than 0 and smaller than all positive rationals due to the Archimedean property of real numbers.[1] Therefore, the compactness theorem which states that "if every finite subset of F is satisfiable, then F is satisfiable" does not hold for $\mathsf{P}\Lambda_{\to}$.

4 The Axiomatization $Ax_{\mathsf{P}\Lambda_{\to}}$

We introduce an axiomatic system for the logic $\mathsf{P}\Lambda_{\to}$ which will be denoted by $Ax_{\mathsf{P}\Lambda_{\to}}$. Inference rules will be divided in two groups, such that inference rules from the first group can be applied only to lambda statements.

Axiom schemes

(1) all instances of the classical propositional tautologies, (atoms are any $\mathsf{P}\Lambda_{\to}$-formulas),
(2) $P_{\geq 0}\alpha$,
(3) $P_{\leq r}\alpha \Rightarrow P_{<s}\alpha$, $s > r$,
(4) $P_{<s}\alpha \Rightarrow P_{\leq s}\alpha$,
(5) $(P_{\geq r}\alpha \wedge P_{\geq s}\beta \wedge P_{\geq 1}(\neg\alpha \vee \neg\beta)) \Rightarrow P_{\geq min\{1, r+s\}}(\alpha \vee \beta)$,
(6) $(P_{\leq r}\alpha \wedge P_{<s}\beta) \Rightarrow P_{<r+s}(\alpha \vee \beta)$, $r + s \leq 1$,
(7) $P_{\geq 1}(\alpha \Rightarrow \beta) \Rightarrow (P_{\geq s}\alpha \Rightarrow P_{\geq s}\beta)$.

[1] For any real number $\epsilon > 0$ there exists an $n \in \mathbb{N}$ such that $\frac{1}{n} < \epsilon$.

Inference Rules I

$$(1)\qquad \frac{M:\sigma\to\tau \qquad N:\sigma}{MN:\tau}\;(\to_E)$$

$$(2)\qquad \frac{\begin{matrix}[x:\sigma]\\ \vdots \\ M:\tau\end{matrix}}{\lambda x.M:\sigma\to\tau}\;(\to_I)$$

$$(3)\qquad \frac{M:\sigma \qquad M=_\beta N}{N:\sigma}\;(\mathrm{eq})$$

Inference Rules II

(1) From $\mathfrak{A}_1$ and $\mathfrak{A}_1 \Rightarrow \mathfrak{A}_2$ infer $\mathfrak{A}_2$,
(2) from α infer $P_{\geq 1}\alpha$,
(3) from the set of premises

$$\{\phi \Rightarrow P_{\geq s-\frac{1}{k}}\alpha \mid k \geq \frac{1}{s}\}$$

infer $\phi \Rightarrow P_{\geq s}\alpha$.

Axiom 2 announces that every formula is satisfied in a set of worlds whose measure is at least 0, and we can easily infer (using $\neg\alpha$ instead of α) that the upper bound is 1, i.e., $P_{\leq 1}\alpha$. Axioms 3 and 4 provide the monotonicity of a measure, Axioms 5 and 6 correspond to the finite additivity of a measure, whereas Axiom 7 ensures that equivalent formulas have equal measures.

Inference Rules I are the rules that correspond to correct inference of typed lambda terms.

Inference Rules II:

- Rule II.1 is modus ponens (MP);
- Rule II.2 is the probability necessitation;
- Rule II.3 is the only infinitary rule of inference, and states that if probability is arbitrary close to s then it is at least s.

Definition 7 (Inference relation)
Let T be a set of formulas and $\mathfrak{A}$ a formula.

1. *$T \vdash \mathfrak{A}$ means that there exists a sequence $\mathfrak{A}_0, \ldots, \mathfrak{A}_{\lambda+1}$ (λ is finite or countable ordinal) of formulas, such that $\mathfrak{A}_{\lambda+1} = \mathfrak{A}$ and for all $i \leqslant \lambda+1$, $\mathfrak{A}_i$ is an axiom-instance, or $\mathfrak{A}_i \in T$, or $\mathfrak{A}_i$ can be derived by some inference rule applied on some previous members of the sequence.*
2. *Instead of $\emptyset \vdash \mathfrak{A}$ we write $\vdash \mathfrak{A}$. Any formula $\mathfrak{A}$ such that $\vdash \mathfrak{A}$ will be called a theorem.*
3. *T is* consistent *if*

- *there is at least a formula* $\alpha \in \mathsf{For_B}$ *and a formula* $\phi \in \mathsf{For_P}$ *that are not deducible from* T *and*
- *for every lambda statement* $M : \sigma$ *such that* $M : \sigma \in T$*, if* x_i*,* $i \in I$ *are all free variables of* M*, then all basic statements of the form* $x_i : \tau_i$ *(needed to adequately derive* $M : \sigma$ *according to and inference rules I) are also in* T*.*

Otherwise, T *is inconsistent.*

4. T *is a maximally consistent set if it is consistent and:*
   (1) for every $\alpha \in \mathsf{For_B}$*, if* $T \vdash \alpha$*, then* $\alpha \in T$ *and* $P_{\geq 1}\alpha \in T$*,*
   (2) for every $\phi \in \mathsf{For_P}$*, either* $\phi \in T$ *or* $\neg\phi \in T$*.*
5. T *is deductively closed if for every* $\mathfrak{A} \in \mathsf{For_{P\Lambda_\to}}$*, if* $T \vdash \mathfrak{A}$*, then* $\mathfrak{A} \in T$*.*

Note that it is not required that for every $\alpha \in \mathsf{For_B}$, either α or $\neg\alpha$ belongs to a maximal consistent set (as it is done for formulas from $\mathsf{For_P}$). It can be proved that, otherwise, in our canonical model, for each α we would have $P_{=1}\alpha$ or $P_{=0}\alpha$, so the probability operator would not make sense.

Theorem 3 (Deduction theorem). *Let* T *be a set of formulas and* $\phi, \psi \in \mathsf{For_P}$*. If* $T \cup \{\phi\} \vdash \psi$ *then* $T \vdash \phi \Rightarrow \psi$*.*

Proof. The proof is given in Appendix. □

Theorem 4 (Soundness). *The axiomatic system* $Ax_{\mathsf{P\Lambda_\to}}$ *is sound with respect to the class of* $\mathsf{P\Lambda}_{\to}^{\mathsf{Meas}}$*-models.*

Proof. The proof is given in Appendix. □

5 Completeness

In order to prove the completeness theorem we start with some auxiliary statements. After that, we show how to extend a consistent set of formulas T to a maximal consistent set of formulas $T^\star$. Finally, we construct the canonical model using the set $T^\star$ such that $\mathcal{M}_{T^\star} \models \mathfrak{A}$ iff $\mathfrak{A} \in T^\star$.

Lemma 1. *Let* T *be a consistent set of formulas.*

(1) For any formula $\phi \in \mathsf{For_P}$*, either* $T \cup \{\phi\}$ *is consistent or* $T \cup \{\neg\phi\}$ *is consistent.*
(2) If $\neg(\phi \Rightarrow P_{\geq s}\alpha) \in T$*, then there is some* $n > \frac{1}{s}$ *such that* $T \cup \{\phi \Rightarrow \neg P_{\geq s - \frac{1}{n}}\alpha\}$ *is consistent.*

Proof. The proof is given in Appendix. □

Lemma 2. *Let* T *be a maximal consistent set of formulas.*

(1) $\psi \in \mathsf{For_P}$*, if* $T \vdash \psi$*, then* $\psi \in T$*.*
(2) For any formula α*, if* $t = sup\{s \mid P_{\geq s}\alpha \in T\}$*, and* $t \in S$*, then* $P_{\geq t}\alpha \in T$*.*

Proof. The proof is given in Appendix. □

Theorem 5. *Every consistent set can be extended to a maximal consistent set.*

Proof. Consider a consistent set T. By $\mathsf{Cn_B}(T)$ we will denote the *consistent* set of all basic formulas that are consequences of T. Let $\phi_0, \phi_1, \dots$ be an enumeration of all formulas from $\mathsf{For_P}$. We define a sequence of sets T_i, $i = 0, 1, 2, \dots$ as follows:

(1) $T_0 = T \cup \mathsf{Cn_B}(T) \cup \{P_{\geq 1}\alpha \mid \alpha \in \mathsf{Cn_B}(T)\}$,
(2) for every $i \geq 0$,
  (a) if $T_i \cup \{\phi_i\}$ is consistent, then $T_{i+1} = T_i \cup \{\phi_i\}$, otherwise
  (b) if ϕ_i is of the form $\psi \Rightarrow P_{\geq s}\beta$, then
    $T_{i+1} = T_i \cup \{\neg\phi_i, \psi \Rightarrow \neg P_{\geq s-\frac{1}{n}}\beta\}$, for some positive integer n, so that T_{i+1} is consistent, otherwise,
  (c) $T_{i+1} = T_i \cup \{\neg\phi_i\}$,
(3) $T^\star = \bigcup_{i=0}^{\infty} T_i$.

The set T_0 is obviously consistent. Note that the existence of the natural number (n) from the step $2(b)$ of the construction is provided by Lemma 1(2), and each T_i is consistent.

It still remains to show that $T^\star$ is a maximal consistent set. The steps (1) and (2) of the above construction ensure that $T^\star$ is maximal.

$T^\star$ obviously does not contain all formulas. If $\alpha \in \mathsf{For_B}$, by the construction of T_0, α and $\neg\alpha$ can not be both in T_0. For a formula $\phi \in \mathsf{For_P}$, the set $T^\star$ does not contain both $\phi = \phi_i$ and $\neg\phi = \phi_j$, because the set $T_{\max\{i,j\}+1}$ is consistent.

Let us prove that $T^\star$ is deductively closed. If a formula $\alpha \in \mathsf{For_B}$ and $T \vdash \alpha$, then by the construction of T_0, $\alpha \in T^\star$ and $P_{\geq 1}\alpha \in T^\star$. Let $\phi \in \mathsf{For_P}$. It can be easily proved (induction on the length of the inference) that if $T^\star \vdash \phi$, then $\phi \in T^\star$. Note the fact that if $\phi = \phi_j$ and $T_i \vdash \phi$ it has to be $\phi \in T^\star$ because $T_{\max\{i,j\}+1}$ is consistent. Suppose that the sequence $\phi_1, \phi_2, \dots, \phi$ is the proof of ϕ from $T^\star$. If the mentioned sequence is finite, there must be some set T_i such that $T_i \vdash \phi$, and $\phi \in T^\star$. Therefore, suppose that the sequence is countably infinite. We can show that, for every i, if ϕ_i is obtained by an application of an arbitrary inference rule, and all the premises belong to $T^\star$, then, also $\phi_i \in T^\star$. If the inference rule is a finitary one, then there must be a set T_j which contains all the premises and $T_j \vdash \phi_i$. So, we conclude that $\phi_i \in T^\star$. Now, consider the infinitary Rule II.3. Let $\phi_i = \psi \Rightarrow P_{\geq s}\alpha$ be obtained from the set of premises $\{\phi_i^k = \psi \Rightarrow P_{\geq s_k}\alpha \mid s_k \in S\}$. By the induction hypothesis, we have that $\phi_i^k \in T^\star$, for every k. If $\phi_i \notin T^\star$, by step (2)(b) of the construction, there are some l and j so that $\neg(\psi \Rightarrow P_{\geq s}\alpha)$, $\psi \Rightarrow \neg P_{\geq s-\frac{1}{l}}\alpha \in T_j$. Thus, we have that for some $j' \geq j$:

- $\psi \wedge \neg P_{\geq s}\alpha \in T_{j'}$,
- $\psi \in T_{j'}$,
- $\neg P_{\geq s-\frac{1}{l}}\alpha \in T_{j'}$,
- $P_{\geq s-\frac{1}{l}}\alpha \in T_{j'}$ Ind. Hyp.

Contradiction with the consistency of a set $T_{j'}$.

Thus, $T^\star$ is a deductively closed set which does not contain all formulas, so it is consistent. □

Definition 8. *If $T^\star$ is a maximally consistent set of formulas, then a tuple $\mathcal{M}_{T^\star} = \langle W, \rho, \xi, H, \mu \rangle$ is defined:*

- $W = \{w = \langle \mathcal{L}(w), \cdot_w, [\![\ ]\!]_w \rangle \mid w \models \mathsf{Cn}_\mathsf{B}(T)\}$ *contains all term models that satisfy the set* $\mathsf{Cn}_\mathsf{B}(T)$*,*
- $\rho_w(x) = [x]$*,*
- $\xi_w(a) = \{[M] \in \mathcal{L}(w) \mid w \models M : a\}$*,*
- $H = \{[\alpha] \mid \alpha \in \mathsf{For}_\mathsf{B}\}$*, where* $[\alpha] = \{w \in W \mid w \models \alpha\}$*,*
- $\mu([\alpha]) = \sup\{s \mid P_{\geq s}\alpha \in T^\star\}$*.*

Lemma 3

(1) H is an algebra of subsets of W,
(2) If $[\alpha] = [\beta]$, then $\mu([\alpha]) = \mu([\beta])$,
(3) $\mu([\alpha]) \geq 0$,
(4) $\mu(W) = 1$, $\mu(\emptyset) = 0$,
(5) $\mu([\alpha]) = 1 - \mu([\neg\alpha])$,
(6) $\mu([\alpha] \cup [\beta]) = \mu([\alpha]) + \mu([\beta])$, for $[\alpha] \cap [\beta] = \emptyset$.

Proof. The proof is given in Appendix. □

Consequence of this Lemma is that $\mathcal{M}_{T^\star}$ is well defined.

Lemma 4. *Let $T^\star$ be a maximal consistent set of formulas. Then, $\mathcal{M}_{T^\star} \in \mathsf{P}\Lambda_\rightarrow^{\mathsf{Meas}}$.*

Proof. Directly from the construction of $\mathcal{M}_{T^\star}$. □

We are now ready to prove the main result of this paper.

Theorem 6 (Strong completeness). *Every consistent set of formulas T is $\mathsf{P}\Lambda_\rightarrow^{\mathsf{Meas}}$-satisfiable.*

Proof. We construct $\mathsf{P}\Lambda_\rightarrow^{\mathsf{Meas}}$-model $\mathcal{M}_{T^\star}$ and show that for every $\mathfrak{A} \in \mathsf{For}_{\mathsf{P}\Lambda_\rightarrow}$, $\mathcal{M}_{T^\star} \models \mathfrak{A}$ iff $\mathfrak{A} \in T^\star$. We use the induction on the complexity of the formula.

(1) $\mathfrak{A}$ is a lambda statement, $\mathfrak{A} = M : \sigma$. If $\mathfrak{A} \in \mathsf{Cn}_\mathsf{B}(T)$, then by definition of $\mathcal{M}_{T^\star}$ we have $\mathcal{M}_{T^\star} \models \mathfrak{A}$. Conversely, suppose $\mathcal{M}_{T^\star} \models \mathfrak{A}$, that is, for all worlds $w \in \mathcal{M}_{T^\star}$, $w \models M : \sigma$, i.e., $[\![M]\!]_\rho^w \in [\![\sigma]\!]_\xi^w$. Let $\mathfrak{B}$ be the set of all basic statements that are in $\mathsf{Cn}_\mathsf{B}(T)$, i.e. $\mathfrak{B}$ is basis and $\mathfrak{B} \subseteq \mathsf{Cn}_\mathsf{B}(T)$. First, let us consider the case when there is an infinite number of variables that are not in $\mathfrak{B}$. We extend $\mathfrak{B}$ to a set, $\mathfrak{B}^+$, of statements in which each type is assigned to an infinite number of variables and no variable is subject of more than one statement and no variable in $\mathfrak{B}^+ \setminus \mathfrak{B}$ occurs in M (the construction of a set $\mathfrak{B}^+$ can be found in [13]). Since no variable from $\mathfrak{B}^+ \setminus \mathfrak{B}$ appears in the set $\mathsf{Cn}_\mathsf{B}(T)$, there is a world, w_0, in which only consequences of $\mathfrak{B}^+$ hold. Let us show that $\xi_{w_0}(a) = \{[N] \mid \mathfrak{B}^+ \vdash N : a\}$ holds, i.e., $\{[N] \mid w_0 \models N : a\} = \{[N] \mid \mathfrak{B}^+ \vdash N : a\}$.

($\subseteq$) Suppose that $[N_1] \notin \{[N] \mid \mathfrak{B}^+ \vdash N : a\}$, that is $\mathfrak{B}^+ \nvdash N_1 : a$. Using Theorem 2, we obtain $\mathfrak{B}^+ \not\models N_1 : a$. Hence, $w_0 \not\models N_1 : a$ and therefore $[N_1] \notin \{[N] \mid w_0 \models N : a\}$.
($\supseteq$) If $[N_1] \in \{[N] \mid \mathfrak{B}^+ \vdash N : a\}$, we have that $\mathfrak{B}^+ \vdash N_1 : a$. Now, by Theorem 1, we have $\mathfrak{B}^+ \models N_1 : a$. Since, $w_0 \models \mathfrak{B}^+$, we obtain $w_0 \models N_1 : a$ and $[N_1] \in \{[N] \mid w_0 \models N : a\}$.
Furthermore, $[N] \in [\![\sigma]\!]_\xi^{w_0} \Leftrightarrow \mathfrak{B}^+ \vdash N : \sigma$ (the proof can be found in [13]). Since $M : \sigma$ holds in every world, whence in w_0 as well, we obtain $\mathfrak{B}^+ \vdash M : \sigma$. Now, the fact that M does not contain any variable from $\mathfrak{B}^+ \setminus \mathfrak{B}$, gives us that $\mathfrak{B} \vdash M : \sigma$, and so $\mathsf{Cn_B}(T) \vdash M : \sigma$, which means that $M : \sigma \in T^\star$.
The case when there is a finite number of variables that are not in $\mathfrak{B}$ can be proved using the same idea as in [13].

(2) $\mathfrak{A}$ is a Boolean combination of lambda statements. If $\mathfrak{A} \in \mathsf{Cn_B}(T)$, then, again, by definition of $\mathcal{M}_{T^\star}$ we have $\mathcal{M}_{T^\star} \models \mathfrak{A}$. Conversely, let $\mathcal{M}_{T^\star} \models \mathfrak{A}$. The goal is to show that $\mathfrak{A} \in T^\star$, i.e., it is enough to show that $T \vdash \mathfrak{A}$. The Axiom 1 and Modus Ponens give us that $\mathfrak{A}$ can be proved from T because of the completeness of classical propositional calculus.

- Next, consider the case $\mathfrak{A} = P_{\geq s}\alpha$. If $P_{\geq s}\alpha \in T^\star$, then $\sup\{r \mid P_{\geq r}\alpha \in T^\star\} = \mu([\alpha]) \geq s$, and so $M_{T^\star} \models P_{\geq s}\alpha$. Conversely, suppose that $M_{T^\star} \models P_{\geq s}\alpha$, i.e. $\sup\{r \mid P_{\geq r}\alpha \in T^\star\} \geq s$. If $\mu([\alpha]) > s$, then by the properties of supremum and monotonicity of μ, we have $P_{\geq s}\alpha \in T^\star$. If $\mu([\alpha]) = s$, then, from Lemma 2, we have that $P_{\geq s}\alpha \in T^\star$.
- Further, let $\mathfrak{A} = \neg\psi \in \mathsf{For_P}$. Then $\mathcal{M}_{T^\star} \models \neg\psi$ iff it is not the case that $\mathcal{M}_{T^\star} \models \psi$ iff $\psi \notin T^\star$ iff $\neg\psi \in T^\star$.
- Finally, let $\mathfrak{A} = \phi \wedge \psi \in \mathsf{For_P}$. Then, $\mathcal{M}_{T^\star} \models \phi \wedge \psi$ iff $\mathcal{M}_{T^\star} \models \phi$ and $\mathcal{M}_{T^\star} \models \psi$ iff $\phi, \psi \in T^\star$ iff $\phi \wedge \psi \in T^\star$. □

6 Conclusion

In this paper, we introduced the logic $\mathsf{P}\Lambda_\rightarrow$ for reasoning about probabilities of simply typed lambda terms. The language of this logic is obtained by adding the operators for probabilities and Boolean connectives to simple type assignment. An axiomatization for this logic is proposed and proved sound and strongly complete. Since this logic is not compact, the axiomatization contains one infinitary rule of inference.

As a topic for a further research, we will work towards simplification of the semantics in order to achieve compactness using finite sets of probability values for those logics. Another goal is to provide finitary axiomatizations for those logics. Also, for a further research we want to consider a case when Axiom 1 is replaced by an Axiom that states that all *intuitionistic* propositional tautologies hold, thus to work in an intuitionistic setting. Furthermore, we want to develop a first order extension of the logic $\mathsf{P}\Lambda_\rightarrow$. Note that such a logic would extend classical first order logic, so the set of all valid formulas is not recursively enumerable [1] and no complete finitary axiomatization is possible in that undecidable framework.

Another line of research is to develop probabilistic reasoning in other type disciplines such as polymorphic, intersection and higher-order types.

Acknowledgements. This work was supported by the SNSF project 200021_165549 *Justifications and non-classical reasoning*, and by the Serbian Ministry of Education, Science and Technological Development through projects ON174026, III 044006 and ON174008.

Appendix Proofs

Proof of Theorem 3. Suppose that $T \cup \{\phi\} \vdash \psi$. We use transfinite induction on the length of a proof.

If the length of a proof is equal to 1, then ψ is either an axiom or $\psi \in T \cup \{\phi\}$.

(a) If ψ is an axiom:
  - $T \vdash \psi$ Ax
  - $T \vdash \psi \Rightarrow (\phi \Rightarrow \psi)$ Ax
  - $T \vdash \phi \Rightarrow \psi$ MP,

(b) If $\psi \in T$:
  - $T \vdash \psi$ Hyp
  - $T \vdash \psi \Rightarrow (\phi \Rightarrow \psi)$ Ax
  - $T \vdash \phi \Rightarrow \psi$ MP,

(c) If $\psi \in \{\phi\}$:
  - $T \vdash \phi \Rightarrow \phi$ Ax.

Now, suppose that the length of a proof is $k > 1$. Formula ψ can belong to the set $T \cup \{\phi\}$, but then the proof is the same as above. Therefore, suppose that the formula ψ is obtained by an application of some inference rule from the Inference Rules II.

First, if ψ is obtained by an application of Rule II.1 from $T, \phi \vdash \psi_1$ and $T, \phi \vdash \psi_1 \Rightarrow \psi$:

- $T \vdash \phi \Rightarrow \psi_1$ Ind. Hyp.
- $T \vdash \phi \Rightarrow (\psi_1 \Rightarrow \psi)$ Ind. Hyp.
- $T \vdash (\phi \Rightarrow (\psi_1 \Rightarrow \psi)) \Rightarrow ((\phi \Rightarrow \psi_1) \Rightarrow (\phi \Rightarrow \psi))$ Taut.
- $T \vdash (\phi \Rightarrow \psi_1) \Rightarrow (\phi \Rightarrow \psi)$ MP
- $T \vdash \phi \Rightarrow \psi$ MP

Next, let us consider the case $\psi = P_{\geq 1}\alpha$ is obtained from $T \cup \{\phi\}$ by an application of Rule II.2. In that case:

- $T, \phi \vdash \alpha$,
- $T, \phi \vdash P_{\geq 1}\alpha$ by IR II.2.

However, since $\alpha \in \mathsf{For_B}$ and $\phi \in \mathsf{For_P}$ (otherwise, $\phi \Rightarrow P_{\geq 1}\alpha$ would not make sense), ϕ cannot affect the proof of α from $T \cup \{\phi\}$, and we have:

(1) $T \vdash \alpha$ Hyp.
(2) $T \vdash P_{\geq 1}\alpha$ IR II.2
(3) $T \vdash P_{\geq 1}\alpha \Rightarrow (\phi \Rightarrow P_{\geq 1}\alpha)$ Taut.
(4) $T \vdash \phi \Rightarrow P_{\geq 1}\alpha$ MP.

Finally, let us consider the case $\psi = \psi_1 \Rightarrow P_{\geq s}\alpha$ is obtained from $T \cup \{\phi\}$ by an application of Rule II.3. Then:

(1) $T, \phi \vdash \psi_1 \Rightarrow P_{\geq s-\frac{1}{k}}\alpha$, for all $k \geq \frac{1}{s}$ Hyp.
(2) $T \vdash \phi \Rightarrow (\psi_1 \Rightarrow P_{\geq s-\frac{1}{k}}\alpha)$ Ind.Hyp.
(3) $T \vdash (\phi \wedge \psi_1) \Rightarrow P_{\geq s-\frac{1}{k}}\alpha$ Taut.
(4) $T \vdash (\phi \wedge \psi_1) \Rightarrow P_{\geq s}\alpha$ IR II.3
(5) $T \vdash \phi \Rightarrow \psi$ Taut. □

Proof of Theorem 4. Our goal is to show that every instance of an axiom scheme holds in every model and that the inference rules preserve the validity. The Axiom 1 holds in every model because of the completeness of classical propositional logic.

By the Definition of the finitely additive probability measure we have that $\mu([\alpha]) \geq 0$ for all $\alpha \in \mathsf{For}_{\mathsf{B}}$. Hence, $\mathcal{M} \models P_{\geq 0}\alpha$, for every model $\mathcal{M}$ and the Axiom 2 is valid.

Let us consider the Axiom 3. Suppose that $P_{\leq r}\alpha$ holds in model $\mathcal{M} = \langle W, \rho, \xi, H, \mu \rangle$ and $s > r$. It means that $\mu([\alpha]) \leq r$. Since $s > r$, we obtain $\mu([\alpha]) < s$, that is $\mathcal{M} \models P_{<s}\alpha$.

Similarly, for the Axiom 4, suppose that $\mathcal{M} \models P_{<s}\alpha$. Then, we have $\mu([\alpha]) < s$, that implies $\mu([\alpha]) \leq s$. Thus, $\mathcal{M} \models P_{\leq s}\alpha$.

Next, let us consider Axiom 5. Suppose that in a model $\mathcal{M} = \langle W, \rho, \xi, H, \mu \rangle$,

$$P_{\geq r}\alpha, P_{\geq s}\beta \text{ and } P_{\geq 1}\neg(\alpha \vee \beta)$$

hold. Then, $\mu([\alpha]) \geq r$, $\mu([\beta]) \geq s$ and $[\alpha]$ and $[\beta]$ are disjoint sets. Since μ is a finitely additive measure, we have that

$$\mu([\alpha] \cup [\beta]) = \mu([\alpha \vee \beta]) = \mu([\alpha]) + \mu([\beta]).$$

Thus, $\mathcal{M} \models P_{\geq min\{1, r+s\}}(\alpha \vee \beta)$, so Axiom 5 holds in the model $\mathcal{M}$.

Now, let us consider the Axiom 6. Suppose that $P_{\leq r}\alpha$, $P_{<s}\beta$ hold in a model $\mathcal{M} = \langle W, \rho, \xi, H, \mu \rangle$. Then, $\mu([\alpha]) \leq r$ and $\mu([\beta]) < s$. From

$$[\alpha] = ([\alpha] \cap (W \setminus [\beta])) \cup [\alpha \wedge \beta],$$

follows that

$$\mu([\alpha]) \geq \mu([\alpha] \cap (W \setminus [\beta])).$$

Since $[\alpha \vee \beta] = ([\alpha] \cap (W \setminus [\beta])) \cup [\beta]$, we have that

$$\mu([\alpha \vee \beta]) \leq \mu([\alpha]) + \mu([\beta]) < r + s.$$

Therefore, $\mathcal{M} \models P_{<r+s}(\alpha \vee \beta)$.

Finally, for the Axiom 7, suppose that $P_{\geq 1}(\alpha \Rightarrow \beta)$ holds in a model $\mathcal{M} = \langle W, \rho, \xi, H, \mu \rangle$. Then, the set of all worlds in which α holds, but β does not hold has the measure 0, i.e., $\mu([\alpha] \cap (W \setminus [\alpha \Rightarrow \beta])) = 0$. From

$$[\alpha] = ([\alpha] \cap (W \setminus [\alpha \Rightarrow \beta])) \cup ([\alpha] \cap [\alpha \Rightarrow \beta])$$

follows that $\mu([\alpha]) = \mu([\alpha] \cap [\alpha \Rightarrow \beta])$ and, since $[\alpha] \cap [\alpha \Rightarrow \beta] \subseteq [\beta]$, we have that $\mu([\alpha]) \leq \mu([\beta])$. Thus, $\mathcal{M} \models P_{\geq s}\alpha \Rightarrow P_{\geq s}\beta$ and Axiom 7 holds in $\mathcal{M}$.

The proof that Inference Rules I are sound can be found in [13].

Inference Rules II:

Rule II.1 is validity-preserving for the same reason as in classical logic.
Rule II.2: suppose that α holds in $\mathcal{M} = \langle W, \rho, \xi, H, \mu \rangle$, then $[\alpha] = W$, and therefore $\mu([\alpha]) = 1$, so $\mathcal{M} \models P_{\geq 1}\alpha$.
Rule II.3: Suppose that $\mathcal{M} \models \phi \Rightarrow P_{\geq s - \frac{1}{k}}\alpha$ whenever $k \geq \frac{1}{s}$. If $\mathcal{M} \not\models \phi$, then obviously $\mathcal{M} \models \phi \Rightarrow P_{\geq s}\alpha$. Otherwise $\mathcal{M} \models P_{\geq s - \frac{1}{k}}\alpha$ for every $k \geq \frac{1}{s}$, so $\mathcal{M} \models P_{\geq s}\alpha$ because of the Archimedean properties of the set of reals. □

Proof of Lemma 1

(1) If $T \cup \{\phi\} \vdash \bot$, and $T \cup \{\neg\phi\} \vdash \bot$, then by Deduction theorem we have $T \vdash \neg\phi$ and $T \vdash \phi$. Contradiction.
(2) Suppose that for all $n > \frac{1}{s}$:

$$T, \phi \Rightarrow \neg P_{\geq s - \frac{1}{n}}\alpha \vdash \bot.$$

Therefore, by Deduction theorem and propositional reasoning, we have

$$T \vdash \phi \Rightarrow P_{\geq s - \frac{1}{n}}\alpha,$$

and by an application of Rule II.3 we obtain $T \vdash \phi \Rightarrow P_{\geq s}\alpha$. Contradiction with the fact that $\neg(\phi \Rightarrow P_{\geq s}\alpha) \in T$. □

Proof of Lemma 2

(1) Consequence of Definition 7.4.
(2) Let $t = sup\{s \mid P_{\geq s}\alpha \in T\} \in \mathsf{S}$. By the monotonicity of a measure, for each $s \in S$, $s < t$, $T \vdash P_{\geq s}\alpha$. Using Inference rule 3, we obtain

$$T \vdash P_{\geq t}\alpha.$$

T is a maximal consistent set of formulas, so, from (1), we have that

$$P_{\geq t}\alpha \in T.$$

□

Proof of Lemma 3

(1) The prove that H is an algebra is straightforward using that $W = [\alpha \vee \neg\alpha]$, $[\alpha]^C = [\neg\alpha]$ and $[\alpha] \cup [\beta] = [\alpha \vee \beta]$.

(2) It suffices to prove that $[\alpha] \subset [\beta]$ implies $\mu([\alpha]) \leq \mu([\beta])$. According to the completeness of the propositional logic, we have that $[\alpha] \subset [\beta]$ means that $\alpha \Rightarrow \beta \in \mathsf{Cn}_{\mathsf{B}}(T)$, and then also $P_{\geq 1}(\alpha \Rightarrow \beta) \in T^{\star}$. By axiom 7, we obtain that for each $s \in S$, $P_{\geq s}\alpha \Rightarrow P_{\geq s}\beta \in T^{\star}$, so $\mu([\alpha]) \leq \mu([\beta])$.

(3) $P_{\geq 0}\alpha$ is an axiom, so $\mu([\alpha]) \geq 0$.

(4) For any $\alpha \in T$, we have that $\alpha \vee \neg\alpha \in \mathsf{Cn}_{\mathsf{B}}(T)$ and $P_{\geq 1}(\alpha \vee \neg\alpha) \in T^{\star}$, therefore, we obtain that $W = [\alpha \vee \neg\alpha]$ and $\mu(W) = 1$. Since $P_{\geq 1}(\alpha \vee \neg\alpha) = P_{\geq 1-0}(\alpha \vee \neg\alpha) = P_{\leq 0}\neg(\alpha \vee \neg\alpha) = P_{\leq 0}(\neg\alpha \wedge \alpha)$ $= \neg P_{>0}(\neg\alpha \wedge \alpha)$, using that $P_{\geq t}\alpha \Rightarrow P_{>s}\alpha$, for $t > s$, we obtain that

$$sup\{s \mid P_{\geq s}(\neg\alpha \wedge \alpha) \in T^{\star}\} = 0,$$

and $\mu(\emptyset) = 0$.

(5) Let $\mu([\alpha]) = \sup\{s \mid P_{\geq s}\alpha \in T^{\star}\} = r$. If $r = 1$, then from Lemma 2 we obtain $P_{\geq 1}\alpha \in T^{\star}$. Therefore, $\neg P_{>0}\neg\alpha \in T^{\star}$. Again, using the fact that $P_{\geq t}\alpha \Rightarrow P_{>s}\alpha$, for $t > s$, we obtain that $\mu([\neg\alpha]) = 0$. Now, suppose that $r < 1$. Then, for each rational number $r' \in (r, 1]$, $\neg P_{\geq r'}\alpha = P_{<r'}\alpha \in T^{\star}$. By Axiom 4, we obtain that $P_{\leq r'}\alpha, P_{\geq 1-r'}\neg\alpha \in T^{\star}$. If there is some rational number $r'' \in [0, r)$ such that $P_{\geq 1-r''}\neg\alpha \in T^{\star}$, then $\neg P_{>r''}\alpha \in T^{\star}$, contradiction. Thus,

$$\sup\{s \mid P_{\geq s}\neg\alpha \in T^{\star}\} = 1 - \sup\{s \mid P_{\geq s}\alpha \in T^{\star}\},$$

i.e., $\mu([\alpha]) = 1 - \mu([\neg\alpha])$.

(6) Let $[\alpha] \cap [\beta] = \emptyset$, and let $\mu([\alpha]) = r$ and $\mu([\beta]) = s$. From the fact that $[\beta] \subset [\neg\alpha]$, using steps (2) and (5), we obtain that $r + s \leq r + (1 - r) = 1$. Suppose that both $r > 0$ and $s > 0$. Using properties of the supremum, for every rational number $r' \in [0, r)$, and for every rational number $s' \in [0, s)$, we have that $P_{\geq r'}\alpha, P_{\geq s'}\beta \in T^{\star}$. By the Axiom 5, we know that $P_{\geq r'+s'}(\alpha \vee \beta) \in T^{\star}$. Therefore, $r + s \leq t_0 = \sup\{t \mid P_{\geq t}(\alpha \vee \beta \in T^{\star})\}$. In the case that $r + s = 1$, the statement holds obviously, so suppose that $r + s < 1$. If $r + s < t_0$, then for every rational number $t' \in (r + s, t_0)$ we have $P_{\geq t'}(\alpha \vee \beta) \in T^{\star}$. There exists rational numbers $r'' > r$ and $s'' > s$, such that:

$$\neg P_{\geq r''}\alpha, P_{<r''}\alpha \in T^{\star} \quad, \neg P_{\geq s''}\alpha, P_{<s''}\alpha \in T^{\star},$$

and

$$r'' + s'' = t' \leq 1.$$

By Axiom 4, we obtain $P_{\leq r''} \in T^{\star}$. Using Axiom 6, we get

$$P_{\leq r''+s''}(\alpha \vee \beta) \in T^{\star}, \quad \neg P_{\geq r''+s''}(\alpha \vee \beta) \in T^{\star},$$

and

$$\neg P_{\geq t'}(\alpha \vee \beta) \in T^{\star}.$$

Contradiction. Hence, $r + s = t_0$ and we obtain that $\mu([\alpha] \cup [\beta]) = \mu([\alpha]) + \mu([\beta])$. Finally, if we suppose that $r = 0$ or $s = 0$, we can reason as above, where $r' = 0$ or $s' = 0$. □

References

1. Abadi, M., Halpern, J.Y.: Decidability and expressiveness for first-order logics of probability. Inf. Comput. **112**(1), 1–36 (1994)
2. Audebaud, P., Paulin-Mohring, C.: Proofs of randomized algorithms in Coq. Sci. Comput. Program. **74**(8), 568–589 (2009)
3. Barendregt, H.P.: The Lambda Calculus: Its Syntax and Semantics. North Holland, New York (1984)
4. Barendregt, H.P., Dekkers, W., Statman, R.: Lambda Calculus with Types (Perspectives in logic). Cambridge University Press, Cambridge (2013)
5. Bizjak, A., Birkedal, L.: Step-indexed logical relations for probability. In: Pitts, A. (ed.) FoSSaCS 2015. LNCS, vol. 9034, pp. 279–294. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46678-0_18
6. Cooper, R., Dobnik, S., Lappin, S., Larsson, S.: A probabilistic rich type theory for semantic interpretation. In: Proceedings of the EACL 2014 Workshop on Type Theory and Natural Language Semantics (TTNLS), pp. 72–79 (2014)
7. Doder, D., Grant, J., Ognjanović, Z.: Probabilistic logics for objects located in space and time. J. Logic Comput. **23**(3), 487–515 (2013)
8. Ehrhard, T., Pagani, M., Tasson, C.: The computational meaning of probabilistic coherence spaces. In: Proceedings of the 26th Annual IEEE Symposium on Logic in Computer Science, LICS 2011, pp. 87–96 (2011)
9. Fagin, R., Halpern, J.Y., Megiddo, N.: A logic for reasoning about probabilities. Inf. Comput. **87**(1/2), 78–128 (1990)
10. Fattorosi-Barnaba, M., Amati, G.: Modal operators with probabilistic interpretations. I. Studia Logica **46**(4), 383–393 (1987)
11. Ghilezan, S., Likavec, S.: Computational interpretations of logics. Zbornik radova, Special Issue Logic and Computer Science, Matematički institut **12**(20), 159–215 (2009)
12. Goodman, N.D.: The principles and practice of probabilistic programming. In: The 40th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2013, pp. 399–402 (2013)
13. Hindley, J.R.: The completeness theorem for typing lambda-terms. Theor. Comput. Sci. **22**, 1–17 (1983)
14. Hindley, J.R.: Basic Simple Type Theory. Cambridge Tracts in Theoretical Computer Science 42. Cambridge University Press, Cambridge (1997)
15. Hindley, J.R., Longo, G.: Lambda-calculus models and extesionality. Math. Logic Q. **26**, 289–310 (1980)
16. Ikodinović, N., Ognjanović, Z., Rašković, M., Marković, Z.: First-order probabilistic logics and their applications. Zbornik radova, Subseries Logic in Computer Science, Matematički institut **18**(26), 37–78 (2015)
17. Kokkinis, I., Maksimović, P., Ognjanović, Z., Studer, T.: First steps towards probabilistic justification logic. Logic J. IGPL **23**(4), 662–687 (2015)
18. Kokkinis, I., Ognjanović, Z., Studer, T.: Probabilistic justification logic. In: Artemov, S., Nerode, A. (eds.) LFCS 2016. LNCS, vol. 9537, pp. 174–186. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-27683-0_13
19. Lago, U.D., Zorzi, M.: Probabilistic operational semantics for the lambda calculus. RAIRO Theor. Inform. Appl. **46**(3), 413–450 (2012)
20. Marković, Z., Ognjanović, Z., Rašković, M.: A probabilistic extension of intuitionistic logic. Math. Logic Q. **49**(4), 415–424 (2003)

21. Milnikel, R.S.: The logic of uncertain justifications. Ann. Pure Appl. Logic **165**(1), 305–315 (2014)
22. Nilsson, N.J.: Probabilistic logic. Artif. Intell. **28**(1), 71–87 (1986)
23. Ognjanović, Z.: Discrete linear-time probabilistic logics: completeness, decidability and complexity. J. Logic Comput. **16**(2), 257–285 (2006)
24. Ognjanović, Z., Rašković, M., Marković, Z.: Probability logics. Zborik radova, Subseries Logic in Computer Science, Matematički institut **12**(20), 35–111 (2009)
25. Ognjanović, Z., Rašković, M., Marković, Z.: Probability Logics: Probability-Based Formalization of Uncertain Reasoning. Springer, Cham (2016)
26. Savić, N., Doder, D., Ognjanović, Z.: Logics with lower and upper probability operators. Int. J. Approx. Reason. **88**, 148–168 (2017)

Polyteam Semantics

Miika Hannula[1], Juha Kontinen[2], and Jonni Virtema[2,3](✉)

[1] University of Auckland, Auckland, New Zealand
m.hannula@auckland.ac.nz
[2] University of Helsinki, Helsinki, Finland
juha.kontinen@helsinki.fi
[3] Hasselt University, Hasselt, Belgium
jonni.virtema@uhasselt.be

Abstract. Team semantics is the mathematical framework of modern logics of dependence and independence in which formulae are interpreted by sets of assignments (teams) instead of single assignments as in first-order logic. In order to deepen the fruitful interplay between team semantics and database dependency theory, we define *Polyteam Semantics* in which formulae are evaluated over a family of teams. We begin by defining a novel polyteam variant of dependence atoms and give a finite axiomatisation for the associated implication problem. We also characterise the expressive power of poly-dependence logic by properties of polyteams that are downward closed and definable in existential second-order logic (ESO). The analogous result is shown to hold for poly-independence logic and all ESO-definable properties.

Keywords: Team semantics · Dependency theory · Expressive power

1 Introduction

Team semantics is the mathematical framework of modern logics of dependence and independence. The origin of team semantics goes back to [15] but its development to its current form began with the publication of the monograph [24]. In team semantics formulae are interpreted by sets of assignments (teams) instead of single assignments as in first-order logic. The reason for this change is that statements such as *the value of a variable x depends on the value of y* do not really make sense for single assignments. Team semantics has interesting connections with database theory and database dependencies [11–13,18]. In order to facilitate the exchange between team semantics and database theory, we introduce a generalisation of team semantics in which formulae are evaluated over a family of teams. We identify a natural notion of poly-dependence that generalises

This research was supported by a Marsden grant from Government funding, administered by the Royal Society of New Zealand, and grants 292767 and 308712 of the Academy of Finland.

S. Artemov and A. Nerode (Eds.): LFCS 2018, LNCS 10703, pp. 190–210, 2018.
https://doi.org/10.1007/978-3-319-72056-2_12

dependence atoms to polyteams and give a finite axiomatisation for its implication problem. We also define polyteam versions of independence, inclusion and exclusion atoms, and characterise the expressive power of poly-dependence and poly-independence logic.

A team X is a set of assignments with a common finite domain $x_1, \dots, x_n$ of variables. Such a team can be viewed as a database table with $x_1, \dots, x_n$ as its attributes. Dependence logic extends the language of first-order logic with atomic formulae $=(\overline{x}, y)$ called *dependence atoms* expressing that value of the variable y is functionally determined by the values of the variables in $\overline{x}$. On the other hand, *independence atoms* $\overline{y} \perp_{\overline{x}} \overline{z}$ [9] express that, for any fixed value of $\overline{x}$, knowing the value of $\overline{z}$ does not tell us anything new about the value of $\overline{y}$. By viewing a team as a database, the atoms $=(\overline{x}, y)$ and $\overline{y} \perp_{\overline{x}} \overline{z}$ correspond to the widely studied functional and embedded multivalued dependencies. Furthermore, inclusion atoms $\overline{x} \subseteq \overline{y}$ and exclusion atoms $\overline{x} | \overline{y}$ of [6] inherit their semantics from the corresponding database dependencies.

Independence, inclusion, and exclusion atoms have very interesting properties in the team semantics setting. For example, inclusion atoms give rise to a variant of dependence logic that corresponds to the complexity class PTIME over finite ordered structures [7] whereas all the other atoms above (and their combinations) give rise to logics that are equi-expressive with existential second-order logic and the complexity class NP. The complexity theoretic aspects of logics in team semantics have been studied extensively during the past few years (see [4] for a survey).

A multiset version of team semantics was recently defined in [3]. Multiteam semantics is motivated by the fact that multisets are widely assumed in database theory and occur in applications. Multiteam semantics can be also used to model and study database, probabilistic, and approximate dependencies in a unified framework (see [3,25]).

The aim of this work is similar to that of [3], i.e., we want to extend the applicability of team semantics. In database theory dependencies are often expressed by so-called embedded dependencies. An *embedded dependency* is a sentence of first-order logic with equality of the form

$$\forall x_1 \dots \forall x_n \big(\phi(x_1, \dots, x_n) \to \exists y_1 \dots \exists y_k \psi(x_1, \dots, x_n, y_1, \dots, y_k)\big),$$

where ϕ and ψ are conjunctions of relational atoms $R(x_1, \dots, x_n)$ and equalities $x = y$. In the literature embedded dependencies have been thoroughly classified stemming from real life applications. Examples of well-known subclasses include *full*, *uni-relational*, *1-head*, *tuple-generating*, and *equality-generating*. For example, an embedded dependency is called *tuple-generating* if it is equality free (for further details see, e.g., [16, Sect. 3]). The uni-relational dependencies can be studied also in the context of team semantics as generalised dependencies [21]. However in many applications, especially in the area of data exchange and data integration, it is essential to be able to express dependencies between different relations.

In the context of data exchange (see e.g. [5]) the relational database is divided into a set of source relations $\mathcal{S}$ and a set of target relations $\mathcal{T}$. Dependencies are used to describe what kind of properties should hold when data is transferred from the source schema to the target schema. In this setting a new taxonomy of embedded dependencies rises: An embedded dependency $\forall\overline{x}\big(\phi(\overline{x}) \rightarrow \exists\overline{y}\psi(\overline{x},\overline{y})\big)$ is *source-to-target* if the relation symbols occurring in ϕ and ψ are from $\mathcal{S}$ and $\mathcal{T}$, respectively. The embedded dependency is *target* if the relation symbols occurring in it are from $\mathcal{T}$. There is no direct way to study these classes of dependencies in the uni-relational setting of team semantics. In this paper we propose a general framework in which these inherently poly-relational dependencies can be studied.

In Sect. 2 we lay the foundations of polyteam semantics. The shift to polyteams is exemplified in Sect. 2.2, by the definition of poly-dependence atoms and an Armstrong type axiomatisation for the associated implication problem. In Sect. 3 polyteam semantics is extended from atoms to complex formulae. Section 4 studies the expressive power of the new logics over polyteams. The main technical results of the section characterises poly-independence (poly-dependence) logic as the maximal logic capable of defining all (downward closed) properties of polyteams definable in existential second-order logic.

2 From Uni-dependencies to Poly-dependencies

We start by defining the familiar dependency notions from the team semantics literature. In Sect. 2.2 we introduce a novel poly-relational version of dependence atoms and establish a finite axiomatisation of its implication problem. We then continue to present poly-relational versions of inclusion, exclusion, and independence atoms, and a general notion of a poly-relational dependency atom. We conclude this section by relating the embedded dependencies studied in database theory to our new setting.

2.1 Dependencies in Team Semantics

Vocabularies τ are sets of relation symbols with prescribed arities. For each $R \in \tau$, let $\mathrm{ar}(R) \in Z_+$ denote the arity of R. A τ-structure is a tuple $\mathfrak{A} = \big(A, (R_i^{\mathfrak{A}})_{R_i\in\tau}\big)$, where A is a set and each $R_i^{\mathfrak{A}}$ is an $\mathrm{ar}(R_i)$-ary relation on A (i.e., $R_i^{\mathfrak{A}} \subseteq A^{\mathrm{ar}(R_i)}$). We use $\mathfrak{A}$, $\mathfrak{B}$, etc. to denote τ-structures and A, B, etc. to denote the corresponding domains.

Let D be a finite set of first-order variables and A be a nonempty set. A function $s\colon D \rightarrow A$ is called an *assignment*. For a variable x and $a \in A$, the assignment $s(a/x)\colon D \cup \{x\} \rightarrow A$ is obtained from s as follows:

$$s(a/x)(y) := \begin{cases} a & \text{if } y = x, \\ s(y) & \text{otherwise.} \end{cases}$$

For an assignment s and a tuple of variables $\overline{x} = (x_1, \ldots, x_n)$, we write $s(\overline{x})$ to denote the sequence $\big(s(x_1), \ldots, s(x_n)\big)$. A *team* is a set of assignments with a common domain D and codomain A. Let $\mathfrak{A}$ be a τ-structure and X a team with codomain A, then we say that X is a team of $\mathfrak{A}$.

The following dependency atoms were introduced in [6,9,24].

Definition 1 (Dependency atoms). *Let $\mathfrak{A}$ be a model and X a team with codomain A. If $\overline{x}, \overline{y}$ are variable sequences, then $=(\overline{x}, \overline{y})$ is a dependence atom with the truth condition:*

$$\mathfrak{A} \models_X =(\overline{x}, \overline{y}) \text{ if for all } s, s' \in X \text{ s.t. } s(\overline{x}) = s'(\overline{x}), \text{ it holds that } s(\overline{y}) = s'(\overline{y}).$$

If $\overline{x}, \overline{y}$ are variable sequences of the same length, then $\overline{x} \subseteq \overline{y}$ is an inclusion atom and $\overline{x} \mid \overline{y}$ an exclusion atom with satisfaction defined as follows:

$$\mathfrak{A} \models_X \overline{x} \subseteq \overline{y} \text{ if for all } s \in X \text{ there exists } s' \in X \text{ such that } s(\overline{x}) = s'(\overline{y}).$$
$$\mathfrak{A} \models_X \overline{x} \mid \overline{y} \text{ if for all } s, s' \in X : s(\overline{x}) \neq s'(\overline{y}).$$

If $\overline{x}, \overline{y}, \overline{z}$ are variable sequences, then $\overline{y} \perp_{\overline{x}} \overline{z}$ is a conditional independence atom with satisfaction defined by

$$\mathfrak{A} \models_X \overline{y} \perp_{\overline{x}} \overline{z} \text{ if for all } s, s' \in X \text{ such that } s(\overline{x}) = s'(\overline{x}) \text{ there exists } s'' \in X \text{ such that } s''(\overline{x}) = s(\overline{x}),\ s''(\overline{y}) = s(\overline{y}), \text{ and } s''(\overline{z}) = s'(\overline{z}).$$

Note that in the previous definitions it is allowed that some or all of the vectors of variables have length 0. For example, $\mathfrak{A} \models_X =(\overline{x})$ holds iff $\forall s \in X$: $s(\overline{x}) = \overline{c}$ holds for some fixed tuple $\overline{c}$, and $\mathfrak{A} \models_X \overline{y} \perp_{\overline{x}} \overline{z}$ holds always if either of the vectors $\overline{y}$ or $\overline{z}$ is of length 0.

All the aforementioned dependency atoms have corresponding variants in relational databases. One effect of this relationship is that the axiomatic properties of these dependency atoms trace back to well-known results in database theory. Armstrong's axioms for functional dependencies constitute a finite axiomatisation for dependence atoms [1,9], and inclusion atoms can be finitely axiomatised using the axiomatisation for inclusion dependencies [2]. On the other hand, the non-axiomatisability and undecidability of the (finite and unrestricted) implication problem for embedded multivalued dependencies both carry over to conditional independence atoms [14,22,23]. Restricting attention to the so-called *pure independence atoms*, i.e., atoms of the form $\overline{x} \perp_{\emptyset} \overline{y}$, a finite axiomatisation is obtained by relating to marginal independence in statistics [8,18].

2.2 The Notion of Poly-dependence

For each $i \in \mathbb{N}$, let $\mathrm{Var}(i)$ denote a distinct countable set of first-order variable symbols. We say that these variables are of sort i. Relating to databases, sorts correspond to table names. Usually we set $\mathrm{Var}(i) = \{x^i_j \mid j \in \mathbb{N}\}$. We write x^i, y^i, x^i_j to denote variables form $\mathrm{Var}(i)$, and $\overline{x}^i$ to denote tuples of variables from $\mathrm{Var}(i)$. Sometimes we drop the index i and write simply x and $\overline{x}$ instead of x^i and $\overline{x}^i$, respectively. Note that $\overline{x}$ is always a tuple of variables of a single sort. In order to simplify notation, we sometimes write $\overline{x}^i$ and $\overline{x}^j$ to denote arbitrary tuples of variables of sort i and j, respectively. We emphasise that $\overline{x}^i$ and $\overline{x}^j$ might be of different length and may consist of distinct variables. Let $\mathfrak{A}$ be a τ-model and let $D_i \subseteq \mathrm{Var}(i)$ for all $i \in \mathbb{N}$. A tuple $\overline{X} = (X_i)_{i\in\mathbb{N}}$ is a *polyteam* of $\mathfrak{A}$ with domain $\overline{D} = (D_i)_{i\in\mathbb{N}}$, if X_i is a team with domain D_i and co-domain A for each $i \in \mathbb{N}$. We identify $\overline{X}$ with $(X_0, \ldots, X_n)$ if X_i is the singleton team consisting with the empty assignment for all i greater than n. Let $\overline{X} = (X_i)_{i\in\mathbb{N}}$ and $\overline{Y} = (Y_i)_{i\in\mathbb{N}}$ be two polyteams. We say that $\overline{X}$ is a *subteam* of $\overline{Y}$ if $X_i \subseteq Y_i$ for all $i \in \mathbb{N}$. By the *union* (resp. *intersection*) of $\overline{X}$ and $\overline{Y}$ we denote the polyteam $(X_i \cup Y_i)_{i\in\mathbb{N}}$ (resp. $(X_i \cap Y_i)_{i\in\mathbb{N}}$). By a slight abuse of notation we write $\overline{X} \cup \overline{Y}$ (resp. $\overline{X} \cap \overline{Y}$) for the union (resp. intersection) of $\overline{X}$ and $\overline{Y}$, and $\overline{X} \subseteq \overline{Y}$ to denote that $\overline{X}$ is a subteam of $\overline{Y}$. For a tuple $\overline{V} = (V_i)_{i\in\mathbb{N}}$ where $V_i \subseteq \mathrm{Var}(i)$, the *restriction* of $\overline{X}$ to $\overline{V}$, written $\overline{X} \upharpoonright \overline{V}$, is defined as $(X_i \upharpoonright V_i)_{i\in\mathbb{N}}$ where $X_i \upharpoonright V_i$ denotes the restriction of X_i to V_i.

Next we generalise dependence atoms to the polyteam setting. In contrast to the standard dependence atoms, poly-dependence atoms declare functional dependence of variables over two teams.

Poly-dependence. Let $\overline{x}^i\overline{y}^i$ and $\overline{u}^j\overline{v}^j$ be sequences of variables such that $\overline{x}^i$ and $\overline{u}^j$, and $\overline{y}^i$ and $\overline{u}^j$ have the same length, respectively. Then $=\!\left(\overline{x}^i, \overline{y}^i / \overline{u}^j, \overline{v}^j\right)$ is a *poly-dependence atom* whose satisfaction relation $\models_{\overline{X}}$ is defined as follows:

$$\mathfrak{A} \models_{\overline{X}} =\!\left(\overline{x}^i, \overline{y}^i / \overline{u}^j, \overline{v}^j\right) \Leftrightarrow \forall s \in X_i \forall s' \in X_j : s(\overline{x}^i) = s'(\overline{u}^j) \text{ implies } s(\overline{y}^i) = s'(\overline{v}^j).$$

Note that the atom $=\!(\overline{x}, \overline{y}/\overline{x}, \overline{y})$ corresponds to the dependence atom $=\!(\overline{x}, \overline{y})$. For empty tuples $\overline{x}^i$ and $\overline{u}^j$ the poly-dependence atom reduces to a "poly-constancy atom" $=\left(\overline{y}^i/\overline{v}^j\right)$. We will later show (Remark 13) that poly-dependence atoms of the form $=\!\left(\overline{x}^i, \overline{y}^i/\overline{u}^i, \overline{v}^i\right)$ can be expressed with formulae using only ordinary dependence atoms. Thus poly-dependence atoms of this form are considered as primitive notions only when $\overline{x}^i\overline{y}^i = \overline{u}^i\overline{v}^i$; otherwise $=\!\left(\overline{x}^i, \overline{y}^i/\overline{u}^i, \overline{v}^i\right)$ is considered as a shorthand for the equivalent formula obtained from Remark 13.

The ability to reason about database dependencies can be employed to facilitate many critical data management tasks such as schema design, query optimisation, and integrity maintenance. Keys, inclusion dependencies, and functional dependencies in particular have a crucial role in all of these processes. A traditional way to approach the interaction between dependencies has been the utilisation of proof systems similar to natural deduction systems in logic. The most significant of all these systems is the Armstrong's axiomatisation for functional dependencies. This inference system consists of only three rules which we depict below using the standard notation for functional dependencies, i.e., $X \to Y$ denotes that an attribute set X functionally determines another attribute set Y.

Definition 2 (Armstrong's axiomatisation [1])

- *Reflexivity: If $Y \subseteq X$, then $X \to Y$*
- *Augmentation: if $X \to Y$, then $XZ \to YZ$*
- *Transitivity: if $X \to Y$ and $Y \to Z$, then $X \to Z$*

Our first development is the generalisation of Armstrong's axiomatisation to the poly-dependence setting. To this end, we assemble the three rules of Armstrong and introduce three auxiliary rules: Union, Symmetry, and Weak Transitivity. Contrary to the Armstrong's proof system, here Union is not reducible to Transitivity and Augmentation because we operate with sequences instead of sets of variables or attributes. Symmetry in turn is imposed by the sequential notation employed by the poly-dependence atom. Weak Transitivity exhibits transitivity of equalities on the right-hand side of a poly-dependence atom, a phenomenon that arises only in the polyteam setting.

Definition 3 (Axiomatisation for poly-dependence atoms)

- *Reflexivity:* $=\big(\overline{x}^i, pr_k(\overline{x}^i)/\overline{y}^j, pr_k(\overline{y}^j)\big)$, *where* $k = 1, \dots, |\overline{x}^i|$ *and* pr_k *takes the* k*th projection of a sequence.*
- *Augmentation: if* $=\big(\overline{x}^i, \overline{y}^i/\overline{u}^j, \overline{v}^j\big)$, *then* $=\big(\overline{x}^i\overline{z}^i, \overline{y}^i\overline{z}^i/\overline{u}^j\overline{w}^j, \overline{v}^j\overline{w}^j\big)$
- *Transitivity: if* $=\big(\overline{x}^i, \overline{y}^i/\overline{u}^j, \overline{v}^j\big)$ *and* $=\big(\overline{y}^i, \overline{z}^i/\overline{v}^j, \overline{w}^j\big)$, *then* $=\big(\overline{x}^i, \overline{z}^i/\overline{u}^j, \overline{w}^j\big)$
- *Union: if* $=\big(\overline{x}^i, \overline{y}^i/\overline{u}^j, \overline{v}^j\big)$ *and* $=\big(\overline{x}^i, \overline{z}^i/\overline{u}^j, \overline{w}^j\big)$ *then* $=\big(\overline{x}^i, \overline{y}^i\overline{z}^i/\overline{u}^j, \overline{v}^j\overline{w}^j\big)$
- *Symmetry: if* $=\big(\overline{x}^i, \overline{y}^i/\overline{u}^j, \overline{v}^j\big)$, *then* $=\big(\overline{u}^j, \overline{v}^j/\overline{x}^i, \overline{y}^i\big)$
- *Weak Transitivity: if* $=\big(\overline{x}^i, \overline{y}^i\overline{z}^i\overline{z}^i/\overline{u}^j, \overline{v}^j\overline{v}^j\overline{w}^j\big)$, *then* $=\big(\overline{x}^i, \overline{y}^i/\overline{u}^j, \overline{w}^j\big)$

This proof system forms a complete characterisation of logical implication for poly-dependence atoms. We use $\models$ to refer to logical implication, i.e., we write $\Sigma \models \sigma$ if $\mathcal{A} \models_{\overline{X}} \Sigma$ implies $\mathcal{A} \models_{\overline{X}} \sigma$ for all models $\mathcal{A}$ and polyteams $\overline{X}$. Given an *axiomatisation* R, that is, a set of axioms and inference rules, we write $\Sigma \vdash_{\mathcal{R}} \sigma$ if $\mathcal{R}$ yields a proof of σ from Σ. Given a class of dependency atoms $\mathcal{C}$, we then say that $\mathcal{R}$ is sound (complete, resp.) for $\mathcal{C}$ if for all finite sets of dependency atoms $\Sigma \cup \{\sigma\}$ from $\mathcal{C}$, $\Sigma \vdash_{\mathcal{R}} \sigma$ implies (is implied by, resp.) $\Sigma \models \sigma$.

Theorem 4. *The axiomatisation of Definition 3 is sound and complete for poly-dependence atoms.*

Proof. The proof of soundness is straightforward and omitted. We show that the axiomatisation is complete, i.e., that $\Sigma \models \sigma$ implies $\Sigma \vdash \sigma$ for a set $\Sigma \cup \{\sigma\}$ of poly-dependence atoms. Assume σ is $=\big(\overline{x}^i, \overline{y}^i/\overline{x}^j, \overline{y}^j\big)$. First we consider the case where $i = j$ in which case σ is a standard dependence atom. Let Σ^* be the subset of Σ consisting of all standard dependence atoms over $\mathrm{Var}(i)$. Since all teams satisfying Σ^* can be extended to a polyteam satisfying Σ by introducing new empty teams, we have that $\Sigma^* \models \sigma$ in the team semantics setting. Since dependence atoms $=(\overline{x}, \overline{y})$ in team semantics correspond to functional dependencies $\{x \in \overline{x}^i\} \to \{y \in \overline{y}^i\}$ in relational databases (see e.g. [9]), Armstrong's complete axiomatisation from Definition 2 yields a deduction of σ_0 from Σ_0^* where Σ_0^* and $\{\sigma_0\}$ are obtained from Σ^* and σ by replacing dependence atoms with their corresponding functional dependencies. Since dependence atoms are provably order-independent (i.e. one derives $=(\overline{x}_0, \overline{x}_1)$ from $=(\overline{y}_0, \overline{y}_1)$ by Reflexivity, Union, and Transitivity if $\overline{x}_i$ and $\overline{y}_i$ list the same variables), the deduction in Armstrong's system can be simulated with the rules in Definition 3. This proves the case $i = j$.

Let us then consider the case $i \neq j$. We will show that $\Sigma \not\vdash \sigma$ implies $\Sigma \not\models \sigma$. Assume $\Sigma \not\vdash \sigma$. Define first a binary relation $\sim$ on $\mathrm{Var}(i) \cup \mathrm{Var}(j)$ such that $a^i \sim a^j$ if $\Sigma \vdash = \big(\overline{x}^i, a^i/\overline{x}^j, a^j\big)$, $a^j \sim a^i$ if $\Sigma \vdash = \big(\overline{x}^j, a^j/\overline{x}^i, a^i\big)$, and $a^i \sim b^i$ ($a^j \sim b^j$, resp.) if $a^i = b^i$ or $\Sigma \vdash = \big(\overline{x}^i, a^i b^i/\overline{x}^j, a^j a^j\big)$ for some a^j ($a^j = b^j$ or $\Sigma \vdash = \big(\overline{x}^j, a^j b^j/\overline{x}^i, a^i a^i\big)$ for some a^i, resp.). We show that $\sim$ is an equivalence relation.

- Reflexivity: Holds by definition.
- Symmetry: First note that $a^i \sim a^j$ and $a^j \sim a^i$ are derivably equivalent by the symmetry rule. Assume that $a^i \sim b^i$ in which case $= \big(\overline{x}^i, a^i b^i/\overline{x}^j, a^j a^j\big)$ is derivable for some a^j. Then derive $= \big(a^i b^i, b^i/a^j a^j, a^j\big)$ and $= \big(a^i b^i, a^i/a^j a^j, a^j\big)$ by using the reflexivity rule, and then $= \big(\overline{x}^i, b^i/\overline{x}^j, a^j\big)$ and $= \big(\overline{x}^i, a^i/\overline{x}^j, a^j\big)$ by using the transitivity rule. Finally derive $= \big(\overline{x}^i, b^i a^i/\overline{x}^j, a^j a^j\big)$ by using the union rule.
- Transitivity: Assume first that $a^i \sim b^i \sim c^i$, where a^i, b^i, c^i and are pairwise distinct. Then $= \big(\overline{x}^i, a^i b^i/\overline{x}^j, a^j a^j\big)$ and $= \big(\overline{x}^i, b^i c^i/\overline{x}^j, b^j b^j\big)$ are derivable for some a^j and b^j. Then analogously to the previous case assemble $= \big(\overline{x}^i, a^i b^i b^i/\overline{x}^j, a^j a^j b^j\big)$ which admits $= \big(\overline{x}^i, a^i/\overline{x}^j, b^j\big)$ by weak transitivity, and detach $= \big(\overline{x}^i, c^i/\overline{x}^j, b^j\big)$ from $= \big(\overline{x}^i, b^i c^i/\overline{x}^j, b^j b^j\big)$. By the union rule we then obtain $= \big(\overline{x}^i, a^i c^i/\overline{x}^j, b^j b^j\big)$ and thus that $a^i \sim c^i$. Since all the other cases are analogous, we observe that $\sim$ is transitive.

Let s be a function that maps each $x \in \mathrm{Var}(i) \cup \mathrm{Var}(j)$ that appears in $\Sigma \cup \{\sigma\}$ to the equivalence class $x/\sim$. We define $\overline{X} = (X_i, X_j)$ where $X_k = \{s \restriction \mathrm{Var}(k)\}$ for $k = i, j$. First notice that $\overline{X} \not\models \sigma$ for, by union, it cannot be the case that $\mathrm{pr}_k(\overline{y}^i) \sim \mathrm{pr}_k(\overline{y}^j)$ for all $k = 1, \ldots, |\overline{y}^i|$. It suffices to show that $\overline{X}$ satisfies each $= (\overline{u}^m, \overline{v}^m/\overline{u}^n, \overline{v}^n)$ in Σ. If $m = n$ or $\{m, n\} \neq \{i, j\}$, the atom is trivially satisfied. Hence, and by symmetry, we may assume that the atom is of the form $= \big(\overline{u}^i, \overline{v}^i/\overline{u}^j, \overline{v}^j\big)$. Assume that $s(\overline{u}^i) = s(\overline{u}^j)$, that is, $\mathrm{pr}_k(\overline{u}^i) \sim \mathrm{pr}_k(\overline{u}^j)$ for all $k = 1, \ldots, |\overline{u}^i|$. We obtain by the union rule that $= \big(\overline{x}^i, \overline{u}^i/\overline{x}^j, \overline{u}^j\big)$ is derivable, and hence by the transitivity rule that $= \big(\overline{x}^i, \overline{v}^i/\overline{x}^j, \overline{v}^j\big)$ is also derivable. Therefore, by using the reflexivity and transitivity rules we conclude that $s(\overline{v}^i) = s(\overline{v}^j)$. □

2.3 A General Notion of a Poly-dependency

Next we consider suitable polyteam generalisations for the dependencies discussed in Sect. 2.1 and also define a general notion of poly-dependency. This generalisation is immediate for inclusion atoms which are inherently multi-relational; relational database management systems maintain referential integrity by enforcing inclusion dependencies specifically between two distinct tables. With poly-inclusion atoms these multi-relational features can now be captured.

Poly-inclusion. Let $\overline{x}^i$ and $\overline{y}^j$ be sequences of variables of the same length. Then $\overline{x}^i \subseteq \overline{y}^j$ is a *poly-inclusion atom* whose satisfaction relation $\models_{\overline{X}}$ is defined as follows:

$$\mathfrak{A} \models_{\overline{X}} \overline{x}^i \subseteq \overline{y}^j \Leftrightarrow \forall s \in X_i \exists s' \in X_j : s(\overline{x}^i) = s'(\overline{y}^j).$$

If $i = j$, then the atom is the standard inclusion atom.

Poly-exclusion. Let $\overline{x}^i$ and $\overline{y}^j$ be sequences of variables of the same length. Then $\overline{x}^i \mid \overline{y}^j$ is a *poly-exclusion atom* whose satisfaction relation $\models_{\overline{X}}$ is defined as follows:

$$\mathfrak{A} \models_{\overline{X}} \overline{x}^i \mid \overline{y}^j \Leftrightarrow \forall s \in X_i, s' \in X_j : s(\overline{x}^i) \neq s'(\overline{y}^j).$$

If $i = j$, then the atom is the standard exclusion atom.

Poly-independence. Let $\overline{x}^i$, $\overline{y}^i$, $\overline{a}^j$,$\overline{b}^j$, $\overline{u}^k$, $\overline{v}^k$, and $\overline{w}^k$ be tuples of variables such that $|\overline{x}^i| = |\overline{a}^j| = |\overline{u}^k|$, $|\overline{y}^i| = |\overline{v}^k|$, $|\overline{b}^j| = |\overline{w}^k|$. Then $\overline{y}^i/\overline{v}^k \perp_{\overline{x}^i,\overline{a}^j/\overline{u}^k} \overline{b}^j/\overline{w}^k$ is a *poly-independence atom* whose satisfaction relation $\models_{\overline{X}}$ is defined as follows:

$$\mathfrak{A} \models_{\overline{X}} \overline{y}^i/\overline{v}^k \perp_{\overline{x}^i,\overline{a}^j/\overline{u}^k} \overline{b}^j/\overline{w}^k \Leftrightarrow \forall s \in X_i, s' \in X_j : s(\overline{x}^i) = s'(\overline{a}^j) \text{ implies}$$
$$\exists s'' \in X_k : s''(\overline{u}^k\overline{v}^k) = s(\overline{x}^i\overline{y}^i) \text{ and } s''(\overline{w}^k) = s'(\overline{b}^j).$$

The atom $\overline{y}/\overline{y} \perp_{\overline{x},\overline{x}/\overline{x}} \overline{z}/\overline{z}$, where all variables are of the same sort, corresponds to the standard independence atom $\overline{y} \perp_{\overline{x}} \overline{z}$. Furthermore, a *pure poly-independence atom* is an atom of the form $\overline{y}^i/\overline{v}^k \perp_{\emptyset,\emptyset/\emptyset} \overline{b}^j/\overline{w}^k$, written using a shorthand $\overline{y}^i/\overline{v}^k \perp \overline{b}^j/\overline{w}^k$.

Poly-independence atoms are closely related to equi-join operators of relational databases as the next example exemplifies.

Example 5. A relational database schema

$$\mathrm{P(\textsc{rojects})} = \{\texttt{project,team}\}, \quad \mathrm{T(\textsc{eams})} = \{\texttt{team,employee}\},$$
$$\mathrm{E(\textsc{mployees})} = \{\texttt{employee,team,project}\},$$

stores information about distribution of employees for teams and projects in a workplace. The poly-independence atom

$$\mathrm{P}[\texttt{project}]/\mathrm{E}[\texttt{project}] \perp_{\mathrm{P}[\texttt{team}],\mathrm{T}[\texttt{team}]/\mathrm{E}[\texttt{team}]} \mathrm{T}[\texttt{employee}]/\mathrm{E}[\texttt{employee}] \quad (1)$$

expresses that the relation Employees includes as a subrelation the natural join of Projects and Teams. If furthermore E[project,team] ⊆ P[project,team] and E[team,employee] ⊆ T[team,employee] hold, then Employees is exactly this natural join.

In addition to the poly-atoms described above we define a notion of a generalised poly-atom, similarly to the notion of generalised atom of [21].

Generalised poly-atoms. Let $(j_1, \ldots, j_n)$ be a sequence of positive integers. A *generalised quantifier* of type $(j_1, \ldots, j_n)$ is a collection Q of relational structures

$(A, R_1, \ldots, R_n)$ (where each R_i is j_i-ary) that is closed under isomorphisms. Then, for any sequence $(\overline{x}_1, \ldots, \overline{x}_n)$ where $\overline{x}_i$ is a length j_i tuple of variables from some $\mathrm{Var}(l_i)$, $A_Q(\overline{x}_1, \ldots, \overline{x}_n)$ is a *generalised poly-atom* of type $(j_1, \ldots, j_n)$. For a model $\mathcal{A}$ and polyteam $\overline{X}$ where $\overline{x}_i \subseteq \mathsf{Dom}(X_{l_i})$, the satisfaction relation with respect to A_Q is defined as follows:

$$\mathcal{A} \models_{\overline{X}} A_Q(\overline{x}_1, \ldots, \overline{x}_n)$$
$$\Leftrightarrow \Big(\mathsf{Dom}(\mathcal{A}), R_1 := \mathrm{rel}(X_{l_1}, \overline{x}_1) \ldots, R_n := \mathrm{rel}(X_{l_n}, \overline{x}_n)\Big) \in Q.$$

By $\mathrm{rel}(X, \overline{x})$, for $\overline{x} = (x_1, \ldots, x_m)$, we denote the relation $\{(s(x_1), \ldots, s(x_m)) \mid s \in X\}$. A poly-atom $A_Q(\overline{x}_1, \ldots, \overline{x}_n)$ is a *uni-atom* if the variable sequences $\overline{x}_1, \ldots, \overline{x}_n$ are of a single sort. Uni-atoms correspond exactly to generalised atoms of [21]. We say that the atom $A_Q(\overline{x}_1, \ldots, \overline{x}_n)$ is definable in a logic $\mathcal{L}$ if the class Q is definable in $\mathcal{L}$. For instance, we notice that a poly-inclusion atom $(x^1, y^1) \subseteq (u^2, v^2)$ is a first-order definable generalised poly-atom of type $(2, 2)$.

2.4 Database Dependencies as Poly-atoms

Embedded dependencies in a multi-relational context can now be studied with the help of generalised poly-atoms and polyteam semantics. Conversely, strong results obtained in the study of database dependencies can be transferred and generalised for stronger results in the polyteam setting. In particular, each embedded dependency can be seen as a defining formula for a generalised poly-atom, and hence the classification of embedded dependencies naturally yield a corresponding classification of generalised poly-atoms. For example, the class

$$\mathcal{C} := \{A_Q(\overline{x}_1, \ldots, \overline{x}_n) \mid Q \text{ is definable by an } \mathsf{FO}(R_1, \ldots, R_n)\text{-sentence in the class of equality-generating dependencies}\}$$

is the class of *equality-generating* poly-atoms. The defining formula of the generalised atom of type (2, 2) that captures the poly-dependence atom of type $=\!\big(x^i, y^i/u^j, v^j\big)$ is

$$\forall x_1 \forall x_2 \forall y_1 \forall y_2 \big((R_1(x_1, x_2) \wedge R_2(y_1, y_2) \wedge x_1 = y_1) \rightarrow x_2 = y_2\big).$$

Thus poly-dependence atoms are included in the class of equality-generating poly-atoms.

In order to study data exchange in the polyteam setting, we first need to define the notions of *source-to-target* and *target* poly-atoms. This classification of poly-atoms requires some more care as it is not enough to consider the defining formulae of the corresponding atoms, but also the variables that the atom is instantiated with. We will return to this topic briefly after we have given semantics for logics that work on polyteams.

3 Polyteam Semantics for Complex Formulae

We next delineate a version of team semantics suitable for the polyteam context. We note here that it is not a priori clear what sort of modifications for connectives and quantifiers one should entertain when shifting from teams to the polyteam setting.

3.1 Syntax and Semantics

Definition 6. *Let τ be a set of relation symbols. The syntax of* poly first-order logic $\mathsf{PFO}(\tau)$ *is given by the following grammar rules:*

$$\phi ::= x = y \mid x \neq y \mid R(\overline{x}) \mid \neg R(\overline{x}) \mid (\phi \wedge \phi) \mid (\phi \vee \phi) \mid (\phi \vee^{j} \phi) \mid \exists x\phi \mid \forall x\phi,$$

where $R \in \tau$ is a k-ary relation symbol, $j \in \mathbb{N}$, $\overline{x} \subseteq \mathit{Var}(i)^k$ and $x, y \in \mathit{Var}(i)$ for some $i, k \in \mathbb{N}$.

We say that $\vee$ is a *global disjunction* whereas $\vee^i$ is a *local disjunction*. Note that in the definition the scope of negation is restricted to atomic formulae. Note also that the restriction of $\mathsf{PFO}(\tau)$ to formulae without the connective $\vee^j$ and using only variables of a single fixed sort is $\mathsf{FO}(\tau)$.

For the definition of the polyteam semantics of PFO, recall the definitions of teams and polyteams from Sects. 2.1 and 2.2, respectively. Let X be a team, A a finite set, and $F \colon X \to \mathcal{P}(A) \setminus \{\emptyset\}$ a function. We denote by $X[A/x]$ the modified team $\{s(a/x) \mid s \in X, a \in A\}$, and by $X[F/x]$ the team $\{s(a/x) \mid s \in X, a \in F(s)\}$. Again note that if restricted to the above fragment of $\mathsf{PFO}(\tau)$ the polyteam semantics below coincides with traditional team semantics, see e.g. [4] for a definition. Thus for $\mathsf{FO}(\tau)$ formulae we may write $\mathfrak{A} \models_{X_i} \phi$ instead of $\mathfrak{A} \models_{(X_i)} \phi$.

Definition 7 (Lax polyteam semantics). *Let $\mathfrak{A}$ be a τ-structure and $\overline{X}$ a polyteam of $\mathfrak{A}$. The satisfaction relation $\models_{\overline{X}}$ for poly first-order logic is defined as follows:*

$$\begin{array}{lll}
\mathfrak{A} \models_{\overline{X}} x = y & \Leftrightarrow & \textit{if } x, y \in \mathit{Var}(i) \textit{ then } \forall s \in X_i : s(x) = s(y) \\
\mathfrak{A} \models_{\overline{X}} x \neq y & \Leftrightarrow & \textit{if } x, y \in \mathit{Var}(i) \textit{ then } \forall s \in X_i : s(x) \neq s(y) \\
\mathfrak{A} \models_{\overline{X}} R(\overline{x}) & \Leftrightarrow & \textit{if } \overline{x} \in \mathit{Var}(i)^k \textit{ then } \forall s \in X_i : s(\overline{x}) \in R^{\mathfrak{A}} \\
\mathfrak{A} \models_{\overline{X}} \neg R(\overline{x}) & \Leftrightarrow & \textit{if } \overline{x} \in \mathit{Var}(i)^k \textit{ then } \forall s \in X_i : s(\overline{x}) \notin R^{\mathfrak{A}} \\
\mathfrak{A} \models_{\overline{X}} (\psi \wedge \theta) & \Leftrightarrow & \mathfrak{A} \models_{\overline{X}} \psi \textit{ and } \mathfrak{A} \models_{\overline{X}} \theta \\
\mathfrak{A} \models_{\overline{X}} (\psi \vee \theta) & \Leftrightarrow & \mathfrak{A} \models_{\overline{Y}} \psi \textit{ and } \mathfrak{A} \models_{\overline{Z}} \theta \textit{ for some } \overline{Y}, \overline{Z} \subseteq \overline{X} \textit{ s.t. } \overline{Y} \cup \overline{Z} = \overline{X} \\
\mathfrak{A} \models_{\overline{X}} (\psi \vee^{j} \theta) & \Leftrightarrow & \mathfrak{A} \models_{\overline{X}[Y_j/X_j]} \psi \textit{ and } \mathfrak{A} \models_{\overline{X}[Z_j/X_j]} \theta, \\
 & & \textit{for some } Y_j, Z_j \subseteq X_j \textit{ s.t. } Y_j \cup Z_j = X_j \\
\mathfrak{A} \models_{\overline{X}} \forall x\psi & \Leftrightarrow & \mathfrak{A} \models_{\overline{X}[X_i[A/x]/X_i]} \psi, \textit{ when } x \in \mathit{Var}(i) \\
\mathfrak{A} \models_{\overline{X}} \exists x\psi & \Leftrightarrow & \mathfrak{A} \models_{\overline{X}[X_i[F/x]/X_i} \psi \textit{ holds for some } F \colon X_i \to \mathcal{P}(A) \setminus \{\emptyset\}, \\
 & & \textit{when } x \in \mathit{Var}(i)
\end{array}$$

The truth of a *sentence* ϕ (i.e., a formula with no free variables) in a model $\mathfrak{A}$ is defined as: $\mathfrak{A} \models \phi$ if $\mathfrak{A} \models_{(\{\emptyset\})} \phi$, where $(\{\emptyset\})$ denotes the polyteam consisting only singleton teams of the empty assignment. We write $\mathrm{Fr}(\phi)$ for the set of free variables in ϕ, and $\mathrm{Fr}_i(\phi)$ for $\mathrm{Fr}(\phi) \cap \mathrm{Var}(i)$.

Polyteam semantics is a conservative extension of team semantics in the same fashion as teams semantics is a conservative extension of Tarski semantics [24].

Proposition 8. *Let $\phi \in \mathsf{FO}(\tau)$ whose variables are all of sort $i \in \mathbb{N}$. Let $\mathfrak{A}$ be a τ-structure and $\overline{X}$ a polyteam of $\mathfrak{A}$. Then $\mathfrak{A} \models_{\overline{X}} \phi \Leftrightarrow \mathfrak{A} \models_{X_i} \phi \Leftrightarrow \forall s \in X_i : \mathfrak{A} \models_s \phi$, where $\models_s$ denotes the ordinary satisfaction relation of first-order logic.*

Example 9. A relational database schema

$$\begin{aligned}
\textsc{Patient} =& \{\texttt{patient\_id,patient\_name}\},\\
\textsc{Case} =& \{\texttt{case\_id,patient\_id,diagnosis\_id,confirmation}\},\\
\textsc{Test} =& \{\texttt{diagnosis\_id,test\_id}\},\\
\textsc{Results} =& \{\texttt{patient\_id,test\_id,result}\}
\end{aligned}$$

stores information about patient cases and their related laboratory tests. In order to maintain consistency of the stored data, database management systems support the use of integrity constraints that are based on functional and inclusion dependencies. For instance, on relation schema $\textsc{Patient}$ the key `patient_id` (i.e. the dependence atom $=(\texttt{patient\_id,patient\_name})$) ensures that no patient id can refer to two different patient names. On $\textsc{Case}$ the foreign key `patient_id` referring to `patient_id` on $\textsc{Patient}$ (i.e. the inclusion atom $\textsc{Case}[\texttt{patient\_id}] \subseteq \textsc{Patient}[\texttt{patient\_id}]$) enforces that patient ids on $\textsc{Case}$ refer to real patients. The introduction of poly-dependence logics opens up possibilities for more expressive data constraints. The poly-inclusion formula

$$\begin{aligned}
\phi_0 =& \texttt{confirmation} \neq \mathit{positive} \ \vee_{\textsc{Case}} \exists x_1 x_2 \big(x_1 \neq x_2 \wedge \\
& \bigwedge_{i=1,2} (\textsc{Case}[\texttt{diagnosis\_id}, x_i] \subseteq \textsc{Test}[\texttt{diagnosis\_id,test\_id}] \wedge \\
& \textsc{Case}[\texttt{patient\_id}, x_i, \mathit{positive}] \subseteq \textsc{Results}[\texttt{patient\_id,test\_id,result}])\big)
\end{aligned}$$

ensures that a diagnosis may be confirmed only if it has been affirmed by two different appropriate tests. The poly-exclusion formula

$$\begin{aligned}
\phi_1 =& \texttt{confirmation} \neq \mathit{negative} \vee_{\textsc{Case}} \\
& \forall x \big(\textsc{Case}[\texttt{diagnosis\_id}, x] \mid \textsc{Test}[\texttt{diagnosis\_id,test\_id}] \vee_{\textsc{Case}} \\
& \textsc{Case}[\texttt{patient\_id}, x, \mathit{positive}] \mid \textsc{Results}[\texttt{patient\_id,test\_id,result}]\big)
\end{aligned}$$

makes sure that a diagnosis may obtain a negative confirmation only if it has no positive indication by any suitable test. Note that both formulae employ local disjunction and quantified variables that refer to $\textsc{Case}$. Interestingly, the illustrated expressive gain is still computationally feasible as both ϕ_0 and ϕ_1 can be enforced in polynomial time. For ϕ_0 note that the data complexity of inclusion logic is in PTIME [7]; for ϕ_1 observe that satisfaction of a formula of the form $\overline{x}^1 \mid \overline{y}^2 \vee_1 \overline{x}^1 \mid \overline{z}^3$ can be decided in PTIME as well.

Poly-dependence logics. *Poly-dependence*, *poly-independence*, *poly-inclusion*, and *poly-exclusion logics* ($\mathsf{PFO}(\mathrm{pdep})$, $\mathsf{PFO}(\mathrm{pind})$, $\mathsf{PFO}(\mathrm{pinc})$, and $\mathsf{PFO}(\mathrm{pexc})$, resp.) are obtained by extending PFO with poly-dependence, poly-independence, poly-inclusion, and poly-exclusion atoms, respectively. In general, given a set of atoms $\mathcal{C}$ we denote by $\mathsf{PFO}(\mathcal{C})$ the logic obtained by extending PFO with the atoms of $\mathcal{C}$. We also consider poly-atoms in the team semantics setting; by $\mathsf{FO}(\mathcal{C})$ we denote the extension of first-order logic by the poly-atoms in $\mathcal{C}$. Similarly, it is also possible to consider atoms of Sect. 2.1 in the polyteam setting by requiring that the variables used with each atom are of a single sort.

3.2 Basic Properties

We say that a formula ϕ is *local* in polyteam semantics if for all $\overline{V} = (V_i)_{i\in\mathbb{N}}$ where $\mathrm{Fr}_i(\phi) \subseteq V_i$ for $i \in \mathbb{N}$, and all models $\mathfrak{A}$ and polyteams $\overline{X}$, we have

$$\mathfrak{A} \models_{\overline{X}} \phi \Leftrightarrow \mathfrak{A} \models_{\overline{X}\upharpoonright\overline{V}} \phi.$$

In other words, the truth value of a local formula depends only on its free variables. Furthermore, a logic $\mathcal{L}$ is called local if all its formulae are local.

Proposition 10 (Locality). *For any set $\mathcal{C}$ of generalised poly-atoms* $\mathsf{PFO}(\mathcal{C})$ *is local.*

Furthermore, the downward closure of dependence logic as well as the union closure of inclusion logic generalise to polyteams.

Proposition 11 (Downward Closure and Union Closure). *Let ϕ be a formula of* $\mathsf{PFO}(\mathrm{pdep})$*, ψ a formula of* $\mathsf{PFO}(pinc)$*, $\mathfrak{A}$ a model, and $\overline{X}, \overline{Y}$ two polyteams. Then $\mathfrak{A} \models_{\overline{X}} \phi$ and $\overline{Y} \subseteq \overline{X}$ implies that $\mathfrak{A} \models_{\overline{Y}} \phi$, and $\mathfrak{A} \models_{\overline{X}} \psi$ and $\mathfrak{A} \models_{\overline{Y}} \psi$ implies that $\mathfrak{A} \models_{\overline{X}\cup\overline{Y}} \psi$.*

The following proposition shows that the substitution of independence (dependence) atoms for any (downwards closed) class of atoms definable in existential second-order logic (ESO) results in no expressive gain.

Proposition 12. *Let $\mathcal{C}$ ($\mathcal{D}$, resp.) be the class of all (all downward closed, resp.)* ESO*-definable poly-atoms. The following equivalences of logics hold:* $\mathsf{FO}(\mathcal{C}) \equiv \mathsf{FO}(\mathrm{ind})$, $\mathsf{FO}(\mathcal{D}) \equiv \mathsf{FO}(\mathrm{dep})$, *and* $\mathsf{FO}(pinc) \equiv \mathsf{FO}(\mathrm{inc})$.

Proof. The claim $\mathsf{FO}(\mathrm{pinc}) \equiv \mathsf{FO}(\mathrm{inc})$ follows directly from the observation that in the team semantics setting poly-inclusion atoms are exactly inclusion atoms. Note that $\mathsf{FO}(\mathrm{ind})$ ($\mathsf{FO}(\mathrm{dep})$, resp.) captures all (all downward closed, resp.) ESO-definable properties of teams (see Theorem 18). It is easy to show (cf. [17, Theorem 6]) that every property of teams definable in $\mathsf{FO}(\mathcal{C})$ ($\mathsf{FO}(\mathcal{D})$, resp.) is ESO-definable (ESO-definable and downward closed, resp.). Thus since $\mathrm{ind} \in \mathcal{C}$ and $\mathrm{dep} \in \mathcal{D}$, we obtain that $\mathsf{FO}(\mathcal{C}) \equiv \mathsf{FO}(\mathrm{ind})$ and $\mathsf{FO}(\mathcal{D}) \equiv \mathsf{FO}(\mathrm{dep})$. □

Remark 13. In particular it follows from the previous proposition that, in the polyteam setting, each occurrence of any (any downward closed, resp.) ESO-definable poly-atom that takes variables of a single sort as parameters may be equivalently expressed by a formula of PFO(ind) (PFO(dep), resp.) that only uses variables of the same single sort.

We end this section by considering the relationship of global and local disjunctions. In particular, we observe that by the introduction of local disjunction its global variant becomes redundant. To facilitate our construction we here allow the use of $\vee^I$, where I is a set on indices, with obvious semantics. We then show that $\vee$ can be replaced by $\vee^I$ and $\vee^I$ by $\vee^i$.

Proposition 14. *For every formula of* PFO *there exists an equivalent formula of* PFO *that only uses disjunctions of type* $\vee^i$*.*

Proof. Let ϕ be a formula of PFO and let I list the sorts of all the variables that occur in ϕ. Let ϕ^* denote the formula obtained from ϕ by substituting all occurrences of $\vee$ by $\vee^I$. It is a direct consequence of the locality property that ϕ and ϕ^* are equivalent.

We will next show how to eliminate disjunctions of type $\vee^I$ from ϕ^*. Let $\phi_0 \vee^I \phi_1$ be a formula of PFO and let $I = \{i_1, \dots, i_n\}$. Define

$$\psi := \exists z_0^{i_1} \exists z_1^{i_1} \dots \exists z_0^{i_n} \exists z_1^{i_n} (\theta_0 \wedge \theta_1),$$

where $z_0^{i_1}$, $z_1^{i_1}$, …, $z_0^{i_n}$, $z_1^{i_n}$ are fresh and distinct variables, and

$$\begin{aligned}\theta_0 := (z_0^{i_1} = z_1^{i_1} \vee^{i_1} (z_0^{i_1} \neq z_1^{i_1} \wedge (z_0^{i_2} = z_1^{i_2} \vee^{i_2} (z_0^{i_2} \neq z_1^{i_2}\\ \wedge (\dots \wedge (z_0^{i_n} = z_1^{i_n} \vee^{i_n} (z_0^{i_n} \neq z_1^{i_n} \wedge \phi_0) \dots),\\ \theta_1 := (z_0^{i_1} \neq z_1^{i_1} \vee^{i_1} (z_0^{i_1} = z_1^{i_1} \wedge (z_0^{i_2} \neq z_1^{i_2} \vee^{i_2} (z_0^{i_2} = z_1^{i_2}\\ \wedge (\dots \wedge (z_0^{i_n} \neq z_1^{i_n} \vee^{i_n} (z_0^{i_n} = z_1^{i_n} \wedge \phi_1) \dots).\end{aligned}$$

The idea above is that the variables $z_0^{i_j}$, $z_1^{i_j}$ are used to encode a split of the team X_j. Using locality it is easy to see that $(\phi_0 \vee^I \phi_1)$ and ψ are equivalent over structures of cardinality at least two. From this the claim follows in a straightforward manner. □

3.3 Data Exchange in the Polyteam Setting

As promised, we now return to the topic of modelling data exchange in our new setting. In this section we restrict our attention to poly-atoms that are embedded dependencies. Our first goal is to define the notions of *source-to-target* and *target* poly-atoms. For this purpose we define a normal form for embedded dependencies. We call an embedded dependency $\forall \overline{x}\big(\phi(\overline{x}) \to \exists \overline{y} \psi(\overline{x}, \overline{y})\big)$ *separated* if the relation symbols that occur in ϕ and ψ are distinct. A poly-atom is called *separated*, if the defining formula is a separated embedded dependency.

In the polyteam setting this is just a technical restriction as non-separated poly-atoms can be always simulated by separated ones. Below we use the syntax $A(\overline{x}_1,\dots,\overline{x}_l,\overline{y}_1,\dots,\overline{y}_k)$ for separated poly-atoms. The idea is that $\overline{x}_i$s project extensions for relations used in the antecedent and $\overline{y}_j$s in the consequent of the defining formula.

Let $\mathcal{S}$ and $\mathcal{T}$ be a set of source relations and target relations from some data exchange instance, respectively. Let $\overline{X} = (S_1,\dots S_n, T_1,\dots,T_m)$ be a polyteam that encodes $\mathcal{S}$ and $\mathcal{T}$ in the obvious manner. We say that an instance of a separated atom $A(\overline{x}_1,\dots,\overline{x}_l,\overline{y}_1,\dots,\overline{y}_k)$ is *source-to-target* if each $\overline{x}_i$ is a tuple of variables of the sort of S_j, for some j, and each $\overline{y}_i$ is a tuple of variables of the sort of T_j, for some j. Analogously the instance $A(\overline{x}_1,\dots,\overline{x}_l,\overline{y}_1,\dots,\overline{y}_k)$ is *target* if each $\overline{x}_i$ and $\overline{y}_j$ is a tuple of variables of the sort of T_p for some p.

Data exchange problems can now be directly studied in the polyteam setting. For example the *existence-of-solution* problem can be reduced to a model checking problem by using first-order quantifiers to *guess* a solution for the problem while the rest of the formula describes the dependences required to be fulfilled in the data exchange problem.

Example 15. A relational database schemas

$$\begin{aligned}
\mathcal{S}: &\quad \mathrm{P(\textsc{rojects})} = \{\texttt{name, employee, employee\_position}\},\\
\mathcal{T}: &\quad \mathrm{E(\textsc{mployees})} = \{\texttt{name, project\_1, project\_2}\}
\end{aligned}$$

are used to store information about employees positions in different projects. We wish to check whether for a given instance of the schema $\mathcal{S}$ there exists an instance of the schema $\mathcal{T}$ that does not lose any information about for which projects employees are tasked to work and that uses the attribute `name` as a key. The $\mathsf{PFO}(\mathrm{pinc},\mathrm{dep})$-formula

$$\begin{aligned}
\phi := \exists x_1 \exists x_2 \exists x_3 \Big(\big(&\mathrm{P}[\texttt{employee, name}] \subseteq \mathrm{E}[x_1, x_2] \\
&\vee_{\mathrm{P}} \mathrm{P}[\texttt{employee, name}] \subseteq \mathrm{E}[x_1, x_3] \big) \wedge {=}(x_1, (x_2, x_3)) \Big),
\end{aligned}$$

when evaluated on a polyteam that encodes an instance of the schema $\mathcal{S}$, expresses that a solution for the data exchange problem exists. The variables x_1, x_2 and x_3 above are of the sort E and are used to encode attribute names `name`, `project_1` and `project_2`, respectively. The dependence atom above enforces that the attribute `name` is a key.

4 Expressiveness

The expressiveness properties of dependence, independence, inclusion, and exclusion logic and their fragments enjoy already comprehensive classifications. Dependence logic and exclusion logic are equi-expressive and capture all downward closed ESO properties of teams [6,19]. Independence logic, whose independence atoms violate downward closure, in turn captures all ESO team properties

[6]. On the other hand, the expressivity of inclusion logic has been characterised by the so-called greatest fixed point logic [7]. In this section we turn attention to polyteams and consider the expressivity of the poly-dependence logics introduced in this paper. Section 4.1 deals with logics with only uni-dependencies whereas in Sect. 4.2 poly-dependencies are considered.

4.1 Uni-dependencies in Polyteam Semantics

The following theorem displays how polyteam semantics over logics with only uni-atoms collapses to standard team semantics.

Theorem 16. *Let $\mathcal{C}$ be a set of uni-atoms. Each formula $\phi(\overline{x}^1, \ldots, \overline{x}^n) \in \mathsf{PFO}(\mathcal{C})$ can be associated with a sequence of formulae $\psi_1(\overline{x}^1), \ldots, \psi_n(\overline{x}^n) \in \mathsf{FO}(\mathcal{C})$ such that for all $\overline{X} = (X_1, \ldots, X_n)$, where X_i is a team with domain $\overline{x}^i$,*

$$\mathcal{M} \models_{\overline{X}} \phi(\overline{x}^1, \ldots, \overline{x}^n) \Leftrightarrow \forall i = 1, \ldots, n : \mathcal{M} \models_{X_i} \psi_i(\overline{x}^i).$$

Similarly, the statement holds vice versa.

Proof. The latter statement is clear as it suffices to set $\phi(\overline{x}^1, \ldots, \overline{x}^n) := \psi_1(\overline{x}^1) \wedge \ldots \wedge \psi_n(\overline{x}^n)$. For the other direction, we define recursively functions f_i that map formulae $\phi(\overline{x}^1, \ldots, \overline{x}^n) \in \mathsf{PFO}(\mathcal{C})$ to formulae $\psi_i(\overline{x}^i) \in \mathsf{FO}(\mathcal{C})$. By Proposition 14 we may assume that only disjunctions of type $\vee^i$, for some $i \in \mathbb{N}$, may occur in ϕ. The functions f_i are defined as follows:

- If $\phi(\overline{x}^j)$ is an atom, then $f_i(\phi) = \begin{cases} \phi & \text{if } i = j, \\ \top & \text{otherwise.} \end{cases}$
- $f_i(\psi \vee^j \theta) = \begin{cases} f_i(\psi) \vee f_i(\theta) & \text{if } i = j, \\ f_i(\psi) \wedge f_i(\theta) & \text{otherwise.} \end{cases}$
- $f_i(\psi \wedge \theta) = f_i(\psi) \wedge f_i(\theta)$.
- For $Q \in \{\exists, \forall\}$, if $f_i(Qx^j\psi) = \begin{cases} Qxf_i(\psi) & \text{if } i = j, \\ f_i(\psi) & \text{otherwise.} \end{cases}$

We set $\psi_i := f_i(\phi)$ and show the claim by induction on the structure of the formula. The cases for atoms and conjunctions are trivial. We show the case for $\vee^i$.

Let $\phi = \psi \vee^j \theta$ and assume that the claim holds for ψ and θ. Now

$$\begin{aligned} \mathfrak{A} \models_{\overline{X}} \phi \quad \text{iff} \quad & \mathfrak{A} \models_{\overline{X}[Y_j/X_j]} \psi \,\text{and}\, \mathfrak{A} \models_{\overline{X}[Z_j/X_j]} \theta, \\ & \text{for some}\, Y_j, Z_j \subseteq X_j \,\text{such that}\, Y_j \cup Z_j = X_j. \end{aligned}$$

By the induction hypothesis, $\mathfrak{A} \models_{\overline{X}[Y_j/X_j]} \psi$ and $\mathfrak{A} \models_{\overline{X}[Z_j/X_j]} \theta$ iff $\mathfrak{A} \models_{Y_j} f_j(\psi)$, $\mathfrak{A} \models_{Z_j} f_j(\theta)$, and $\mathfrak{A} \models_{X_i} f_i(\psi), \mathfrak{A} \models_{X_i} f_i(\theta)$ for each $i \neq j$. Thus we obtain that $\mathfrak{A} \models_{\overline{X}} \phi$ holds iff

$$\mathfrak{A} \models_{X_j} f_j(\psi) \vee f_j(\theta), \text{ and } \mathfrak{A} \models_{X_i} f_i(\psi) \text{ and } \mathfrak{A} \models_{X_i} f_i(\theta) \text{ for each } i \neq j.$$

The above can be rewritten as

$$\mathfrak{A} \models_{X_j} f_j(\psi) \vee f_j(\theta), \text{ and } \mathfrak{A} \models_{X_i} f_i(\psi) \wedge f_i(\theta) \text{ for each } i \neq j.$$

The claim now follows, since $f_j(\psi) \vee f_j(\theta) = f_j(\psi \vee^j \theta)$ and $f_i(\psi) \wedge f_i(\theta) = f_i(\psi \vee^j \theta)$, for $i \neq j$.

The cases for the quantifiers are similar.

This theorem implies that poly-atoms which describe relations between two teams are beyond the scope of uni-logics. The following proposition illustrates this for $\mathsf{PFO}(\text{dep})$.

Proposition 17. *The poly-constancy atom* $=\big(x^1/x^2\big)$ *cannot be expressed in* $\mathsf{PFO}(\text{dep})$.

Proof. Assume that $=\big(x^1/x^2\big)$ can be defined by some $\phi(x^1, x^2) \in \mathsf{PFO}(\text{dep})$. By Theorem 16 there are $\mathsf{FO}(\text{dep})$-formulae $\psi_1(x^1)$ and $\psi_2(x^2)$ such that for all $\overline{X} = (X_1, X_2)$, where X_i is a team with domain x^i, it holds that

$$\mathcal{M} \models_{\overline{X}} =\big(x^1/x^2\big) \Leftrightarrow \forall i = 1, 2 : \mathcal{M} \models_{X_i} \psi_i(x^i). \tag{2}$$

Define teams $X_1 := \{x^1 \mapsto 0\}$, $X_2 := \{x^2 \mapsto 0\}$, $Y_1 := \{x^1 \mapsto 1\}$, and $Y_2 := \{x^2 \mapsto 1\}$. Now clearly $\mathcal{M} \models_{(X_1,X_2)} =\big(x^1/x^2\big)$, and $\mathcal{M} \models_{(Y_1,Y_2)} =\big(x^1/x^2\big)$. Hence by (2), we obtain first that $\mathcal{M} \models_{X_1} \psi_i(x^1)$ and $\mathcal{M} \models_{Y_2} \psi_i(x^2)$, and then that $\mathcal{M} \models_{(X_1,Y_2)} =\big(x^1/x^2\big)$, which is a contradiction. □

Using Theorem 16 we may now compare and characterise the expressivity of $\mathsf{PFO}(\text{dep})$ and $\mathsf{PFO}(\text{ind})$ in terms of existential second-order logic. To this end, let us first recall the ESO characterisations of open dependence and independence logic formulae. Note that $\text{rel}(X)$ refers to a relation $\{s(x_1, \ldots, x_n) \mid s \in X\}$ where $x_1, \ldots, x_n$ is some enumeration of $\mathsf{Dom}(X)$.

Theorem 18 ([6,19]). *Let $\phi(\overline{x})$ be an independence logic (dependence logic, resp.) formula, and let R be an $|\overline{x}|$-ary relation. Then there is an (downward closed with respect to R, resp.)* ESO*-sentence $\psi(R)$ such that for all teams $X \neq \emptyset$ where* $\mathsf{Dom}(X) = \overline{x}$,

$$\mathcal{M} \models_X \phi(\overline{x}) \Leftrightarrow (\mathcal{M}, R := \text{rel}(X)) \models \psi(R)$$

The same statement holds also vice versa.

It is now easy to see that Theorems 16 and 18 together imply that $\mathsf{PFO}(\text{dep})$ captures all conjunctions of downward closed ESO properties of teams whereas $\mathsf{PFO}(\text{ind})$ captures all such properties.

Theorem 19. *Let $\phi(\overline{x}^1, \ldots, \overline{x}^n)$ be a* $\mathsf{PFO}(\text{ind})$ *(*$\mathsf{PFO}(\text{dep})$*, resp.) formula where $\overline{x}^i$ is a sequence of variables from $Var(i)$. Let R_i be an $|\overline{x}^i|$-ary relation symbol for $i = 1, \ldots, n$. Then there are (downward closed with respect to R_i, resp.)* ESO*-sentences $\psi_1(R_1), \ldots, \psi_n(R_n)$ such that for all polyteams $\overline{X} = (X_1, \ldots, X_n)$ where* $\mathsf{Dom}(X_i) = \overline{x}^i$ *and $X_i \neq \emptyset$*

$$\begin{aligned} &\mathcal{M} \models_{\overline{X}} \phi(\overline{x}^1, \ldots, \overline{x}^n) \\ &\qquad \Leftrightarrow (\mathcal{M}, R_1 := \text{rel}(X_1), \ldots, R_n := \text{rel}(X_n)) \models \ \psi_1(R_1) \wedge \ldots \wedge \psi_n(R_n). \end{aligned}$$

The same statement holds also vice versa.

4.2 Poly-dependencies in Polyteam Semantics

Next we consider poly-dependencies in polyteam semantics.

Lemma 20. *The following equivalences hold:*

$$\begin{aligned}
=\!\big(\overline{x}^1,\overline{y}^1/\overline{u}^2,\overline{v}^2\big) &\equiv \overline{y}^1/\overline{y}^1 \perp_{\overline{x}^1,\overline{u}^2/\overline{x}^1} \overline{v}^2/\overline{y}^1, && (3)\\
=\!\big(\overline{x}^1,y^1/\overline{u}^2,v^2\big) &\equiv \forall z^1(y^1 = z^1 \vee^1 \overline{x}^1 z^1 \mid \overline{u}^2 v^2), && (4)\\
\overline{x}^1 \subseteq \overline{u}^2 &\equiv \overline{x}^1/\overline{u}^2 \perp \emptyset/\emptyset, && (5)\\
\overline{x}^1 \subseteq \overline{u}^2 &\equiv \forall \overline{v}^2(\overline{x}^1 \mid \overline{v}^2 \vee^2 \overline{v}^2 \subseteq \overline{u}^2), && (6)\\
\overline{x}^1 \mid \overline{u}^2 &\equiv \exists y^1 z^1 v^2 w^2(=\!\big(\overline{x}^1, y^1 z^1/\overline{u}^2, v^2 w^2\big) && (7)\\
&\qquad \wedge y^1 = z^1 \wedge v^2 \neq w^2),\\
\overline{x}^1 \mid \overline{u}^2 &\equiv \exists \overline{y}^1(\overline{u}^2 \subseteq \overline{y}^1 \wedge \overline{x}^1 \mid \overline{y}^1), && (8)
\end{aligned}$$

$$\begin{aligned}
\overline{y}^2/\overline{y}^1 \perp_{\overline{x}^2,\overline{x}^3/\overline{x}^1} \overline{z}^3/\overline{z}^1 \equiv\ & \forall \overline{p}^2\overline{q}^2 \exists u^2 v^2 \forall \overline{p}^3\overline{q}^3\overline{r}^3 \exists u^3 v^3 \Big(&& (9)\\
& =\!\big(\overline{p}^2\overline{q}^2, u^2v^2/\overline{p}^3\overline{q}^3, u^3v^3\big)\\
& \wedge \big(u^2 = v^2 \vee^1 (u^2 \neq v^2 \wedge \overline{x}^2\overline{y}^2 \mid \overline{p}^2\overline{q}^2)\big)\\
& \wedge \big(u^3 \neq v^3 \vee^2 \overline{x}^3\overline{z}^3 \mid \overline{p}^3\overline{r}^3 \vee^2 \overline{p}^3\overline{q}^3\overline{r}^3 \subseteq \overline{x}^1\overline{y}^1\overline{z}^1\big)\Big).
\end{aligned}$$

Proof. The equivalences (3)–(8) are straightforward and (9) is analogous to the corresponding translation in the team semantics setting (see [6]). □

The following theorem compares the expressive powers of different polyteam-based logics. Observe that the expressivity of the logics with two poly-dependency atoms remains the same even if either one of the atoms has the standard team semantics interpretation.

Theorem 21. *The following equivalences of logic hold:*

(1) $\mathsf{PFO}(\mathrm{pdep}) \equiv \mathsf{PFO}(\mathrm{pexc})$,
(2) $\mathsf{PFO}(\mathrm{pind}) \equiv \mathsf{PFO}(\mathrm{pexc},\mathrm{inc}) \equiv \mathsf{PFO}(pinc,\mathrm{exc}) \equiv \mathsf{PFO}(\mathrm{pdep},\mathrm{inc})$
$\equiv \mathsf{PFO}(pinc,\mathrm{dep}) \equiv \mathsf{PFO}(\mathrm{pdep},\mathrm{ind}) \equiv \mathsf{PFO}(\mathrm{pexc},\mathrm{ind}) \equiv \mathsf{PFO}(pinc,\mathrm{ind})$.

Proof. Item (1) follows by Eqs. (4) and (7). Item (2) follows from the below list of relationships:

- $\mathsf{PFO}(\mathrm{pind}) \subseteq \mathsf{PFO}(\mathrm{pexc},\mathrm{inc})$ by (4), (6), and (9).
- $\mathsf{PFO}(\mathrm{pexc},\mathrm{inc}) \equiv \mathsf{PFO}(\mathrm{pinc},\mathrm{exc})$ by (6) and (8).
- $\mathsf{PFO}(\mathrm{pexc},\mathrm{inc}) \equiv \mathsf{PFO}(\mathrm{pdep},\mathrm{inc})$ by (4) and (7).
- $\mathsf{PFO}(\mathrm{pinc},\mathrm{exc}) \equiv \mathsf{PFO}(\mathrm{pinc},\mathrm{dep})$, since exclusion (dependence, resp.) atoms can be described in $\mathsf{FO}(\mathrm{dep})$ ($\mathsf{FO}(\mathrm{exc})$, resp.) [6].
- $\mathsf{PFO}(\mathrm{pdep},\mathrm{inc}) \subseteq \mathsf{PFO}(\mathrm{pdep},\mathrm{ind})$, $\mathsf{PFO}(\mathrm{pexc},\mathrm{inc}) \subseteq \mathsf{PFO}(\mathrm{pexc},\mathrm{ind})$, and $\mathsf{PFO}(\mathrm{pinc},\mathrm{dep}) \subseteq \mathsf{PFO}(\mathrm{pinc},\mathrm{ind})$ since inclusion atoms can be described in $\mathsf{FO}(\mathrm{ind})$ [6] and dependence atoms by independence atoms [9].

- $\mathsf{PFO}(\text{pdep}, \text{ind}) \subseteq \mathsf{PFO}(\text{ind})$, $\mathsf{PFO}(\text{pexc}, \text{ind}) \subseteq \mathsf{PFO}(\text{ind})$, and $\mathsf{PFO}(\text{pinc}, \text{ind}) \subseteq \mathsf{PFO}(\text{pind})$ by (3), (5), and (7). □

Next we show the analogue of Theorem 18 for polyteams.

Theorem 22. *Let $\phi(R_1, \ldots, R_n)$ be an ESO-sentence. There is a $\mathsf{PFO}(\text{pdep}, \text{inc})$ formula $\phi^*(\overline{x}^1, \ldots, \overline{x}^n)$, where $|\overline{x}^i| = \text{ar}(R_i)$, such that for all polyteams $\overline{X} = (X_1, \ldots, X_n)$ with $\mathsf{Dom}(X_i) = \overline{x}^i$ and $X_i \neq \emptyset$,*

$$\mathcal{M} \models_{\overline{X}} \phi^*(\overline{x}^1, \ldots, \overline{x}^n) \Leftrightarrow (\mathcal{M}, R_1 := \text{rel}(X_1), \ldots, R_n := \text{rel}(X_n)) \models \phi(R_1, \ldots, R_n).$$

The statement holds also vice versa.

Proof. The direction from $\mathsf{PFO}(\text{pdep}, \text{inc})$ to ESO is proven by a translation similar to the one from dependence logic to ESO in [24]. We show only the opposite direction. Analogously to [6], we can rewrite $\phi(R_1, \ldots, R_n)$ as

$$\exists \overline{f} \forall \overline{u} \big(\bigwedge_{i=1}^{n} (R_i(\overline{u}_i) \leftrightarrow f_{2i-1}(\overline{u}_i) = f_{2i}(\overline{u}_i)) \wedge \psi(\overline{u}, \overline{f}) \big)$$

where $\overline{f} = f_1, \ldots, f_{2n}, \ldots, f_m$ is a list of function variables, ψ is a quantifier-free formula in which no R_i appears, each $\overline{u}_i$ is a subsequence of $\overline{u}$, and each f_i occurs only as $f_i(\overline{u}_{j_i})$ for some fixed tuple $\overline{u}_{j_i}$ of variables. For instance, $j_i = i/2$ for even $i \leq 2n$.

Let $\overline{b}^i$ be sequences of variables of sort i such that $|\overline{b}^i| = |\overline{u}_i|$, and let $\overline{u}^1\overline{y}^1$ be a sequence of variables of sort 1 such that $\overline{u}^1$ is a copy of $\overline{u}$ and $\overline{y}^1 = y^1_1, \ldots, y^1_m$. We define $\phi^*(\overline{x}^1, \ldots, \overline{x}^n)$ as the formula

$$\forall \overline{b}^1 \exists z^1_0 z^1_1 \ldots \forall \overline{b}^n \exists z^n_0 z^n_1 \forall \overline{u}^1 \exists \overline{y}^1 \big(\theta_0 \wedge \theta_1 \wedge \psi'(\overline{u}^1, \overline{y}^1) \big)$$

where

$$\theta_0 := \bigwedge_{i=1}^{n} {=}\Big(\overline{b}^i, z^i_0\Big) \wedge {=}\Big(\overline{b}^i, z^i_1\Big) \wedge ((\overline{b}^i \subseteq \overline{x}^i \wedge z^i_0 = z^i_1) \vee^i (\overline{x}^i \mid \overline{b}^i \wedge z^i_0 \neq z^i_1)),$$

$$\theta_1 := \bigwedge_{i=1}^{n} {=}\Big(\overline{u}^1_i, y^1_{2i-1}/\overline{b}^i, z^i_0\Big) \wedge {=}\Big(\overline{u}^1_i, y^1_{2i}/\overline{b}^i, z^i_1\Big) \wedge \bigwedge_{i=n+1}^{m} {=}\big(\overline{u}^1_{j_i}, y^1_i\big),$$

and $\psi'(\overline{u}^1, \overline{y}^1)$ is obtained from $\psi(\overline{u}, \overline{f})$ by replacing $\overline{u}$ pointwise with $\overline{u}^1$ and each $f_i(\overline{u}_{j_i})$ with y^1_i. Above, θ_0 amounts to the description of the characteristic functions f_{2i-1} and f_{2i}. We refer the reader to [6] to check that $\mathcal{M} \models_{\overline{X}} \theta_0$ iff for all i the functions $s(\overline{b}^i) \mapsto s(z^i_0)$ and $s(\overline{b}^i) \mapsto s(z^i_1)$ determined by the assignments $s \in X_i$ agree on $s(\overline{b}^i)$ exactly when $s(\overline{b}^i) \in \text{rel}(X_i)$. The poly-dependence atoms in θ_1 then transfer these functions over to the first team, and the dependence atoms in ψ_1 describe the remaining functions. As in [6], it can now be seen that ϕ^* correctly simulates ϕ. Since exclusion atoms can be expressed in dependence logic, the claim then follows. □

By item (2) of Theorem 21 the result of Theorem 22 extends to a number of other logics as well. For instance, we obtain that poly-independence logic captures all ESO properties of polyteams. The proof of Theorem 22 can be now easily adapted to show that poly-exclusion and poly-dependence logic capture all downward closed ESO properties of polyteams.

Theorem 23. *Let $\phi(R_1, \ldots, R_n)$ be an ESO-sentence that is downward closed with respect to R_i. Then there is a $\mathsf{PFO}(\mathrm{pdep})$-formula $\phi^*(\overline{x}^1, \ldots, \overline{x}^n)$, where $|\overline{x}^i| = \mathrm{ar}(R_i)$, such that for all polyteams $\overline{X} = (X_1, \ldots, X_n)$ with $\mathsf{Dom}(X_i) = \overline{x}^i$ and $X_i \neq \emptyset$,*

$$\mathcal{M} \models_{\overline{X}} \phi^*(\overline{x}^1, \ldots, \overline{x}^n) \Leftrightarrow (\mathcal{M}, R_1 := \mathrm{rel}(X_1), \ldots, R_n := \mathrm{rel}(X_n)) \models \ \phi(R_1, \ldots, R_n).$$

The statement holds also vice versa.

Proof. The direction from $\mathsf{PFO}(\mathrm{pdep})$ to ESO is again similar to the standard translation of [24]. For the other direction, let $\phi(R_1, \ldots, R_n)$ be an ESO-sentence in which the relations R_i appear only negatively. As in the proof of Theorem 22 and by downward closure we may transform it to an equivalent form (see [19] for details)

$$\exists \overline{f} \forall \overline{u} \big(\bigwedge_{i=1}^{n} (\neg R_i(\overline{u}_i) \vee f_{2i-1}(\overline{u}_i) = f_{2i}(\overline{u}_i)) \wedge \psi(\overline{u}, \overline{f}) \big)$$

Now the translation $\phi(\overline{x}^1, \ldots, \overline{x}^n)$ is defined analogously to the proof of Theorem 22 except for θ_0 which is redefined as

$$\theta_0 := \bigwedge_{i=1}^{n} =\!\left(\overline{b}^i, z^i_0\right) \wedge =\!\left(\overline{b}^i, z^i_1\right) \wedge (\overline{x}^i \mid \overline{b}^i \vee^i z^i_0 = z^i_1).$$

Finally the claim follows by eliminating the exclusion atoms from θ_0.

5 Conclusion

In this article we have laid the foundations of polyteam semantics in order to facilitate the fruitful exchange of ideas and results between team semantics and database theory. Our results show that many of the familiar properties and results from team semantics carry over to the polyteam setting. In particular, we identified a natural polyteam analogue of dependence atoms and gave a complete axiomatisation for the associated implication problem. It is an interesting task to develop axiomatic characterisations for these new logics (cf. [10,20]). Another interesting issue is to study the expressive power of various syntactic fragments of logics over polyteams.

References

1. Armstrong, W.W.: Dependency structures of data base relationships. In: Proceedings of IFIP World Computer Congress, pp. 580–583 (1974)
2. Casanova, M.A., Fagin, R., Papadimitriou, C.H.: Inclusion dependencies and their interaction with functional dependencies. J. Comput. Syst. Sci. **28**(1), 29–59 (1984)
3. Durand, A., Hannula, M., Kontinen, J., Meier, A., Virtema, J.: Approximation and dependence via multiteam semantics. In: Gyssens, M., Simari, G. (eds.) FoIKS 2016. LNCS, vol. 9616, pp. 271–291. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-30024-5_15
4. Durand, A., Kontinen, J., Vollmer, H.: Expressivity and complexity of dependence logic. In: Abramsky, S., Kontinen, J., Väänänen, J., Vollmer, H. (eds.) Dependence Logic: Theory and Applications, pp. 5–32. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-31803-5_2
5. Fagin, R., Kolaitis, P.G., Miller, R.J., Popa, L.: Data exchange: semantics and query answering. Theoret. Comput. Sci. **336**(1), 89–124 (2005)
6. Galliani, P.: Inclusion and exclusion dependencies in team semantics: on some logics of imperfect information. Ann. Pure Appl. Logic **163**(1), 68–84 (2012)
7. Galliani, P., Hella, L.: Inclusion logic and fixed point logic. In: Proceedings of CSL, pp. 281–295 (2013)
8. Geiger, D., Paz, A., Pearl, J.: Axioms and algorithms for inferences involving probabilistic independence. Inf. Comput. **91**(1), 128–141 (1991)
9. Grädel, E., Väänänen, J.A.: Dependence and independence. Studia Logica **101**(2), 399–410 (2013)
10. Hannula, M.: Axiomatizing first-order consequences in independence logic. Ann. Pure Appl. Logic **166**(1), 61–91 (2015)
11. Hannula, M.: Reasoning about embedded dependencies using inclusion dependencies. In: Davis, M., Fehnker, A., McIver, A., Voronkov, A. (eds.) LPAR 2015. LNCS, vol. 9450, pp. 16–30. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48899-7_2
12. Hannula, M., Kontinen, J.: A finite axiomatization of conditional independence and inclusion dependencies. Inf. Comput. **249**, 121–137 (2016)
13. Hannula, M., Kontinen, J., Link, S.: On the finite and general implication problems of independence atoms and keys. J. Comput. Syst. Sci. **82**(5), 856–877 (2016)
14. Herrmann, C.: On the undecidability of implications between embedded multivalued database dependencies. Inf. Comput. **122**(2), 221–235 (1995)
15. Hodges, W.: Compositional semantics for a language of imperfect information. J. Interest Group Pure Appl. Logics **5**(4), 539–563 (1997)
16. Kanellakis, P.C.: Elements of relational database theory. In: Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics (B), pp. 1073–1156. MIT Press, Cambridge (1990)
17. Kontinen, J., Kuusisto, A., Virtema, J.: Decidability of predicate logics with team semantics. In: Proceedings of MFCS 2016, pp. 60:1–60:14 (2016)
18. Kontinen, J., Link, S., Väänänen, J.: Independence in database relations. In: Libkin, L., Kohlenbach, U., de Queiroz, R. (eds.) WoLLIC 2013. LNCS, vol. 8071, pp. 179–193. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39992-3_17
19. Kontinen, J., Väänänen, J.: On definability in dependence logic. J. Logic Lang. Inf. **3**(18), 317–332 (2009)

20. Kontinen, J., Väänänen, J.: Axiomatizing first-order consequences in dependence logic. Ann. Pure Appl. Logic **164**(11), 1101–1117 (2013)
21. Kuusisto, A.: A double team semantics for generalized quantifiers. J. Logic Lang. Inf. **24**(2), 149–191 (2015)
22. Sagiv, Y., Walecka, S.F.: Subset dependencies and a completeness result for a subclass of embedded multivalued dependencies. J. ACM **29**(1), 103–117 (1982)
23. Parker Jr., D.S., Parsaye-Ghomi, K.: Inferences involving embedded multivalued dependencies and transitive dependencies. In: Proceedings of the 1980 ACM SIGMOD International Conference on Management of Data, pp. 52–57 (1980)
24. Väänänen, J.: Dependence Logic. Cambridge University Press, New York (2007)
25. Väänänen, J.: The logic of approximate dependence. In: Başkent, C., Moss, L.S., Ramanujam, R. (eds.) Rohit Parikh on Logic, Language and Society, vol. 11, pp. 227–234. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-47843-2_12

On the Sharpness and the Single-Conclusion Property of Basic Justification Models

Vladimir N. Krupski(✉)

Faculty of Mechanics and Mathematics, Lomonosov Moscow State University,
Moscow 119992, Russia
krupski@lpcs.math.msu.su

Abstract. Justification Awareness Models, JAMs, were proposed by S. Artemov as a tool for modelling epistemic scenarios like Russell's Prime Minister example. It was demonstrated that the sharpness and the single-conclusion property of a model play essential role in the epistemic usage of JAMs. The problem to axiomatize these properties using the propositional justification language was left opened. We propose the solution and define a decidable justification logic J_{ref} that is sound and complete with respect to the class of all sharp single-conclusion justification models.

Keywords: Modal logic · Justification logic
Justification awareness models · Single-conclusion property
Sharpness property

1 Introduction

Justification Awareness Models (*JAM*) were introduced in [3] (see also [4]) as a flexible tool for modelling epistemic scenarios like Russell's Prime Minister example.[1] A *JAM* consists of a basic model for justification logic J^- (see [2]), supplied with the means to distinguish acceptable (i.e. meaningful) and knowledge-producing justifications.

In this paper we consider the first component. It is referred in [3] as *a basic justification model*. The language of the model extends the usual propositional language by new atoms, *justification assertions*, of the form $t : F$ with the intended meaning "t is a justification of F". Justifications are terms built from atomic ones by a binary operation · (application) that reflects logical reasonings via *Modus ponens* rule, so the following property is assumed:

$$s{:}(F \rightarrow G) \rightarrow (t{:}F \rightarrow [s{\cdot}t]{:}G). \tag{1}$$

Justification logic J^- is the extension of the classical propositional logic by Application axiom (1) and a basic justification model (up to some details of

[1] In [3] they were referred as *JEM*s, Justification Epistemic Models. Later the terminology was changed, see [4].

S. Artemov and A. Nerode (Eds.): LFCS 2018, LNCS 10703, pp. 211–220, 2018.
https://doi.org/10.1007/978-3-319-72056-2_13

the formulation, see Sect. 2.1) corresponds to a single world in the constructive canonical model for J^-. In such a model a justification t denotes the set of formulas justified by t and the justification assertion $t : F$ means that F is a member of this set. The application $\cdot$ denotes a binary operation on sets of formulas that satisfy the condition (1).

The epistemic usage of *JAM*s involves the detailed analysis of the term structure of a model. The following properties of a model, the *sharpness* and the *single-conclusion property*, are pointed out in [3] as essential.

Sharpness. Consider a model with some true justification assertion of the form $[s{\cdot}t]{:}G$. It is a claim that G follows by logical reasoning using *Modus ponens* rule from some facts already justified by s and t respectively. One should treat it as nonsense when there is no such facts. The sharpness condition eliminates this possibility. It requires that application should be interpreted by the following operation on sets of formulas:

$$S \rhd T = \{G \mid F \to G \in S \text{ and } F \in T \text{ for some } F\}.$$

So, in a sharp model the application means application of *Modus ponens* rule and nothing more.

Single-conclusion justifications.[2] A model is single-conclusion if for every justification t there exists at most one formula that is justified by t. This requirement admits the treatment of justifications as objects, not only as parts of justification assertions. The decision whether a justification t is meaningful or knowledge-producing can be made on the basis of the analysis of t itself and does not depend on the context where it is used. The justified statement v_t can be restored from it, so for meaningful t the justification assertion $t{:}F$ implies $v_t = F$ and $t{:}v_t$.

Justification logic J^- is sound and complete with respect to the class of all basic justification models (see [2,3]). How to axiomatize the class of all basic justification models that are sharp and single-conclusion? This question was stated as an open problem in [3]. We provide the solution.

The key idea is to distinguish between the language of a model and the language of the logic. Both of them are justification languages but in the first one atoms are treated as constants whereas in the second one they are syntactical variables that admit substitution. An interpretation of the logical language in a model is an infinite substitution that replaces syntactical variables with corresponding expressions of the language of the model, the translation need not be injective. This approach gives the possibility to axiomatize the single-conclusion property of a model via Unification axioms (see [1,7,8] where they are used for axiomatization of the single-conclusion property of arithmetical proof predicates).

In the presence of Unification axioms the sharpness property can be expressed using reference constructions v_t. We add them to the logical language. Reference constructions in the justification language were considered in [9,10] where the general technique was developed and used in the context of Logic of Proofs. We simplify the exposition and adjust it to the case of J^- and the particular

[2] In [3] they were referred as injective justifications. In [4] the terminology was changed.

reference construction “the judgement justified by t”. As a result we obtain a decidable justification logic J_{ref} and prove that it is sound and complete with respect to the class of all sharp single-conclusion basic justification models.

2 Preliminaries

2.1 Basic Justification Models

Let P^0 (atomic propositions) and J^0 (atomic justifications) be disjoint countable sets of identifiers. The justification language $L(P^0, J^0)$ has two sorts of expressions — justification terms (Tm^0) and formulas (Fm^0), defined by the following grammar:

$$Tm^0 ::= J^0 \mid Tm^0 \cdot Tm^0, \qquad Fm^0 ::= \bot \mid P^0 \mid Fm^0 \rightarrow Fm^0 \mid Tm^0\colon Fm^0.$$

A *basic justification model* is defined in [3] as a pair $\langle L(P^0, J^0), *\rangle$ where $*$ is an interpretation that consists of two parts, $*\colon Fm^0 \rightarrow \{0,1\}$, $*\colon Tm^0 \rightarrow 2^{Fm^0}$. It has the following properties:

$$\bot^* = 0, \qquad (F \rightarrow G)^* = 1 \Leftrightarrow (F^* = 0 \text{ or } G^* = 1),$$

$$(t\colon F)^* = 1 \Leftrightarrow F \in t^*, \qquad s^* \rhd t^* \subseteq (s \cdot t)^*.$$

The class of all basic models can be axiomatized by the system J^- (see [2]) which is asserted in [3] to be the base system of justification epistemic logic. Basic models correspond to possible worlds in the canonical model of J^-, so J^- is sound and complete with respect to this semantics (see [3,4]).

A basic model is called *sharp* when $s^* \rhd t^* = (s \cdot t)^*$ for all $t, s \in Tm^0$. It is *single-conclusion* if for all $t \in Tm^0$ the set t^* contains no more than one formula. These properties of a model become essential when we analyze the term structure of justifications in more details. Single-conclusion justifications can be used as pointers (see [7,9,10] for details). Below we exploit this ability in order to axiomatize the sharpness property.

2.2 Unification

We recall the unification technique developed in [9,10]. Let $P = \{p_0, p_1, \ldots\}$ and $J = \{x_0, x_1, \ldots\}$ be sets of syntactical (first-order) variables of two sorts. The language $L^v(P, J)$ is the extension of $L(P, J)$ by the additional second-order function variable v of type $Tm \rightarrow Fm$. It is defined by the grammar

$$Tm ::= J \mid Tm \cdot Tm, \qquad Fm ::= \bot \mid P \mid Fm \rightarrow Fm \mid Tm\colon Fm \mid v(Tm),$$

so expressions of the form $v(t)$ are additional first-order variables indexed by terms, $v_t = v(t)$. Below we use this notation for better readability.[3]

[3] In [9,10] these variables are called *reference constructions*. In the context of Single-Conclusion Logic of Proofs they represent syntactical operations that restore some parts of a formula given its proof. It will be seen that v corresponds to the proof goal operation that extracts a formula from its proof.

Members of $\mathit{Expr} = \mathit{Tm} \cup \mathit{Fm}$ will be considered as terms in the signature $\Omega = \{\bot, \to, :, \cdot\}$ and will be called *expressions*. In this context *a substitution* is a sort preserving homomorphism of free term algebras of signature Ω, i.e. a function on Expr that maps terms into terms, formulas into formulas and commutes with symbols from Ω.

We admit infinite substitutions too. A substitution θ is completely defined by its values on atomic expressions from the set $Var = J \cup P \cup v(\mathit{Tm})$. Let

$$\mathit{Dom}(\theta) = \{z \in Var \mid z\theta \neq z\}, \quad Var(\theta) = \mathit{Dom}(\theta) \ \cup \bigcup_{z \in \mathit{Dom}(\theta)} Var(z\theta),$$

where $Var(e)$ denotes the set of all $z \in Var$ that occur in $e \in \mathit{Expr}$.

A substitution θ is called *comprehensive* if $t_1\theta = t_2\theta$ implies $v_{t_1}\theta = v_{t_2}\theta$ for all $t_1, t_2 \in \mathit{Tm}$.

A *conditional unification problem* is a finite set of conditional equalities

$$A_i = B_i \Rightarrow C_i = D_i, \qquad A_i, B_i, C_i, D_i \in \mathit{Expr}, \ \ i = 1, \ldots, n. \tag{2}$$

Its solution, or *unifier*, is a comprehensive idempotent ($\theta^2 = \theta$) substitution $\theta \colon \mathit{Expr} \to \mathit{Expr}$ such that $A_i\theta = B_i\theta$ implies $C_i\theta = D_i\theta$ for $i = 1, \ldots, n$. The conditional unification problem is called *unifiable* when such a unifier does exist.

The classical (unconditional) first-order unification is a special case of these definitions. In our case the main results of the classical unification theory are also valid. It was established in [8] for the first-order conditional unification; the case of a language with reference constructions of the form $v(t)$ was considered in [9,10] where the following statements were proved:[4]

- The unifiability property for conditional unification problems of the form (2) is decidable.
- Any unifiable problem of the form (2) has a unifier θ that is *the most general unifier* (m.g.u) in the following *weak sense*: any substitution θ' that unifies (2) has the form $\theta' = \theta\lambda$ for some substitution λ. (Note that not every substitution of the form $\theta\lambda$ must unify (2).)
- The m.g.u. of (2) can be computed effectively given A_i, B_i, C_i, D_i, $i = 1, \ldots, n$.
- The computation of θ can be detailed in the following way. Let V be the set of all variables $v \in Var$ that occur in (2). It is possible to compute a finite substitution θ_0 with $\mathit{Dom}(\theta_0) \subseteq V$ such that

$$z\theta = \begin{cases} z\theta_0, & \text{if } z \in V, \\ z, & \text{if } z \in (P \cup J) \setminus V, \\ (v_{t\theta_0})\theta_0, & \text{if } z = v_t \in v(\mathit{Tm}) \setminus V. \end{cases} \tag{3}$$

We may also assume that θ_0 is *conservative*, i.e.

$$Var(\theta_0) \subseteq V \cup \{v_{t\theta_0} \mid v_t \in V\}. \tag{4}$$

[4] The general second-order unification problem is known to be undecidable [5,6]. In our case it is decidable. The problem is more simple because there is no nested occurrences of the function variable v in the language.

The finite substitution θ_0 (together with the finite set V) can be used as a finite representation of the most general unifier θ. We will call it *the finite part* of θ. It can be computed by the variable elimination method, so if two conditional unification problems S and S' are unifiable, $S \subseteq S'$, and θ is a m.g.u. of S with the finite part θ_0, then it is possible to choose a m.g.u. θ' of S' with the finite part θ'_0 for which $Dom(\theta_0) \subseteq Dom(\theta'_0)$. In this case we will write $\theta \preceq \theta'$. Note that if $\theta \preceq \theta'$ and $Dom(\theta_0) = Dom(\theta'_0)$ then S' has the same unifiers as S.

Definition 1. Let S be the conditional unification problem (2) and $A, B \in$ *Expr*. We shall write $A = B \, mod \, S$ when $A\theta = B\theta$ for every unifier θ of S.

Lemma 2 ([9,10]). *The relation* $A = B \, mod \, S$ *is decidable.*

Proof. The unifiability property of S is decidable. If S is not unifiable then $A = B \, mod \, S$ holds for every $A, B \in$ *Expr*. For unifiable S one should restore the most general unifier θ of S and test the equality $A\theta = B\theta$. □

With a formula of the form $G = \bigwedge_{i=1}^{n} t_i : F_i$ we associate a conditional unification problem:

$$t_i = t_j \Rightarrow F_i = F_j, \quad i, j = 1, \ldots, n. \tag{5}$$

We shall write $A = B \, mod \, G$ when $A = B \, mod \, S$ and S is the conditional unification problem (5).

3 Referential Justification Logic J_{ref}

The idea to express the injectivity of justifications via unification first appeared in [1]. Later it was developed in order to axiomatize the single-conclusion property of arithmetical proof predicates (see [7–10]). It was used for axiomatization of symbolic models of single-conclusion proof logics in [11,12]. The concept of an single-conclusion basic justification model is more general, so we extend this approach.

We will distinguish between the language of a basic justification model and the language $L^v(P, J)$ that will be used to formulate the properties of the model.

Definition 3. *An interpretation* of the language $L^v(P, J)$ in a basic justification model $M = \langle L(P^0, J^0), * \rangle$ is a comprehensive (infinite) substitution σ that maps terms and formulas of the language $L^v(P, J)$ into terms and formulas of the language $L(P^0, J^0)$ respectively, $\sigma : Tm \rightarrow Tm^0$, $\sigma : Fm \rightarrow Fm^0$. We also require that $v_t\sigma \in (t\sigma)^*$ when $(t\sigma)^*$ is nonempty. The corresponding validity relation for formulas $F \in Fm$ is defined in the usual way:

$$\langle \sigma, M \rangle \models F \quad \text{iff} \quad (F\sigma)^* = 1.$$

Referential justification logic J_{ref} in the language $L^v(P, J)$ is defined by the following calculus:

(A0) axioms of the classical propositional logic,

(A1) $s\!:\!(F \to G) \to (t\!:\!F \to [s \cdot t]\!:\!G)$, (Application)

(A2) $\bigwedge\limits_{i=1}^{n} t_i\!:\!F_i \to (F \leftrightarrow G) \quad \text{if} \quad F = G \, mod \bigwedge\limits_{i=1}^{n} t_i\!:\!F_i$, (Unification)

(A3) $t\!:\!F \to t\!:\!v_t$, (Assignment)

(A4) $[s \cdot t]\!:\!v_{s\cdot t} \ \to \ s\!:\!(v_t \to v_{s\cdot t}) \wedge t\!:\!v_t$. (Sharpness)

Inference rule: $F \to G,\ F \vdash G$. (Modus ponens)

J_{ref} extends the justification logic J^-. The set of its axioms is decidable by Lemma 2. We will prove that J_{ref} is sound and complete with respect to the class of all interpretations in sharp and single-conclusion basic justification models.

Unification axioms (A2) reflect the single-conclusion property (see [8,10]). Assignment axioms (A3), together with Unification, provide the correct values for reference variables v_t when $t : F$ is valid (the statement v_t restored from t must be equivalent to F). The last axiom scheme (A4) makes it possible to reconstruct logical reasonings given the term structure of justifications. It means the sharpness property.

Theorem 4. *Let σ be an interpretation of $L^v(P, J)$ in a sharp single-conclusion basic justification model $M = \langle L(P^0, J^0), *\rangle$ and $F \in Fm$. Then $\mathsf{J}_{ref} \vdash F$ implies $\langle \sigma, M\rangle \models F$.*

Proof. It is sufficient to prove that the translations of axioms (A0)-(A4) are valid in M. For (A0), (A1) it follows from the fact that M is a model for J^-.

Case (A2). Suppose that $\langle \sigma, M\rangle \models \bigwedge\limits_{i=1}^{n} t_i\!:\!F_i$, so

$$(t_i\sigma)^* = \{F_i\sigma\}, \quad i = 1, \ldots, n.$$

There exists a unifier θ of (5) such that

$$e_1\sigma = e_2\sigma \Leftrightarrow e_1\theta = e_2\theta \tag{6}$$

holds for all expressions e_1, e_2 occurring in (A2). Indeed, let V be the finite set of all variables $v \in Var$ that occur in (A2) and σ_0 be the restriction of σ to V,

$$z\sigma_0 = \begin{cases} z\sigma, & z \in V, \\ z, & z \in Var \setminus V. \end{cases}$$

Consider a substitution $\theta_0 = \sigma_0\lambda$ where λ is an injective substitution that maps P^0 into $(P \setminus V)$ and J^0 into $(J \setminus V)$. The substitution θ_0 maps *Expr* into *Expr* and is idempotent, because any expression of the form $e\theta_0$ does not contain variables from $Dom(\theta_0) \cup Dom(\lambda)$. It satisfies the limited comprehension condition ($t_1\theta_0 = t_2\theta_0 \Rightarrow v_{t_1}\theta_0 = v_{t_2}\theta_0$) only for terms that occur in (A2). The full-scale comprehension will be forced by the transformation (3). The corresponding substitution θ is comprehensive and idempotent. It coincides with θ_0 on variables from V, so the equivalence (6) follows from the injectivity of λ.

We claim that θ is a unifier of (5). Indeed,

$$t_i\theta = t_j\theta \;\Rightarrow\; (t_i\sigma)^* = (t_j\sigma)^* \;\Rightarrow\; F_i\sigma = F_j\sigma \;\Rightarrow\; F_i\theta = F_j\theta.$$

But $F = G\,mod \bigwedge_{i=1}^{n} t_i{:}F_i$ implies $F\theta = G\theta$ and $F\sigma = G\sigma$. Thus, F and G denote the same formula in the language $L(P^0, J^0)$, so $\langle\sigma, M\rangle \models (F \leftrightarrow G)$.

Case (A3) follows from the definition of the translation. If $\langle\sigma, M\rangle \models t{:}\,F$ then $v_t\sigma = F\sigma$ because M is single-conclusion, so $t{:}F$ and $t{:}v_t$ denote the same formula in the language $L(P^0, J^0)$.

Case (A4). Suppose $\langle\sigma, M\rangle \models [s \cdot t] : v_{s\cdot t}$. Then $v_{s\cdot t}\sigma \in (s\sigma \cdot t\sigma)^*$. By the sharpness property of M, there exists a formula F such that $F \in (t\sigma)^*$ and $(F \to v_{s\cdot t}\sigma) \in (s\sigma)^*$. But $v_t\sigma \in (t\sigma)^*$ because $(t\sigma)^*$ is nonempty, so $F = v_t\sigma$ by the single-conclusion property of M. Thus, $\langle\sigma, M\rangle \models s{:}(v_t \to v_{s\cdot t}) \wedge t{:}v_t$. □

4 Completeness

Theorem 5. *Let* $\mathsf{J}_{ref} \nvdash F$. *There exists an interpretation* σ *of the language* $L^v(P, J)$ *in a sharp single-conclusion basic justification model* M *such that* $\langle\sigma, M\rangle \not\models F$.

The completeness proof is based on the saturation procedure from [9,10] where its general form for languages with reference constructions is developed. We will use a simplified version that fits the language $L^v(P, J)$.

Let (θ, Γ, Δ) be the global data structure, where $\theta : \mathit{Expr} \to \mathit{Expr}$ is a substitution[5] and $\Gamma, \Delta \subset Fm$ are finite sets of formulas. The saturation is a nondeterministic procedure that starts from a formula $F \in Fm$. It initializes the data structure: $\theta := id$, $\Gamma := \emptyset$, $\Delta := \{\bot, F\}$. Then it applies repeatedly the following blocks of instructions:

1. For every $X \to Y \in \Gamma$ that has not been discharged by the rule 1 before nondeterministically add Y to Γ or add X to Δ. Discharge $X \to Y$ and all its descendants (its substitutional instances that will be added to Γ by block 3 later). For every $X \to Y \in \Delta$ add X to Γ and add Y to Δ. Repeat these actions until Γ, Δ will not change. If $\Gamma \cap \Delta \neq \emptyset$ then terminate with failure else go to 2.
2. For every $t : X \in \Gamma$ add $t : v_t$ to Γ. For every term t that occurs in some formula from $\Gamma \cup \Delta$ do: if $t\theta{:}v_{t\theta} \in \Gamma$ add $t{:}v_t$ to Γ. For every $[s \cdot t]{:}X \in \Gamma$ also add $s{:}(v_t \to X)$ and $t{:}v_t$ to Γ. For every pair $s{:}(X \to Y)$, $t{:}X \in \Gamma$ do: if the term $s \cdot t$ occurs in some formula from $\Gamma \cup \Delta$ then add $[s \cdot t]{:}Y$ to Γ. Repeat these actions until Γ will not change. If $\Gamma \cap \Delta \neq \emptyset$ then terminate with failure else go to 3.

[5] θ is an infinite substitution of the form (3). We store the finite part of it.

3. Combine a formula $t_1 : F_1 \wedge \ldots \wedge t_n : F_n$ where $t_i : F_i$, $i = 1, \ldots, n$ are all formulas of the form $t{:}X$ from Γ. Test the corresponding unification problem (5) for unifiability. If it is not unifiable then terminate with failure. If it is unifiable then compute an m.g.u. $\theta' \succeq \theta$ of (5) and update $\Gamma := \Gamma \cup \Gamma\theta'$, $\Delta := \Delta \cup \Delta\theta'$. If $\Gamma \cap \Delta \neq \emptyset$ then terminate with failure. Otherwise compare the finite parts θ'_0 and θ_0. If $Dom(\theta'_0) = Dom(\theta_0)$ then set $\theta := \theta'$ and terminate with success; else update $\theta := \theta'$ and go to 1.

Consider a computation of the saturation procedure. Any action in it that changes the data structure (θ, Γ, Δ) will be called a *saturation step*. There are steps of type 1, 2 or 3 depending on the block involved.

Lemma 6. *Every computation of the saturation procedure terminates.*

Proof. Consider a computation starting from F. Suppose that it does not terminate with failure. It is sufficient to prove that it contains a finite number of steps.

Let

$$V^i = V_1^i \cup V_2^i, \quad V_1^i \subset (P \cup J), \quad V_2^i \subset v(Tm)$$

be the set of all variables occurring in $\Gamma \cup \Delta$ and T^i be the set of all terms occurring in $\Gamma \cup \Delta$ at some state i of the computation.

The computation does not change the set V_1^i because all substitutions constructed by steps of type 3 are conservative (see (4)). All variables of a term $t \in T^i$ belong to V_1^i. Steps of types 1,2 do not change the set T^i. Steps of type 3 may extend the set T^i by terms of the form $t\theta'$, $t \in T^i$, but the choice of $\theta' \succeq \theta$ together with the idempotency of m.g.u.'s imply that sets T^i will stabilize after some steps too. One more iteration after it will stabilize the set V_2. Consider the part of the computation after it.

Consider two consecutive iterations of blocks 1–3. Suppose that at the start of the second one there exists a formula $X \to Y \in \Gamma \cup \Delta$ that is not discharged. It is obtained at the previous iteration from some variable $p \in \Gamma \cup \Delta$ by substitution θ executed by block 3,

$$X \to Y = p\theta, \quad p \in P \cup v(Tm).$$

Formula $X \to Y$ and all its descendants will be discharged at the second iteration by block 1. It means that p will be never used in this role later because later the substitution will be updated as $\theta' = \theta\lambda$ and $p\,\theta' = p\,\theta\lambda = p\,\theta^2\lambda = (X \to Y)\theta'$, so $p\,\theta'$ will be a descendant of $X \to Y$ and must be already discharged. Thus, the number of iterations with active steps of type 1 does not exceed the maximal cardinality of sets V^i plus one. Two iterations after the last active step of type 1 will stabilize the conditional unification problem (5) extracted from Γ and terminate the computation with success. □

Let the initial formula F be fixed. All computations starting from F form a saturation tree. It has no infinite paths by Lemma 6. Its branching is bounded, so the saturation tree is finite.

Lemma 7. *If all computations starting from F terminate with failure then $\mathsf{J}_{ref} \vdash F$.*

Proof. Consider a node of the saturation tree. Let Γ, Δ be the contents of the data structure at that node. One can establish by the straightforward induction on the depth of the node that $\mathsf{J}_{ref} \vdash \bigwedge \Gamma \to \bigvee \Delta$. For the root node it implies $\mathsf{J}_{ref} \vdash F$. □

Proof of Theorem 5**.** Suppose $\mathsf{J}_{ref} \nvdash F$. By Lemma 7, there exists a successful computation of the saturation procedure starting from F. Let (θ, Γ, Δ) be the resulting contents of the data structure, $Expr' = \{e\theta \mid e \in Expr\}$,

$$Var' = Var \cap Expr', \quad Tm' = Tm \cap Expr', \quad Fm' = Fm \cap Expr',$$

$$\Gamma' = \Gamma \cap Expr', \qquad \Delta' = \Delta \cap Expr'.$$

The substitution θ is idempotent, so the set $Expr'$ consists of all fixed points of θ. For every term $t \in Tm'$ the set Fm' contains at most one formula of the form $t{:}X$ because θ is comprehensive.

Completion. We construct the set $\Gamma'' \supseteq \Gamma'$, $\Gamma'' \cap \Delta' = \emptyset$, and the substitution $\lambda \colon Expr' \to Expr'$ as follows. Consider a pair of formulas $s{:}(X \to Y)$, $t{:}X \in \Gamma'$ such that $[s \cdot t]{:}Y \notin \Gamma'$. By the restriction from saturation block 2, $[s \cdot t]{:}Y \notin \Delta'$, the variable $v_{s\cdot t}$ does not occur in formulas from $\Gamma' \cup \Delta'$ and $v_{s\cdot t} \in Var'$. Add $[s \cdot t]{:}Y$ to Γ' and set $v_{s\cdot t}\lambda := Y$. Note that the set of all variables occurring in formulas from $\Gamma' \cup \Delta'$ remains unchanged. Repeat this step until Γ' will not change and define Γ'' as the least fixed point of it.

The substitution λ defined by this process is idempotent, $Dom(\lambda) \subset v(Tm')$, $Var(\lambda) \subset Var'$ and $X\lambda = X$ for $X \in \Gamma' \cup \Delta'$. Let

$$P^0 = \{p \in Var' \mid p\lambda = p\}, \qquad J^0 = Var' \cap J.$$

Consider the language $L(P^0, J^0)$ with the interpretation $*$ defined by Γ'':

$$p^* = 1 \Leftrightarrow p \in \Gamma'' \quad \text{ for } p \in P^0,$$

$$t^* = \{X \mid t{:}X \in \Gamma''\} \quad \text{ for } t \in Tm^0.$$

By the construction, it is a basic justification model M that is sharp and single-conclusion. The sharpness condition is forced by saturation block 2 and the completion procedure. The model is single-conclusion because for each t the set Γ' contains at most one formula of the form $t{:}X$ and the completion procedure preserves this property.

Lemma 8 (Truth lemma). *If $G \in \Gamma''$ then $G^* = 1$, if $G \in \Delta'$ then $G^* = 0$.*

Proof. Straightforward induction on the complexity of G. Note that $\Gamma'' \cap \Delta' = \emptyset$. If G is atomic or has the form $t{:}X$ then the statement follows from the definition of $*$. For G of the form $X \to Y$ it is forced by saturation block 1. In this case $G \in \Gamma' \cup \Delta'$, so it will be discharged by block 1 at some step. □

The substitution $\sigma = \theta\lambda$ is an interpretation of the language $L^v(P, J)$ in M. Indeed, it is idempotent because $Var(\lambda) \subset Var'$ and both substitutions θ and λ are idempotent. It is comprehensive because θ is comprehensive and $Dom(\lambda) \subseteq v(Tm')$. As a consequence, the equality $v_t\sigma = v_{t\sigma}\sigma$ holds for each $t \in Tm$.

Suppose $(t\sigma)^* \neq \emptyset$ for some $t \in Tm$. Then $t\sigma = t\theta = t'$, $(t')^* = \{X'\}$ and $t' : X' \in \Gamma''$ for some $t' \in Tm'$, $X' \in Fm'$. If $t' : X' \in \Gamma'$ then, by saturation block 2, $t' : v_{t'} \in \Gamma'$, and $v_{t'}\theta = X'\theta = X'$ by saturation block 3. But in this case $v_t\sigma = v_{t'}\theta$ because $X'\lambda = X'$. If $t' : X' \in \Gamma'' \setminus \Gamma'$ then $v_{t'}\lambda = F'$ by the definition of λ and $v_t\sigma = v_{t'}\lambda$. In both cases $v_t\sigma = X' \in (t\sigma)^*$.

We have $F\sigma = F\theta \in \Delta'$. By Truth lemma, $(F\sigma)^* = 0$, so $\langle\sigma, M\rangle \not\models F$. □

Corollary 9. *The logic J_{ref} is decidable.*

Proof. $\mathsf{J}_{ref} \vdash F$ iff all computations of the saturation procedure starting from F terminate with failure. The saturation tree is finite and can be restored from F. □

Comment. Basic justification models that are single-conclusion but not necessarily sharp can be axiomatized in the language $L(P, J)$ without function variable v by axioms (A0)–(A3). The definition of a unifier used in (A3) should be simplified by omitting the comprehension condition and all other items that involve expressions of the form v_t. The corresponding justification logic is also decidable.

Acknowledgements. I would like to thank Sergei Artemov who attracted my attention to the problem.

References

1. Artemov, S., Straßen, T.: Functionality in the basic logic of proofs. Technical report IAM 92–004, University of Bern (1993)
2. Artemov, S.: The logic of justification. Rev. Symb. Log. **1**(4), 477–513 (2008)
3. Artemov, S.: Epistemic Modeling with Justifications. arXiv:1703.07028v1 (2017)
4. Artemov, S.: Justification awareness models. In: Artemov, S., Nerode, A. (eds.) LFCS 2018. LNCS, vol. 10703, pp. 22–36. Springer, Cham (2018)
5. Farmer, W.M.: Simple second-order languages for which unification is undecidable. Theor. Comput. Sci. **87**, 25–41 (1991)
6. Goldfarb, W.G.: The undecidability of the second-order unification problem. Theor. Comput. Sci. **13**, 225–230 (1981)
7. Krupski, V.N.: Operational logic of proofs with functionality condition on proof predicate. In: Adian, S., Nerode, A. (eds.) Logical Foundations of Computer Science 1997. LNCS, vol. 1234, pp. 167–177. Springer, Heidelberg (1997)
8. Krupski, V.N.: The single-conclusion proof logic and inference rules specification. Ann. Pure Appl. Log. **113**(1–3), 181–206 (2001)
9. Krupski, V.N.: Reference constructions in the single-conclusion proof logic. J. Log. Comput. **16**(5), 645–661 (2006)
10. Krupski, V.N.: Referential logic of proofs. Theor. Comput. Sci. **357**, 143–199 (2006)
11. Krupski, V.N.: Symbolic models for single-conclusion proof logics. In: Ablayev, F., Mayr, E.W. (eds.) CSR 2010. LNCS, vol. 6072, pp. 276–287. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13182-0_26
12. Krupski, V.N.: On symbolic models for single-conclusion logic of proofs. Sb. Math. **202**(5), 683–695 (2011)

Founded Semantics and Constraint Semantics of Logic Rules

Yanhong A. Liu(✉) and Scott D. Stoller

Computer Science Department, Stony Brook University,
Stony Brook, NY 11794, USA
liu@cs.stonybrook.edu

Abstract. Logic rules and inference are fundamental in computer science and have been studied extensively. However, prior semantics of logic languages can have subtle implications and can disagree significantly.

This paper describes a simple new semantics for logic rules, *founded semantics*, and its straightforward extension to another simple new semantics, *constraint semantics*, that unify the core of different prior semantics. The new semantics support unrestricted negation, as well as unrestricted existential and universal quantifications. They are uniquely expressive and intuitive by allowing assumptions about the predicates and rules to be specified explicitly. They are completely declarative and relate cleanly to prior semantics. In addition, founded semantics can be computed in linear time in the size of the ground program.

Keywords: Datalog · Unrestricted negation
Existential and universal quantifications · Fixed-point semantics
Constraints · Well-founded semantics · Stable model semantics
Fitting (Kripke-Kleene) semantics · Supported model semantics

1 Introduction

Logic rules and inference are fundamental in computer science, especially for solving complex modeling, reasoning, and analysis problems in critical areas such as program analysis and verification, security, and decision support.

The semantics of logic rules and their efficient computations have been a subject of significant study, especially for complex rules that involve recursion and unrestricted negation and quantifications. Many different semantics and computation methods have been proposed, e.g., see surveys [1,2]. Even those used in many Prolog-based systems and Answer Set Programming systems—negation as failure [3], well-founded semantics (WFS) [4], and stable model semantics (SMS) [5]—have subtle implications and differ significantly. Is it possible to create a simple semantics that also unifies these different semantics?

This work was supported in part by NSF under grants CCF-1414078, CNS-1421893, IIS-1447549, CCF-1248184, CCF-0964196, and CCF-0613913; and ONR under grants N000141512208 and N000140910651.

S. Artemov and A. Nerode (Eds.): LFCS 2018, LNCS 10703, pp. 221–241, 2018.
https://doi.org/10.1007/978-3-319-72056-2_14

In practice, different semantics may be useful under different assumptions about the facts, rules, and reasoning used. For example, an application may have complete information about some predicates, i.e., sets and relations, but not other predicates. Capturing such situations is important for increasingly larger and more complex applications. Any semantics that is based on a single set of assumptions for all predicates cannot best model such applications. How can a semantics be created to support all different assumptions and still be simple and easy to use?

This paper describes a simple new semantics for logic rules, *founded semantics*, and its straightforward extension to another simple new semantics, *constraint semantics*.

- The new semantics support unrestricted negation (both stratified and non-stratified), as well as unrestricted combinations of existential and universal quantifications.
- They allow each predicate to be specified explicitly as certain (each assertion of the predicate has one of two values: true, false) or uncertain (has one of three values: true, false, undefined), and as complete (all rules defining the predicate are given) or not.
- Completion rules are added for predicates that are complete, as explicit rules for inferring the negation of those predicates using the negation of the hypotheses of the given rules.
- Founded semantics infers all true and false values that are founded, i.e., rooted in the given true or false values and exactly following the rules, and it completes certain predicates with false values and completes uncertain predicates with undefined values.
- Constraint semantics extends founded semantics by allowing undefined values to take all combinations of true and false values that satisfy the constraints imposed by the rules.

Founded semantics and constraint semantics unify the core of previous semantics and have three main advantages:

1. They are expressive and intuitive, by allowing assumptions about predicates and rules to be specified explicitly, by including the choice of uncertain predicates to support common-sense reasoning with ignorance, and by adding explicit completion rules to define the negation of predicates.
2. They are completely declarative. Founded semantics takes the given rules and completion rules as recursive definitions of the predicates and their negation, and is simply the least fixed point of the recursive functions. Constraint semantics takes the given rules and completion rules as constraints, and is simply the set of all solutions that are consistent with founded semantics.
3. They relate cleanly to prior semantics, including stratified semantics [6], first-order logic, Fitting semantics (also called Kripke-Kleene semantics) [7], supported models [6], as well as WFS and SMS, by precisely capturing corresponding assumptions about the predicates and rules.

Additionally, founded semantics can be computed in linear time in the size of the ground program, as opposed to quadratic time for WFS.

Finally, founded semantics and constraint semantics can be extended to allow uncertain, complete predicates to be specified as closed—making an assertion of the predicate false if inferring it to be true (respectively false) using the given rules and facts requires assuming itself to be true (respectively false)—and thus match WFS and SMS, respectively.

2 Motivation for Founded Semantics and Constraint Semantics

Founded semantics and constraint semantics are designed to be intuitive and expressive. For rules with no negation or with restricted negation, which have universally accepted semantics, the new semantics are consistent with the accepted semantics. For rules with unrestricted negation, which so far lack a universally accepted semantics, the new semantics unify the core of prior semantics with two basic principles:

1. Assumptions about certain and uncertain predicates, with true (T) and false (F) values, or possibly undefined (U) values, and about whether the rules defining each predicate are complete must be made explicit.
2. Any easy-to-understand semantics must be consistent with one where everything inferred that has a unique T or F value is rooted in the given T or F values and following the rules.

This section gives informal explanations.

Rules with no negation. Consider a set of rules with no negation in the hypotheses, e.g., a rule can be "`q(x) if p(x)`" but not "`q(x) if not p(x)`" for predicates `p` and `q` and variable `x`. The meaning of the rules, given a set of facts, e.g., a fact `p(a)` for constant `a`, is the set of all facts that are given or can be inferred by applying the rules to the facts, e.g., {`p(a),q(a)`} using the example rule and fact given. In particular,

1. Everything is either T or F, i.e., T as given or inferred facts, or F as otherwise. So one can just explicitly express what are T, and the rest are F.
2. Everything inferred must be founded, i.e., rooted in the given facts and following the rules. So anything that always depends on itself, e.g., `p(a)`, given only the rule "`p(x) if p(x)`", is not T.

In technical terms, the semantics is *2-valued*, and the set of all facts, i.e., true assertions, is the *minimum model*, equal to the *least fixed point* of applying the rules starting from the given facts.

Rules with restricted negation. Consider rules with negation in the hypotheses, but with each negation only on a predicate all of whose facts can be inferred without using rules that contain negation of that predicate, e.g., one can have "`q(x) if not p(x)`" but not "`p(x) if not p(x)`". The meaning of the rules is as for rules with no negation except that a rule with negation is applied only after all facts of the negated predicates have been inferred. In other words,

> The true assertions of any predicate do not depend on the negation of that predicate. So a negation could be just a test after all facts of the negated predicate are inferred. The rest remains the same as for rules with no negation.

In technical terms, this is *stratified negation*; the semantics is still 2-valued, the minimum model, and the set of all true assertions is the least fixed point of applying the rules in order of the *strata*.

Rules with unrestricted negation. Consider rules with unrestricted negation in the hypotheses, where a predicate may cyclically depend on its own negation, e.g., "`p(x) if not p(x)`". Now the value of a negated assertion needs to be established before all facts of the negated predicate have been inferred. In particular,

> There may not be a unique T or F value for each assertion. For example, given only rule "`p(x) if not p(x)`", `p(a)` cannot be T because inferring it following the rule would require itself be F, and it cannot be F because it would lead to itself being T following the rule. That is, there may not be a 2-valued model.

In technical terms, the negation may be *non-stratified*. There are two best solutions to this that generalize a unique 2-valued model: a unique 3-valued model and a set of 2-valued models, as in well-founded semantics (WFS) and stable model semantics (SMS), respectively.

In a unique 3-valued model, when a unique T or F value cannot be established for an assertion, a third value, *undefined* (U), is used. For example, given only rule "`p(x) if not p(x)`", `p(a)` is U, in both WFS and founded semantics.

– With the semantics being 3-valued, when one cannot infer that an assertion is T, one should be able to express whether it is F or U when there is a choice. For example, given only rule "`p(x) if p(x)`", `p(a)` is not T, so `p(a)` may in general be F or U.
– WFS requires that such an assertion be F, even though common sense generally says that it is U. WFS attempts to be the same as in the case of 2-valued semantics, even though one is now in a 3-valued situation.
– Founded semantics supports both, allowing one to choose explicitly when there is a choice. Founded semantics is more expressive by supporting the choice. It is also more intuitive by supporting the common-sense choice for expressing ignorance.

For a set of 2-valued models, similar considerations motivate our constraint semantics. In particular, given only rule "`p(x) if not p(x)`", the semantics is the empty set, i.e., there is no model, in both SMS and constraint semantics, because no model can contain `p(a)` or `not p(a)`, for any `a`, because `p(a)` cannot be T or F as discussed above. However, given only rule "`p(x) if p(x)`", SMS requires that `p(a)` be F in all models, while constraint semantics allows the choice of `p(a)` being F in all models or being T in some models and F in other models.

Certain or uncertain. Founded semantics and constraint semantics first allow a predicate to be declared *certain* (i.e., each assertion of the predicate has one of two values: T, F) or *uncertain* (i.e., each assertion of the predicate has one of three values: T, F, U) when there is a choice. If a predicate is defined (as conclusions of rules) with use of non-stratified negation, then it must be declared uncertain, because it might not have a unique 2-valued model. Otherwise, it may be declared certain or uncertain.

- For a certain predicate, everything T must be given or inferred by following the rules, and the rest are F, in both founded semantics and constraint semantics.
- For an uncertain predicate, everything T or F must be given or inferred, and the rest are U in founded semantics. Constraint semantics then extends everything U to be combinations of T and F that satisfy all the rules and facts as constraints.

Complete or not. Founded semantics and constraint semantics then allow an uncertain predicate that is in the conclusion of a rule to be declared *complete*, i.e., all rules with that predicate in the conclusion are given.

- If a predicate is complete, then completion rules are added to define the negation of the predicate explicitly using the negation of the hypotheses of all given rules and facts of that predicates.
- Completion rules, if any, and given rules are used together to infer everything T and F. The rest are U in founded semantics, and are combinations of T and F in constraint semantics as described above.

Closed or not. Finally, founded semantics and constraint semantics can be extended to allow an uncertain, complete predicate to be declared *closed*, i.e., an assertion of the predicate is made F, called *self-false*, if inferring it to be T (respectively F) using the given rules and facts requires assuming itself to be T (respectively F).

- Determining self-false assertions is similar to determining unfounded sets in WFS. Repeatedly computing founded semantics and self-false assertions until a least fixed point is reached yields WFS.
- Among combinations of T and F values for assertions with U values in WFS, removing each combination that has self-false assertions that are not already F in that combination yields SMS.

Correspondence to prior semantics, more on motivation. Table 1 summarizes corresponding declarations that capture different assumptions under prior semantics; formal definitions and proofs for these and for additional relationships

Table 1. Correspondence between prior semantics and the new semantics, with declarations for all predicates, capturing different assumptions under prior semantics. Stratified semantics is given only for rules that do not use non-stratified negation, whereas the other semantics are given for rules with unrestricted negation.

Prior semantics	New	Certain?	Complete?	Closed?	Theorem
Stratified	Founded	Yes	(Implied yes)	(Implied yes)	5
	Constraint				
First-Order Logic	Constraint	No	No	(Implied no)	6
Fitting (Kripke-Kleene)	Founded	No except for extensional predicates	Yes	No	7
Supported	Constraint				11
WFS	Founded	Any allowed	Yes	Yes	17
SMS	Constraint				18

appear in the following sections. Founded semantics and constraint semantics allow additional combinations of declarations besides those in the table.

Some observations from the table may help one better understand founded semantics and constraint semantics.

- The 4 wide rows cover all combinations of allowed declarations (for all predicates).
- Wide row 1 is a special case of wide row 4, because being certain implies being complete and closed. So one could prefer to use only the latter two choices and omit the first choice. However, being certain is uniquely important, both for conceptual simplicity and practical efficiency:
  (1) It covers the vast class of database applications that do not use non-stratified negation, for which stratified semantics is universally accepted. It does not need to be understood by explicitly combining the latter two more sophisticated notions.
  (2) It allows founded semantics to match WFS for all example programs we found in the literature, with predicates being certain when possible and complete otherwise, but without the last, most sophisticated notion of being closed; and the semantics can be computed in linear time.
- Wide rows 2 and 3 allow the assumption about predicates that are uncertain, not complete, or not closed to be made explicitly.

In a sense, WFS uses F for both false and some kinds of ignorance (no knowledge of something must mean it is F), uses T for both true and some kinds of ignorance inferred through negation of F, and uses U for conflict, remaining kinds of ignorance from T and F, and imprecision; SMS resolves the ignorance in U, but not the ignorance in F and T. In contrast,

- founded semantics uses T only for true, F only for false, and U for conflict, ignorance, and imprecision;
- constraint semantics further differentiates among conflict, ignorance, and imprecision—corresponding to there being no model, multiple models, and a unique model, respectively, consistent with founded semantics.

After all, any easy-to-understand semantics must be consistent with the T and F assertions that can be inferred by exactly following the rules and completion rules starting from the given facts.

- Founded semantics is the maximum set of such T and F assertions, as a least fixed point of the given rules and completion rules if any, plus U for the remaining assertions.
- Constraint semantics is the set of combinations of all T and F assertions that are consistent with founded semantics and satisfy the rules as constraints.

Founded semantics without closed predicates can be computed easily and efficiently, as a least fixed point, contrasting with an alternating fixed point or iterated fixed point for computing WFS.

3 Language

We first consider Datalog with unrestricted negation in hypotheses. We extend it in Sect. 7 to allow unrestricted combinations of existential and universal quantifications and other features.

Datalog with unrestricted negation. A *program* in the core language is a finite set of rules of the following form, where any P_i may be preceded with $\neg$, and any P_i and Q over all rules may be declared certain or uncertain, and declared complete or not:

$$Q(X_1,\ldots,X_a) \;\leftarrow\; P_1(X_{11},\ldots,X_{1a_1}) \;\wedge\; \cdots \;\wedge\; P_h(X_{h1},\ldots,X_{ha_h}) \qquad (1)$$

Symbols $\leftarrow$, $\wedge$, and $\neg$ indicate backward implication, conjunction, and negation, respectively; h is a natural number, each P_i (respectively Q) is a predicate of finite number a_i (respectively a) of arguments, each X_{ij} and X_k is either a constant or a variable, and each variable in the arguments of Q must also be in the arguments of some P_i.

If $h = 0$, there is no P_i or X_{ij}, and each X_k must be a constant, in which case $Q(X_1,\ldots,X_a)$ is called a *fact*. For the rest of the paper, "rule" refers only to the case where $h \geq 1$, in which case each $P_i(X_{i1},\ldots,X_{ia_i})$ or $\neg P_i(X_{i1},\ldots,X_{ia_i})$ is called a *hypothesis* of the rule, and $Q(X_1,\ldots,X_a)$ is called the *conclusion* of the rule. The set of hypotheses of the rule is called the *body* of the rule.

A predicate declared certain means that each assertion of the predicate has a unique true (T) or false (F) value. A predicate declared uncertain means that each assertion of the predicate has a unique true, false, or undefined (U) value. A predicate declared complete means that all rules with that predicate in the conclusion are given in the program.

A predicate in the conclusion of a rule is said to be *defined* using the predicates or their negation in the hypotheses of the rule, and this defined-ness relation is transitive.

- A predicate must be declared uncertain if it is defined transitively using its own negation, or is defined using an uncertain predicate; otherwise, it may be declared certain or uncertain and is by default certain.
- A predicate may be declared complete or not only if it is uncertain and is in the conclusion of a rule, and it is by default complete.

In examples with no explicit specification of declarations, default declarations are used.

Rules of form (1) without negation are captured exactly by Datalog [8,9], a database query language based on the logic programming paradigm. Recursion in Datalog allows queries not expressible in relational algebra or relational calculus. Negation allows more sophisticated logic to be expressed directly. However, unrestricted negation in recursion has been the main challenge in defining the semantics of such a language, e.g., [1,2], including whether the semantics should be 2-valued or 3-valued, and whether the rules are considered complete or not.

Example. We use `win`, the win-not-win game, as a running example, with default declarations: `move` is certain, and `win` is uncertain and complete. A move from position `x` to position `y` is represented by a fact `move(x,y)`. The following rule captures the win-not-win game: a position `x` is winning if there is a move from `x` to some position `y` and `y` is not winning. Arguments `x` and `y` are variables.

```
win(x) ← move(x,y) ∧ ¬ win(y)
```

Note that the declarations for predicates `move` and `win` are different. Other choices of declarations can lead to different results, e.g., see the last example under least fixed point in Sect. 4. ■

Additional examples are given in Appendices A and B of [10].

Notations. In arguments of predicates, we use letter sequences for variables, and use numbers and quoted strings for constants.

In presenting the semantics, in particular the completion rules, we use equality and the notations below for existential and universal quantifications, respectively, in the hypotheses of rules, and use negation in the conclusions.

$$\begin{array}{ll} \exists\ X_1,\ldots,X_n \ \mid\ Y & \text{existential quantification} \\ \forall\ X_1,\ldots,X_n \ \mid\ Y & \text{universal quantification} \end{array} \qquad (2)$$

The quantifications return T iff for some or all, respectively, combinations of values of $X_1,\ldots,X_n$, the value of Boolean expression Y is T. The domain of each quantified variable is the set of all constants in the program.

4 Formal Definition of Founded Semantics and Constraint Semantics

Atoms, literals, and projection. Let π be a program. A predicate is *intensional* in π if it appears in the conclusion of at least one rule; otherwise, it is *extensional*. An *atom* of π is a formula formed by applying a predicate symbol in π to constants in π. A *literal* of π is an atom of π or the negation of an atom of π. These are called *positive literals* and *negative literals*, respectively. The literals p and $\neg p$ are *complements* of each other. A set of literals is *consistent* if it does not contain a literal and its complement. The *projection* of a program π onto a set S of predicates, denoted $\mathit{Proj}(\pi, S)$, contains all facts of π whose predicates are in S and all rules of π whose conclusions contain predicates in S.

Interpretations, ground instances, models, and derivability. An *interpretation* of π is a consistent set of literals of π. Interpretations are generally 3-valued: a literal p is *true* (T) in interpretation I if it is in I, is *false* (F) in I if its complement is in I, and is *undefined* (U) in I if neither it nor its complement is in I. An interpretation of π is *2-valued* if it contains, for each atom A of π, either A or its complement. An interpretation I is *2-valued for predicate* P if, for each atom A for P, I contains A or its complement. Interpretations are ordered by set inclusion $\subseteq$.

A *ground instance* of a rule R is any rule that can be obtained from R by expanding universal quantifications into conjunctions over all constants in the domain, and then instantiating the remaining variables with constants. For example, `q(a)` $\leftarrow$ `p(a)` $\wedge$ `r(b)` is a ground instance of `q(x)` $\leftarrow$ `p(x)` $\wedge$ $\exists$ `y` $\mid$ `r(y)`. An interpretation is a *model* of a program if it contains all facts in the program and satisfies all rules of the program, interpreted as formulas in 3-valued logic [7], i.e., for each ground instance of each rule, if the body is true, then so is the conclusion. The *one-step derivability* operator T_π for program π performs one step of inference using rules of π, starting from a given interpretation. Formally, $C \in T_\pi(I)$ iff C is a fact of π or there is a ground instance R of a rule of π with conclusion C such that each hypothesis of R is true in interpretation I.

Dependency graph. The *dependency graph* $\mathit{DG}(\pi)$ of program π is a directed graph with a node for each predicate of π, and an edge from Q to P labeled $+$ (respectively, $-$) if a rule whose conclusion contains Q has a positive (respectively, negative) hypothesis that contains P. If the node for predicate P is in a cycle containing only positive edges, then P has *circular positive dependency* in π; if it is in a cycle containing a negative edge, then P has *circular negative dependency* in π.

Founded semantics. Intuitively, the *founded model* of a program π, denoted $\mathit{Founded}(\pi)$, is the least set of literals that are given as facts or can be inferred by repeated use of the rules. We define $\mathit{Founded}(\pi) = \mathit{UnNameNeg}(\mathit{LFPbySCC}(\mathit{NameNeg}(\mathit{Cmpl}(\pi))))$, where functions Cmpl, $\mathit{NameNeg}$, $\mathit{LFPbySCC}$, and $\mathit{UnNameNeg}$ are defined as follows.

Completion. The completion function, $\mathit{Cmpl}(\pi)$, returns the *completed program* of π. Formally, $\mathit{Cmpl}(\pi) = \mathit{AddInv}(\mathit{Combine}(\pi))$, where *Combine* and *AddInv* are defined as follows.

The function $\mathit{Combine}(\pi)$ returns the program obtained from π by replacing the facts and rules defining each uncertain complete predicate Q with a single *combined rule* for Q, defined as follows. Transform the facts and rules defining Q so they all have the same conclusion $Q(V_1, \ldots, V_a)$, where $V_1, \ldots, V_a$ are fresh variables (i.e., not occurring in the given rules defining Q), by replacing each fact or rule $Q(X_1, \ldots, X_a) \leftarrow H_1 \wedge \cdots \wedge H_h$ with $Q(V_1, \ldots, V_a) \leftarrow (\exists\, Y_1, \ldots, Y_k \mid V_1 = X_1 \wedge \cdots \wedge V_a = X_a \wedge H_1 \wedge \cdots \wedge H_h)$, where $Y_1, \ldots, Y_k$ are all variables occurring in the given fact or rule. Combine the resulting rules for Q into a single rule defining Q whose body is the disjunction of the bodies of those rules. This combined rule for Q is logically equivalent to the original facts and rules for Q. Similar completion rules are used in Clark completion [3] and Fitting semantics [7].

Example. For the `win` example, the rule for `win` becomes the following. For readability, we renamed variables to transform the equality conjuncts into tautologies and then eliminated them.

```
win(x) ← ∃ y | (move(x,y) ∧ ¬ win(y))
```

■

The function $\mathit{AddInv}(\pi)$ returns the program obtained from π by adding, for each uncertain complete predicate Q, a *completion rule* that derives negative literals for Q. The completion rule for Q is obtained from the inverse of the combined rule defining Q (recall that the inverse of $C \leftarrow B$ is $\neg C \leftarrow \neg B$), by putting the body of the rule in negation normal form, i.e., using laws of predicate logic to move negation inwards and eliminate double negations, so that negation is applied only to atoms.

Example. For the `win` example, the added rule is

```
¬ win(x) ← ∀ y | (¬ move(x,y) ∨ win(y))
```

■

Least fixed point. The least fixed point is preceded and followed by functions that introduce and remove, respectively, new predicates representing the negations of the original predicates.

The function $\mathit{NameNeg}(\pi)$ returns the program obtained from π by replacing each negative literal $\neg P(X_1, \ldots, X_a)$ with $\texttt{n.}P(X_1, \ldots, X_a)$, where the new predicate $\texttt{n.}P$ represents the negation of predicate P.

Example. For the `win` example, this yields:

```
win(x) ← ∃ y | (move(x,y) ∧ n.win(y))
n.win(x) ← ∀ y | (n.move(x,y) ∨ win(y))
```

■

The function $\mathit{LFPbySCC}(\pi)$ uses a least fixed point to infer facts for each strongly connected component (SCC) in the dependency graph of π, as follows. Let $S_1, \ldots, S_n$ be a list of the SCCs in dependency order, so earlier SCCs do not depend on later ones; it is easy to show that any linearization of the dependency order leads to the same result for *LFPbySCC*. For convenience, we overload S to also denote the set of predicates in the SCC.

Define $\mathit{LFPbySCC}(\pi) = I_n$, where I_0 is the empty set and $I_i = \mathit{AddNeg}(\mathit{LFP}(T_{I_{i-1} \cup \mathit{Proj}(\pi, S_i)}), S_i)$ for $i \in 1..n$. *LFP* is the least fixed point operator. The least fixed point is well-defined, because the one-step derivability function $T_{I_{i-1} \cup \mathit{Proj}(\pi, S_i)}$ is monotonic, because the program π does not contain negation. The function $\mathit{AddNeg}(I, S)$ returns the interpretation obtained from interpretation I by adding *completion facts* for certain predicates in S to I; specifically, for each certain predicate P in S, for each combination of values $v_1, \ldots, v_a$ of arguments of P, if I does not contain $P(v_1, \ldots, v_a)$, then add $\mathtt{n.}P(v_1, \ldots, v_a)$.

Example. For the `win` example, the least fixed point calculation

1. infers `n.win(x)` for any `x` that does not have `move(x,y)` for any `y`, i.e., has no move to anywhere;
2. infers `win(x)` for any `x` that has `move(x,y)` for some `y` and `n.win(y)` has been inferred;
3. infers more `n.win(x)` for any `x` such that any `y` having `move(x,y)` has `win(y)`;
4. repeatedly does 2 and 3 above until a fixed point is reached. ■

The function $\mathit{UnNameNeg}(I)$ returns the interpretation obtained from interpretation I by replacing each atom $\mathtt{n.}P(X_1, \ldots, X_a)$ with $\neg P(X_1, \ldots, X_a)$.

Example. For the `win` example, positions `x` for which `win(x)` is T, F, and U, respectively, in the founded model correspond exactly to the well-known winning, losing, and draw positions, respectively. In particular,

1. a losing position is one that either does not have a move to anywhere or has moves only to winning positions;
2. a winning position is one that has a move to a losing position; and
3. a draw position is one not satisfying either case above, i.e., it is in a cycle of moves that do not have a move to a losing position, called a *draw cycle*, or is a position that has only sequences of moves to positions in draw cycles. ■

Example. Suppose the running example uses the declaration that `move` is uncertain instead of the default of being certain. This means that moves not in the given `move` have U values, not allowing any `n.win` or `win` facts to be inferred. Therefore, the founded semantics infers that `win` is U for all positions. ■

Constraint semantics. Constraint semantics is a set of 2-valued models based on founded semantics. A *constraint model* of π is a consistent 2-valued interpretation M such that M is a model of $\mathit{Cmpl}(\pi)$ and $\mathit{Founded}(\pi) \subseteq M$. We define $\mathit{Constraint}(\pi)$ to be the set of constraint models of π. Constraint models can be computed from $\mathit{Founded}(\pi)$ by iterating over all assignments of true and false to atoms that are undefined in $\mathit{Founded}(\pi)$, and checking which of the resulting interpretations satisfy all rules in $\mathit{Cmpl}(\pi)$.

Example. For `win`, draw positions (i.e., positions for which `win` is undefined) are in draw cycles, i.e., cycles that do not have a `move` to a `n.win` position, or are positions that have only a sequence of moves to positions in draw cycles.

1. If some SCC has draw cycles of only odd lengths, then there is no satisfying assignment of T and F to `win` for positions in the SCC, so there are no constraint models of the program.
2. If some SCC has draw cycles of only even lengths, then there are two satisfying assignments of T and F to `win` for positions in the SCC, with the truth values alternating between T and F around each cycle, and with the second truth assignment obtained from the first by swapping T and F. The total number of constraint models of the program is exponential in the number of such SCCs. ■

5 Properties of Founded Semantics and Constraint Semantics

Proofs of theorems appear in Appendix C of [10].

Consistency and correctness. The most important properties are consistency and correctness.

Theorem 1. The founded model and constraint models of a program π are consistent.

Theorem 2. The founded model of a program π is a model of π and $\mathit{Cmpl}(\pi)$. The constraint models of π are 2-valued models of π and $\mathit{Cmpl}(\pi)$.

Same SCC, same certainty. All predicates in an SCC have the same certainty.

Theorem 3. For every program, for every SCC S in its dependence graph, all predicates in S are certain, or all of them are uncertain.

Higher-order programming. Higher-order logic programs, in languages such as HiLog, can be encoded as first-order logic programs by a semantics-preserving transformation that replaces uses of the original predicates with uses of a single predicate `holds` whose first argument is the name of an original predicate [11]. For example, `win(x)` is replaced with `holds(win,x)`. This transformation merges a set of predicates into a single predicate, facilitating higher-order programming. We show that founded semantics and constraint semantics are preserved by merging of *compatible* predicates, defined below, if a simple type system is used to distinguish the constants in the original program from the new constants representing the original predicates.

We extend the language with a simple type system. A type denotes a set of constants. Each predicate has a type signature that specifies the type of each argument. A program is well-typed if, in each rule or fact, (1) each constant belongs to the type of the argument where the constant occurs, and (2) for each variable, all its occurrences are as arguments with the same type. In the semantics, the values of predicate arguments are restricted to the appropriate type.

Predicates of program π are *compatible* if they are in the same SCC in $DG(\pi)$ and have the same arity, same type signature, and (if uncertain) same completeness declaration. For a set S of compatible predicates of program π with arity a and type signature $T_1, \ldots, T_a$, the *predicate-merge transformation* $Merge_S$ transforms π into a program $Merge_S(\pi)$ in which predicates in S are replaced with a single fresh predicate `holds` whose first parameter ranges over S, and which has the same completeness declaration as the predicates in S. Each atom A in a rule or fact of π is replaced with $MergeAtom_S(A)$, where the function $MergeAtom_S$ on atoms is defined by: $MergeAtom_S(P(X_1, \ldots, X_a))$ equals `holds`(“P”, $X_1, \ldots, X_a$) if $P \in S$ and equals $P(X_1, \ldots, X_a)$ otherwise. We extend $MergeAtom_S$ pointwise to a function on sets of atoms and a function on sets of sets of atoms. The predicate-merge transformation introduces S as a new type. The type signature of `holds` is $S, T_1, \ldots, T_a$.

Theorem 4. Let S be a set of compatible predicates of program π. Then $Merge_S(\pi)$ and π have the same founded semantics, in the sense that $Founded(Merge_S(\pi)) = MergeAtom_S(Founded(\pi))$. $Merge_S(\pi)$ and π also have the same constraint semantics, in the sense that $Constraint(Merge_S(\pi)) = MergeAtom_S(Constraint(\pi))$.

6 Comparison with Other Semantics

Stratified semantics. Let $Stratified(\pi)$ denote the unique 2-valued model of a program with stratified negation, as discussed in Sect. 2.

Theorem 5. For a program π with stratified negation and in which all predicates are certain, $Founded(\pi) = Stratified(\pi)$.

First-order logic. The next theorem relates constraint models with the interpretation of a program as a set of formulas in first-order logic.

Theorem 6. For a program π in which all predicates are uncertain and not complete, the constraint models of π are exactly the 2-valued models of π.

Fitting semantics. The *Fitting model* of a program π, denoted $Fitting(\pi)$, is the least model of a formula in 3-valued logic [7]; Sect. 6 of [10] summarizes the definition.

Theorem 7. For a program π in which all extensional predicates are certain, and all intensional predicates are uncertain and complete, $Founded(\pi) = Fitting(\pi)$.

Theorem 8. (a) For a program π in which all intensional predicates are uncertain and complete, $Founded(\pi) \subseteq Fitting(\pi)$. (b) If, furthermore, some extensional predicate is uncertain, and some positive literal p for some uncertain extensional predicate does not appear in π, then $Founded(\pi) \subset Fitting(\pi)$.

Theorem 9. (a) For a program π in which all predicates have default declarations as certain or uncertain and complete or not, $\mathit{Fitting}(\pi) \subseteq \mathit{Founded}(\pi)$. (b) If, furthermore, $\mathit{Fitting}(\pi)$ is not 2-valued for some certain intensional predicate P, then $\mathit{Fitting}(\pi) \subset \mathit{Founded}(\pi)$.

Well-founded semantics. The *well-founded model* of a program π, denoted $\mathit{WFS}(\pi)$, is the least fixed point of a monotone operator W_π on interpretations [4]; Sect. 6 of [10] summarizes the definition.

Theorem 10. For every program π, $\mathit{Founded}(\pi) \subseteq \mathit{WFS}(\pi)$.

Supported models. Supported model semantics of a logic program π is a set of 2-valued models [6], denoted $\mathit{Supported}(\pi)$; Sect. 6 of [10] summarizes the definition.

Theorem 11. For a program π in which all extensional predicates are certain, and all intensional predicates are uncertain and complete, $\mathit{Supported}(\pi) = \mathit{Constraint}(\pi)$.

Theorem 12. For a program π in which all intensional predicates are uncertain and complete, $\mathit{Supported}(\pi) \subseteq \mathit{Constraint}(\pi)$.

Theorem 13. For a program π in which all predicates have default declarations as certain or uncertain and complete or not, $\mathit{Constraint}(\pi) \subseteq \mathit{Supported}(\pi)$.

Stable models. Gelfond and Lifschitz define *stable model semantics* (SMS) of logic programs [5]. They define the *stable models* of a program π to be the 2-valued interpretations of π that are fixed points of a particular transformation. Let $\mathit{SMS}(\pi)$ denote the set of stable models of π.

Theorem 14. For a program π in which all predicates have default declarations as certain or uncertain, $\mathit{SMS}(\pi) \subseteq \mathit{Constraint}(\pi)$.

Example. For the `win` example with default declarations, Fitting semantics and WFS are the same as founded semantics in Sect. 4, and supported model semantics and SMS are the same as constraint semantics in Sect. 4. Additional examples can be found in Appendix B of [10]. ■

7 Computational Complexity and Extensions

Computing founded semantics and constraint semantics

Theorem 15. Computing founded semantics is linear time in the size of the ground program.

Proof. First ground all given rules, using any grounding. Then add completion rules, if any, by adding an inverse rule for each group of the grounded given rules that have the same conclusion, yielding ground completion rules of the same asymptotic size as the grounded given rules.

Now compute the least fixed point for each SCC of the resulting ground rules using a previous method [12]. To do so, first introduce a new intermediate predicate and rule for each conjunction and disjunction in the rules, yielding a new set of rules of the same asymptotic size. In computing the least fixed point, each resulting rule incurs at most one rule firing because there are no variables in the rule, and each firing takes worst-case $O(1)$ time. Thus, the total time is worst-case linear in the size of all ground rules and therefore in the size of the grounded given rules. ■

The size of the ground program is polynomial in the size n of input data, i.e., the given facts, because each variable in each rule can be instantiated at most $O(n)$ times (because the domain size is at most n), and there is a fixed number of variables in each rule, and a fixed size of the given rules. Precisely, the size of the ground program is in the worst case $O(n^k \times r)$, where k is the maximum number of variables in a rule, and r is the size of the given rules.

Computing constraint semantics may take exponential time in the size of the input data, because in the worst case, all assertions of all predicates may have U values in founded semantics, and there is an exponential number of combinations of T and F values of all assertions, where each combination may be checked for whether it satisfies the constraints imposed by all rules.

These complexity analyses also apply to the extensions below except that computing founded semantics with closed predicates may take quadratic time in the size of the ground program, because of repeated computation of founded semantics and self-false assertions.

Closed predicate assumption. We can extend the language to support declaration of uncertain complete predicates as *closed*. Informally, this means that an atom A of the predicate is false in an interpretation I, called *self-false* in I, if every ground instance of rules that concludes A, or recursively concludes some hypothesis of that rule instance, has a hypothesis that is false or, recursively, is self-false in I. Self-false atoms are elements of unfounded sets [4].

Formally, $\mathit{SelfFalse}_\pi(I)$, the set of self-false atoms of program π with respect to interpretation I, is defined in the same way as the greatest unfounded set of π with respect to I, except replacing "some positive hypothesis of R is in U" with "some positive hypothesis of R for a closed predicate is in U". The founded semantics of this extended language is defined by repeatedly computing the semantics as per Sect. 4 and then setting self-false atoms to false, until a least fixed point is reached. Formally, the founded semantics is $\mathit{FoundedClosed}(\pi) = \mathit{LFP}(F_\pi)$, where $F_\pi(I) = \mathit{Founded}(\pi \cup I) \cup \neg \cdot \mathit{SelfFalse}_\pi(\mathit{Founded}(\pi \cup I))$.

The constraint semantics for this extended language includes only interpretations that contain the negative literals required by the closed declarations. Formally, a *constraint model* of a program π with closed declarations is a consistent 2-valued interpretation M such that M is a model of $\mathit{Cmpl}(\pi)$,

$\mathit{FoundedClosed}(\pi) \subseteq M$, and $\neg \cdot \mathit{SelfFalse}_{\pi}(M) \subseteq M$. Let $\mathit{ConstraintClosed}(\pi)$ denote the set of constraint models of π.

The next theorem states that changing predicate declarations from uncertain, complete, and closed to certain when allowed, or vice versa, preserves founded and constraint semantics. Theorem 3 implies that this change needs to be made for all predicates in an SCC.

Theorem 16. Let π be a program. Let S be an SCC in its dependence graph containing only predicates that are uncertain, complete, and closed. Let π' be a program identical to π except that all predicates in S are declared certain. Note that, for the declarations in both programs to be allowed, predicates in SCCs that follow S in dependency order must be uncertain, predicates in SCCs that precede S in dependency order must be certain, and predicates in S must not have circular negative dependency. Then $\mathit{FoundedClosed}(\pi) = \mathit{FoundedClosed}(\pi')$ and $\mathit{ConstraintClosed}(\pi) = \mathit{ConstraintClosed}(\pi')$.

Theorem 17. For a program π in which every uncertain predicate is complete and closed, $\mathit{FoundedClosed}(\pi) = \mathit{WFS}(\pi)$.

Theorem 18. For a program π in which every uncertain predicate is complete and closed, $\mathit{ConstraintClosed}(\pi) = \mathit{SMS}(\pi)$.

Note, however, that founded semantics for default declarations (certain when possible and complete otherwise) allows the number of repetitions for computing self-false atoms to be greatly reduced, even to zero, compared with WFS that does repeated computation of unfounded sets.

In all examples we have found in the literature, and all natural examples we have been able to think of, founded semantics for default declarations, without closed predicate assumption, infers the same result as WFS. However, while founded semantics computes a single least fixed point without the outer repetition and is worst-case linear time, WFS computes an alternating fixed point or iterated fixed point and is worst-case quadratic. In fact, we have not found any natural example showing that an actual quadratic-time alternating or iterated fixed-point for computing WFS is needed.[1]

Unrestricted quantifications in hypotheses. We extend the language to allow unrestricted combinations of existential and universal quantifications as well as negation, conjunction, and disjunction in hypotheses. The domain of each quantified variable is the set of all constants in the program.

[1] Even a contrived example that demonstrates the worst-case quadratic-time computation of WFS has been challenging to find. For example, the quadratic-time example in [13] turns out to be linear in XSB; after significant effort between us and Warren, we found a much more sophisticated example that appears to take quadratic time, but a remaining bug in XSB makes the correctness of its computation unclear.

Example. For the `win` example, the following two rules may be given instead:

```
win(x) ← ∃ y | move(x,y) ∧ lose(y)
lose(x) ← ∀ y | ¬ move(x,y) ∨ win(y)
```
■

The semantics in Sect. 4 is easily extended to accommodate this extension: these constructs simply need to be interpreted, using their 3-valued logic semantics [7], when defining one-step derivability. Theorems 1–3 hold for this extended language. The other semantics discussed in Sect. 6 are not defined for this extension, thus we do not have theorems relating to them.

Negation in facts and conclusions. We extend the language to allow negation in given facts and in conclusions of given rules; such facts and rules are said to be *negative*. The Yale shooting example in Appendix B of [10] is a simple example.

The definition of founded semantics applies directly to this extension, because it already introduces and handles negative rules, and it already infers and handles negative facts. Note that *Combine* combines only positive facts and positive rules to form combined rules; negative facts and negative rules are copied unchanged into the completed program.

With this extension, a program and hence its founded model may be inconsistent; for example, a program could contain or imply `p` and ¬`p`. Thus, Theorem 1 does not hold for such programs. When the founded model is inconsistent, the inconsistent literals in it can easily be reported. When the founded model is consistent, the definition of constraint semantics applies directly, and Theorems 2 and 3 hold. The other semantics discussed in Sect. 6 are not defined for this extended language, so we do not have theorems relating to them.

8 Related Work and Conclusion

There is a large literature on logic language semantics and efficient computations. Several overview articles [1,2,14,15] give a good sense of the challenges when there is unrestricted negation. We discuss major prior semantics here.

Clark [3] describes completion of logic programs to give a semantics for negation as failure. Numerous others, e.g., [16–21], describe similar additions. Fitting [7] presents a semantics, called Fitting semantics or Kripke-Kleene semantics, that aims to give a least 3-valued model. Apt et al. [6] defines supported model semantics, which is a set of 2-valued models; the models correspond to extensions of the Fitting model. Apt et al. [6] introduces stratified semantics. WFS [4] also gives a 3-valued model but aims to maximize false values. SMS [5] also gives a set of 2-valued models and aims to maximize false values. Other formalisms and semantics include partial stable models, also called stationary models [14], and FO(ID), for first-order logic with inductive definitions [22]. There are also many studies that relate different semantics, e.g., [23,24].

Our founded semantics, which extends to constraint semantics, is unique in that it allows predicates to be specified as certain or uncertain, as complete or not, and as closed or not. These choices clearly and explicitly capture the different assumptions one can have about the predicates, rules, and reasoning, including the well-known closed-world assumption vs open-world assumption—i.e., whether or not all rules and facts about a predicate are given in the program—and allow both to co-exist naturally. These choices make our new semantics more expressive and intuitive. Instead of using many separate semantics, one just need to make the assumptions explicit; the same underlying logic is used for inference. In this way, founded semantics and constraint semantics unify different semantics.

In addition, founded semantics and constraint semantics are completely declarative, as a least fixed point and as constraint satisfaction, respectively. Our default declarations without closed predicates lead to the same semantics as WFS and SMS for all natural examples we have found. Additionally, founded semantics without closed predicates can be computed in linear time in the size of the ground program, as opposed to quadratic time for WFS.

There are many directions for future study, including additional relationships with prior semantics, further extensions, efficient implementations, and applications.

Acknowledgment. We thank David S. Warren, Michael Kifer, Anil Nerode, Tuncay Tekle, Molham Aref, Marc Denecker, Cordell Green, Goyal Gupta, Bob Kowalski, Fangzhen Lin, Alberto Pettorossi, Maurizio Proietti, Neng-Fa Zhou, and many others for helpful comments and discussions on logic languages, semantics, and efficient computations.

A Comparison of Semantics for Well-Known Small Examples and More

Table 2 shows well-known example rules and more for tricky boundary cases in the semantics, where all uncertain predicates that are in a conclusion are declared complete, but not closed, and shows different semantics for them.

- Programs 1 and 2 contain only negative cycles. All three of Founded, WFS, and Fitting agree. All three of Constraint, SMS, and Supported agree.
- Programs 3 and 4 contain only positive cycles. Founded for certain agrees with WFS; Founded for uncertain agrees with Fitting. Constraint for certain agrees with SMS; Constraint for uncertain agrees with Supported.
- Programs 5 and 6 contain no cycles. Founded for certain agrees with WFS and Fitting; Founded for uncertain has more undefined. Constraint for certain agrees with SMS and Supported; Constraint for uncertain has more models.
- Programs 7 and 8 contain both negative and positive cycles. For program 7 where ¬ q and q are disjunctive, all three of Founded, WFS, and Fitting agree; Constraint and Supported agree, but SMS has no model. For program 8 where ¬ q and q are conjunctive, Founded and Fitting agree, but WFS has q being F; all three of Constraint, SMS, and Supported agree.

For all 8 programs, with default complete but not closed predicates, we have the following:

Table 2. Different semantics for programs where all uncertain predicates that are in a conclusion are declared complete, but not closed. "uncertain" means all predicates in the program are declared uncertain. "certain" means all predicates in the program that can be declared certain are declared certain; "–" means no predicates can be declared certain, so the semantics is the same as "uncertain". p, $\overline{\text{p}}$ and $\underline{\text{p}}$ mean p is T, F, and U, respectively.

	Program	Founded (not closed)		WFS	Fitting (Kripke-Kleene)	Constraint (not closed)		SMS	Supported
		Uncertain	Certain			Uncertain	Certain		
1	$q \leftarrow \neg\, q$	$\{\underline{q}\}$	–	$\{\underline{q}\}$	$\{\underline{q}\}$	No model	–	No model	No model
2	$q \leftarrow \neg\, p$ $p \leftarrow \neg\, q$	$\{\underline{p}, \underline{q}\}$	–	$\{\underline{p}, \underline{q}\}$	$\{\underline{p}, \underline{q}\}$	$\{p, \overline{q}\}$,$\{\overline{p}, q\}$	–	$\{p, \overline{q}\}$,$\{\overline{p}, q\}$	$\{p, \overline{q}\}$,$\{\overline{p}, q\}$
3	$q \leftarrow q$	$\{\underline{q}\}$	$\{\overline{q}\}$	$\{\overline{q}\}$	$\{\underline{q}\}$	$\{q\}$,$\{\overline{q}\}$	$\{\overline{q}\}$	$\{\overline{q}\}$	$\{q\}$,$\{\overline{q}\}$
4	$q \leftarrow p$ $p \leftarrow q$	$\{\underline{p}, \underline{q}\}$	$\{\overline{p}, \overline{q}\}$	$\{\overline{p}, \overline{q}\}$	$\{\underline{p}, \underline{q}\}$	$\{p, q\}$,$\{\overline{p}, \overline{q}\}$	$\{\overline{p}, \overline{q}\}$	$\{\overline{p}, \overline{q}\}$	$\{p, q\}$,$\{\overline{p}, \overline{q}\}$
5	$q \leftarrow \neg\, p$	$\{\underline{p}, \underline{q}\}$	$\{\overline{p}, q\}$	$\{\overline{p}, q\}$	$\{\overline{p}, q\}$	$\{p, \overline{q}\}$,$\{\overline{p}, q\}$	$\{\overline{p}, q\}$	$\{\overline{p}, q\}$	$\{\overline{p}, q\}$
6	$q \leftarrow p$	$\{\underline{p}, \underline{q}\}$	$\{\overline{p}, \overline{q}\}$	$\{\overline{p}, \overline{q}\}$	$\{\overline{p}, \overline{q}\}$	$\{p, q\}$,$\{\overline{p}, \overline{q}\}$	$\{\overline{p}, \overline{q}\}$	$\{\overline{p}, \overline{q}\}$	$\{\overline{p}, \overline{q}\}$
7	$q \leftarrow \neg\, q$ $q \leftarrow q$	$\{\underline{q}\}$	–	$\{\underline{q}\}$	$\{\underline{q}\}$	$\{q\}$	–	no model	$\{q\}$
8	$q \leftarrow \neg\, q$ $\wedge\, q$	$\{\underline{q}\}$	–	$\{\overline{q}\}$	$\{\underline{q}\}$	$\{\overline{q}\}$	–	$\{\overline{q}\}$	$\{\overline{q}\}$

- If all predicates are the default certain or uncertain, then Founded agrees with WFS, and Constraint agrees with SMS, with one exception for each:
  (1) Program 7 concludes q whether q is F or T, so SMS having no model is an extreme outlier among all 6 semantics and is not consistent with common sense.
  (2) Program 8 concludes q if q is F and T, so Founded semantics with q being U is imprecise, but Constraint has q being F. WFS has q being F because it uses F for ignorance.
- If predicates not in any conclusion are certain (not shown in Table 2 but only needed for q in programs 5 and 6), and other predicates are uncertain, then Founded equals Fitting, and Constraint equals Supported, as captured in Theorems 7 and 11, respectively.
- If all predicates are uncertain, then Founded has all values being U, capturing the well-known unclear situations in all these programs, and Constraint gives all different models except for programs 2 and 5, and programs 4 and 6, which are pair-wise equivalent under completion, capturing exactly the differences among all these programs.

Finally, if all predicates in these programs are not complete, then Founded and Constraint are the same as in Table 2 except that Constraint for uncertain becomes equivalent to truth values in first-order logic: programs 1 and 8 have an additional model, {q}, program 6 has an additional model, {$\overline{\mathrm{p}}$, q}, and programs 2 and 5 have an additional model, {p,q}.

References

1. Apt, K.R., Bol, R.N.: Logic programming and negation: a survey. J. Log. Program. **19**, 9–71 (1994)
2. Fitting, M.: Fixpoint semantics for logic programming: a survey. Theor. Comput. Sci. **278**(1), 25–51 (2002)
3. Clark, K.L.: Negation as failure. In: Gallaire, H., Minker, J. (eds.) Logic and Databases, pp. 293–322. Plenum Press, New York (1978)
4. Van Gelder, A., Ross, K., Schlipf, J.S.: The well-founded semantics for general logic programs. J. ACM **38**(3), 620–650 (1991)
5. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: Proceedings of the 5th International Conference and Symposium on Logic Programming, pp. 1070–1080. MIT Press (1988)
6. Apt, K.R., Blair, H.A., Walker, A.: Towards a theory of declarative knowledge. In: Foundations of Deductive Databases and Logic Programming, pp. 89–148. Morgan Kaufman (1988)
7. Fitting, M.: A Kripke-Kleene semantics for logic programs. J. Log. Program. **2**(4), 295–312 (1985)
8. Ceri, S., Gottlob, G., Tanca, L.: Logic Programming and Databases. Springer, Heidelberg (1990)
9. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases: The Logical Level. Addison-Wesley, Reading (1995)

10. Liu, Y.A., Stoller, S.D.: The founded semantics and constraint semantics of logic rules. Computing Research Repository arXiv:1606.06269 [cs.LO] (Revised 2017) (2016)
11. Chen, W., Kifer, M., Warren, D.S.: HiLog: a foundation for higher-order logic programming. J. Log. Program. **15**(3), 187–230 (1993)
12. Liu, Y.A., Stoller, S.D.: From datalog rules to efficient programs with time and space guarantees. ACM Trans. Program. Lang. Syst. **31**(6), 1–38 (2009)
13. Zukowski, U.: Flexible computation of the well-founded semantics of normal logic programs. Ph.D. thesis, Faculty of Computer Science and Mathematics, University of Passau (2001)
14. Przymusinski, T.C.: Well-founded and stationary models of logic programs. Ann. Math. Artif. Intell. **12**(3), 141–187 (1994)
15. Ramakrishnan, R., Ullman, J.D.: A survey of deductive database systems. J. Log. Program. **23**(2), 125–149 (1995)
16. Lloyd, J.W., Topor, R.W.: Making Prolog more expressive. J. Log. Program. **1**(3), 225–240 (1984)
17. Sato, T., Tamaki, H.: Transformational logic program synthesis. In: Proceedings of the International Conference on Fifth Generation Computer Systems, pp. 195–201 (1984)
18. Jaffar, J., Lassez, J.-L., Maher, M.J.: Some issues and trends in the semantics of logic programming. In: Shapiro, E. (ed.) ICLP 1986. LNCS, vol. 225, pp. 223–241. Springer, Heidelberg (1986). https://doi.org/10.1007/3-540-16492-8_78
19. Chan, D.: Constructive negation based on the completed database. In: Proceedings of the 5th International Conference and Symposium on Logic Programming, pp. 111–125. MIT Press (1988)
20. Foo, N.Y., Rao, A.S., Taylor, A., Walker, A.: Deduced relevant types and constructive negation. In: Proceedings of the 5th International Conference and Symposium on Logic Programming, pp. 126–139 (1988)
21. Stuckey, P.J.: Constructive negation for constraint logic programming. In: Proceedings of the 6th Annual IEEE Symposium on Logic in Computer Science, pp. 328–339 (1991)
22. Denecker, M., Ternovska, E.: A logic of nonmonotone inductive definitions. ACM Trans. Comput. Log. **9**(2), 14 (2008)
23. Dung, P.M.: On the relations between stable and well-founded semantics of logic programs. Theor. Comput. Sci. **105**(1), 7–25 (1992)
24. Lin, F., Zhao, Y.: Assat: computing answer sets of a logic program by sat solvers. Artif. Intell. **157**(1–2), 115–137 (2004)

Separating the Fan Theorem and Its Weakenings II

Robert S. Lubarsky(✉)

Department of Mathematical Sciences, Florida Atlantic University,
Boca Raton, FL 33431, USA
Robert.Lubarsky@alum.mit.edu

Abstract. Varieties of the Fan Theorem have recently been developed in reverse constructive mathematics, corresponding to different continuity principles. They form a natural implicational hierarchy. Earlier work showed all of these implications to be strict. Here we re-prove one of the strictness results, using very different arguments. The technique used is a mixture of realizability, forcing in the guise of Heyting-valued models, and Kripke models.

Keywords: Fan Theorems · Kripke models · Forcing
Heyting-valued models · Formal topology · Recursive realizability

AMS 2010 MSC: 03C90 · 03E70 · 03F50 · 03F60 · 03D80

1 Introduction

The Fan Theorem states that, in $2^{<\omega}$, every bar (i.e. set of nodes which contains a member of every (infinite) path) is uniform (i.e. contains a member of every (infinite) path by some fixed level of $2^{<\omega}$). It has been important in the foundation of constructive mathematics every since it was first articulated (by Brouwer), and so it is no surprise that with the development of reverse mathematics in recent years it has become an important principle there. In particular, various weakenings of it have been shown to be equivalent to some principles involving continuity and compactness [2,4,9]. These weakenings all involve strengthening the hypothesis, by restricting which bars they apply to. The strictest version, FAN_Δ or Decidable Fan, is to say that the bar B in question is **decidable**: every node is either in B or not. Another natural version, $\text{FAN}_{\Pi^0_1}$ or Π^0_1 Fan, is to consider Π^0_1 **bars**: there is a decidable set $C \subseteq 2^{<\omega} \times \mathbb{N}$ such that $\sigma \in B$ iff, for all $n \in \mathbb{N}$, $(\sigma, n) \in C$. Nestled in between these two is FAN_c or c-Fan, which is based on the notion of a ***c*-bar**, which is a particular kind of Π^0_1 bar: for some decidable set $C \subseteq 2^{<\omega}$, $\sigma \in B$ iff every extension of σ is in C. It is easy to see that the implications all hold over a weak base theory. What about the reverse implications? (We always include the implication of FAN_Δ from basic set theory when discussing the converses of the conditionals above.)

S. Artemov and A. Nerode (Eds.): LFCS 2018, LNCS 10703, pp. 242–255, 2018.
https://doi.org/10.1007/978-3-319-72056-2_15

There had been several proofs that some of the converses did not hold [1,3,6]. These were piecemeal, in that each applied to only one converse, or even just a weak form of the converse, and used totally different techniques, so that there was no uniform view of the matter. This situation changed with [10], which provided a family of Kripke models showing the non-reversal of all the implications. It was asked there whether those models were in some sense the right, or canonical, models for this purpose; implicit was the question whether the other common modeling techniques, realizability and Heyting-valued models, could provide the same separations.

Here we do not answer those questions. We merely bring the discussion along, by providing a different kind of model. It should be pointed out early on that, at this point, the only separation provided is that FAN_Δ does not imply FAN_c, although we see no reason the arguments could not be extended to the other versions of Fan.

There are several ways that the model here differs from those of [10]. In the earlier paper, a tree with no simple paths was built over a model of classical ZFC via forcing, and the non-implications were shown by hiding that tree better or worse in various models of IZF. In particular, we showed there that FAN_Δ does not imply FAN_c by including that tree as the complement of a c-bar in a gentle enough way that no new decidable bars were introduced. Here, we start with a model of $\neg\mathrm{FAN}_\Delta$, and extend it by including paths that miss decidable (former) bars. If this is done to all decidable bars, FAN_Δ can be made to hold. If this is done gently enough, counter-examples to FAN_c will remain as counter-examples.

The other difference is in the techniques used. It is like a Kripke model built using Heyting-valued extensions of a realizability model. This is not the first time that some of these techniques have been combined (see [11] for references and discussion). This is the first time we are aware of that all three have been combined. Perhaps that in and of itself makes this work to be of some interest.

This work was started while the author was a fellow at the Isaac Newton Institute's fall 2015 program in the Higher Infinite. The author warmly thanks them for their support and hospitality during that time. Thanks are due also to Andrew Swan, a conversation with whom led to this work. Thanks go in addition to Francois Dorais and Noah Schweber for their input on Math Overflow about Francois's example of a c-bar which is not decidable.

2 Kripke Structures of Constructive Models

While the general theory of models of constructive systems is often itself presented constructively (for example in [7,8,12]), particular models are often built within a classical meta-theory, because essential use is made of classical constructions (as in [5] or [10], for instance). For Kripke models, that means, working classically, giving a partial order, and associating to each node a classical model (for the similarity type in question), along with a family of transition functions; this then determines a model of constructive logic. Our current setting is different.

A warning shot is given by the fact that the root of this model is, effectively, Kleene's recursive realizability model $\mathcal{M}_{K_1}$; to the degree we work within this model, we must work constructively, and not classically. This point could be finessed, though, by insisting that $\mathcal{M}_{K_1}$ was itself built within a classical theory.

More crucially, the structures at each node will be determined by Heyting-valued extensions of $\mathcal{M}_{K_1}$. These structures are no longer of the right type. A structure of the language of set theory would, among other things, determine, for objects a and b, whether $a \in b$. A Heyting-valued model, though, determines whether $\mathcal{H} \Vdash$ "$a \in b$", where $\mathcal{H}$ is a value in the Heyting algebra $\mathcal{T}$ in question.

To address these matters, a structure at a node will be, in addition to a Heyting-valued model, also a Heyting value $\mathcal{H}$ from the Heyting algebra, and we will let a formula be true there only if $\mathcal{H}$ forces it. To have forcing truth still be valid within this Kripke structure, we will allow an extension of a node to be determined in part by a strengthening of the Heyting value $\mathcal{H}$.

Before giving the formal definitions, we sketch the idea. We will need to iterate taking Heyting-valued extensions. By way of notation, $\mathcal{T}$ will be taken to be a typical complete Heyting algebra, to be consistent with the notation below. Some of these Heyting algebras $\mathcal{T}$ will show up only in some previously constructed Heyting-valued extension. So we assume we have a definable collection of Heyting algebras, say with definition $\phi(\mathcal{T})$, which IZF proves to be a set. To each node p will be associated a string $\langle(\mathcal{T}_0, \mathcal{O}_0), (\mathcal{T}_1, \mathcal{O}_1), ..., (\mathcal{T}_n, \mathcal{O}_n)\rangle$ such that each Heyting value $\mathcal{O}_i$ is not $\perp$ and forces that the next $\mathcal{T}_{i+1}$ is an allowable Heyting algebra, as given by ϕ. A child of p is determined by, optionally, extending some of the $\mathcal{O}_i$'s, and, optionally, including another pair $(\mathcal{T}_{n+1}, \mathcal{O}_{n+1})$ onto the string.

More formally, let $\phi(x)$ be a formula such that IZF proves "if $\phi(x)$ then x is a complete Heyting algebra, and ϕ is satisfied by only set-many objects." We will have occasion to consider ϕ as evaluated in a Heyting-valued extension, and so as applied to a term. Even if there are only set-many objects satisfying ϕ in this extension, there could still be class-many such terms. In order to keep this construction fully set-sized, we will implicitly allow only minimal terms $\mathcal{T}$ to be applied to ϕ. In more detail, suppose we assert in some context that $\phi(\mathcal{T})$. If this context is the ground model $\mathcal{M}_{K_1}$, then there are only set-many such $\mathcal{T}$'s within this model. Else the context will be some Heyting-valued extension, given by a Heyting value $\mathcal{H}$ – $\mathcal{H} \Vdash \phi(\mathcal{T})$ – within some other context. This latter context will then satisfy "for any other term t of rank less than that of $\mathcal{T}$, $\mathcal{H} \Vdash t \neq \mathcal{T}$". Similarly for the members of, or the Heyting values in, $\mathcal{T}$.

Definition 1. *Definition of the nodes, and their associated models, by induction on ω.*

The unique node of length 0 is the empty sequence $\langle\rangle$, with associated model $\mathcal{M}_{\langle\rangle} = \mathcal{M}_{K_1}$.

Inductively, given the set of nodes of length n, a node p of length $n+1$ will be a string of the form $\langle(\mathcal{T}_0, \mathcal{O}_0), (\mathcal{T}_1, \mathcal{O}_1), ..., (\mathcal{T}_n, \mathcal{O}_n)\rangle$ such that $p \upharpoonright n$ is a node, and, in $\mathcal{M}_{K_1}$, $\mathcal{O}_0 \Vdash$ "$\mathcal{O}_1 \Vdash$ ..."$\mathcal{O}_{n-1} \Vdash$ "$\phi(\mathcal{T}_n)$ and $\mathcal{O}_n \neq \perp$ is a Heyting value in the Heyting algebra $\mathcal{T}_n$"" ... ". The model $\mathcal{M}_p$ associated to p is the forcing *extension by $\mathcal{T}_n$, with truth determined by $\mathcal{O}_n$, as evaluated within the model for $p \restriction n$.*

We abbreviate the iterated forcing $\mathcal{O}_0 \Vdash$ "$\mathcal{O}_1 \Vdash$ ..."$\mathcal{O}_{n-1} \Vdash \psi$" ..." as $\langle \mathcal{O}_0, ..., \mathcal{O}_{n-1}\rangle \Vdash_H \psi$. The reason for the subscript H is to emphasize that this notion of truth is given by iterated forcing. In contrast, for example, truth in $\mathcal{M}_{K_1}$ is given by realizers, and will be written as $e \Vdash_r \psi$. One important instance of that will be iterated forcing over $\mathcal{M}_{K_1}$. So truth in the model given by forcing with $\mathcal{T}$ over $\mathcal{M}_{K_1}$ would be written as $e \Vdash_r$ "$\mathcal{O} \Vdash_H \psi$".

By our various conventions, there are only set-many nodes.

By way of notation, we will typically suppress mention of the $\mathcal{T}_i$'s, as they are implicit in the choice of the $\mathcal{O}_i$'s. The opens of a node will (at least sometimes) be written as $\mathcal{O}^p_i$, so that p of length n will be $\langle \mathcal{O}^p_0, ..., \mathcal{O}^p_{n-1}\rangle$.

Definition 2. *The partial order on the set of nodes. For q to be an extension of p, written $q \geq p$, q has to be at least as long as p, and, for i less than the length of p, $q \restriction i \Vdash_H \mathcal{O}^q_i \leq \mathcal{O}^p_i$. (For this to make sense, implicitly $q \restriction i \Vdash_H \mathcal{T}^q_i = \mathcal{T}^p_i$.)*

We leave it to the reader to show that this is indeed a partial order. Notice that an extension of a node is indicated with the standard notation for partial orders, $\geq$, in contrast with the strengthening of a Heyting value, which is indicated with the standard notation for forcing, $\leq$.

This p.o. is in $\mathcal{M}_{K_1}$. Since a model embeds into any Heyting-valued extension, the p.o. is also in any of the models associated with a node. Furthermore, consider the p.o. restricted to a node (i.e. the extensions of any node, including itself). This restriction is definable in the node's model, uniformly from the node. That is, given any node as a parameter, the node's model can figure out the rest of the p.o.

We will need the notion of a node being covered by a set of nodes, akin to an open set in a topological space being covered by a collection of open sets, or, more generally, a member of a Heyting algebra being (less that) the join of a subset of the algebra.

Definition 3. *We define p of length n being covered by $P = \{p_j \mid j \in J\}$ by induction on n.*

- *For $n = 0, \langle\rangle$ is covered by only $\{\langle\rangle\}$.*
- *For $n = 1, p$ of the form $\langle(\mathcal{T}, \mathcal{O})\rangle$ is covered by P if each p_j also has length 1, and $p_j \geq p$ (so p_j is of the form $\langle(\mathcal{T}, \mathcal{O}_j)\rangle$, the point being that $\mathcal{T}$ is the same), and $\{\mathcal{O}_j \mid j \in J\}$ covers $\mathcal{O}$ in the sense of $\mathcal{T}$: $\mathcal{O} \leq \bigvee\{\mathcal{O}_j \mid j \in J\}$.*
- *For a length $n + 1 > 1$, some conditions are immediate analogues: each p_j extends p in the Kripke order, and each p_j has length $n + 1$. Furthermore, letting $P \restriction n$ be $\{p_j \restriction n \mid j \in J\}$, we have that $P \restriction n$ is to cover $p \restriction n$. Finally, we want to view P as a term for a set in the model associated with $p \restriction n$. Recall that a term for a Heyting-valued model is an arbitrary collection of pairs $\langle \mathcal{O}, \sigma\rangle$, where $\mathcal{O}$ is a member of the Heyting algebra and σ is (inductively) a term. If we are considering a two-step iteration, then σ is (a term for) a*

pair $\langle\hat{\mathcal{O}},\tau\rangle$, where $\hat{\mathcal{O}}$ is a value from the second Heyting algebra. This can be abbreviated by $\langle(\mathcal{O},\hat{\mathcal{O}}),\tau\rangle$. Whereas each p_j is of the form $\langle\mathcal{O}_0,\ldots,\mathcal{O}_n\rangle$, it induces a set $alt-p_j := \langle(\mathcal{O}_0,\ldots,\mathcal{O}_{n-1}),\mathcal{O}_n\rangle$, which is a term, in the language for an n-fold forcing iteration, with value (forced to be) an open set in $\mathcal{T}_n$. Of course, the n-fold iteration in question is just the model associated with $p \restriction n$. So, letting P_n be $\{alt-p_j \mid j\in J\}$, $p\restriction n \Vdash_H P_n$ is a collection of open sets of $\mathcal{T}_n$. Our final condition is that $p\restriction n \Vdash_H P_n$ covers $p(n)$.

(At some point, we might need that the transitivity condition is satisfied: if P covers p, and each $p_j \in P$ is covered by P_j, then $\bigcup_{j\in J} P_j$ covers p.)

We are finally in a position to define the model. Working within $\mathcal{M}_{K_1}$, inductively on the ordinals α, we define the members $\mathcal{M}^p_\alpha$ of the model at node p of rank α (where we associate α with its canonical image in each of the associated models), along with the transition functions f_{pq} from $\mathcal{M}^p_\alpha$ to $\mathcal{M}^q_\alpha$. We will usually drop the subscripts and just write f as a polymorphic transition function. Similarly, we will not adorn f with any α, since the definition of f will be uniform in α. Do not confuse the associated models $\mathcal{M}_p$ from above with the $\mathcal{M}^p$ about to be defined.

Definition 4. *The universe $\mathcal{M}^p$ of the model at node p. First we define $\mathcal{M}^p_\alpha$ inductively on ordinals α. A member σ of $\mathcal{M}^p_\alpha$ is a function with domain the p.o. restricted to p (i.e. p and its extensions). Furthermore, $\sigma(q) \subseteq \bigcup_{\beta<\alpha}\mathcal{M}^q_\beta$. In order to fulfill the basic Kripke condition, if $\tau\in\sigma(q)$, and $r\geq q$, then $f(\tau)\in\sigma(r)$. If $q\geq p$, then $f(\sigma)=\sigma\restriction\mathcal{P}^{\geq q}$. Let $\mathcal{M}^p$ be $\bigcup_\alpha \mathcal{M}^p_\alpha$.*

(If you're wondering whether there are any such members, or whether instead the definition is vacuous, consider the constant function, with domain the entire tree, which always returning the empty set; this is an object of rank 0, and represents the empty set. The reader is invited to think through now what, inductively, represents any natural number n, and why there is (something playing the role of) ω within this formalism.)

Because this model has aspects of both a Kripke and a Heyting-valued model, it is in actuality neither. So to give the semantics, we cannot rely on any standard definition already extant in the literature. Rather, we have to give an independent, inductive definition of satisfaction. In this case we do not subscript $\Vdash$, because it is our main notion of truth; if we ever need to disambiguate, it will be written as $\Vdash_K$. Furthermore, we will refer to it as a Kripke model, because it is similar to one, and so we can distinguish it verbally from the various Heyting-valued models considered and from the realizability model in which this is all taking place.

Definition 5. *Implicitly in what follows, when we write "$p\Vdash\phi$", the parameters in ϕ are all in $\mathcal{M}^p$. Also implicit is the application of the transition function f, as need be. For future reference, $\Vdash$ and $\Vdash_K$ are the same thing.*

- *$p\Vdash\sigma\in\tau$ iff p is covered by some P, and for all $p_j\in P$ there is a $\sigma_j\in\tau(p_j)$ such that $p_j\Vdash\sigma=\sigma_j$.*

- *$p \Vdash \sigma = \tau$ iff for all $q \geq p$ and all $\rho \in \sigma(q)$, $q \Vdash \rho \in \tau$, and vice versa.*
- *$p \Vdash \phi \wedge \psi$ iff $p \Vdash \phi$ and $p \Vdash \psi$.*
- *$p \Vdash \phi \vee \psi$ iff p is covered by some Q, and for each $q \in Q$ either $q \Vdash \phi$ or $q \Vdash \psi$.*
- *$p \Vdash \phi \rightarrow \psi$ iff for all $q \geq p$ if $q \Vdash \phi$ then $q \Vdash \psi$.*
- *$p \Vdash \perp$ never.*
- *$p \Vdash \forall x\ \phi(x)$ iff for all $q \geq p$ and $\sigma \in \mathcal{M}^q$ $q \Vdash \phi(\sigma)$.*
- *$p \Vdash \exists x\ \phi(x)$ iff p is covered by some Q and for all $q \in Q$ there is some σ such that $q \Vdash \phi(\sigma)$.*

Now that we have a semantics, we can state what we would like the semantics to do for us. Although, as we already emphasized, we cannot rely on established theorems to tell us anything about our $\Vdash$, since the techniques used are a mixture of standard methods for building models of constructivism, it should come as no surprise that we have such a model. Since all of these results are only to be expected, the proofs are omitted.

Lemma 1. *Each node satisfies the equality axioms.*

Lemma 2. *If Q covers p, and for each $q \in Q$ we have $q \Vdash \phi$, then $p \Vdash \phi$.*

Proof. By a straightforward induction on ϕ.

Corollary 3. *Each node satisfies constructive logic.*

Theorem 4. *This structure models IZF.*

3 FAN$_{\Delta}$ Does Not Imply FAN$_c$

For the moment, we will work simply under IZF.

Our primary task is now to define the right ϕ, the class of Heyting algebras we will use to build the nodes. They will be induced by the possible counterexamples B to FAN$_{\Delta}$: B is a decidable set of binary strings, but is not uniform. It is safe to assume that B is closed upwards. Mostly we're interested in when B is in addition a bar, there famously being such a creature in Kleene's recursive realizability model. The reason that we do not include being a bar in this definition is that would then be another condition to check before being able to use B. This is more than just a matter of convenience, or saving a little work. When we're working within a Heyting-valued extension of a realizability model, say, different conditions might decide whether B is a bar differently, and if B had to be a bar then we'd need to find an infinite path through those conditions along which B became a bar, meaning either there is such a path, or we'd have to find a non-uniform bar forcing such a path, and all of a sudden the thicket starts to look impenetrable. Although it seems unaesthetic to force paths that we really don't need, this is a small price to pay for having a theorem with a proof.

Let T be the complement of B. So T is a decidable, infinite tree. We will generically shoot a branch through T.

We will define a formal topology S from T. To help make this paper self-contained, we present a definition of a formal topology. Such definitions are not uniform in the literature. Here we will use the one from [8], Sect. 2.1.

Definition 6. *A formal topology is a poset $(S, \leq)$ and a relation $\lhd$ between elements and subsets of S. (One should think of the elements of S as open sets, with $\leq$ as containment and $\lhd$ as covering.) The axioms are:*

- *if $a \in p$ then $a \lhd p$,*
- *if $a \leq b$ and $b \lhd p$ then $a \lhd p$,*
- *if $a \lhd p$ and $\forall x \in p\ x \lhd q$ then $a \lhd q$, and*
- *if $a \lhd p$ and $a \lhd q$ then $a \lhd \downarrow p \cap \downarrow q$,*

where $\downarrow p$ is the downward closure of p.

Definition 7. *The formal topology induced by B:*

Let B be a decidable, upwards-closed, non-uniform set of binary strings, and T its complement in $2^{<\omega}$. A member of S is a union of finitely many basic members of S. A basic member of S, $\mathcal{O}_\sigma$, is given by a node $\sigma \in T$, and is the set of all nodes in T compatible with σ, that is, all initial segments and extensions, when it is infinite. A witness that $\mathcal{O} \in S$, that is, a finite set Σ such that $\mathcal{O} = \bigcup_{\sigma \in \Sigma} \mathcal{O}_\sigma$, is called a base *for $\mathcal{O}$; note that bases are not unique. The partial order $\leq$ on S is just the subset relation $\subseteq$.*

A subset $\mathcal{U}$ of S covers $\mathcal{O} \in S$, $\mathcal{O} \lhd \mathcal{U}$, if it is not the case that there is no finite length n such that, for all $\sigma \in T$ of length n, either $\sigma \notin \mathcal{O}$ or, for some initial segment τ of σ and for some $\mathcal{O}_\mathcal{U} \in \mathcal{U}$, we have $\mathcal{O}_\tau \subseteq \mathcal{O}$ and $\mathcal{O}_\tau \subseteq \mathcal{O}_\mathcal{U}$. In symbols, $\mathcal{U}$ covers $\mathcal{O}$ iff

$$\neg\neg\exists n\ \forall\sigma \in T\ \ |\ \sigma\ | = n \rightarrow (\sigma \notin \mathcal{O} \vee \exists\tau \subseteq \sigma\ \exists\mathcal{O}_\mathcal{U} \in \mathcal{U}\ \mathcal{O}_\tau \subseteq (\mathcal{O} \cap \mathcal{O}_\mathcal{U})).$$

For any such n, we say that $\mathcal{U}$ covers $\mathcal{O}$ by length n.

Remarks: By choosing the base to be empty, $\emptyset \in S$.

If the set of nodes compatible with σ is finite, then σ does not determine an open set; alternatively, we could allow the induced set to be open, and it will be covered by the empty set.

For $\sigma \in T$ and $\mathcal{O} \in S$ it is decidable from a base for $\mathcal{O}$ whether $\sigma \in \mathcal{O}$.

Note that if $\mathcal{U}$ covers $\mathcal{O}$ by n then $\mathcal{U}$ covers $\mathcal{O}$ by any $k \geq n$. The reason for the double-negation in the definition of covering should become clear, when it is used, in Theorems 6 and 7.

Proposition 5. $(S, \leq, \lhd)$ *from above constitutes a formal topology.*

The reason for this formal topology is so that we can take the Heyting-valued model $\mathcal{M}_T$ over it.

We do not know whether the next theorem is true in general (meaning provable in IZF). So for the moment, we work in the recursive realizability model. That is, the model $\mathcal{M}_T$ is taken as being built within it.

Theorem 6. *Working within the recursive realizability model, in $\mathcal{M}_T$, the generic G is (identifiable with) an infinite branch through T.*

Proof. We can identify the generic G with $\{\langle \mathcal{O}_\sigma, \tau\rangle \mid \tau \subseteq \sigma, \mathcal{O}_\sigma$ a basic open set$\}$. We want to show that $\mathcal{O}_\emptyset \Vdash_H$ "for all k there is a unique σ of length k with $\sigma \in G$." Since the natural numbers in the sense of $\mathcal{M}_T$ can be identified with those of $\mathcal{M}_{K_1}$, which are themselves just those of V, it suffices to fix a k in the sense of V. It is easy to see that if $\mathcal{O}_\sigma$ is a basic open set with σ of length k then $\mathcal{O}_\sigma \Vdash_H$ "σ is the unique member of G of length k." Let $\mathcal{U}$ be $\{\mathcal{O}_\sigma \mid \sigma$ has length k and $\mathcal{O}_\sigma$ is a basic open set$\}$. It suffices to show that $\mathcal{U}$ covers $\mathcal{O}_\emptyset$.

Because of the double negation in the definition of covering, when showing that $\mathcal{U}$ covers $\mathcal{O}_\emptyset$ it is not necessary to get the n as a computable function of k; rather, any realizer will do. So it's just a matter of finding an n in the ground model V such that the rest (of the definition of covering) is easily seen to be forced. Toward this end, let n be large enough so that, whenever T beneath σ of length k is finite, T contains no descendants of σ of length n. In other words, go through level k of T, take all those nodes whose subtrees will eventually die, of which there are only finitely many, and then go out far enough that all of them have died already. Now given a node τ of T of length n, $\tau \restriction k$ and $\mathcal{O}_{\tau\restriction k}$ are the desired witnesses.

The preceding lemma will help us with the analysis of the Kripke model at nodes of length 1. Of course, we need to consider longer nodes too. Hence we must prove the corresponding lemma for base models a finite iteration of these extensions.

Theorem 7. *Let p be some node of the Kripke model, with associated model $\mathcal{M}_p$. Suppose that, within $\mathcal{M}_p$, T is an infinite, decidable tree. Then, working within $\mathcal{M}_p$, in $\mathcal{M}_T$, the generic G is (identifiable with) an infinite branch through T.*

Proof. For ease of exposition, we take p to be $\langle (T_0, \mathcal{O}_{\langle\rangle})\rangle$; that is, to have length 1 and to have the open set be the entire tree. We leave the general case to the reader.

By way of notation, let $\mathcal{O}^0$ refer to open sets in the formal topology in $\mathcal{M}_{K_1}$ induced by T_0, and $\mathcal{O}^1$ refer to open sets in the formal topology in $\mathcal{M}_p$ for T. As in the previous theorem, the natural numbers of $\mathcal{M}_T$ can be identified with those of V. So let k be a natural number, and in $\mathcal{M}_p$ let $\mathcal{U}^1$ be $\{\mathcal{O}^1_\sigma \mid \sigma$ has length k and $\mathcal{O}^1_\sigma$ is a basic open set$\}$. As before, it suffices to show that $\mathcal{U}^1$ covers $\mathcal{O}^1_\emptyset$; actually, we must show that $\mathcal{O}^0_\emptyset \Vdash_H$ "$\mathcal{U}^1$ covers $\mathcal{O}^1_\emptyset$"; actually, what we really must show is that there is some e which realizes the above forcing assertion, uniformly in k.

Recall that "$\mathcal{U}^1$ covers $\mathcal{O}^1_\emptyset$" is an abbreviation of "not not there is an n such that $\mathcal{U}^1$ covers $\mathcal{O}^1_\emptyset$ by n." So we must realize "for every $\mathcal{O}^0_\sigma$ extending $\mathcal{O}^0_\emptyset, \mathcal{O}^0_\sigma \not\Vdash_H$ there is no n such that $\mathcal{U}^1$ covers $\mathcal{O}^1_{\langle\rangle}$ by n." Toward that end, suppose $f \Vdash_r \mathcal{O}^0_\sigma$ is an open set of the space T_0. Then we must have chosen e so that $\{e\}(k, f) \Vdash_r$ "$\mathcal{O}^0_\sigma \not\Vdash_H$ there is no n such that $\mathcal{U}^1$ covers $\mathcal{O}^1_{\langle\rangle}$ by n."

By the realizability semantics, everything realizes a negation, as long as nothing realizes the statement being negated. So we must show, in V, that nothing realizes "$\mathcal{O}^0_\sigma \Vdash_H$ there is no n such that $\mathcal{U}^1$ covers $\mathcal{O}^1_{\langle\rangle}$ by n."

Suppose, toward a contradiction, that g does realize that statement. Unpacking the semantics further, $g \Vdash_r$ "for every extension $\mathcal{O}^0_\tau$ of $\mathcal{O}^0_\sigma$, $\mathcal{O}^0_\tau \nVdash$ there is an n such that $\mathcal{U}^1$ covers $\mathcal{O}^1_{\langle\rangle}$ by n." Let $h \Vdash_r \mathcal{O}^0_\tau$ extends $\mathcal{O}^0_\sigma$. Then $\{g\}(h) \Vdash_r$ "$\mathcal{O}^0_\tau \nVdash$ there is an n such that $\mathcal{U}^1$ covers $\mathcal{O}^1_{\langle\rangle}$ by n." The only way to realize a negation is if nothing realizes the statement being negation. So nothing realizes "$\mathcal{O}^0_\tau \Vdash$ there is an n such that $\mathcal{U}^1$ covers $\mathcal{O}^1_{\langle\rangle}$ by n." We will have our desired contradiction once we find a $\mathcal{O}^0_\tau$ extending $\mathcal{O}^0_\sigma$, an integer n and a realizer of "$\mathcal{O}^0_\tau \Vdash \mathcal{U}^1$ covers $\mathcal{O}^1_{\langle\rangle}$ by n."

We will construct a finite sequence of basic open sets of T_0, starting with $\mathcal{O}^0_\sigma$ and ending with the desired $\mathcal{O}^0_\tau$. The steps of this procedure are indexed by the binary sequences of length k. At each step, indexed by say ρ, extend the current open set if possible to force T beneath ρ to be finite, and then again to force a level n_ρ witnessing this finiteness (i.e. that ρ has no extension in T of length n_ρ); whenever this is not possible, the open set at hand already forces T beneath ρ to be infinite. Let n be the maximum of the n_ρ's. Working beneath $\mathcal{O}^0_\tau$, if π of length n is ever forced to be in T, then $\pi \restriction k$ and $\mathcal{O}^1_{\pi\restriction k}$ will be as desired.

Corollary 8. *For p a node in the Kripke partial order, with final entry $(T, \mathcal{O})$, and B the complement of T, $p \Vdash_K B$ is not a bar.*

Proof. Let G be the generic for forcing with T. The function (with domain the partial order from p onwards) with constant output G (more accurately, the canonical image of G in the input's associated model) witnesses that B is not a bar.

The next two theorems finish this paper.

Theorem 9. $\mathcal{M} \models FAN_\Delta$.

Proof. The idea is simple enough. If, at a node, B is forced to be a decidable bar, then B must also be forced to be uniform, because, if not, the node would have an extension given by forcing with the complement of B, showing that B could not have been a bar. We need to check the details though, to guard against things like the use of classical logic and to make sure we're using the semantics of the model at hand. For better or worse, I know of no other way to do this than to unravel the statement to be shown, using the semantics given.

We need to show $\langle\rangle \Vdash \mathrm{FAN}_\Delta$, working within $\mathcal{M}_{K_1}$, meaning we must find a realizer e for the statement $\langle\rangle \Vdash \mathrm{FAN}_\Delta$. As a reminder, $\langle\rangle$ is the empty sequence, the bottom node in the partial order underlying the model. For reference, FAN_Δ is the assertion "for all B, if B is an upwards-closed decidable bar (in $2^{<\omega}$), then B is uniform, i.e. there is a natural number n such that all binary sequences of length n are in B."

The Hypothesis: Unpacking the meaning of $\Vdash$, we need to show that within $\mathcal{M}_{K_1}$, if $B \in \mathcal{M}^p$ then $p \Vdash$ "if B is such a bar then B is uniform." That means that if $t \Vdash_r p$ is a node and $B \in \mathcal{M}^p$ then $\{e\}(t) \Vdash_r$ "$p \Vdash$ "if B is such a bar then B is uniform"." To save on notation, we will suppress mention of t. This means that we must show $e \Vdash_r$ "for all $q \geq p$, if $q \Vdash B$ is such a bar then $q \Vdash B$ is uniform." Again suppressing the realizer that $q \geq p$, we must show $e \Vdash_r$ "if $q \Vdash B$ is such a bar then $q \Vdash B$ is uniform." So, suppose $f \Vdash_r$ "$q \Vdash B$ is such a bar;" we must have that $\{e\}(f) \Vdash_r$ "$q \Vdash B$ is uniform."

Unpacking some more, there is a realizer g, easily computable from f, with $g \Vdash_r$ "$q \Vdash B$ is decidable;" and that means $g \Vdash_r$ "$q \Vdash$ for all $\sigma \in 2^{<\omega}$, either $\sigma \in B$ or $\sigma \notin B$." Since $2^{<\omega}$ does not change from node to node, that means $g \Vdash_r$ "for all $\sigma \in 2^{<\omega}, q \Vdash$ (either $\sigma \in B$ or $\sigma \notin B$)." Identifying a realizer that $\sigma \in 2^{<\omega}$ with σ itself, that becomes "for all $\sigma \in 2^{<\omega}$ $\{g\}(\sigma) \Vdash_r q \Vdash (\sigma \in B \vee \sigma \notin B)$." And that means that "for all $\sigma \in 2^{<\omega}$ there is a Q_σ such that $\{g\}(\sigma) \Vdash_r$ (Q_σ covers q and for each $r \in Q_\sigma$ either $r \Vdash \sigma \in B$ or $r \Vdash \sigma \notin B$)."

The Conclusion: Having just unpacked the hypothesis, we will now analyze the conclusion. Recall what we need to show: $\{e\}(f) \Vdash_r$ "$q \Vdash B$ is uniform;" i.e. $\{e\}(f) \Vdash_r$ "$q \Vdash$ there is a bound n witnessing that B is uniform;" which is $\{e\}(f) \Vdash_r$ "there is a cover R of q, and for all $r \in R$ there is some object n such that $r \Vdash n$ is a natural number witnessing the uniformity of B." That comes down to the existence of a set R such that $\{e\}(f) \Vdash_r$ "R covers q, and for all $r \in R$ there is some object n such that $r \Vdash n$ is a natural number witnessing the uniformity of B."

Since this is complicated enough, we're now going to build up somewhat slowly. We will examine several cases, based on the length of q. Since e has access to a realizer that q is a node, e has access to q's length, and so can make this case distinction.

Case I: For the first pass, suppose $q = \langle\rangle$. Every element in the cover Q_σ must have the same length as the thing covered, which in this case is 0, so $Q_\sigma = \{\langle\rangle\}$. So we have $\{g\}(\sigma) \Vdash_r$ (either $\langle\rangle \Vdash \sigma \in B$ or $\langle\rangle \Vdash \sigma \notin B$)." Similarly for R: we must have $\{e\}(f) \Vdash_r$ "there is some n such that $\langle\rangle \Vdash n$ is a natural number witnessing the uniformity of B." The obvious algorithm to find a uniform bound for B is to run through the various $\{g\}(\sigma)$'s until one finds such a bound. All there is left to do in this case is to show that this algorithm terminates. If not, then every K_1 realizer will realize that B is not uniform. So, in $\mathcal{M}_{K_1}$, letting T be the complement of B, $\langle(S_T, \mathcal{O}_{\langle\rangle})\rangle$ is a node (where S_T is the formal topology induced by T). By the corollary, $\langle(S_T, \mathcal{O}_{\langle\rangle})\rangle \Vdash$ "B is not a bar," contradicting the assumption that $f \Vdash_r$ "$\langle\rangle \Vdash B$ is (such) a bar."

Case II: For our second pass, suppose that q has length 1. So the notion that Q_σ as a set of nodes covers q as a node reduces to covering in the sense of the Heyting algebra S_T, where $q = \langle(S_T, \mathcal{O})\rangle$. Letting $\mathcal{U}_\sigma$ be the set of second components of the members of Q_σ (actually, a member of Q_σ being a sequence of length 1, we are identifying such a sequence with its sole entry), $\{g\}(\sigma)$ yields a realizer that $\mathcal{U}_\sigma$ covers $\mathcal{O}$ in the sense of S_T. Now recall that the definition of covering begins with a double negation; unpacking what it is to realize a double negation, since something realizes that $\mathcal{U}_\sigma$ covers $\mathcal{O}$, everything does; and there is a realizer that $\mathcal{U}_\sigma$ covers $\mathcal{O}$ by some witnessing length n, but those latter two items are not uniformly computable from $\{f\}(\sigma)$.

We need to define a set R with which we can work easily. Towards this end, say that a binary sequence ρ is sufficiently long if it is at least as long as each σ in the given fixed base Σ for $\mathcal{O}$. (That is, e computes a realizer that q is a node. Being a node includes that $\mathcal{O}$ is open in S_T, meaning that $\mathcal{O}$ has a finite base, which is then witnessed by the realizers at hand.) By way of notation, for $r \geq q$ of length 1, r will be $\langle (S_T, \mathcal{O}_r)\rangle$. In V, let R be $\{\langle h, r\rangle \mid h \Vdash_r$ "$r \geq q, \mathcal{O}_r = \mathcal{O}_\rho \leq \mathcal{O}, \rho$ is sufficiently long, and there is an $n \leq\mid \rho \mid$ such that $r \Vdash n$ is a natural number witnessing the uniformity of B"$\}$. (The reason to insist that $r \leq q$ is that, in order for r to force B to be uniform, r already has to force that B is a set of binary strings; recalling that our assumption is no more than that q forces B to be a bar, we remain on the safe side by working beneath q.)

From a realizer that r is in R, it is easy (and uniform) to compute an n and a realizer that r forces n to witness the uniformity of B. So there is a uniform realizer, not even depending on f, of the second part of what $\{e\}(f)$ is supposed to realize. Take such a realizer for the second part of $\{e\}(f)$. As for the first part of $\{e\}(f)$, it is supposed to realizes that R covers q. Working in $\mathcal{M}_{K_1}$, let $\mathcal{U}$ be $\{\mathcal{O}_r \mid r \in R\}$. We need to realize that $\mathcal{U}$ covers $\mathcal{O}$.

Recall that the notion of covering begins with a double negation. So if anything realizes that, then everything does. Hence we need only the existence of an n and a realizer that n witnesses that $\mathcal{U}$ covers $\mathcal{O}$ by length n; n and the realizer do not have to be computed.

In V, either there are such an n and a realizer, or there aren't. If there are, we are done (with this case of q having length 1). If not, then we would like to force with the complement of $\mathcal{U}$. Now, it does us little good to do such forcing over $\mathcal{M}_{K_1}$; rather, we need to work beneath q, or, more accurately, in $\mathcal{M}_T$. Toward this end, let B_R be the term in $\mathcal{M}_{K_1}$ for the $\mathcal{M}_T$-set which is (the canonical embedding of) $\{\rho \mid \mathcal{O}_\rho \in \mathcal{U} \vee \rho \notin \mathcal{O}\}$. We would like B_R to induce an extension of $\mathcal{M}_T$. That is, for the complement T_R of B_R, we would like that $q^\frown\langle (S_{T_R}, \mathcal{O}_{\langle\rangle})\rangle$ is an extension of q. (Please note that $T_R = \mathcal{O}\backslash\{\rho \mid \mathcal{O}_\rho \in \mathcal{U}\}$.) We will argue later why that essentially does it for this case; in the meantime, our intermediate goal is to show that $\mathcal{O} \Vdash_H$ "B_R is a decidable, upwards closed set of binary strings, which is not uniform."

Easily, B_R is forced to be an upwards closed set of binary strings.

Turning to B_R being decidable, it is not clear that it (rather, its $\mathcal{M}_{K_1}$ version) is so in $\mathcal{M}_{K_1}$. After all, the definition of B_R depends on R, which itself depends on B; even though p forces B to be decidable, it is not clear that translates to an algorithm for deciding facts about forcing the uniformity of B. Serendipitously, we do not need decidability of B_R in $\mathcal{M}_{K_1}$, but only in $\mathcal{M}_T$ (which, recall, is q's associated model), as forced by $\mathcal{O}$. We will be able to leverage the difference between decidability in $\mathcal{M}_{K_1}$ and in $\mathcal{M}_T$ to get the latter.

Unpacking the assertion $\mathcal{O} \Vdash_H$ "B_R is decidable," we must show that for all $\rho \in 2^{<\omega}$, $\mathcal{O} \Vdash \rho \in B_R \vee \rho \notin B_R$. And that means that for all such ρ there is a cover R_ρ of $\mathcal{O}$ such that, for all $\mathcal{O}_s \in R_\rho$, either $\mathcal{O}_s \Vdash \rho \in B_R$ or $\mathcal{O}_s \Vdash \rho \notin B_R$.

Given such a ρ, first determine whether $\rho \in \mathcal{O}$, using the decidability of T and the finite base of $\mathcal{O}$. If so, then determine whether ρ is sufficiently long. If so, consider all binary σ of the same length as ρ. Using the meet operation of the Heyting-algebraic structure of S_T, applied to the (finitely many) $\mathcal{U}_\sigma$'s, there is a cover Q of $\mathcal{O}$ each member of which decides B of level $\mid \rho \mid$ (i.e. decides membership in B of all strings of the same length as ρ). It is not hard to see that this Q suffices for our R_ρ.

With decidability out of the way, we now show that $\mathcal{O} \Vdash_H B_R$ is not uniform. That is, we are interested in evaluating whether B_R is uniform in $\mathcal{M}_T$. That means that for no $\hat{\mathcal{O}} \leq \mathcal{O}$ do we have that $\hat{\mathcal{O}} \Vdash B_R$ is uniform. Suppose to the contrary we had such an $\hat{\mathcal{O}}$. Then any witness to the uniformity of B_R is then a witness to the uniformity of $\mathcal{U}$ as a cover of $\mathcal{O}$, which is assumed not to exist. So $\mathcal{O} \Vdash_H B_R$ is not uniform.

So we have an extension node of q, achieved by forcing over T_R. That induces a generic path through T_R. From this path, we can interpret B as a non-uniform tree. B is already assumed to be forced to be upwards closed and decidable. So now B induces an extension in the Kripke partial order. This gets a path avoiding B. So B cannot be forced to be a bar. This contradiction finishes the proof of this case.

Case III: It is time to finish the proof of this theorem. So, let q be a node. If need be, we will work inductively on the length of q, so q can be taken to have length at least 1. To recall, we have a realizer f that $q \Vdash B$ is an upwards closed decidable bar. We are searching for a realizer that $q \Vdash B$ is uniform. As above, we will need a set R such that it is easily realizable that R covers q, and for all $r \in R$ there is an n such that $r \Vdash n$ witnesses the uniformity of B. By our notational conventions, we can omit mention of the spaces $\mathcal{T}_i$ in q, and consider q to be $\langle \mathcal{O}^q_0, \ldots, \mathcal{O}^q_m \rangle (m \geq 0)$; similarly for extensions of q. This means that nodes can also be taken to be iterated forcing conditions, thereby eliminating the need for notation to translate between nodes and their corresponding conditions.

For $r \leq q$ as a forcing condition, we say that r is in normal form if (for all i) $r \restriction i \Vdash_H \mathcal{O}^r_i = \mathcal{O}_{\rho_i}$, for some specific $\rho_i \in 2^{<\omega}$. Let the length of r be the maximum of m and the length of each ρ_i occurring in r. Let R be $\{\langle h, r \rangle \mid h \Vdash_r$ "r is in normal form, and for some n at most the length of r, $r \Vdash n$ witnesses the uniformity of B"$\}$. All that remains to do is to realize that R covers q, the rest of what needs showing being trivial. For that, we show by induction on n up to $m+1$, that $R \restriction n$ covers $q \restriction n$.

For $n = 0$, this is just that R is non-empty. This is a simple case, which can be handled by methods similar to those that follow, and so is left to the reader. For $n \neq 0$, we need to show that $q \restriction n \Vdash_H R_n$ covers $q(n)$, i.e. $q \restriction n \Vdash_H$ "not there is a witness k that R_n covers $q(n)$ by length k." Toward that end, let r extend $q \restriction n$ (in the iterated forcing $\mathcal{T}_0 \times \ldots \times \mathcal{T}_{n-1}$). We need to show that $r \not\Vdash_H$ "there is no such witness k." Aiming toward a contradiction, suppose instead that $r \Vdash_H$ "there is no such witness k."

Now viewing r and q as nodes in the Kripke partial order, let $r \vee q$ be the least common extension of r and q in this p.o. We would like to show that $r \vee q \Vdash_K$

"there is no such k" (i.e. in the sense of the Kripke model). Let $s \geq_K r \vee q$. Assume toward a contradiction that $s \Vdash_K$ there is such a k. Since s is covered by a set forcing a particular such witness, we can assume that s itself forces that R_n covers $q(n)$ by length k. Now extend s in the Kripke p.o. (to, by abuse of notation, s) to decide on membership in $q(n)$ of each σ of length k, and, for each such σ forced to be in $q(n)$, witnesses $\tau \subseteq \sigma$ and $\mathcal{O} \in R_n$ to the covering property. Keeping in mind that $q(n)$ and R_n are sets in the model associated with $q \restriction n$, these same facts about R_n and $q(n)$ are therefore forced by $s \restriction n$ in the sense of iterated forcing. Since $s \restriction n \leq r$, this is the desired contradiction. Hence we conclude that $r \vee q \Vdash_K$ "there is no such k."

Let B_{R_n} be a term (in $r \vee q$'s associated model) for $\{\rho \mid \mathcal{O}_\rho \in R_n \text{ or } \rho \notin q(n)\}$. We would like to show that $r \vee q \Vdash_H$ "B_{R_n} is a decidable, upwards closed, non-uniform set of binary strings." Once we do that, B_{R_n} induces a Heyting-valued extension, in which B is not witnessed to be uniform. So then B induces a Heyting-valued extension in which it is not a bar. This is the ultimate contradiction toward which we were aiming.

It is easy to see that B_{R_n} is forced to be an upwards closed set of binary strings. For the non-uniformity, use the previous result that there is no witness k to R_n covering $q(n)$.

All we need left to do in this entire[1] proof is to show that B_{R_n} is forced to be decidable. As in the previous case (for q of length 1), for any finite binary string ρ, we must find a set S covering $r \vee q$ such that each $s \in S$ forces either $\rho \in B_{R_n}$ or $\rho \notin B_{R_n}$. As before, cover $r \vee q$ with a set, each member of which decides on a finite base for q, as well as on B up to the level of the length of ρ. This suffices.

Theorem 10. *$\mathcal{M} \models \neg FAN_c$.*

Proof. Recall that a c-fan is based on a decidable set of C, which can be taken to be a computable assignment of "in" and "out" to all the nodes. A node is in the bar if it and all of its successors are assigned "in", and out of the bar, or in the tree, if one of its successors is "out".

Consider the following c-fan, due to Francois Dorais. Let K be some complete c.e. set, with enumeration K_s (K at stage s). Let C be such that all nodes on level n are labeled "in" except for the unique node consistent with K_n (i.e. convert K_n into a characteristic function). It is easy to see that the characteristic function of K, if it exists, is the unique branch missing the induced c-set B, which is a bar in the realizability model, because, as is well known, K's characteristic function does not exist in that model. We must, and need only, show that B remains a bar in $\mathcal{M}$, the idea being that generically K is not added by forcing to the model.

Suppose toward a contradiction that $p \Vdash K$ witnesses that B is not a bar, in that K is an infinite path avoiding B. What that comes down to is that for every natural number n there is a covering Q_n of p such that every $q \in Q$

[1] If the reader is feeling any frustration at the length and detail of this proof, it might amuse them to know that in an earlier draft, the spot above contained an expletive.

forces a unique binary string of length n to be in K and that all such strings in K cohere. It is easy to see that p could not be $\langle\rangle$, because $\langle\rangle$ is covered by only $\{\langle\rangle\}$, at which point K could be externalized from $\mathcal{M}^{\langle\rangle}$ to $\mathcal{M}_{K_1}$, which as already remarked is not the case.

More generally, when $q \Vdash \sigma \in K$, then σ really is an initial part of K's characteristic function. After all, if not, then at some stage s an element k is enumerated into K such that $\sigma(k) = 0$. So all descendants of σ of length $t \geq s$ are in C, and hence in the bar B. That would mean that q could not force an extension of σ of length s to be in K, contradicting K being forced to be an infinite path. Hence any such σ in an initial segment of K. That means that all such σ's among all such q's cohere. By taking their union, the characteristic function of K can be built in $\mathcal{M}_{K_1}$.

References

1. Beeson, M.: Foundations of Constructive Mathematics. Springer, Heidelberg (1985)
2. Berger, J.: The logical strength of the uniform continuity theorem. In: Beckmann, A., Berger, U., Löwe, B., Tucker, J.V. (eds.) CiE 2006. LNCS, vol. 3988, pp. 35–39. Springer, Heidelberg (2006). https://doi.org/10.1007/11780342_4
3. Berger, J.: A separation result for varieties of Brouwer's Fan Theorem. In: Proceedings of the 10th Asian Logic Conference (ALC 10), Kobe University in Kobe, Hyogo, Japan, 1–6 September 2008, pp. 85–92 (2010)
4. Diener, H., Loeb, I.: Sequences of real functions on [0, 1] in constructive reverse mathematics. Ann. Pure Appl. Logic **157**(1), 50–61 (2009)
5. Diener, H., Lubarsky, R.: Notions of cauchyness and metastability. In: Artemov, S., Nerode, A. (eds.) Symposium on Logical Foundations in Computer Science 2018, LNCS. Springer, Heidelberg (2018, to appear)
6. Fourman, M.P., Hyland, J.M.E.: Sheaf models for analysis. In: Fourman, M., Mulvey, C., Scott, D. (eds.) Applications of Sheaves. LNM, vol. 753, pp. 280–301. Springer, Heidelberg (1979). https://doi.org/10.1007/BFb0061823
7. Fourman, M.P., Scott, D.S.: Sheaves and logic. In: Fourman, M., Mulvey, C., Scott, D. (eds.) Applications of Sheaves. LNM, vol. 753, pp. 302–401. Springer, Heidelberg (1979). https://doi.org/10.1007/BFb0061824
8. Gambino, N.: Heyting-valued interpretations for Constructive Set Theory. Ann. Pure Appl. Logic **137**, 164–188 (2006)
9. Julian, W., Richman, F.: A uniformly continuous function on [0,1] that is everywhere different from its infimum. Pac. J. Math. **111**(2), 333–340 (1984)
10. Lubarsky, R., Diener, H.: Separating the Fan Theorem and its weakenings. J. Symbolic Logic **79**, 792–813 (2014)
11. van Oosten, J.: Realizability: An Introduction to its Categorical Side. Elsevier, Amsterdam (2008)
12. Troelstra, A.S., van Dalen, D.: Constructivism in Mathematics, Vol. 2. In: Studies in Logic, vol. 123, Elsevier (1988)

Dialectica Categories for the Lambek Calculus

Valeria de Paiva[1(✉)] and Harley Eades III[2]

[1] Nuance Communications, Sunnyvale, CA, USA
valeria.depaiva@gmail.com
[2] Computer Science, Augusta University, Augusta, GA, USA
harley.eades@gmail.com

Abstract. We revisit the old work of de Paiva on the models of the Lambek Calculus in dialectica models making sure that the syntactic details that were sketchy on the first version got completed and verified. We extend the Lambek Calculus with a κ modality, inspired by Yetter's work, which makes the calculus commutative. Then we add the of-course modality !, as Girard did, to re-introduce weakening and contraction for all formulas and get back the full power of intuitionistic and classical logic. We also present the categorical semantics, proved sound and complete. Finally we show the traditional properties of type systems, like subject reduction, the Church-Rosser theorem and normalization for the calculi of extended modalities, which we did not have before.

Keywords: Lambek calculus · Dialectica models
Categorical semantics · Type theory · Structural rules
Non-commutative · Linear logic

1 Introduction

Lambek introduced his homonymous calculus (originally called the 'Syntactic Calculus') for proposed applications in Linguistics. However the calculus got much of its cult following and reputation by being a convenient, well-behaved prototype of a Gentzen sequent calculus without any structural rules.

This note recalls a Dialectica model of the Lambek Calculus presented by the first author in the Amsterdam Colloquium in 1991 [10]. Here, like then, we approach the Lambek Calculus from the perspective of Linear Logic, so we are interested in the basic sequent calculus with no structural rules, except associativity of tensors. In that earlier work we took for granted the syntax of the calculus and only discussed the exciting possibilities of categorical models of linear-logic-like systems. Many years later we find that the work on models is still interesting and novel, and that it might inform some of the most recent work on the relationship between categorial grammars and notions of distributional semantics [8].

Moreover, using the Agda proof assistant [6] we verify the correctness of our categorical model (Sect. 5.1), and we add the type theoretical (Sect. 6) notions that were left undiscussed in the Amsterdam Colloquium presentation. All of

S. Artemov and A. Nerode (Eds.): LFCS 2018, LNCS 10703, pp. 256–272, 2018.
https://doi.org/10.1007/978-3-319-72056-2_16

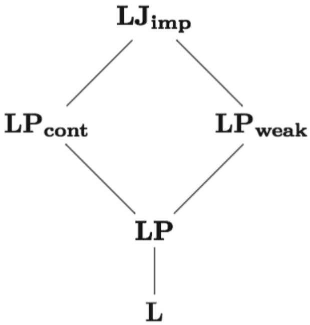
the syntax in this paper was checked using Ott [23]. The goal is to show that our work can shed new light on some of the issues that remained open. Mostly we wanted to check the correctness of the semantic proposals put forward since Szabo's seminal book [24] and, for future work, on the applicability and fit of the original systems to their intended uses.

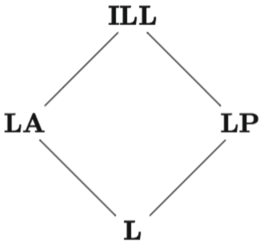
Overview. The Syntactic Calculus was first introduced by Joachim Lambek in 1958 [19]. Since then the rechristened Lambek Calculus has had as its main motivation providing an explanation of the mathematics of sentence structure, starting from the author's algebraic intuitions. The Lambek Calculus is the core of logical Categorial Grammar. The first use of the term "categorial grammar" seems to be in the title of Bar-Hillel, Gaifman and Shamir (1960), but categorial grammar began with Ajdukiewicz (1935) quite a few years earlier. After a period of ostracism, around 1980 the Lambek Calculus was taken up by logicians interested in Computational Linguistics, especially the ones interested in Categorial Grammars.

The work on Categorial Grammar was given a serious impulse by the advent of Girard's Linear Logic at the end of the 1980s. Girard [12] showed that there is a full embedding, preserving proofs, of Intuitionistic Logic into Linear Logic with a modality "!". This meant that Linear Logic, while paying attention to resources, could always code up faithfully Classical Logic and hence one could, as Girard put it, 'have one's cake and eat it', paying attention to resources, if desired, but always forgetting this accounting, if preferred. This meant also that several new systems of resource logics were conceived and developed and these refined resource logics were applied to several areas of Computer Science.

In Computational Linguistics, the Lambek Calculus has seen a significant number of works written about it, apart from a number of monographs that deal with logical and linguistic aspects of the generalized type-logical approach. For a shorter introduction, see Moortgat's entry on the Stanford Encyclopedia of Philosophy on Type Logical Grammar [20]. Type Logical Grammar situates the type-logical approach within the framework of Montague's Universal Grammar and presents detailed linguistic analyses for a substantive fragment of syntactic and semantic phenomena in the grammar of English. Type Logical Semantics offers a general introduction to natural language semantics studied from a type-logical perspective.

This meant that a series of systems, implemented or not, were devised that used the Lambek Calculus or its variants as their basis. These systems can be as expressive as Intuitionistic Logic and the claim is that they are more precise i.e. they make finer distinctions. Some of the landscape of calculi can be depicted as follows:





From the beginning it was clear that the Lambek Calculus is the multiplicative fragment of non-commutative Intuitionistic Linear Logic. In the diagrams **L** stands for the Lambek Calculus, as expounded in [19] but with the unit I added for the tensor connective (there was a certain amount of dispute on that, as the original system did not introduce the constant corresponding to the nullary case of the tensor product, here written as I). The system **LP** is the Lambek Calculus with permutation, sometimes called the van Benthem calculus. We are calling **LA** the Lambek Calculus with additives, the more usual algebraic connectives corresponding to meet and join. Hence adding both permutation and additives to the Lambek Calculus we get to intuitonistic linear logic. On the other diagram to the Lambek Calculus with permutation we add either weakening ($\mathbf{LP_{weak}}$) or contraction ($\mathbf{LP_{cont}}$) or both to get the implicational fragment of Intuitionsitic Propositional Logic.

The Lambek Calculus also has the potential for many applications in other areas of computer science, such as, modeling processes. Linear Logic has been at the forefront of the study of process calculi for many years [1,14,22]. We can think of the commutative tensor product of linear logic as a parallel operator. For example, given a process A and a process B, then we can form the process $A \otimes B$ which runs both processes in parallel. If we remove commutativity from the tensor product we obtain a sequential composition instead of parallel composition. That is, the process $A \rhd B$ first runs process A and then process B in that order. Paraphrasing Vaughan Pratt, "The sequential composition operation has no evident counterpart in type theory" see page 11 of [22]. We believe that the Lambek Calculus will lead to filling this hole, and the results of this paper as a means of obtaining a theory with both a parallel operator and a sequential composition operator. This work thus has a potential to impact research in programming languages and computer security where both linear logic and sequential composition play important roles.

There are several interesting questions, considered for Linear Logic, that could also be asked of the Lambek Calculus or its extensions. One of them, posed by Morrill et al. is whether we can extend the Lambek Calculus with a modality that does for the structural rule of *(exchange)* what the modality *of course* '!' does for the rules of *(weakening)* and *(contraction)*. A preliminary proposal, which answers this question affirmatively, is set forward in this paper. The answer was provided in semantical terms in the first version of this work.

Here we provide also the more syntactic description of these modalities. Building up from work of Ciabattoni, Galatos and Terui in [7] and others that describe how to transform systems of axioms into cut-free sequent rules, we aim to refine the algebraization of proof theory.

2 Related Work

Lamarche and Retoré [18] give an overview of proof nets for the Lambek Calculus where they discuss the addition of various exchange rules to the calculus. For example, the following permutation and cycle rules:

$$\frac{A_1, A_2, \Gamma \vdash B}{A_2, A_1, \Gamma \vdash B}\ \textsc{perm} \qquad \frac{A_1, A_2, \Gamma \vdash B}{A_1, \Gamma, A_2, \Gamma \vdash B}\ \textsc{cycl}$$

Taken in isolation each rule does not imply general exchange, but taken together they do. Thus, it is possible to take a restricted notion of exchange to the Lambek Calculus without taking general exchange. However, applications where one needs a completely non-commutative tensor product and a commutative tensor product cannot be modeled in the Lambek Calculus with these rules.

Polakow and Pfenning [21] combine intuitionistic, commutative linear, and non-commutative linear logic into a system called Ordered Linear Logic (OLL). Polakow and Pfenning then extend OLL into two new systems: a term assignment of OLL called OLLi, and a logical framework in the tradition of LF called OLF.

OLL's sequent is of the form $\Gamma, \Delta, \Omega \vdash A$ where Γ is a multiset of intuitionistic assumptions, Δ is a multiset of commutative linear assumptions, and Ω is a list of non-commutative linear assumptions. Furthermore, OLL contains logical connectives from each logic. For example, there are two different tensor products and three different implications. Thus, the systems developed here are a simplification of OLL showing how to get all of these systems using modalities.

Greco and Palmigiano [13] give a type of logic called a proper display logic for the Lambek Calculus with exponentials. However, they decompose the linear exponential into an adjunction in the spirit Benton's LNL models. In this paper we concentrate on sequent calculi rather than display logic.

3 The Lambek Calculus

The Lambek Calculus, formerly the Syntactic Calculus L, due to Lambek [19], was created to capture the logical structure of sentences. Lambek introduced what we think of as a substructural logic with an operator denoting concatenation, $A \otimes B$, and two implications relating the order of phrases, $A \leftharpoonup B$ and $A \rightharpoonup B$. The first implication corresponds to a phrase of type A when followed by a phrase of type B, and the second is a phrase of type B when preceded by a phrase of type A.

The Lambek Calculus, defined in Fig. 1, can be presented as a non-commutative intuitionistic multiplicative linear logic. Contexts are sequences

$$\frac{}{A \vdash A}\ \text{AX} \qquad \frac{}{\cdot \vdash I}\ \text{UR} \qquad \frac{\Gamma_2 \vdash A \qquad \Gamma_1, A, \Gamma_3 \vdash B}{\Gamma_1, \Gamma_2, \Gamma_3 \vdash B}\ \text{CUT} \qquad \frac{\Gamma_1, \Gamma_2 \vdash A}{\Gamma_1, I, \Gamma_2 \vdash A}\ \text{UL}$$

$$\frac{\Gamma, A, B, \Gamma' \vdash C}{\Gamma, A \otimes B, \Gamma' \vdash C}\ \text{TL} \qquad \frac{\Gamma_1 \vdash A \qquad \Gamma_2 \vdash B}{\Gamma_1, \Gamma_2 \vdash A \otimes B}\ \text{TR}$$

$$\frac{\Gamma_2 \vdash A \qquad \Gamma_1, B, \Gamma_3 \vdash C}{\Gamma_1, A \rightharpoonup B, \Gamma_2, \Gamma_3 \vdash C}\ \text{IRL} \qquad \frac{\Gamma_2 \vdash A \qquad \Gamma_1, B, \Gamma_3 \vdash C}{\Gamma_1, \Gamma_2, B \leftharpoonup A, \Gamma_3 \vdash C}\ \text{ILL}$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \rightharpoonup B}\ \text{IRR} \qquad \frac{A, \Gamma \vdash B}{\Gamma \vdash B \leftharpoonup A}\ \text{ILR}$$

Fig. 1. The Lambek Calculus: L

of formulas, and we denote mapping the modalities over an arbitrary context by $!\Gamma$ and $\kappa\Gamma$.

Because the operator $A \otimes B$ denotes the type of concatenations the types $A \otimes B$ and $B \otimes A$ are not equivalent, and hence, L is non-commutative which explains why implication must be broken up into two operators $A \leftharpoonup B$ and $A \rightharpoonup B$. In the following subsections we give two extensions of L: one with the well-known modality of-course of linear logic which adds weakening and contraction, and a second with a new modality adding exchange.

4 Extensions to the Lambek Calculus

The linear modality, $!A$, read "of-course A" was first proposed by Girard [12] as a means of encoding non-linear logic in both classical and intuitionistic forms in linear logic. For example, non-linear implication $A \rightharpoonup B$ is usually encoded into linear logic by $!A \multimap B$. Since we have based L on non-commutative intuitionistic linear logic it is straightforward to add the of-course modality to L. The rules for the of-course modality are defined by the following rules:

$$\frac{\Gamma_1, !A, \Gamma_2, !A, \Gamma_3 \vdash B}{\Gamma_1, !A, \Gamma_2, \Gamma_3 \vdash B}\ \text{C} \qquad \frac{\Gamma_1, \Gamma_2 \vdash B}{\Gamma_1, !A, \Gamma_2 \vdash B}\ \text{W} \qquad \frac{!\Gamma \vdash B}{!\Gamma \vdash !B}\ \text{BR} \qquad \frac{\Gamma_1, A, \Gamma_2 \vdash B}{\Gamma, !A, \Gamma_2 \vdash B}\ \text{BL}$$

The rules C and W add contraction and weakening to L in a controlled way. Then the other two rules allow for linear formulas to be injected into the modality; and essentially correspond to the rules for necessitation found in S4 [5]. Thus, under the of-course modality the logic becomes non-linear. We will see in Sect. 5.1 that these rules define a comonad. We call the extension of L with the of-course modality $\text{L}_!$.

As we remarked above, one leading question of the Lambek Calculus is: can exchange be added in a similar way to weakening and contraction? That is, can we add a new modality that adds the exchange rule to L in a controlled way?

The answer to this question is positive, and the rules for this new modality are as follows:

$$\frac{\kappa\Gamma \vdash B}{\kappa\Gamma \vdash \kappa B}\ \textsc{Er} \qquad \frac{\Gamma_1, A, \Gamma_2 \vdash B}{\Gamma_1, \kappa A, \Gamma_2 \vdash B}\ \textsc{El} \qquad \frac{\Gamma_1, \kappa A, B, \Gamma_2 \vdash C}{\Gamma_1, B, \kappa A, \Gamma_2 \vdash C}\ \text{E1} \qquad \frac{\Gamma_1, A, \kappa B, \Gamma_2 \vdash C}{\Gamma_1, \kappa B, A, \Gamma_2 \vdash C}\ \text{E2}$$

The first two rules are similar to of-course, but the last two add exchange to formulas under the κ-modality. We call L with the exchange modality L_κ. Thus, unlike intuitionistic linear logic where any two formulas can be exchanged, L_κ restricts exchange to only formulas under the exchange modality. Just like of-course the exchange modality is modeled categorically as a comonad; see Sect. 5.1.

5 Categorical Models

We now turn to the categorical models, ones where one considers different proofs of the same theorem. Since the Lambek Calculus itself came from its categorical models, biclosed monoidal categories, there is no shortage of these models. However, Girard's insight of relating logical systems via modalities should also be considered in this context.

Lambek's work on monoidal biclosed categories happened almost three decades before Girard introduced Linear Logic, hence there were no modalities or exponentials in Lambek's setting. The categorical modelling of the modalities (of-course! and why-not?) was the difficult issue with Linear Logic. This is where there are design decisions to be made.

5.1 Dialectica Lambek Spaces

A sound and complete categorical model of the Lambek Calculi can be given using a modification of de Paiva's dialectica categories [9]. Dialectica categories arose from de Paiva's thesis on a categorical model of Gödel's Dialectica interpretation, hence the name. Dialectica categories were one of the first sound categorical models of intuitionistic linear logic, with linear modalities. We show in this section that they can be adapted to become a sound and complete model for the Lambek Calculus, with both the exchange and of-course modalities. We call this model *dialectica Lambek spaces.*

Due to the complexities of working with dialectica categories we have formally verified[1] this section in the proof assistant Agda [6]. Dialectica categories arise as constructions over a given monoidal category. Suppose $\mathcal{C}$ is such a category. Then in complete generality the objects of the dialectica category over $\mathcal{C}$ are triples (U, X, α) where U and X are objects of $\mathcal{C}$, and $\alpha : A \longrightarrow U \otimes X$ is a subobject of the tensor product in $\mathcal{C}$ of U and X. Thus, we can think of α as a relation over $U \otimes X$. If we specialize the category $\mathcal{C}$ to the category of sets and

[1] The complete formalization can be found online at https://github.com/heades/dialectica-spaces/blob/Lambek/NCDialSets.agda.

functions, Set, then we obtain what is called a dialectica space. Dialectica spaces are a useful model of full intuitionistic linear logic [15].

Morphisms between objects (U, X, α) and (V, Y, β) are pairs (f, F) where $f : U \longrightarrow V$ and $F : Y \longrightarrow X$ are morphisms of $\mathcal{C}$ such that the pullback condition $(U \otimes F)^{-1}(\alpha) \leq (f \otimes Y)^{-1}(\beta)$ holds. In dialectica spaces this condition becomes $\forall u \in U.\forall y \in Y.\alpha(u, F(y)) \leq \beta(f(u), y)$. The latter reveals that we can think of the condition on morphisms as a weak form of an adjoint condition. Finally, through some nontrivial reasoning on this structure we can show that this is indeed a category; for the details see the formal development. Dialectica categories are related to the Chu construction [11] and to Lafont and Streicher's category of games GAME_κ [17].

To some extent the underlying category $\mathcal{C}$ controls the kind of structure we can expect in the dialectica category over $\mathcal{C}$. However, de Paiva showed [11] that by changing the relations used in the objects and the order used in the 'adjoint condition' (which also controls the type of structure) we can obtain a non-symmetric tensor in the dialectica category, if the structure of the underlying category and the structure of the underlying relations are compatible. She also showed that one can abstract the notion of relation out as a parameter in the dialectica construction, so long as this has enough structure, i.e. so long as you have an algebra (that she called a lineale) to evaluate the relations at. We denote this construction by $\mathsf{Dial}_L(\mathcal{C})$ where L is the lineale controlling the relations coming from the monoidal category $\mathcal{C}$. For example, $\mathsf{Dial}_2(\mathsf{Set})$ is the category of usual dialectica spaces of sets over the Heyting (or Boolean) algebra 2.

This way we can see dialectica categories as a framework of categorical models of various logics, varying the underlying category $\mathcal{C}$ as well as the underlying lineale or algebra of relations L. Depending on which category we start with and which structure we use for the relations in the construction we will obtain different models for different logics.

The underlying category we will choose here is the category Set, but the structure we will define our relations over will be a biclosed poset, defined in the next definition.

Definition 1. *Suppose $(M, \leq, \circ, e)$ is an ordered non-commutative monoid. If there exists a largest $x \in M$ such that $a \circ x \leq b$ for any $a, b \in M$, then we denote x by $a \rightharpoonup b$ and called it the **left-pseudocomplement** of a w.r.t b. Additionally, if there exists a largest $x \in M$ such that $x \circ a \leq b$ for any $a, b \in M$, then we denote x by $b \leftharpoonup a$ and called it the **right-pseudocomplement** of a w.r.t b.*

*A **biclosed poset**, $(M, \leq, \circ, e, \rightharpoonup, \leftharpoonup)$, is an ordered non-commutative monoid, $(M, \leq, \circ, e)$, such that $a \rightharpoonup b$ and $b \leftharpoonup a$ exist for any $a, b \in M$.*

Now using the previous definition we define dialectica Lambek spaces.

Definition 2. *Suppose $(M, \leq, \circ, e, \rightharpoonup, \leftharpoonup)$ is a biclosed poset. Then we define the category of **dialectica Lambek spaces**, $\mathsf{Dial}_M(\mathsf{Set})$, as follows:*

- *objects, or dialectica Lambek spaces, are triples (U, X, α) where U and X are sets, and $\alpha : U \times X \longrightarrow M$ is a generalized relation over M, and*

- *maps that are pairs* $(f, F) : (U, X, \alpha) \longrightarrow (V, Y, \beta)$ *where* $f : U \longrightarrow V$ *and* $F : Y \longrightarrow X$ *are functions such that the weak adjointness condition* $\forall u \in U. \forall y \in Y. \alpha(u, F(y)) \leq \beta(f(u), y)$ *holds.*

Notice that the biclosed poset is used here as the target of the relations in objects, but also as providing the order relation in the weak adjoint condition on morphisms. This will allow the structure of the biclosed poset to lift up into $\mathsf{Dial}_M(\mathsf{Set})$.

We will show that $\mathsf{Dial}_M(\mathsf{Set})$ is a model of the Lambek Calculus with modalities. First, we show that it is a model of the Lambek Calculus without modalities. Thus, we must show that $\mathsf{Dial}_M(\mathsf{Set})$ is monoidal biclosed.

Definition 3. *Suppose* (U, X, α) *and* (V, Y, β) *are two objects of* $\mathsf{Dial}_M(\mathsf{Set})$*. Then their tensor product is defined as follows:*

$$(U, X, \alpha) \otimes (V, Y, \beta) = (U \times V, (V \to X) \times (U \to Y), \alpha \otimes \beta)$$

where $- \to -$ *is the function space from* Set*, and* $(\alpha \otimes \beta)((u, v), (f, g)) = \alpha(u, f(v)) \circ \beta(g(u), v)$*.*

The identity of the tensor product just defined is $I = (\mathbb{1}, \mathbb{1}, e)$, where $\mathbb{1}$ is the terminal object in Set, and e is the unit of the biclosed poset. It is straightforward to show that the tensor product is functorial, one can define the left and right unitors, and the associator for tensor; see the formalization for the definitions. In addition, all of the usual monoidal diagrams hold [9]. Take note of the fact that this tensor product is indeed non-commutative, because the non-commutative multiplication of the biclosed poset is used to define the relation of the tensor product.

The tensor product has two right adjoints making $\mathsf{Dial}_M(\mathsf{Set})$ biclosed.

Definition 4. *Suppose* (U, X, α) *and* (V, Y, β) *are two objects of* $\mathsf{Dial}_M(\mathsf{Set})$*. Then two internal-homs can be defined as follows:*

$$\begin{array}{l}(U, X, \alpha) \rightharpoonup (V, Y, \beta) = ((U \to V) \times (Y \to X), U \times Y, \alpha \rightharpoonup \beta)\\ (V, Y, \beta) \leftharpoonup (U, X, \alpha) = ((U \to V) \times (Y \to X), U \times Y, \alpha \leftharpoonup \beta)\end{array}$$

These two definitions are functorial, where the first is contravariant in the first argument and covariant in the second, but the second internal-hom is covariant in the first argument and contravariant in the second. The relations in the previous two definitions prevent these two from collapsing into the same object, because of the use of the left and right pseudocomplement. It is straightforward to show that the following bijections hold:

$$\mathsf{Hom}(A \otimes B, C) \cong \mathsf{Hom}(B, A \rightharpoonup C) \quad \mathsf{Hom}(A \otimes B, C) \cong \mathsf{Hom}(A, C \leftharpoonup B)$$

Therefore, $\mathsf{Dial}_M(\mathsf{Set})$ is biclosed, and we obtain the following result.

Theorem 1. $\mathsf{Dial}_M(\mathsf{Set})$ *is a sound and complete model for the Lambek Calculus L without modalities.*

We now extend $\mathsf{Dial}_M(\mathsf{Set})$ with two modalities: the usual modality, of-course, denoted $!A$, and the exchange modality denoted κA. However, we must first extended biclosed posets to include an exchange operation.

Definition 5. *A **biclosed poset with exchange** is a biclosed poset $(M, \leq, \circ, e, \rightharpoonup, \leftharpoonup)$ equipped with an unary operation $\kappa : M \to M$ satisfying the following:*

$$\begin{array}{rl}
\textit{(Compatibility)} & a \leq b \textit{ implies } \kappa a \leq \kappa b \textit{ for all } a, b, c \in M \\
\textit{(Minimality)} & \kappa a \leq a \textit{ for all } a \in M \\
\textit{(Duplication)} & \kappa a \leq \kappa\kappa a \textit{ for all } a \in M \\
\textit{(Left Exchange)} & \kappa a \circ b \leq b \circ \kappa a \textit{ for all } a, b \in M \\
\textit{(Right Exchange)} & a \circ \kappa b \leq \kappa b \circ a \textit{ for all } a, b \in M
\end{array}$$

Compatibility results in $\kappa : M \to M$ being a functor in the biclosed poset, and the remainder of the axioms imply that κ is a comonad extending the biclosed poset with left and right exchange.

We can now define the two modalities in $\mathsf{Dial}_M(\mathsf{Set})$ where M is a biclosed poset with exchange; clearly we know $\mathsf{Dial}_M(\mathsf{Set})$ is also a model of the Lambek Calculus without modalities by Theorem 1 because M is a biclosed poset.

Definition 6. *Suppose (U, X, α) is an object of $\mathsf{Dial}_M(\mathsf{Set})$ where M is a biclosed poset with exchange. Then the **of-course** and **exchange** modalities can be defined as $!(U, X, \alpha) = (U, U \to X^*, !\alpha)$ and $\kappa(U, X, \alpha) = (U, X, \kappa\alpha)$ where X^* is the free commutative monoid on X, $(!\alpha)(u, f) = \alpha(u, x_1) \circ \cdots \circ \alpha(u, x_i)$ for $f(u) = (x_1, \ldots, x_i)$, and $(\kappa\alpha)(u, x) = \kappa(\alpha(u, x))$.*

This definition highlights a fundamental difference between the two modalities. The definition of the exchange modality relies on an extension of biclosed posets with essentially the exchange modality in the category of posets. However, the of-course modality is defined by the structure already present in $\mathsf{Dial}_M(\mathsf{Set})$, specifically, the structure of Set.

Both of the modalities have the structure of a comonad. That is, there are monoidal natural transformations $\varepsilon_! : !A \longrightarrow A$, $\varepsilon_\kappa : \kappa A \longrightarrow A$, $\delta_! : !A \longrightarrow !!A$ and $\delta_\kappa : \kappa A \longrightarrow \kappa\kappa A$ which satisfy the appropriate diagrams; see the formalization for the full proofs. Furthermore, these comonads come equipped with arrows $e : !A \longrightarrow I$, $d : !A \longrightarrow !A \otimes !A$, $\beta L : \kappa A \otimes B \longrightarrow B \otimes \kappa A$, and $\beta R : A \otimes \kappa B \longrightarrow \kappa B \otimes A$. Thus, we arrive at the following result.

Theorem 2. *Suppose M is a biclosed poset with exchange. Then $\mathsf{Dial}_M(\mathsf{Set})$ is a sound and complete model for the Lambek Calculi $L_!$, L_κ, and $L_{!\kappa}$.*

6 Type Theory for Lambek Systems

In this section we introduce typed calculi for each of the logics discussed so far. Each type system is based on the term assignment for Intuitionistic Linear Logic introduced in [2]. We show that they are all strongly normalizing and confluent, but we do not give full detailed proofs of each of these properties, because they are straightforward consequences of the proofs of strong normalization and confluence for intuitionistic linear logic. In fact, we will reference Bierman's thesis often within this section. The reader may wish to review Sect. 3.5 on page 88 of [4].

6.1 The Typed Lambek Calculus: λL

The first system we cover is the Lambek Calculus without modalities. This system can be seen as the initial core of each of the other systems we introduce below, and thus, we will simply extend the results here to three other systems.

The syntax for patterns, terms, and contexts are described by the following grammar:

$$\begin{array}{rl}
\text{(patterns)} & p := - \mid x \mid \mathsf{unit} \mid p_1 \otimes p_2 \\
\text{(terms)} & t := x \mid \mathsf{unit} \mid t_1 \otimes t_2 \mid \lambda_l x : A.t \mid \lambda_r x : A.t \mid \mathsf{app}_l\ t_1\ t_2 \mid \\
 & \mathsf{app}_r\ t_1\ t_2 \mid \mathsf{let}\ t_1\ \mathsf{be}\ p\ \mathsf{in}\ t_2 \\
\text{(contexts)} & \Gamma := \cdot \mid x : A \mid \Gamma_1, \Gamma_2
\end{array}$$

Contexts are sequences of pairs of free variables and types. Patterns are only used in the let-expression which is itself used to eliminate logical connectives within the left rules of L. All variables in the pattern of a let-expression are bound. The remainder of the terms are straightforward.

$$\frac{}{x : A \vdash x : A}\ \text{T\_VAR} \qquad \frac{}{\cdot \vdash \mathsf{unit} : I}\ \text{T\_UR}$$

$$\frac{\Gamma_2 \vdash t_1 : A \qquad \Gamma_1, x : A, \Gamma_3 \vdash t_2 : B}{\Gamma_1, \Gamma_2, \Gamma_3 \vdash [t_1/x]t_2 : B}\ \text{T\_CUT}$$

$$\frac{\Gamma_1, \Gamma_2 \vdash t : A}{\Gamma_1, x : I, \Gamma_2 \vdash \mathsf{let}\ x\ \mathsf{be}\ \mathsf{unit}\ \mathsf{in}\ t : A}\ \text{T\_UL}$$

$$\frac{\Gamma, x : A, y : B, \Gamma' \vdash t : C}{\Gamma, z : A \otimes B, \Gamma' \vdash \mathsf{let}\ z\ \mathsf{be}\ x \otimes y\ \mathsf{in}\ t : C}\ \text{T\_TL} \qquad \frac{\Gamma_1 \vdash t_1 : A \qquad \Gamma_2 \vdash t_2 : B}{\Gamma_1, \Gamma_2 \vdash t_1 \otimes t_2 : A \otimes B}\ \text{T\_TR}$$

$$\frac{\Gamma_2 \vdash t_1 : A \qquad \Gamma_1, x : B, \Gamma_3 \vdash t_2 : C}{\Gamma_1, z : A \rightharpoonup B, \Gamma_2, \Gamma_3 \vdash [\mathsf{app}_r\ z\ t_1/x]t_2 : C}\ \text{T\_IRL}$$

$$\frac{\Gamma_2 \vdash t_1 : A \qquad \Gamma_1, x : B, \Gamma_3 \vdash t_2 : C}{\Gamma_1, \Gamma_2, z : B \leftharpoonup A, \Gamma_3 \vdash [\mathsf{app}_l\ z\ t_1/x]t_2 : C}\ \text{T\_ILL} \qquad \frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda_r x : A.t : A \rightharpoonup B}\ \text{T\_IRR}$$

$$\frac{x : A, \Gamma \vdash t : B}{\Gamma \vdash \lambda_l x : A.t : B \leftharpoonup A}\ \text{T\_ILR}$$

Fig. 2. Typing rules for the Typed Lambek Calculus: λL

The typing rules can be found in the in Fig. 2 and the reduction rules in Fig. 3. The typing rules are as one might expect. The reduction rules were extracted from the cut-elimination procedure for L.

$$\frac{}{\mathsf{app}_l\ (\lambda_l x : A.t_2)\ t_1 \rightsquigarrow [t_1/x]t_2}\ \textsc{R\_BetaL} \qquad \frac{}{\mathsf{app}_r\ (\lambda_r x : A.t_2)\ t_1 \rightsquigarrow [t_1/x]t_2}\ \textsc{R\_BetaR}$$

$$\frac{}{\mathsf{let}\ t_1\ \mathsf{be\ unit\ in}\ [\mathsf{unit}/z]t_2 \rightsquigarrow [t_1/z]t_2}\ \textsc{R\_BetaU}$$

$$\frac{}{\mathsf{let}\ t_1 \otimes t_2\ \mathsf{be}\ x \otimes y\ \mathsf{in}\ t \rightsquigarrow [t_1/x][t_2/y]t}\ \textsc{R\_BetaT1}$$

$$\frac{}{\mathsf{let}\ t_1\ \mathsf{be}\ x \otimes y\ \mathsf{in}\ [x \otimes y/z]t_2 \rightsquigarrow [t_1/x]t_2}\ \textsc{R\_BetaT2}$$

$$\frac{}{[\mathsf{let}\ t_1\ \mathsf{be\ unit\ in}\ t_2/z]t_3 \rightsquigarrow \mathsf{let}\ t_1\ \mathsf{be\ unit\ in}\ [t_2/z]t_3}\ \textsc{R\_NatU}$$

$$\frac{}{[\mathsf{let}\ t_1\ \mathsf{be}\ x \otimes y\ \mathsf{in}\ t_2/z]t_3 \rightsquigarrow \mathsf{let}\ t_1\ \mathsf{be}\ x \otimes y\ \mathsf{in}\ [t_2/z]t_3}\ \textsc{R\_NatT}$$

$$\frac{}{\mathsf{let\ unit\ be\ unit\ in}\ t \rightsquigarrow t}\ \textsc{R\_LetU}$$

Fig. 3. Rewriting rules for the Lambek Calculus: λL

We denote the reflexive and transitive closure of the $\rightsquigarrow$ by $\rightsquigarrow^*$. We call a term with no β-redexes a normal form, and we denote normal forms by n. In the interest of space we omit the congruence rules from the definition of the reduction relation; we will do this for each calculi introduced throughout this section. The other typed calculi we introduce below will be extensions of λL, thus, we do not reintroduce these rules each time for readability.

Strong normalization. It is well known that intuitionistic linear logic (ILL) is strongly normalizing, for example, see Bierman's thesis [4] or Benton's beautiful embedding of ILL into system F [3].

It is fairly straightforward to define a reduction preserving embedding of λL into ILL. Intuitionistic linear logic can be obtained from λL by replacing the rules T_IRl, T_ILl, T_IRr, and T_ILr with the following two rules:

$$\frac{\Gamma_2 \vdash t_1 : A \qquad \Gamma_1, x : B, \Gamma_3 \vdash t_2 : C}{\Gamma_1, z : A \multimap B, \Gamma_2, \Gamma_3 \vdash [z\ t_1/x]t_2 : C}\ \textsc{T\_Il} \qquad \frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x : A.t : A \multimap B}\ \textsc{T\_Ir}$$

In addition, contexts are considered multisets, and hence, exchange is handled implicitly. Then we can reuse the idea of Benton's embeddings to show type preservation and type reduction.

At this point we define the following embeddings.

Definition 7. *We embed types and terms of λL into ILL as follows:*

Types:

$$\begin{array}{rcl} I^{\mathsf{e}} = I & & (A \rightharpoonup B)^{\mathsf{e}} = A^{\mathsf{e}} \multimap B^{\mathsf{e}} \\ (A \otimes B)^{\mathsf{e}} = A^{\mathsf{e}} \otimes B^{\mathsf{e}} & & (A \leftharpoonup B)^{\mathsf{e}} = A^{\mathsf{e}} \multimap B^{\mathsf{e}} \end{array}$$

Terms:

$$\begin{array}{rl} x^{\mathsf{e}} = x & (\lambda_l x : A.t)^{\mathsf{e}} = \lambda x : A.t^{\mathsf{e}} \\ \mathsf{unit}^{\mathsf{e}} = \mathsf{unit} & (\lambda_r x : A.t)^{\mathsf{e}} = \lambda x : A.t^{\mathsf{e}} \\ (t_1 \otimes t_2)^{\mathsf{e}} = {t_1}^{\mathsf{e}} \otimes {t_2}^{\mathsf{e}} & (\mathsf{app}_l\ t_1\ t_2)^{\mathsf{e}} = {t_1}^{\mathsf{e}}\ {t_2}^{\mathsf{e}} \\ (\mathsf{let}\ t_1\ \mathsf{be}\ p\ \mathsf{in}\ t_2)^{\mathsf{e}} = \mathsf{let}\ {t_1}^{\mathsf{e}}\ \mathsf{be}\ p\ \mathsf{in}\ {t_2}^{\mathsf{e}} & (\mathsf{app}_r\ t_1\ t_2)^{\mathsf{e}} = {t_1}^{\mathsf{e}}\ {t_2}^{\mathsf{e}} \end{array}$$

The previous embeddings can be extended to contexts in the straightforward way, and to sequents as follows:

$$(\Gamma \vdash t : A)^{\mathsf{e}} = \Gamma^{\mathsf{e}} \vdash t^{\mathsf{e}} : A^{\mathsf{e}}$$

We can now prove strong normalization using the embedding preserves.

Theorem 3 (Strong Normalization)

- *If $\Gamma \vdash t : A$ in λL, then $\Gamma^{\mathsf{e}} \vdash t^{\mathsf{e}} : A^{\mathsf{e}}$ in ILL.*
- *If $t_1 \rightsquigarrow t_2$ in λL, then ${t_1}^{\mathsf{e}} \rightsquigarrow {t_2}^{\mathsf{e}}$ in ILL.*
- *If $\Gamma \vdash t : A$, then t is strongly normalizing.*

Proof. The first two cases hold by straightforward induction on the form of the assumed typing or reduction derivation. They then imply the third.

Confluence. The Church-Rosser property is well known to hold for ILL modulo commuting conversions, for example, see Theorem 19 of [4] on page 96. Since λL is essentially a subsystem of ILL, it is straightforward, albeit lengthly, to simply redefine Bierman's candidates and carry out a similar proof as Bierman's (Theorem 19 on page 96 of ibid.).

Theorem 4 (Confluence). *The reduction relation, $\rightsquigarrow$, modulo the commuting conversions is confluent.*

6.2 The Typed Lambek Calculus: $\boldsymbol{\lambda}\mathbf{L}_!$

The calculus we introduce in this section is an extension of λL with the of-course modality $!A$. This extension follows from ILL exactly. The syntax of types and terms of λL are extended as follows:

$$\begin{array}{lll} \text{(types)} & A & := \cdots \mid !A \\ \text{(terms)} & t & := \cdots \mid \mathsf{copy}\ t'\ \mathsf{as}\ t_1, t_2\ \mathsf{in}\ t \mid \mathsf{discard}\ t'\ \mathsf{in}\ t \mid \mathsf{promote}_!\ t'\ \mathsf{for}\ t''\ \mathsf{in}\ t \mid \\ & & \quad \mathsf{derelict}_!\ t \end{array}$$

The new type and terms are what one might expect, and are the traditional syntax used for the of-course modality. We add the following typing rules to λL:

$$\frac{\Gamma_1, x :!A, \Gamma_2, y :!A, \Gamma_3 \vdash t : B}{\Gamma_1, z :!A, \Gamma_2, \Gamma_3 \vdash \mathsf{copy}\ z\ \mathsf{as}\ x, y\ \mathsf{in}\ t : B}\ \text{T\_C} \qquad \frac{\Gamma_1, \Gamma_2 \vdash t : B}{\Gamma_1, x :!A, \Gamma_2 \vdash \mathsf{discard}\ x\ \mathsf{in}\ t : B}\ \text{T\_W}$$

$$\frac{\overrightarrow{x} :!\Gamma \vdash t : B}{\overrightarrow{y} :!\Gamma \vdash \mathsf{promote}_!\ \overrightarrow{y}\ \mathsf{for}\ \overrightarrow{x}\ \mathsf{in}\ t :!B}\ \text{T\_Br} \qquad \frac{\Gamma_1, x : A, \Gamma_2 \vdash t : B}{\Gamma_1, y :!A, \Gamma_2 \vdash [\mathsf{derelict}_!\ y/x]t : B}\ \text{T\_Bl}$$

Finally, the reduction rules can be found in Fig. 4. The equality used in the R_BetaC rule is definitional, meaning, that the rule simply gives the terms on the right side of the equation the name on the left side, and nothing more. This makes the rule easier to read.

Strong normalization. Showing strong normalization for $\lambda L_!$ easily follows by a straightforward extension of the embedding we gave for λL.

Definition 8. *The following is an extension of the embedding of λL into ILL resulting in an embedding of types and terms of $\lambda L_!$ into ILL. First, we define $(!A)^{\mathsf{e}} = !A^{\mathsf{e}}$, then the following defines the embedding of terms:*

$$\begin{aligned}(\mathsf{copy}\ t'\ \mathsf{as}\ t_1, t_2\ \mathsf{in}\ t)^{\mathsf{e}} &= \mathsf{copy}\ t'^{\mathsf{e}}\ \mathsf{as}\ {t_1}^{\mathsf{e}}, {t_2}^{\mathsf{e}}\ \mathsf{in}\ t^{\mathsf{e}}\\ (\mathsf{discard}\ t'\ \mathsf{in}\ t)^{\mathsf{e}} &= \mathsf{discard}\ t'^{\mathsf{e}}\ \mathsf{in}\ t^{\mathsf{e}}\\ (\mathsf{promote}_!\ t'\ \mathsf{for}\ t''\ \mathsf{in}\ t)^{\mathsf{e}} &= \mathsf{promote}_!\ t'^{\mathsf{e}}\ \mathsf{for}\ t''^{\mathsf{e}}\ \mathsf{in}\ t^{\mathsf{e}}\\ (\mathsf{derelict}_!\ t)^{\mathsf{e}} &= \mathsf{derelict}_!\ t^{\mathsf{e}}\end{aligned}$$

Just as before this embedding is type preserving and reduction preserving.

Theorem 5 (Type and Reduction Preserving Embedding)

- *If $\Gamma \vdash t : A$ in $\lambda L_!$, then $\Gamma^{\mathsf{e}} \vdash t^{\mathsf{e}} : A^{\mathsf{e}}$ in ILL.*
- *If $t_1 \rightsquigarrow t_2$ in $\lambda L_!$, then ${t_1}^{\mathsf{e}} \rightsquigarrow {t_2}^{\mathsf{e}}$ in ILL.*
- *If $\Gamma \vdash t : A$, then t is strongly normalizing.*

Proof. The first two cases hold by straightforward induction on the form of the assumed typing or reduction derivation. They then imply the third.

Confluence. The Church-Rosser property also holds for $\lambda L_!$, and can be shown by straightforwardly applying a slightly modified version of Bierman's proof [4] just as we did for λL. Thus, we have the following:

Theorem 6 (Confluence). *The reduction relation, $\rightsquigarrow$, modulo the commuting conversions is confluent.*

$$\frac{}{\mathsf{derelict}_!\ (\mathsf{promote}_!\ \overrightarrow{t}\ \mathsf{for}\ \overrightarrow{x}\ \mathsf{in}\ t_1) \rightsquigarrow [\overrightarrow{t}/\overrightarrow{x}]t_1}\ \text{R\_BetaDR}$$

$$\frac{}{\mathsf{discard}\ (\mathsf{promote}_!\ \overrightarrow{t}\ \mathsf{for}\ \overrightarrow{x}\ \mathsf{in}\ t_1)\ \mathsf{in}\ t_2 \rightsquigarrow \mathsf{discard}\ \overrightarrow{t}\ \mathsf{in}\ t_2}\ \text{R\_BetaDI}$$

$$\frac{t_1' = \mathsf{promote}_!\ \overrightarrow{w}\ \mathsf{for}\ \overrightarrow{x}\ \mathsf{in}\ t_1 \qquad t_1'' = \mathsf{promote}_!\ \overrightarrow{z}\ \mathsf{for}\ \overrightarrow{x}\ \mathsf{in}\ t_1}{\mathsf{copy}\ (\mathsf{promote}_!\ \overrightarrow{t}\ \mathsf{for}\ \overrightarrow{x}\ \mathsf{in}\ t_1)\ \mathsf{as}\ w, z\ \mathsf{in}\ t_2 \rightsquigarrow \mathsf{copy}\ \overrightarrow{t}\ \mathsf{as}\ \overrightarrow{w}, \overrightarrow{z}\ \mathsf{in}\ [t_1'/w][t_1''/z]t_2}\ \text{R\_BetaC}$$

$$\frac{}{[\mathsf{discard}\ t\ \mathsf{in}\ t_1/x]t_2 \rightsquigarrow \mathsf{discard}\ t\ \mathsf{in}\ [t_1/x]t_2}\ \text{R\_NatD}$$

$$\frac{}{[\mathsf{copy}\ t\ \mathsf{as}\ x, y\ \mathsf{in}\ t_1/x]t_2 \rightsquigarrow \mathsf{copy}\ t\ \mathsf{as}\ x, y\ \mathsf{in}\ [t_1/x]t_2}\ \text{R\_NatC}$$

Fig. 4. Rewriting rules for The Typed Lambek Calculus: $\lambda L_!$

6.3 The Typed Lambek Calculus: λL_κ

The next calculus we introduce is also an extension of λL with a modality that adds exchange to λL_κ denoted κA. It is perhaps the most novel of the calculi we have introduced.

The syntax of types and terms of λL are extended as follows:

$$\begin{array}{ll}\text{(types) } A := \cdots \mid \kappa A \\ \text{(terms) } t \ := \cdots \mid \mathsf{exchange_l}\ t_1, t_2\ \mathsf{with}\ x, y\ \mathsf{in}\ t_3 \mid \mathsf{exchange_r}\ t_1, t_2\ \mathsf{with}\ x, y\ \mathsf{in}\ t_3 \mid \\ \qquad\qquad \mathsf{promote}_\kappa\ t'\ \mathsf{for}\ t''\ \mathsf{in}\ t \mid \mathsf{derelict}_\kappa\ t\end{array}$$

The syntax for types has been extended to include the exchange modality, and the syntax of terms follow suit. The terms $\mathsf{exchange_l}\ t_1, t_2\ \mathsf{with}\ x, y\ \mathsf{in}\ t_3$ and $\mathsf{exchange_r}\ t_1, t_2\ \mathsf{with}\ x, y\ \mathsf{in}\ t_3$ are used to explicitly track uses of exchange throughout proofs.

We add the following typing rules to λL:

$$\frac{\Gamma_1, x_1 : \kappa A, y_1 : B, \Gamma_2 \vdash t : C}{\Gamma_1, y_2 : B, x_2 : \kappa A, \Gamma_2 \vdash \mathsf{exchange_l}\ y_2, x_2\ \mathsf{with}\ x_1, y_1\ \mathsf{in}\ t : C}\ \text{T\_E1}$$

$$\frac{\Gamma_1, x_1 : A, y_1 : \kappa B, \Gamma_2 \vdash t : C}{\Gamma_1, y_2 : \kappa B, x_2 : A, \Gamma_2 \vdash \mathsf{exchange_r}\ y_2, x_2\ \mathsf{with}\ x_1, y_1\ \mathsf{in}\ t : C}\ \text{T\_E2}$$

$$\frac{\overrightarrow{x} : \kappa\Gamma \vdash t : B}{\overrightarrow{y} : \kappa\Gamma \vdash \mathsf{promote}_\kappa\ \overrightarrow{y}\ \mathsf{for}\ \overrightarrow{x}\ \mathsf{in}\ t : \kappa B}\ \text{T\_Er} \qquad \frac{\Gamma_1, x : A, \Gamma_2 \vdash t : B}{\Gamma_1, y : \kappa A, \Gamma_2 \vdash [\mathsf{derelict}_\kappa\ y/x]t : B}\ \text{T\_El}$$

The reduction rules are in Fig. 5, and are vary similar to the rules from $\lambda L_!$.

$$\frac{}{\mathsf{derelict}_\kappa\ (\mathsf{promote}_\kappa\ \overrightarrow{t}\ \mathsf{for}\ \overrightarrow{x}\ \mathsf{in}\ t_1) \rightsquigarrow [\overrightarrow{t}/\overrightarrow{x}]t_1}\ \text{R\_BetaEDR}$$

$$\frac{}{[\mathsf{exchange_l}\ t_1, t_2\ \mathsf{with}\ x, y\ \mathsf{in}\ t_3/z]t_4 \rightsquigarrow \mathsf{exchange_l}\ t_1, t_2\ \mathsf{with}\ x, y\ \mathsf{in}\ [t_3/z]t_4}\ \text{R\_NatEl}$$

$$\frac{}{[\mathsf{exchange_r}\ t_1, t_2\ \mathsf{with}\ x, y\ \mathsf{in}\ t_3/z]t_4 \rightsquigarrow \mathsf{exchange_r}\ t_1, t_2\ \mathsf{with}\ x, y\ \mathsf{in}\ [t_3/z]t_4}\ \text{R\_NatEr}$$

Fig. 5. Rewriting rules for The Typed Lambek Calculus: λL_κ

Strong normalization. Similarly, we show that we can embed λL_κ into ILL, but the embedding is a bit more interesting.

Definition 9. *The following is an extension of the embedding of λL into ILL resulting in an embedding of types and terms of λL_κ into ILL. First, we define $(\kappa A)^{\mathsf{e}} = !A^{\mathsf{e}}$, and then the following defines the embedding of terms:*

$$\begin{aligned}
(\mathsf{exchange}_{\mathsf{l}}\ t_1, t_2\ \mathsf{with}\ x, y\ \mathsf{in}\ t_3)^{\mathsf{e}} &= [{t_2}^{\mathsf{e}}/x][{t_1}^{\mathsf{e}}/y]{t_3}^{\mathsf{e}}\\
(\mathsf{exchange}_{\mathsf{r}}\ t_1, t_2\ \mathsf{with}\ x, y\ \mathsf{in}\ t_3)^{\mathsf{e}} &= [{t_2}^{\mathsf{e}}/x][{t_1}^{\mathsf{e}}/y]{t_3}^{\mathsf{e}}\\
(\mathsf{promote}_{\kappa}\ t'\ \mathsf{for}\ t''\ \mathsf{in}\ t)^{\mathsf{e}} &= \mathsf{promote}_{!}\ t'^{\mathsf{e}}\ \mathsf{for}\ t''^{\mathsf{e}}\ \mathsf{in}\ t^{\mathsf{e}}\\
(\mathsf{derelict}_{\kappa}\ t)^{\mathsf{e}} &= \mathsf{derelict}_{!}\ t^{\mathsf{e}}
\end{aligned}$$

The embedding translates the exchange modality into the of-course modality of ILL. We do this so as to preserve the comonadic structure of the exchange modality. One might think that we could simply translate the exchange modality to the identity, but as Benton showed [3], this would result in an embedding that does not preserve reductions. Furthermore, the left and right exchange terms are translated away completely, but this works because ILL contains exchange in general, and hence, does not need to be tracked explicitly. We now have strong normalization and confluence.

Theorem 7 (Strong Normalization)

- *If $\Gamma \vdash t : A$ in $\lambda L_!$, then $\Gamma^{\mathsf{e}} \vdash t^{\mathsf{e}} : A^{\mathsf{e}}$ in ILL.*
- *If $t_1 \rightsquigarrow t_2$ in $\lambda L_!$, then ${t_1}^{\mathsf{e}} \rightsquigarrow {t_2}^{\mathsf{e}}$ in ILL.*
- *If $\Gamma \vdash t : A$, then t is strongly normalizing.*
- *The reduction relation, $\rightsquigarrow$, modulo the commuting conversions is confluent.*

Proof. The first two cases hold by straightforward induction on the form of the assumed typing or reductions derivation. They then imply the third case.

6.4 The Typed Lambek Calculus: $\lambda L_{!\kappa}$

If we combine all three of the previous typed Lambek Calculi, then we obtain the typed Lambek Calculus $\lambda L_{!\kappa}$. The main characteristics of this system are that it provides the benefits of the non-symmetric adjoint structure of the Lambek Calculus with the ability of having exchange, and the of-course modality, but both are carefully tracked within the proofs.

Strong normalization for this calculus can be proved similarly to the previous calculi by simply merging the embeddings together. Thus, both modalities of $\lambda L_{!\kappa}$ would merge into the of-course modality of ILL. The Church-Rosser property also holds for $\lambda L_{!\kappa}$ by extending the proof of confluence for ILL by Bierman [4] just as we did for the other systems. Thus, we have the following results.

Theorem 8 (Strong Normalization). *If $\Gamma \vdash t : A$, then t is strongly normalizing.*

Theorem 9 (Confluence). *The reduction relation, $\rightsquigarrow$, modulo the commuting conversions is confluent.*

7 Conclusions

We have recalled how to use biclosed posets and sets to construct dialectica-like models of the Lambek Calculus. This construction is admittedly not the easiest one, which is the reason why we use automated tools to verify our definitions, but it has one striking advantage. It shows how to introduce modalities to recover the expressive power of intuitionistic (and a posteriori classical) propositional logic to the system. We know of no other construction of models of Lambek Calculus that does model modalities, not using their syntactic properties. (The traditional view in algebraic semantics is to consider idempotent operators for modalities like !). The categorical semantics here has been described before [10], but the syntactic treatment of the lambda-calculi, in the style of [2] had not been done and there were doubts about its validity, given the results of Jay [16]. We are glad to put this on a firm footing, using another one of Benton's ideas: his embedding of intuitionistic linear logic into system F. Finally, we envisage more work, along the lines of algebraic proof theory, for modalities and non-symmetric type systems.

Acknowledgments. The authors would like to thank the anonymous reviewers for their feedback which did make this a better paper. The second author was partially supported by the NSF grant #1565557.

References

1. Abramsky, S.: Proofs as processes. Theor. Comput. Sci. **135**(1), 5–9 (1994)
2. Benton, N., Bierman, G., de Paiva, V., Hyland, M.: A term calculus for Intuitionistic Linear Logic. In: Bezem, M., Groote, J.F. (eds.) TLCA 1993. LNCS, vol. 664, pp. 75–90. Springer, Heidelberg (1993). https://doi.org/10.1007/BFb0037099
3. Benton, P.N.: Strong normalisation for the linear term calculus. J. Funct. Program. **5**(1), 65–80 (1995)
4. Bierman, G.M.: On Intuitionistic Linear Logic. PhD thesis, Wolfson College, Cambridge, December 1993
5. Bierman, G.M., de Paiva, V.C.V.: On an intuitionistic modal logic. Stud. Logica **65**(3), 383–416 (2000)
6. Bove, A., Dybjer, P., Norell, U.: A brief overview of Agda-a functional language with dependent types. TPHOLs **5674**, 73–78 (2009)
7. Ciabattoni, A., Galatos, N., Terui, K.: Algebraic proof theory for substructural logics: Cut-elimination and completions. Ann. Pure Appl. Logic **163**(3), 266–290 (2012)
8. Coecke, B., Grefenstette, E., Sadrzadeh, M.: Lambek vs. Lambek: Functorial vector space semantics and string diagrams for Lambek calculus. Ann. Pure Appl. Logic **164**(11), 1079–1100 (2013)
9. de Paiva, V.: The dialectica categories. PhD thesis, Computer Laboratory, University of Cambridge, PhD Thesis, 1990. Computer Laboratory, University of Cambridge, PhD Thesis (1990)
10. de Paiva, V.: A Dialectica model of the Lambek calculus. In: 8th Amsterdam Logic Colloquium (1991)
11. De Paiva, V.: Dialectica and chu constructions: Cousins? Theory Appl. Categories **17**(7), 127–152 (2007)
12. Girard, J.-Y.: Linear logic. Theor. Comput. Sci. **50**(1), 1–101 (1987)
13. Greco, G., Palmigiano, A.: Linear Logic Properly Displayed. ArXiv e-prints, November 2016

14. Honda, K., Laurent, O.: An exact correspondence between a typed pi-calculus and polarised proof-nets. Theor. Compu. Sci. **411**(22), 2223–2238 (2010)
15. Hyland, M., de Paiva, V.: Full intuitionistic linear logic (extended abstract). Ann. Pure Appl. Logic **64**(3), 273–291 (1993)
16. Barry Jay, C.: Coherence in category theory and the Church-Rosser property. Notre Dame J. Formal Logic **33**(1), 140–143 (1991). 12
17. Lafont, Y., Streicher, T.: Games semantics for linear logic. In: Proceedings of Sixth Annual IEEE Symposium on Logic in Computer Science, 1991, LICS 1991, pp. 43–50. IEEE (1991)
18. Lamarche, F., Retoré, C.: Proof nets for the Lambek calculus - an overview. In: Proceedings of the Third Roma Workshop. Proofs and Linguistic Categories, pp. 241–262 (1996)
19. Lambek, J.: The mathematics of sentence structure. In: American Mathematical Monthly, pp. 154–170 (1958)
20. Moortgat, M.: Typelogical grammar. In: Zalta, E.N. (ed.) The Stanford Encyclopedia of Philosophy, Spring 2014 edition (2014)
21. Polakow, J.: Ordered linear logic and applications. PhD thesis, Carnegie Mellon University (2001)
22. Pratt, V.R.: Types as processes, via Chu spaces. Electr. Notes Theor. Comput. Sci. **7**, 227–247 (1997)
23. Sewell, P., Nardelli, F., Owens, S., Peskine, G., Ridge, T., Sarkar, S., Strnisa, R.: Ott: Effective tool support for the working semanticist. J. Funct. Program. **20**, 71–122 (2010)
24. Szabo, M.E.: Algebra of Proofs. Studies in Logic and the Foundations of Mathematics, vol. 88, North-Holland (1978, 1979)

From Epistemic Paradox to Doxastic Arithmetic

V. Alexis Peluce(✉)

The Graduate Center of the City University of New York, New York City, USA
vpeluce@gradcenter.cuny.edu

Abstract. The logical analysis of epistemic paradoxes—including, for example, the Moore and Gödel-Buridan paradoxes—has traditionally been performed assuming the whole range of corresponding modal logic principles: $\{\mathsf{D}, \mathsf{T}, \mathsf{4}, \mathsf{5}\}$. In this paper, it is discovered precisely which of those principles (including also the law of excluded middle, LEM) are responsible for the paradoxical behavior of the Moore, Gödel-Buridan, Dual Moore, and Commissive Moore sentences. Further, by reproducing these paradoxes intuitionistically we reject a conjecture that these paradoxes are caused by the LEM. An exploration of the Gödel-Buridan sentence prompts the inquiry into a system, Doxastic Arithmetic, DA, designed to represent the arithmetical beliefs of an agent who accepts all specific arithmetical proofs and yet believes in the consistency of their own beliefs. For these reasons, DA may be regarded as an epistemic way to circumvent limitations of Gödel's Second Incompleteness Theorem.

Keywords: Modal logic · Epistemology · Paradoxes
Doxastic arithmetic

1 Introduction

Many find the Moore sentence paradoxical. How can it be the case, it might be asked, that one consistently assert: "it is raining but I do not believe that it is raining?" If one *asserts* it, the reasoning would go, then it is strange to say that they do not also believe it. But if they believe the conjunction, then we would be inclined to say that they believe that it is raining. If we already have that it was not the case that our agent believed it was raining, we will have a contradiction.

More clearly, the Moore sentence, M, is the following:

$$A \wedge \neg \Box A$$

Perhaps what we find paradoxical about our situation is not that there is some problem with M itself, but that something goes wrong with the related *believed* Moore sentence, $\Box M$:

$$\Box(A \wedge \neg \Box A)$$

V. A. Peluce—The author would like to thank Sergei Artemov for discussion and guidance throughout the development of this project.

S. Artemov and A. Nerode (Eds.): LFCS 2018, LNCS 10703, pp. 273–288, 2018.
https://doi.org/10.1007/978-3-319-72056-2_17

Jaakko Hintikka put forth such a response to the Moore paradox in his *Knowledge and Belief*, [10], p. 67. He writes, "In short, the gist of Moore's paradox may be said (somewhat elliptically) to lie in the fact that $[M]$ is necessarily *unbelievable* by the speaker." We will use *paradoxicality* in this sense of Hintikka, to mean:

Hintikka Paradoxicality: A formula F is Hintikka Paradoxical in some logic L if and only if F is satisfiable in L, while $\Box F$ is not satisfiable in L.[1]

The goal of this paper is to investigate exactly the circumstances in which four related sentences, the Moore sentence $(A \wedge \neg\Box A)$, the Gödel-Buridan sentence $(A \leftrightarrow \neg\Box A)$, the—introduced in this paper—Dual Moore sentence $(\neg A \wedge \Box A)$, and the Commissive Moore sentence $(A \wedge \Box\neg A)$, are paradoxical. Note that for A we have in mind some *atomic* proposition. Call these sentences M, G, DM, and CM, respectively. We aim to carry out this investigation in the general epistemic context, that is, we will consider the modal logics both of belief and of knowledge. Specifically, assuming iK (K over intuitionistic logic), we aim to discover all subsets of principles from the set $\mathcal{P}$ where:

$$\mathcal{P} = \{LEM, \mathsf{D}, \mathsf{T}, \mathsf{4}, \mathsf{5}\}$$

that make M, G, DM, and CM paradoxical in Hintikka's sense. By starting with iK rather than K itself, we extend our analysis to capture *LEM—the Law of Excluded Middle*—as well.

Inquiry into these sentences, specifically the Gödel-Buridan sentence prompts further exploration of a system for representing arithmetical beliefs of an agent. We call this system Doxastic Arithmetic, or DA. The idea is that DA allows us to model beliefs of an agent who accepts all specific arithmetical proofs and yet believes in the consistency of their own beliefs; it thus may be regarded as an epistemic way to sidestep constraints imposed by Gödel's Second Incompleteness Theorem.

2 Moore's Paradox

The Moore sentence M is the sentence $A \wedge \neg\Box A$. M is paradoxical in the sense that, in certain contexts, it is satisfiable while $\Box M$ is not. It is obvious that the context of knowledge is one such context. More interesting for our purposes, however, is the context of belief in which $\Box$ is not factive.

In this section it is shown that in $\mathsf{iKD4}$, the intuitionistic version of $\mathsf{KD4}$, $\Box M$ is not satisfiable. It is then shown that $\Box M$ is satisfiable in $\mathsf{K45}$ and KD.

[1] This notion is closely related to Hintikka's similar concept of *doxastic indefensibility*, [10], p. 71. Utterance, being of central importance to Hintikka's concept, does not enjoy the same status in our notion, however. Hintikka writes, "[W]e shall call the set $\{p_1, p_2, \ldots p_k\}$ *doxastically indefensible for the person referred to by this term to utter* if and only if the sentence $\mathcal{B}_a(p_1 \;\&\; p_2 \;\&\ldots\&\; p_k)$ is indefensible *simpliciter*." A natural question is: what Hintikka meant by a formula's indefensibility *simpliciter*? Given [10], p. 32, a natural answer is *inconsistency*.

Along with the well-known observation that $\mathsf{iKT} \vdash \neg\Box M$, this provides an exact characterization of the combination of principles from $\mathcal{P}$ which make M paradoxical in our sense: M is paradoxical over a logic $L = \mathsf{K} + X$ where $X \subseteq \mathcal{P}$ iff X contains either T or $\mathsf{D} + \mathsf{4}$. This thus suggests that even though the Moore sentence shares in some paradoxicality in the context of belief, neither $\mathsf{5}$ nor the law of excluded middle are to blame. Even more D or $\mathsf{4}$ alone are not strong enough to make M paradoxical in this sense.

Theorem 1. *M is Hintikka Paradoxical over a logic $L = \mathsf{iK} + X$ where $X \subseteq \mathcal{P}$ iff X contains either* $\mathsf{D} + \mathsf{4}$, *or* T.

Proof. This follows from Lemmas 1, 2, 3, 4, and 5. ■

Lemma 1. *M is satisfiable in* $\mathsf{S5}$

Proof. Consider the model $\mathcal{M} = \langle W, \mathcal{R}, \Vdash \rangle$. Let W $= \{1, 2\}$. Let $\mathcal{R} = \{(1, 1), (1, 2), (2, 2), (2, 1)\}$. Here $\mathcal{R}$ is transitive, reflexive, and also euclidean; hence this will be an $\mathsf{S5}$ model. Let also $1 \Vdash A$ and $2 \Vdash \neg A$. Hence, also $1 \Vdash \neg\Box A$. But $A \wedge \neg\Box A$ is M, the Moore sentence (Fig. 1). ■

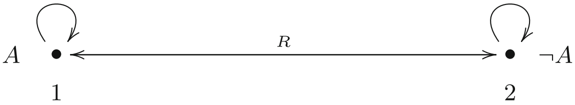


Fig. 1. $\mathsf{S5}$ model of M

Lemma 2. $\mathsf{iKT} \vdash \neg\Box M$[2]

Proof. It suffices to establish that $\mathsf{iKT} + \Box\ M \vdash \bot$

1. $\Box(A \wedge \neg\Box A)$ - Assumption.
2. $\Box A \wedge \Box\neg\Box A$ - K reasoning on 1.
3. $\Box A \wedge \neg\Box A$ - T reasoning on 2.
4. $\bot$ - 3. ■

Lemma 3. $\mathsf{iKD4} \vdash \neg\Box M$[3]

Proof. It suffices to establish that $\mathsf{iKD4} + \Box\ M \vdash \bot$

1. $\Box(A \wedge \neg\Box A)$ - Assumption.
2. $\Box A \wedge \Box\neg\Box A$ - K reasoning on 1.
3. $\Box\Box A \wedge \Box\neg\Box A$ - $\mathsf{4}$ reasoning on 2.
4. $\Box\Box A \wedge \neg\Box\Box A$ - D reasoning on 3.
5. $\bot$ - From 4. ■

Lemma 4. *$\Box M$ is satisfiable in* $\mathsf{KD5}$

Proof. Consider the three world model $\mathcal{M} = \langle W, \mathcal{R}, \Vdash \rangle$. Let W $= \{1, 2, 3\}$. Let $\mathcal{R} = \{(1, 2), (2, 2), (2, 3), (3, 3)\}$. This is a D model insofar as every world can access another world and is a $\mathsf{5}$ model because $\mathcal{R}$ is euclidean. This is not a $\mathsf{4}$ model, since $\mathcal{R}$ is not transitive. Let $2 \Vdash A$ and $3 \Vdash \neg A$. So, $2 \Vdash M$ and $1 \Vdash \Box M$ (Fig. 2). ■

[2] This sort of result is offered in [10].

[3] In its classical form, this sort of observation is not a new one, see for instance, [11], p. 2. It is not clear, however, that this has been demonstrated in $\mathsf{iKD4}$.

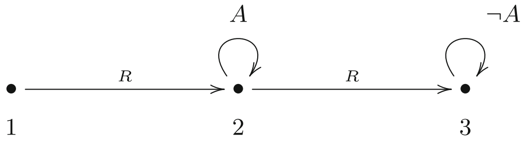


Fig. 2. KD5 model of $\Box M$

Lemma 5. *$\Box M$ is satisfiable in* K45

Proof. Consider the model $\mathcal{M} = \langle W, \mathcal{R}, \Vdash \rangle$. Let W $= \{1\}$. Let $\mathcal{R} = \emptyset$. This is trivially a 4 and a 5 model. Clearly, $1 \Vdash \Box M$ (Fig. 3). ■

Fig. 3. K45 model of $\Box M$

3 The Gödel-Buridan Paradox

The philosophical tradition, see Tyler Burge's [5] (1978),[4] attributes the investigation into an interesting paradox to Jean Buridan. In that context, it is more recently discussed in Michael Caie's [6,7], though it has obvious ties to the work of Kurt Gödel.[5] The paradox is the following sentence:

I don't believe that this sentence is true.

In addition to a modality for belief, both a device for self-reference and a truth predicate are employed. In this part of our discussion, however, we can set the second and third of these features aside (though self-reference à la Gödel will come up again in Sect. 7). We thus understand the Gödel-Buridan sentence, G, as the sentence $A \leftrightarrow \neg\Box A$.[6] Now, in which cases is G paradoxical?

Theorem 2. *G is Hintikka Paradoxical over a logic $L =$ iK $+ X$ where $X \subseteq \mathcal{P}$ iff X contains either* T *or* D + 4*.*

[4] Burge writes that "In Sophism 13 of Chapter VIII Buridan supposes the following proposition is written on the wall: Socrates knows the proposition written on the wall to be doubted by him. Socrates reads it, thinks it through and is unsure (doubts) whether or not it is true. Further, he knows that he doubts it. Buridan asks whether or not the proposition is true." [5], p. 22.

[5] Caie in [6], pp. 16–17, conjectures that the law of excluded middle is to blame for the refutability of $\Box G$ in the context of belief. This is refuted by Theorem 2.

[6] It would be more precise to use notation G_A here since G actually depends on A for the reasons mentioned. Since we said we would set those considerations aside for now, we stick to a shorter notation.

Proof. This follows from Lemmas 6, 7, 8, 9, and 10. ■

Lemma 6. *G is satisfiable in* $\mathsf{S5}$

Proof. Consider the model $\mathcal{M} = \langle W, \mathcal{R}, \Vdash \rangle$. Let W $= \{1, 2\}$. Let $\mathcal{R} = \{(1, 1), (1, 2), (2, 1), (2, 2)\}$. Each world can access each world, so this is an $\mathsf{S5}$ model. Let $1 \Vdash A$ and $2 \Vdash \neg A$. Clearly, $1 \Vdash G$. ■

The figure for this model is the same as the one in Lemma 1.

Lemma 7. $\mathsf{iKT} \vdash \neg\Box G$[7]

Proof. It suffices to show that $\mathsf{iKT} + \Box G \vdash \bot$

1. $\Box(A \rightarrow \neg\Box A)$ - Assumption.
2. $\Box(\neg\Box A \rightarrow A)$ - Assumption.
3. $\Box(\Box A \rightarrow A)$ - Necessitation on T.
4. $\Box(\Box A \rightarrow \neg\Box A)$ - From 1 and 3 by K reasoning.
5. $\Box(\Box A \rightarrow (\Box A \wedge \neg\Box A))$ - From 4 by K reasoning.
6. $\Box\neg\Box A$ - From 5.
7. $\neg\Box A$ - T on 6.
8. $\Box\neg\Box A \rightarrow \Box A$ - K reasoning on 2.
9. $\Box A$ - Modus Ponens on 6 and 8.
10. $\bot$ - 7 and 9. ■

Lemma 8. $\mathsf{iKD4} \vdash \neg\Box G$

Proof. It suffices to show that $\mathsf{iKD4} + \Box B \vdash \bot$

1. $\Box(A \leftrightarrow \neg\Box A)$ - Definition of $\Box G$.
2. $\Box(A \rightarrow (A \wedge \neg\Box A))$ - propositional reasoning on 1.
3. $\Box A \rightarrow \Box(A \wedge \neg\Box A)$ - K reasoning on 2.
4. $\Box A \rightarrow (\Box A \wedge \Box\neg\Box A)$ - K reasoning on 3.
5. $\Box A \rightarrow (\Box\Box A \wedge \Box\neg\Box A)$ - 4 and K reasoning on 4.
6. $\Box A \rightarrow \Box(\Box A \wedge \neg\Box A)$ - K reasoning on 5.
7. $\Box A \rightarrow \Box\bot$ - K reasoning on 6.
8. $\Box A \rightarrow \bot$ - D and K reasoning on 7.
9. $\neg\Box A$ - 8.
10. $\Box\neg\Box A$ - 9 and Necessitation.[8]
11. $\Box A \leftrightarrow \Box\neg\Box A$ - K reasoning on 1.
12. $\Box A$ - 10 and 11, Modus Ponens.
13. $\bot$ - 9 and 12 and propositional logic or 8 and 12 and Modus Ponens. ■

Lemma 9. *$\Box G$ is satisfiable in* $\mathsf{KD5}$

[7] The author thanks Sergei Artemov for this proof.

[8] Necessitation is not always unproblematic, especially in the epistemic context. See, for instance, Artemov's [2]. It is admissible in this case, however, because $\neg\Box A$ follows using only $\mathsf{iKD4}$ and $\Box G$.

Proof. Consider the three world model $\mathcal{M} = \langle W, \mathcal{R}, \Vdash \rangle$. Let $\mathrm{W} = \{1, 2, 3\}$. Let $\mathcal{R} = \{(1, 2), (2, 2), (2, 3), (3, 3)\}$. This is a D model insofar as every world can access another world and is a $\mathsf{5}$ model because $\mathcal{R}$ is euclidean. This is not a $\mathsf{4}$ model, since $\mathcal{R}$ is not transitive. Let $2 \Vdash A$ and $3 \Vdash \neg A$. So, $2 \Vdash G$ and $1 \Vdash \Box G$. ■

This figure is the same as the one in Lemma 4.

Lemma 10. *$\Box G$ is satisfiable in* $\mathsf{K45}$.

Proof. Consider the model $\mathcal{M} = \langle W, \mathcal{R}, \Vdash \rangle$. Let $\mathrm{W} = \{1\}$. Let $\mathcal{R} = \emptyset$. This is trivially a $\mathsf{4}$ and a $\mathsf{5}$ model. Clearly, $1 \Vdash \Box G$. ■

This figure is the same as the one in Lemma 5.

4 Moore, Gödel-Buridan, and Dual Moore

In the previous sections we explored the conditions in which M and G were paradoxical. There are similarities between those two sentences, though those similarities can be overstated.[9] In this section, we show how G can be read as a disjunction of M and something else. That something else, which we will call the Dual Moore, is itself an interesting formula.

The reader will recall that G was the sentence:

$$A \leftrightarrow \neg \Box A$$

The following is a classical tautology:

$$((X \wedge \neg Y) \vee (\neg X \wedge Y)) \leftrightarrow (X \leftrightarrow \neg Y)$$

We can prove both of sides of the biconditional classically. In intuitionistic logic, however, the → holds while ← fails. Substituting A for X and $\Box A$ for Y, we get:

$$(A \wedge \neg \Box A) \vee (\neg A \wedge \Box A)$$

[9] Caie [7], p. 36, discusses the Moore sentence as relates to G. There are some things having to do with that discussion, though, that are worth clarifying. He writes, "We've considered two classes of sentences, the Moore-paradoxical and the Burge-Buridan sentences. It's worth noting, however, that the latter class is really a subclass of the former. In general, a Moore-paradoxical sentence is one that has the following form $\phi \wedge \neg \mathcal{B}\phi$. A Burge-Buridan sentence, on the other hand, has the form $\neg \mathcal{B}T(\beta)$, where β refers to that very sentence. On the surface, of course, this does not seem to have the form of a Moore-paradoxical sentence. However, given the plausible assumption that $T(\beta)$ and $\neg \mathcal{B}T(\beta)$ are logically equivalent, then we get that $\neg \mathcal{B}T(\beta)$ is, in fact, equivalent to $T(\beta) \wedge \neg \mathcal{B}T(\beta)$. Thus a Burge-Buridan sentence, while not having the overt form of a Moore-paradoxical sentence, is equivalent to a Moore-paradoxical sentence." Though G does in fact share features with M, Caie's claim is too strong. For as we have shown, G is equivalent not to the Moore sentence M but to the disjunction of the Moore and Dual Moore, DM.

The sentence on the left side of the disjunction is just the Moore sentence, M, which has been the focal point of much discussion in epistemology. Call the sentence on the right side the Dual Moore or *DM*:

$$\neg A \wedge \Box A$$

5 The Dual Moore

We now turn our investigation to the Dual Moore sentence. As was seen in Sect. 4, *DM* is one disjunct of G, the other being M. Now, M is paradoxical even in the context of knowledge (insofar as M is satisfiable in $\mathsf{S5}$ while $\Box M$ is not). Unlike M, the Dual Moore does not make it to the knowledge context even without a $\Box$. This suggests that *DM* is truly a paradox of belief. In this section, we explore the properties of *DM*. We prove that *DM* is not satisfiable in any extension of T (tying *DM* uniquely to *belief*), that *DM* is satisfiable in $\mathsf{KD45}$, that $\Box DM$ is not satisfiable both $\mathsf{iKD4}$ and $\mathsf{iKD5}$, and that $\Box DM$ is satisfiable in both $\mathsf{K45}$ and KD.

Before moving on, though, we stop to acknowledge a related sentence, the Commissive Moore, $A \wedge \Box\neg A$. This sentence will be the topic of Sect. 6. It will turn out that though *DM* is classically equivalent to the Commissive Moore in one of its forms, it is not intuitionistically equivalent, meaning thus that it is worth examining the Dual and Commissive Moores separately.

First, it is obvious that *DM* is not satisfiable, and hence not paradoxical, in T. We now prove the following:

Theorem 3. *DM is Hintikka Paradoxical over a logic* $L = \mathsf{iK} + X$ *where* $X \subseteq \mathcal{P}$ *iff* X *contains either* $\mathsf{D} + \mathsf{4}$ *or* $\mathsf{D} + \mathsf{5}$.

Proof. This follows from the fact that *DM* is not satisfiable in T and Lemmas 11, 12, 13, 14, and 15. ■

Lemma 11. *DM is satisfiable in* $\mathsf{KD45}$.

Proof. Consider the model $\mathcal{M} = \langle W, \mathcal{R}, \Vdash \rangle$. Let $\mathrm{W} = \{1, 2\}$. Let $\mathcal{R} = \{(1,2),(2,2)\}$. Here $\mathcal{R}$ is transitive, serial, and also euclidean; hence this will be a $\mathsf{KD45}$ model. Let $1 \Vdash \neg A$ and $2 \Vdash A$. Hence, also $1 \Vdash \Box A$. But $\neg A \wedge \Box A$ is *DM*, the Dual Moore sentence (Fig. 4). ■

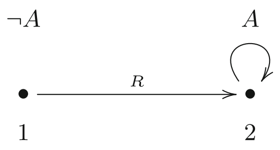


Fig. 4. $\mathsf{KD45}$ model of *DM*

Lemma 12. *$\Box DM$ is not satisfiable in* iKD4.

Proof. It suffices to check that $\mathsf{iKD4} + \Box DM \vdash \bot$

1. $\Box(\neg A \wedge \Box A)$ - Assumption.
2. $\Box\neg A \wedge \Box\Box A$ - K reasoning on 1.
3. $\Box\Box\neg A \wedge \Box\Box A$ - 4 reasoning on 2.
4. $\Box\Box(\neg A \wedge A)$ - K reasoning on 3.
5. $\bot$ - 4 D reasoning on 4. ■

Lemma 13. *$\Box DM$ is not satisfiable in* iKD5.

Proof. It suffices to check that $\mathsf{iKD5} + \Box DM \vdash \bot$

1. $\Box(\neg A \wedge \Box A)$ - Assumption.
2. $\Box\neg A \wedge \Box\Box A$ - K reasoning on 1.
3. $\neg\Box A \wedge \Box\Box A$ - D reasoning on 2.
4. $\Box\neg\Box A \wedge \Box\Box A$ - 5 reasoning on 3.
5. $\Box(\neg\Box A \wedge \Box A)$ - K reasoning on 4.
6. $\neg\Box\bot$ - D.
7. $\bot$ - 5 and 6. ■

Lemma 14. *$\Box DM$ is satisfiable in* K45.

Proof. Consider the model $\mathcal{M} = \langle W, \mathcal{R}, \Vdash\rangle$. Let W $= \{1\}$. Let $\mathcal{R} = \emptyset$. This model is trivially a K45 model since $\mathcal{R}$ is transitive and euclidean. Clearly, $1 \Vdash \Box DM$. ■

This figure is the same as the one in Lemma 5.

Lemma 15. *$\Box DM$ is satisfiable in* KD.

Proof. Consider the model $\mathcal{M} = \langle W, \mathcal{R}, \Vdash\rangle$. Let W $= \{1, 2, 3\}$. Let $\mathcal{R} = \{(1, 2), (2, 3), (3, 3)\}$. This is a D model since $\mathcal{R}$ is serial. Let $2 \Vdash \neg A$ and $3 \Vdash A$. Hence, $1 \Vdash \Box DM$ (Fig. 5). ■

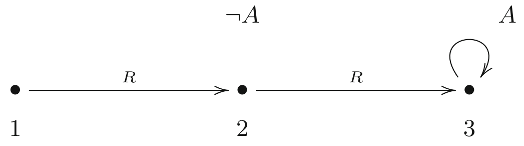


Fig. 5. KD model of $\Box DM$

6 The Commissive Moore

It might be pointed out that DM looks very much like another sentence, one also discussed by Moore [13]: the Commissive Moore, or CM.[10] It is the following:

$$A \wedge \Box\neg A$$

Within classical logic, DM and CM may be regarded equivalent. We should not, however, expect this equivalence to hold intuitionistically. For this reason, CM is worth studying independently.

First, it is immediate that CM is not satisfiable in T.

Theorem 4. *CM is Hintikka Paradoxical over a logic $L = \mathsf{iK} + X$ where $X \subseteq \mathcal{P}$ iff X contains either $\mathsf{D} + \mathsf{4}$, or $\mathsf{D} + \mathsf{5}$.*

Proof. This is clear from the fact that CM is not satisfiable in T, and Lemmas 16, 17, 18, 19, and 20. ■

Lemma 16. *CM is satisfiable in $\mathsf{KD45}$.*

Proof. Consider the model $\mathcal{M} = \langle W, \mathcal{R}, \Vdash\rangle$. Let $\mathrm{W} = \{1, 2\}$. Let $\mathcal{R} = \{(1, 2), (2, 2)\}$. Here $\mathcal{R}$ is transitive, serial, and also euclidean; hence this will be a $\mathsf{KD45}$ model. Let $1 \Vdash A$ and $2 \Vdash \neg A$. Hence, also $1 \Vdash \Box\neg A$. But $A \wedge \Box\neg A$ is CM, the Commissive Moore sentence (Fig. 6). ■

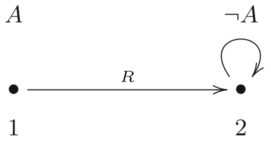


Fig. 6. $\mathsf{KD45}$ model of CM

Lemma 17. *$\Box CM$ is not satisfiable in* $\mathsf{iKD4}$

Proof. It suffices to show that $\mathsf{iKD4} + \Box CM \vdash \bot$.

1. $\Box(A \wedge \Box\neg A)$ - Assumption.
2. $\Box A \wedge \Box\Box\neg A$ - K reasoning on 1.
3. $\Box\Box A \wedge \Box\Box\neg A$ - $\mathsf{4}$ reasoning on 2.
4. $\Box\Box(A \wedge \neg A)$ - K reasoning on 3.
5. $\bot$ - D reasoning on 4. ■

Lemma 18. *$\Box CM$ is not satisfiable in* $\mathsf{iKD5}$

Proof. It suffices to show that $\mathsf{iKD5} + \Box CM \vdash \bot$.

[10] For an overview of its discussion in the literature, see Green and Williams [12].

1. $\Box(A \wedge \Box\neg A)$ - Assumption.
2. $\Box A \wedge \Box\Box\neg A$ - K reasoning on 1.
3. $\neg\Box\neg A \wedge \Box\Box\neg A$ - D reasoning on 2.
4. $\Box\neg\Box\neg A \wedge \Box\Box\neg A$ - 5 reasoning on 3.
5. $\Box(\neg\Box\neg A \wedge \Box\neg A)$ - K reasoning on 4.
6. $\bot$ - D reasoning on 5. ■

Lemma 19. *$\Box CM$ is satisfiable in* K45

Proof. Consider the model $\mathcal{M} = \langle W, \mathcal{R}, \Vdash \rangle$. Let W $= \{1\}$. Let $\mathcal{R} = \emptyset$. Here $\mathcal{R}$ is trivially transitive and euclidean; hence this will be a K45 model. Clearly, $1 \Vdash \Box(A \wedge \Box\neg A)$. ■

This figure is the same as the one in Lemma 5.

Lemma 20. *$\Box CM$ is satisfiable in* KD*.*

Proof. Consider the model $\mathcal{M} = \langle W, \mathcal{R}, \Vdash \rangle$. Let W $= \{1, 2, 3\}$. Let $\mathcal{R} = \{(1, 2), (2, 3), (3, 3)\}$. This is a D model since $\mathcal{R}$ is serial. Let $2 \Vdash A$ and $3 \Vdash \neg A$. Hence, $1 \Vdash \Box CM$ (Fig. 7). ■

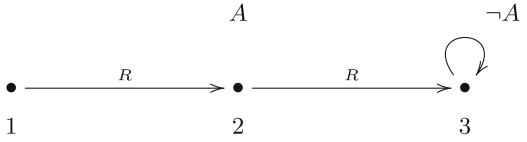


Fig. 7. KD model of $\Box CM$

7 Doxastic Arithmetic

What use can these epistemic-logical insights be put to? An intuitively appealing project is to create a model of the epistemic states of an idealized mathematician. In this section, we outline one method of achieving this goal: *Doxastic Arithmetic*, or DA. We begin by stating DA. As will be seen, the epistemic notion we make use of, $\Box(x)$, interpreted as "the agent believes that x," is explicitly a predicate and *not* a modal connective. We discuss the merits of this approach. Next, we prove some properties about DA, including the facts that $\Box(x)$ respects both proofs and Σ sentences but is not complete with respect to $Pr(x)$. We then prove the fixed point lemma for DA. This illuminates a key difference between DA and systems that make use of $\Box$ as a normal modal connective; in those systems, no such fixed point is provable. We then note that DA has an arithmetical interpretation, hence DA is consistent.

A brief typographical note: following George Boolos [4], p. 45, we make use of square bracket notation for Gödel numbers of formulas. Where $P(x)$ is some predicate in the language of DA and F is a formula of DA with n free variables, the formula $P[F]$ is a formula of DA with the same n variables free as F. Note thus that if F is a sentence, then $P[F]$ also has no free variables.

Where A is an atomic formula of Peano Arithmetic, PA, and F and G are formulas of DA, the following provides the syntax of DA:

$$\Phi = \bot \mid A \mid F \rightarrow G \mid F \wedge G \mid F \vee G \mid \neg F \mid \exists x F \mid \forall x F \mid \Box[F]$$

Where s is a term, formula, or symbol of DA, the Gödel number of s is $\ulcorner s \urcorner$. Note that $\Box(x)$ is a fresh designated unary predicate, and not a connective. This distinguishes the current approach from those that make use of a $\Box$ that is a connective (see, for instance, Shapiro in [14] who takes the connective approach for different reasons). The consequences of this will be seen in Theorems 7 and 8.

The axioms and rules of DA are defined as follows:

1. Axioms and rules of PA
2. $\Box[F \rightarrow G] \rightarrow (\Box[F] \rightarrow \Box[G])$
3. $\vdash F$ then $\vdash \Box[F]$
4. $\neg\Box[\bot]$

(1) is self-explanatory. We see in (2) an expression of K. We understand this as the assumption that our mathematician's beliefs are closed with respect to deduction. This does involve some idealization, but this much seems intuitive. (3) says that if $\mathsf{DA} \vdash F$, that is, F is provable in DA using no assumptions, then our agent believes that F. This is the analog of the modal Necessitation rule. Lastly, (4) is the built-in consistency axiom. We read it as "The agent does not believe that a contradiction holds."

At this point we are in the position to prove some properties of DA. First, we will see that $\Box(x)$ respects proof, which is to say that if t is a specific proof of F in DA, then our agent believes F. We symbolize that t is a proof of F with $\mathit{Proof}(t, \ulcorner F \urcorner)$. More formally, this says that t is a finite sequence of axioms, formulas used in Modus Ponens, or ones in Universal Generalization.

Theorem 5. Belief Respects Proof: *For each t,* $\mathsf{DA} \vdash \mathit{Proof}(t, \ulcorner F \urcorner) \rightarrow \Box[F]$

Proof. There are two cases: either $\mathit{Proof}(t, \ulcorner F \urcorner)$ is true or it is not. If $\mathit{Proof}(t, \ulcorner F \urcorner)$ is true, then it holds that t is a code for a proof of F. But then $\mathsf{DA} \vdash F$. By DA's Necessitation, it follows that $\mathsf{DA} \vdash \Box[F]$, and so $\mathsf{DA} \vdash \mathit{Proof}(t, \ulcorner F \urcorner) \rightarrow \Box[F]$. If $\mathit{Proof}(t, \ulcorner F \urcorner)$ is not true, then $\neg\mathit{Proof}(t, \ulcorner F \urcorner)$ is true. Because $\neg\mathit{Proof}(t, \ulcorner F \urcorner)$ is a provably primitive recursive formula, it follows that $\mathsf{DA} \vdash \neg\mathit{Proof}(t, \ulcorner F \urcorner)$. For any G, then, $\mathsf{DA} \vdash \mathit{Proof}(t, \ulcorner F \urcorner) \rightarrow G$. Thus, $\mathsf{DA} \vdash \mathit{Proof}(t, \ulcorner F \urcorner) \rightarrow \Box[F]$. ■

Next, we note that $\Box(x)$ is complete with respect to Σ sentences.

Theorem 6. Σ Completeness of Belief: *For each Σ sentence σ,* $\mathsf{DA} \vdash \sigma \rightarrow \Box[\sigma]$

Proof. By Theorem 5, we know that $\mathsf{DA} \vdash \mathit{Proof}(t, \ulcorner \sigma \urcorner) \rightarrow \Box[\sigma]$ But then, because it holds that $\mathsf{DA} \vdash \sigma \rightarrow \mathit{Proof}(t, \ulcorner \sigma \urcorner)$ (see, for instance, Boolos' [4], pp. 46–49), we have thus shown that $\Box(x)$ is Σ complete. ■

Interestingly, the fixed point lemma is also provable for $\neg\Box(x)$.[11]

Theorem 7. Fixed Point Lemma for $\Box$: *For some* DA*-formula* G*, it holds that* $\mathsf{DA} \vdash G \leftrightarrow \neg\Box[G]$.

Proof. We already know that for each predicate $P(x)$ in the language of PA, there is a formula G for which it holds that (cf., Boolos [4], p. 53):

$$\mathsf{PA} \vdash G \leftrightarrow P[G]$$

Then, DA can prove this as well with $\neg\Box(x)$. We thus get the fixed point lemma for $\neg\Box(x)$ in DA, that for some G:

$$\mathsf{DA} \vdash G \leftrightarrow \neg\Box[G]$$

■

This brings out an important difference between the approach taken in this paper, that of taking $\Box(x)$ to be a predicate, and that of interpreting $\Box$ as a connective. The fixed point lemma does not hold in systems with $\Box$ as a modality compatible with $\mathsf{D} + \mathsf{4}$.

Theorem 8. *For every normal modal logic* L*, where* $\mathsf{LD4} \nvdash \bot$*, there is no* A *such that* $\mathsf{L} \vdash A \leftrightarrow \neg\Box A$

Proof. Assume that $\mathsf{L} \vdash A \leftrightarrow \neg\Box A$, then $\mathsf{LD4} \vdash A \leftrightarrow \neg\Box A$. But then, $\mathsf{LD4} \vdash \Box(A \leftrightarrow \neg\Box A)$ by $\mathsf{Necessitation}$. In Lemma 8, we saw that $\mathsf{LD4} \vdash \neg\Box(A \leftrightarrow \neg\Box A)$. So, $\mathsf{LD4} \vdash \bot$. Since $\mathsf{LD4} \nvdash \bot$, it follows thus that $\mathsf{L} \nvdash A \leftrightarrow \neg\Box A$. ■

We now define an arithmetical interpretation * of DA.

Definition 1. *An arithmetical interpretation of* DA *in* PA *is a pair* $(B(x),\ ^*)$ *in which* $B(x)$ *is an arithmetical predicate such that for any* PA*-formulas* F *and* G *it holds that:*

1. $\mathsf{PA} \vdash B[F \rightarrow G] \rightarrow (B[F] \rightarrow B[G])$
2. $\mathsf{PA} \vdash F$ *then* $\mathsf{PA} \vdash B[F]$
3. $\mathsf{PA} \vdash \neg B[\bot]$
And * *is a mapping from the language of* DA *to that of* PA *such that:*
4. $F^* = F$*, for each* $\Box$*-free formula*
5. * *commutes with Boolean connectives and quantifiers.*
6. $(\Box[F])^* = B[F^*]$

We now prove that DA has an interpretation in PA.

Lemma 21. DA *has an interpretation in* PA.

[11] We follow the proofs in Boolos' *The Logic of Provability* [4], pp. 53–54 and Hartry Field's *Saving Truth from Paradox* [9], pp. 26–27.

Proof. Consider now one of the systems with non-Gödelian proof predicates. Take Sol Feferman's system F (see [8] and Albert Visser's [16], pp. 173–178, esp. 174. See [16] also for other options of suitable systems with non-Gödelian proof predicates.). Feferman's system has a predicate $\Delta(x)$ which satisfies the following:

$1f$. $\mathsf{PA} \vdash \Delta[F \rightarrow G] \rightarrow (\Delta[F] \rightarrow \Delta[G])$
$2f$. $\mathsf{PA} \vdash F$ then $\mathsf{PA} \vdash \Delta[F]$
$3f$. $\mathsf{PA} \vdash \neg\Delta[\bot]$

Thus, interpreting $B(x)$ as $\Delta(x)$, we see that DA in fact has an interpretation in PA. ■

Lemma 22. *Let* $(B(x), {}^*)$ *be an interpretation of* DA *in* PA*. Then, for any* DA*-formula* F*, if* $\mathsf{DA} \vdash F$*, then* $\mathsf{PA} \vdash F^*$*.*

Proof. Induction on derivability in DA, that is, assume $\mathsf{DA} \vdash F$ and consider the following cases:

Case 1: F is an instance of a logical or arithmetical axiom. Then, F^* is also an instance of the same logical or arithmetical axiom. So $\mathsf{PA} \vdash F^*$.

Case 2: F is an instance of modal axiom 1 or 3. By Definition 1 it is clear that $\mathsf{PA} \vdash F^*$.

Case 3: F follows by Modus Ponens or Universal Generalization. These are the same in both PA and DA. In the first case, we assume that $\mathsf{DA} \vdash G$, $\mathsf{DA} \vdash G \rightarrow F$ and also that $\mathsf{PA} \vdash G^*$, $\mathsf{PA} \vdash G^* \rightarrow F^*$. But then it is clear that $\mathsf{PA} \vdash F^*$. The argument is similar for Universal Generalization.

Case 4: F follows by DA's Necessitation. That is, $F = \Box G$ and $\mathsf{DA} \vdash \Box G$. By induction hypothesis, $\mathsf{PA} \vdash G^*$. It follows then that $\mathsf{PA} \vdash B[G^*]$. But $(\Box[G])^* = B[G^*]$. So, $\mathsf{PA} \vdash (\Box[G])^*$. Hence, $\mathsf{PA} \vdash F^*$. ■

There is an important corollary to be drawn. Namely, the consistency of DA:

Corollary 1. Consistency of DA: $\mathsf{DA} \nvdash \bot$

Proof. By Lemma 22, if $\mathsf{DA} \vdash \bot$ then $\mathsf{PA} \vdash \bot^*$. Since $\bot^* = \bot$, it would follow that $\mathsf{PA} \vdash \bot$. But, we know that $\mathsf{PA} \nvdash \bot$.[12] So, $\mathsf{DA} \nvdash \bot$. ■

Insofar as Robinson Arithmetic Q is sufficient for the fixed point lemma, Q taken along with a KD4 unary predicate $\Box(x)$ will be inconsistent. The proof of this that we will present is nearly identical to that of Lemma 8; due to the significance of Lemma 23, we include the proof here too.

Lemma 23. KD4 + Q is inconsistent.

Proof. It suffices to show that $\mathsf{K4} + \mathsf{D} + \mathsf{Necessitation} + G \vdash \bot$.

1. $\Box\bot \rightarrow \bot$ - D.

[12] See [3] for some discussion.

2. $A \to \neg\Box A$ - First half of G.
3. $\Box(A \to \neg\Box A)$ - Necessitation on 2.
4. $\Box A \to \Box\neg\Box A$ - K reasoning on 3.
5. $\Box A \to (\Box A \wedge \Box\neg\Box A)$ - Propositional reasoning on 4.
6. $\Box A \to (\Box\Box A \wedge \Box\neg\Box A)$ - 4 reasoning on 5.
7. $\Box A \to \Box(\Box A \wedge \neg\Box A)$ - K reasoning on 6.
8. $\Box A \to \Box\bot$ - 7.
9. $\Box A \to \bot$ - 1 and 8.
10. $\neg\Box A$ - 9.
11. $\neg\Box A \to A$ - Second half of G.
12. A - Modus Ponens with 10 and 11.
13. $\Box A$ - Necessitation on 12.
14. $\bot$ - 10 and 13. ■

A corollary of Lemma 23, then, is that there is no interpretation of a predicate $\Box(x)$ with these properties in PA:

Corollary 2. *There is no interpretation of $\Box(x)$ in PA, where $\Box(x)$ is understood explicitly as a KD4 predicate.*

Proof. Assume that there were such an interpretation. By the fixed point lemma, we would be able to construct a G such that PA $\vdash G \leftrightarrow \neg\Box[G]$. By Lemma 23, this would mean that PA $\vdash \bot$. But PA is not inconsistent. So, there is no interpretation of $\Box(x)$ in PA. ■

It is worth noting some advantages that DA will enjoys. First, DA allows for a fixed point lemma for $\Box(x)$, as we saw in Theorem 7, and therefore incorporates Gödel's argument. Contrast this with systems that make use of $\Box$ as a modal connective, which we saw in Theorem 8 will not admit of any fixed point provided they are normal and consistent with KD4.

It might be objected: our agent should also believe that they believe everything that they believe. So, since $\Box(x)$ of DA does not have an analog of the 4 axiom, DA falls short as a representation of such an agent.

But should an agent simply believe that they believe everything that they believe? For at least one reason, related to the sorts of reasons that motivate explicit approaches to epistemic logic more generally, the answer seems to be "no." The idea would be that, following reasoning in Arthemov's [1], p. 6, we do not simply grant that because our agent believes in F based on some evidence, they also believe that they believe in F based on that evidence. The thought is that some further justification or piece of evidence would be required to warrant this higher order belief.

Is it obvious that we should build an evidence based belief predicate into the model of our agent? For instance, the argument might go, "think of people from (place that is viewed negatively). They believe all sorts of falsehoods for no reason!" That such a tendency to dismiss the beliefs of others as being without reason exists is undeniable. The question is whether or not this sort of dismissal is metaphorical in some sense. After all, it does seem plausibly interpreted instead as "those people believe all sorts of falsehoods for terrible reasons" or even "those people believe all sorts of falsehoods for no *good* reasons." Hence it seems not implausible—or at least this specific appeal to natural language does not make it seem implausible—to endorse an evidence based picture of belief.

Another advantage of DA is that it underlies the class of arithmetical provability predicates with built-in consistency. These systems are studied extensively in Visser's [16] and Shavrukov's [15]. As DA is not tied down to any one specific arithmetic interpretation of $\Box(x)$, it is an abstract and more general version of those systems.

The current formulation of DA is intentionally kept independent of any specific proof predicate to provide a code-free axiomatization of arithmetical beliefs. Two natural candidates, however, come to mind for connecting the belief modality with standard proof predicates:

$$Pr[F] \to \Box[F] \tag{1}$$

or

$$\Box[F] \to Pr[F] \tag{2}$$

for the Gödel proof predicate $Pr(x)$ in the reference system. However, (1) makes this version of DA inconsistent insofar as $\vdash \neg\Box[\bot] \to \neg Pr[\bot]$ and $\vdash \neg\Box[\bot]$ yields $\vdash \neg Pr[\bot]$. This would mean that the Gödel consistency of the reference system is internally provable, which, by Gödel's Incompleteness Theorem, in turn yields the inconsistency of the reference system. Adding (2) is consistent, however, since both the Rosser and Feferman provability predicates satisfy (2) (see [16], pp. 166, 169, and 173–175).

8 Conclusions

In this paper we have discovered the minimal conditions in which *M*, *G*, *DM* and *CM* are paradoxical in Hintikka's sense. The Moore sentence was minimally paradoxical in iKT and iKD4. Like the Moore, the Gödel-Buridan sentence was also minimally paradoxical in iKT and iKD4. Interestingly, the Dual Moore sentence was minimally paradoxical in iKD4 or iKD5, but *not* iKT. The Commissive Moore was also minimally paradoxial in iKD4 and iKD5. We noted that classically, the Dual and Commissive Moores will be equivalent though clearly this equivalence will fail to hold in the intuitionistic context.

One line of analysis suggested by the first part of this paper was pursued in the second part of this paper. Specifically, we built a system DA of Doxastic Arithmetic that models the agent's arithmetical beliefs. The agent accepts all specific arithmetical proofs and yet believes in the consistency of their own beliefs. We proved important properties of this system, including its fixed point lemma and its consistency, and then discussed its merits and motivations.

References

1. Artemov, S.: Explicit provability and constructive semantics. Bull. Symbolic Logic **7**, 1–36 (2001)
2. Artemov, S.: Knowing the model. Published online at: arXiv:1610.04955 [math.LO] (2016)
3. Bernays, P.: On the original Gentzen proof for number theory. In: Intuitionism and Proof Theory. Proceedings of the Summer Conference at Buffalo, NY, 1968, vol. 60, pp. 409–417 (1970)
4. Boolos, G.: The Logic of Provability. Cambridge University Press, Cambridge (1995)
5. Burge, T.: Buridan and epistemic paradox. Philos. Stud. Int. J. Philos. Anal. Tradit. **34**, 21–35 (1978)
6. Caie, M.: Belief and indeterminacy. Philos. Rev. **121**, 1–54 (2012)
7. Caie, M.: Doxastic logic. In: Weisberg, J., Pettigrew, R. (eds.) Open Handbook of Formal Epistemology (2017). Forthcoming. https://sites.google.com/site/caiemike/
8. Feferman, S.: Arithmetization of metamathematics in a general setting. Fundamenta Mathematica **49**, 35–92 (1960)
9. Field, H.: Saving Truth from Paradox. Oxford University Press, Oxford (2008)
10. Hintikka, J.: Knowledge and Belief: An Introduction to the Logic of the Two Notions. Cornell University Press, Ithaca (1962)
11. Holliday, W., Icard, III, T.F.: Moorean phenomena in epistemic logic. In: Goranko, V., Shehtman, V. (eds.) Advances in Modal Logic, vol. 8, pp. 178–199 (2010)
12. Green, M., Williams, J.N.: Introduction. In: Green, M., Williams, J.N. (eds.) Moore's Paradox: New Essays on Belief, Rationality, and the First Person. Oxford University Press (2007)
13. Moore, G.E.: Russell's theory of descriptions. In: Schilpp, P. (ed.) The Philosophy of G.E. Moore, pp. 175–225 (1944)
14. Shapiro, S.: Epistemic and intuitionistic arithmetic. Stud. Logic Found. Math. **118**, 11–46 (1985)
15. Shavrukov, V.Y.: A smart child of Peano's. Notre Dame J. Formal Logic **35**(2), 161–185 (1994)
16. Visser, A.: Peano's smart children: a provability logical study of systems with built-in consistency. Notre Dame J. Formal Logic **30**(2), 161–196 (1998)

A Natural Proof System for Herbrand's Theorem

Benjamin Ralph(✉)

University of Bath, Bath, UK
b.d.ralph@bath.ac.uk
http://people.bath.ac.uk/bdr25

Abstract. The reduction of undecidable first-order logic to decidable propositional logic via Herbrand's theorem has long been of interest to theoretical computer science, with the notion of a Herbrand proof motivating the definition of expansion proofs. The problem of building a natural proof system around expansion proofs, with composition of proofs and cut-free completeness, has been approached from a variety of different angles. In this paper we construct a simple deep inference system for first-order logic, KSh2, based around the notion of expansion proofs, as a starting point to developing a rich proof theory around this foundation. Translations between proofs in this system and expansion proofs are given, retaining much of the structure in each direction.

Keywords: Structural proof theory · First-order logic
Deep inference · Herbrand's theorem · Expansion proofs

1 Introduction

A focus on the existential witnesses created in proofs has long been central to first-order proof theory. If one ignores all other information about a first-order proof except for the details of existential introduction rules, one still has an important kernel of the proof, in some sense the part of the proof that is inherently first-order, as opposed to merely propositional. Herbrand, in [13], innovated an approach to first-order proof theory that isolates this first-order content of the proof, and today the notion of a *Herbrand proof* is common, a proof-theoretic object that shows the carrying out of the following four steps, usually but not always in this order:

1. Expansion of existential subformulae.
2. Prenexification/elimination of universal quantifiers.
3. Term assignment.
4. Propositional tautology check.

This research has been supported by EPSRC Project EP/K018868/1 *Efficient and Natural Proof Systems* and ANR project FISP ANR-15-CE25-0014-01.

S. Artemov and A. Nerode (Eds.): LFCS 2018, LNCS 10703, pp. 289–308, 2018.
https://doi.org/10.1007/978-3-319-72056-2_18

For example, we have the following theorem from [8] which exactly follows this scheme:

Theorem 1 (Herbrand's theorem). *A first-order formula A is valid if and only if A has a Herbrand proof. A Herbrand proof of A consists of a prenexification $A^\star$ of a strong $\vee$-expansion of A plus a witnessing substitution σ for $A^\star$.*

Or take the presentation of Herbrand's theorem in a deep inference system in [7]:

Theorem 2 (Herbrand's theorem). *For each proof of a formula S in system* SKSgr *there is a substitution σ, a propositional formula P, a context $Q\{\ \}$ consisting only of quantifiers and a* Herbrand proof*:*

$$\begin{array}{l} \quad\overline{\Vert}\,{\scriptstyle \mathsf{KS}\cup\{\mathsf{ai}\uparrow\}} \\ \forall \boldsymbol{x} P\sigma \\ \quad\Vert\,{\scriptstyle \{\mathsf{n}\downarrow\}} \\ Q\{P\} \\ \quad\Vert\,{\scriptstyle \{\mathsf{gr}\downarrow\}} \\ \quad S' \\ \quad\Vert\,{\scriptstyle \{\mathsf{qc}\downarrow\}} \\ \quad S \end{array}$$

One obvious difference between the two formulations is that while the first definition of a Herbrand proof does not involve a proof in any commonly used proof system, the second definition is based around a factorisation of a proof in deep inference. Thus, the second definition gives us more opportunities to manipulate, compose and identify Herbrand proofs as proof theoretic objects.

The basic observation of this paper is that defining Herbrand proofs in a deep inference setting is easier and more natural than doing so in Gentzen-style systems (in particular the Sequent Calculus and Natural Deduction). This is because the steps (1), (2) and (3) as defined above are standard inference rules in first-order deep inference proof systems, and, while it is obviously possible to include them as *ad hoc* rules, they are not natural for Gentzen-style systems, especially carried out in this order.

To put it another way: if we want to build a proof theory around Herbrand's theorem, in which the propositional and first-order content of a cut-free proof is separated in a natural way, then deep inference is a superior setting to the sequent calculus, in some concrete senses. To substantiate this claim, we define two inter-translatable classes of deep inference proofs. The first class comprises analytic Herbrand proofs, defined similarly to those in Theorem 2 above, and we borrow a result from [7] to show that the class is complete for FOL. We show a tight correspondence between the second class and expansion proofs, a minimalistic formalism for first-order (and higher-order) proofs that ignores all but the most essential first-order structure. This correspondence suggests the second class as a good candidate for canonical first-order proofs. Therefore, the translation between the two classes enables us to see Herbrand proofs as canonical first-order proofs.

It should be noted that, while a translation between expansion proofs and first-order deep inference proofs has not previously been shown, Straßburger has developed a notion of expansion proofs for MLL2, and provided a similar translation between these structures and a deep inference proof system for that logic [18,19].

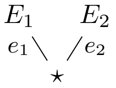
2 Expansion Proofs

In [17], Miller generalises the concept of the Herbrand expansion to higher order logic, representing the witness information in a tree structure, and explicit transformations between these 'expansion proofs' and cut-free sequent proofs are provided. Miller's presentation of expansion proofs lacked some of the usual features of a formal proof system, crucially composition by an eliminable cut, but extensions in this direction have been carried out by multiple authors. In [12], Heijltjes presents a system of 'proof forests', a graphical formalism of expansion proofs with cut and a local rewrite relation that performs cut elimination. Similar work has been carried out by McKinley [16] and more recently by Hetzl and Weller [14] and Alcolei et al. [1]. As expansion proofs and the related formalisms only represent the first-order content of a proof, we will first define expansion proofs in order to guide the definition of the proof systems.

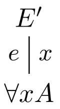
Remark 1. Throughout the paper, we use $\star$ in place of $\wedge$ and $\vee$, and Q in place of $\forall$ and $\exists$ if both cases can be combined into one. For clarity, we will sometimes distinguish between connectives in expansion trees, $\star_E$, and in formulae/derivations, $\star_F$.

Definition 1. *We define* expansion trees*, the two functions Sh (shallow) and Dp (deep) from expansion trees to formulae, a set of* eigenvariables $EV(E)$ *for each expansion tree, and a partial function Lab from edges to terms, following [9,12,17]:*

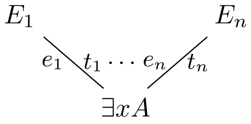
1. *Every literal A (including the units* t *and* f*) is an expansion tree. $Sh(A) := A$, $Dp(A) := A$, and $EV(A) = \emptyset$.*
2. *If E_1 and E_2 are expansion trees with $EV(E_1) \cap EV(E_2) = \emptyset$, then $E_1 \star E_2$ is an expansion tree, with $Sh(E_1 \star_E E_2) := Sh(E_1) \star_F Sh(E_2)$, $Dp(E_1 \star_E E_2) := Dp(E_1) \star_F Dp(E_2)$, and $EV(E_1 \star E_2) = EV(E_1) \cup EV(E_2)$. We call $\star$ a $\star$-*node *and each unlabelled edge e_i connecting the $\star$-node to E_i a $\star$-*edge*. We represent $E_1 \star E_2$ as:*



3. *If E' is an expansion tree s.t. $Sh(E') = A$ and $x \notin EV(E')$, then $E = \forall x A +^x E'$ is an expansion tree with $Sh(E) := \forall x A$, $Dp(E) := Dp(E')$, and $EV(E) := EV(E') \cup \{x\}$. We call $\forall x A$ a $\forall$-*node *and the edge e connecting the $\forall$-node and E' a $\forall$-*edge*, with $Lab(e) = x$. We represent E as:*



4. *If $t_1, \ldots, t_n$ are terms ($n \geq 0$), and $E_1, \ldots, E_n$ are expansion trees s.t. $x \notin EV(E_i)$ and $EV(E_i) \cap EV(E_j) = \emptyset$ for all $1 \leq i < j \leq n$, and $Sh(E_i) = A\{x \Leftarrow t_i\}$, then $E = \exists xA +^{t_1} E_1 +^{t_2} \cdots +^{t_n} E_n$ is an expansion tree, where $Sh(E) := \exists xA$, $Dp(E) := Dp(E_1) \vee \cdots \vee Dp(E_n)$, and $EV(E) = \bigcup_1^n EV(E_i)$. We call $\exists xA$ an* $\exists$-node *and each edge e_i connecting the $\exists$-node with E_i an* $\exists$-edge*, with $Lab(e_i) = t_i$. We represent E as:*



Remark 2. Let ρ be a permutation of $[1 \ldots n]$. We consider the expansion trees $\exists xA +^{t_1} E_1 +^{t_2} \cdots +^{t_n} E_n$ and $\exists xA +^{t_{\rho(1)}} E_{\rho(1)} \cdots +^{t_{\rho(n)}} E_{\rho(n)}$ equal. Our trees are also presented the other way up to usual, e.g. [12]. This is so that they are the same way up as the deep inference proofs we will translate them to below.

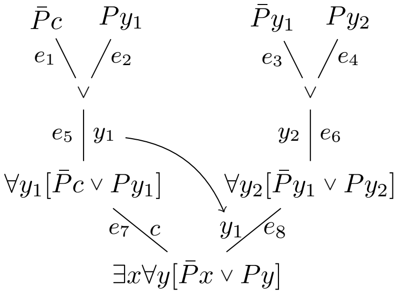
Definition 2. *Let E be an expansion tree and let $<_E^-$ be the relation on the edges in E defined by:*

- *$e <_E^- e'$ if the node directly below e is the node directly above e'.*
- *$e <_E^- e'$ if e is an $\exists$-edge with $Lab(e) = t$, there is an x which is free in t, e' is a $\forall$-edge and $Lab(e') = x$. In this case, we say e'* points to *e.*

The dependency relation *of E, $<_E$, is the transitive closure of $<_E^-$.*

Definition 3. *An expansion tree E is* correct *if $<_E$ is acyclic and $Dp(E)$ is a tautology. We can then call E an* expansion proof *of $Sh(E)$.*

Example 1. Below is an expansion tree E, with $Sh(E) = \exists x \forall y[\bar{P}x \vee Py]$ and $Dp(E) = [\bar{P}c \vee Py_1] \vee [\bar{P}y_1 \vee Py_2]$. The tree is presented with all edges explicitly named, to define the dependency relation below, as well as the labels for the $\exists$-edges and $\forall$-edges.


$\bar{P}c$
Py_1
$\bar{P}y_1$
Py_2
e_1
e_2
e_3
e_4
$\vee$
$\vee$
e_5
y_1
y_2
e_6
$\forall y_1[\bar{P}c \vee Py_1]$
$\forall y_2[\bar{P}y_1 \vee Py_2]$
e_7
c
y_1
e_8
$\exists x \forall y[\bar{P}x \vee Py]$

The dependency relation is generated by the following inequalities: $e_3, e_4 < e_6 < e_8$ and $e_1, e_2 < e_5 < e_7$ and $e_8 < e_5$. e_5 points to e_8. As this dependency relation is acyclic and $[\bar{P}c \vee Py_1] \vee [\bar{P}y_1 \vee Py_2]$ is a tautology, E is correct, and thus an expansion proof.

3 Proof Systems

3.1 Motivation for the Proof Systems

What features would a proof system, PS, designed around Expansion Proofs, EP, have? Say we have a translation $\pi : EP \to PS$.

Firstly, we might want that composition of proofs in PS matches closely to composition of expansion proofs, that something close to functoriality of π holds:

$$\pi(E_1 \star_E E_2) \approx \pi(E_1) \star_F \pi(E_2)$$

For Gentzen-style systems this will prove difficult, as there is no natural way to compose two proofs by disjunction.

A second attractive feature would be that we could isolate a part of the proof system that is relevant to Herbrand's theorem, stating and proving it as a factorisation of proofs, where the first order content of the proof is isolated from the propositional content:

$$\pi(E) = \begin{array}{c} {\scriptstyle \pi^{Up}(E)} \Big\| {\scriptstyle Prop} \\ Dp(E) \\ {\scriptstyle \pi^{Lo}(E)} \Big\| {\scriptstyle FO} \\ Sh(E) \end{array}$$

Interestingly, this is impossible in the usual sequent calculus systems, which we can see by considering Brünnler's second restriction on contraction. Consider the following property of proof systems:

"Proofs can be separated into two phases (seen bottom-up): The lower phase only contains instances of contraction. The upper phase contains instances of the other rules, but no contraction. No formulae are duplicated in the upper phase" [6].

Brünnler shows that a standard sequent calculus proof system with multiplicative rules cannot satisfy this property. The suggested way round this restriction is to use systems with *deep contraction*. In fact, this restriction on sequent calculus systems is shown by McKinley in [15] to create a gap in Buss's proof of Herbrand's theorem in [8]. The faulty proof assumes that if one restricts contraction to only existential formulae, one retains completeness (assuming a multiplicative $\wedge R$ rule). That this is false can be seen by considering the sequent below, where the application of any multiplicative $\wedge R$ rule leads to an invalid sequent:

$$\vdash \forall x A \wedge \forall x B, \left[\exists x \bar{A} \vee \exists x \bar{B}\right] \wedge \left[\exists x \bar{A} \vee \exists x \bar{B}\right]$$

Thus to achieve the desired factorisation property, we either need to add unrestricted contraction by the back-door, say by using additive $\wedge R$ rules, or use a system with deep contraction.

Both the above considerations suggest the formalism of *open deduction* for proofs of Herbrand's theorem [10]. It is a deep inference formalism which allows for free composition of proofs by $\wedge$ and $\vee$ at the propositional level, satisfying the first desideratum. Also, the deep contraction rule is certainly a natural rule for a deep inference system, allowing us to satisfy the second desideratum.

3.2 Open Deduction

As discussed above, we will work in the open deduction formalism. Open deduction differs from the sequent calculus in that we build up complex derivations with connectives and quantifiers in the same way that we build up formulae [10]. We can compose two derivations horizontally with $\star$, quantify over derivations, and compose derivations vertically with an inference rule.

Definition 4. *An open deduction derivation is inductively defined in the following way:*

- *Every atom $Pt_1 \ldots t_n$ is a derivation, where P is an n-ary predicate, and t_i are terms. The units* t *and* f *are also derivations.*

If $\begin{array}{c} A \\ \phi \,\Vert \\ B \end{array}$ *and* $\begin{array}{c} C \\ \psi \,\Vert \\ D \end{array}$ *are derivations, then:*

- $\begin{array}{c} A \star C \\ \phi\star\psi \,\Vert \\ B \star D \end{array} = \begin{array}{c} A \\ \phi \,\Vert \\ B \end{array} \star \begin{array}{c} C \\ \psi \,\Vert \\ D \end{array}$ *and* $\begin{array}{c} QxA \\ Qx\phi \,\Vert \\ QxB \end{array} = Qx \begin{bmatrix} A \\ \phi \,\Vert \\ B \end{bmatrix}$ *are derivations.*
- $\begin{array}{c} A \\ \chi \,\Vert \\ D \end{array} = \begin{array}{c} A \\ \phi \,\Vert \\ \rho \dfrac{B}{C} \\ \psi \,\Vert \\ D \end{array}$ *is a derivation, if* $\rho \dfrac{B}{C}$ *is an instance of ρ.*

When we write $\begin{array}{c} A \\ \phi \,\Vert\, S \\ B \end{array}$*, it means that every inference rule in ϕ is an element of the set S or an equality rule.*

Remark 3. Formulae are just derivations built up with no vertical composition. Open deduction and the *calculus of structures* (the better known deep inference formalism) polynomially simulate each other [11].

Definition 5. *We define a* section *of a derivation in the following way:*

- *Every atom a has one section, a.*
- *If A is a section of ϕ, and B is a section of ψ, then $A \star B$ is a section of $\phi \star \psi$, and QxA is a section of $Qx\phi$.*
- *If A is a section of* $\begin{array}{c} B \\ {\scriptstyle \phi_1}\,\| \\ C \end{array}$ *or* $\begin{array}{c} D \\ {\scriptstyle \phi_2}\,\| \\ E \end{array}$ *and* $\phi = \begin{array}{c} B \\ {\scriptstyle \phi_1}\,\| \\ {\scriptstyle \rho}\,\dfrac{C}{D} \\ {\scriptstyle \phi_2}\,\| \\ E \end{array}$ *then A is a section of ϕ.*

The premise and conclusion of a derivation are, respectively, the uppermost and lowermost section of the derivation. A proof *of A is a derivation with premise* t *and conclusion A, sometimes written* $\begin{array}{c} {\scriptstyle \phi}\,\overline{\|} \\ A \end{array}$.

Definition 6. *We define the rewriting system* Seq *as containing the following two rewrites S_l and S_r:*

$$\begin{array}{c} A \\ \| \\ =\dfrac{K\left\{ {\scriptstyle \rho_1}\dfrac{A_1}{B_1}\right\}\{A_2\}}{K\{B_1\}\left\{ {\scriptstyle \rho_2}\dfrac{A_2}{B_2}\right\}} \\ \| \\ B \end{array} \quad \xleftarrow{S_l} \quad \begin{array}{c} A \\ \| \\ K\left\{ {\scriptstyle \rho_1}\dfrac{A_1}{B_1}\right\}\left\{ {\scriptstyle \rho_2}\dfrac{A_2}{B_2}\right\} \\ \| \\ B \end{array} \quad \xrightarrow{S_r} \quad \begin{array}{c} A \\ \| \\ =\dfrac{K\{A_1\}\left\{ {\scriptstyle \rho_2}\dfrac{A_2}{B_2}\right\}}{K\left\{ {\scriptstyle \rho_1}\dfrac{A_1}{B_1}\right\}\{B_2\}} \\ \| \\ B \end{array}$$

If ϕ is in normal form w.r.t. Seq*, we say ϕ is in* sequential form. *If $\phi \longrightarrow^*_{\mathsf{Seq}} \psi$ and ψ is in sequential form, we say that ψ is a* sequentialisation *of ϕ.*

Proposition 1. *A derivation ϕ is in sequential form iff. It is in the following form, where ρ_i are all the non-equality rules:*

$$\begin{array}{rc} & A \\ \hline = & K_1\left\{ {\scriptstyle \rho_1}\dfrac{A_1}{B_1}\right\} \\ \hline = & \vdots \\ \hline = & K_n\left\{ {\scriptstyle \rho_n}\dfrac{A_n}{B_n}\right\} \\ \hline = & B \end{array}$$

Definition 7. *A* closed *derivation is one where every section of the derivation is a sentence (i.e. a formula with no free variables).*

3.3 KSh1 and Herbrand Proofs

We can now define an open deduction proof system for Herbrand proofs. We will extend the propositional system KS [5]:

$$\mathsf{KS} = \begin{array}{c} \boxed{\begin{array}{c} \begin{array}{ccc} \mathsf{ai}{\downarrow}\,\dfrac{\mathsf{t}}{a \vee \bar{a}} & \mathsf{ac}{\downarrow}\,\dfrac{a \vee a}{a} & \mathsf{aw}{\downarrow}\,\dfrac{\mathsf{f}}{a} \end{array} \\ \\ \begin{array}{cc} \mathsf{s}\,\dfrac{A \wedge [B \vee C]}{(A \wedge B) \vee C} & \mathsf{m}\,\dfrac{(A \wedge B) \vee (C \wedge D)}{[A \vee C] \wedge [B \vee D]} \end{array} \end{array}} \\ + \\ \boxed{\begin{array}{ccc ccc} A \wedge \mathsf{t} & = & A & A \vee \mathsf{f} & = & A \\ \mathsf{t} \vee \mathsf{t} & = & \mathsf{t} & \mathsf{f} \wedge \mathsf{f} & = & \mathsf{f} \\ A \wedge (B \wedge C) & = & (A \wedge B) \wedge C & A \vee [B \vee C] & = & [A \vee B] \vee C \\ A \wedge B & = & B \wedge A & A \vee B & = & B \vee A \end{array}} \end{array}$$

We will write $=\dfrac{A}{B}$ if B can be obtained from A by equality rules (treating multiple instances of different equality rules as one instance of a general equality rule).

We now introduce rules for the first three steps of a Herbrand Proof:

1. For expansion of existential subformulae, we have the rule:

$$\mathsf{qc}{\downarrow}\,\frac{\exists x A \vee \exists x A}{\exists x A}$$

For technical reasons, we insist that all three instances of $\exists x A$ in $\mathsf{qc}{\downarrow}$ are α-equivalent with unique bound variables, but as above we will sometimes refer to them all as $\exists x A$ for simplicity.

2. For prenexification, we have four rules, where B is free for x:

$$\mathsf{r1}{\downarrow}\,\frac{\forall x[A \vee B]}{\forall x A \vee B} \quad \mathsf{r2}{\downarrow}\,\frac{\forall x(A \wedge B)}{\forall x A \wedge B} \quad \mathsf{r3}{\downarrow}\,\frac{\exists x[A \vee B]}{\exists x A \vee B} \quad \mathsf{r4}{\downarrow}\,\frac{\exists x(A \wedge B)}{\exists x A \wedge B}$$

We consider commutative variants of these rules valid instances, e.g. $\mathsf{r1}{\downarrow}\,\dfrac{\forall x[B \vee A]}{B \vee \forall x A}$.

3. For term assignment, we have the rule:

$$\mathsf{n}{\downarrow}\,\frac{A\{x \Leftarrow t\}}{\exists x A}$$

Definition 8. *We define a proof system for FOL,* $\mathsf{KSh1}$*:*

$$\mathsf{KSh1} = \mathsf{KS} + \boxed{\begin{array}{lll} \mathsf{r1}\!\downarrow \dfrac{\forall x[A \vee B]}{[\forall x A \vee B]} & \mathsf{r2}\!\downarrow \dfrac{\forall x(A \wedge B)}{(\forall x A \wedge B)} & \mathsf{n}\!\downarrow \dfrac{A\{x \Leftarrow t\}}{\exists x A} \\[2ex] \mathsf{r3}\!\downarrow \dfrac{\exists x[A \vee B]}{[\exists x A \vee B]} & \mathsf{r4}\!\downarrow \dfrac{\exists x(A \wedge B)}{(\exists x A \wedge B)} & \mathsf{qc}\!\downarrow \dfrac{\exists x A \vee \exists x A}{\exists x A} \end{array}}$$

$$+$$

$$\boxed{\begin{array}{rclrcl} \forall x A & = & \forall z A\{x \Leftarrow z\} & \exists z A & = & \exists z A\{x \Leftarrow z\} \\ \forall x \forall y A & = & \forall y \forall x A & \exists x \exists y A & = & \exists y \exists x A \\ \forall x \mathsf{t} & = & \mathsf{t} = \exists x \mathsf{t} & \forall x \mathsf{f} & = & \mathsf{f} = \exists x \mathsf{f} \end{array}}$$

Where z does not occur in A for the top two equalities.

For an example of a $\mathsf{KSh1}$ proof, see (Fig. 1).

$$\mathsf{qc}\!\downarrow \dfrac{\mathsf{r1}\!\downarrow \dfrac{\mathsf{r1}\!\downarrow \dfrac{= \dfrac{\mathsf{t}}{\forall y_1 \forall y_2 \left[= \dfrac{\mathsf{ai}\!\downarrow \dfrac{\mathsf{t}}{Py_1 \vee \bar{P}y_1} \vee \left[\mathsf{aw}\!\downarrow \dfrac{\mathsf{f}}{\bar{P}c} \vee \mathsf{aw}\!\downarrow \dfrac{\mathsf{f}}{Py_2} \right]}{\left[\bar{P}c \vee Py_1\right] \vee \left[\bar{P}y_1 \vee Py_2\right]} \right]}}{\forall y_1 \left[\left[\bar{P}c \vee Py_1\right] \vee \mathsf{n}\!\downarrow \dfrac{\forall y_2 \left[\bar{P}y_1 \vee Py_2\right]}{\exists x_2 \forall y_2 \left[\bar{P}x_2 \vee Py_2\right]} \right]}}{\mathsf{n}\!\downarrow \dfrac{\forall y_1 \left[\bar{P}c \vee Py_1\right]}{\exists x_1 \forall y_1 \left[\bar{P}x_1 \vee Py_1\right]} \vee \exists x_2 \forall y_2 \left[\bar{P}x_2 \vee Py_2\right]}}{\exists x \forall y \left[\bar{P}x \vee Py\right]}$$

Fig. 1. A $\mathsf{KSh1}$ proof of a variant of the "drinking principle', $\exists x \forall y[\bar{P}x \vee Py]$, popularised by Smullyan: "There is someone in the pub such that, if he is drinking, then everyone in the pub is drinking."

Proposition 2. $\mathsf{KSh1}$ *is sound and complete.*

Proof. Soundness is trivial. Completeness follows from the proof of Herbrand's theorem in [7] (as stated in the introduction) and the observation that with the four rules $\{\mathsf{r1}\!\downarrow, \mathsf{r2}\!\downarrow, \mathsf{r3}\!\downarrow, \mathsf{r4}\!\downarrow\}$ we can simulate the general retract rule:

$$\mathsf{gr}\!\downarrow \frac{Q\{P\{A\}\}}{P\{Q\{A\}\}},$$

where $Q\{\ \}$ is a sequence of quantifiers and $P\{\ \}$ is a propositional context with no variables bound by any quantifier in $Q\{\ \}$.

Following [7], we define a Herbrand proof in the context of KSh1 in the following way.

Definition 9. *A closed* KSh1 *proof is a* Herbrand proof *if it is in the following form:*

$$\begin{array}{c} \overline{\Vert}\, \mathsf{KS} \\ \forall \boldsymbol{x} B\{\boldsymbol{y} \Leftarrow \boldsymbol{t}\} \\ \Vert\, \{\mathsf{n}\downarrow\} \\ Q\{B\} \\ \Vert\, \{\mathsf{r1}\downarrow,\mathsf{r2}\downarrow,\mathsf{r3}\downarrow,\mathsf{r4}\downarrow\} \\ A' \\ \Vert\, \{\mathsf{qc}\downarrow\} \\ A \end{array}$$

where $Q\{\ \}$ is a context consisting only of quantifiers and B is quantifier-free.

For an example of a Herbrand proof, see (Fig. 2).

Proposition 3. *Every proof in* KSh1 *can be converted to a Herbrand Proof.*

$$\mathsf{qc}\downarrow\dfrac{\mathsf{r3}\downarrow\dfrac{\mathsf{n}\downarrow\dfrac{=\dfrac{\mathsf{t}}{\forall y_1 \forall y_2\left[=\dfrac{\mathsf{ai}\downarrow\dfrac{\mathsf{t}}{Py_1 \vee \bar{P}y_1} \vee \left[\mathsf{aw}\downarrow\dfrac{\mathsf{f}}{\bar{P}c} \vee \mathsf{aw}\downarrow\dfrac{\mathsf{f}}{Py_2}\right]}{[\bar{P}c \vee Py_1] \vee [\bar{P}y_1 \vee Py_2]}\right]}}{\exists x_1\left[\mathsf{r1}\downarrow\dfrac{\forall y_1\left[\mathsf{r3}\downarrow\dfrac{\mathsf{n}\downarrow\dfrac{[\bar{P}x_1 \vee Py_1] \vee [\bar{P}y_1 \vee Py_2]}{\exists x_2\left[\mathsf{r1}\downarrow\dfrac{\forall y_2[[\bar{P}x_1 \vee Py_1] \vee [\bar{P}x_2 \vee Py_2]]}{[\bar{P}x_1 \vee Py_1] \vee \forall y_2[\bar{P}x_2 \vee Py_2]}\right]}}{[\bar{P}x_1 \vee Py_1] \vee \exists x_2 \forall y_2[\bar{P}x_2 \vee Py_2]}\right]}{\forall y_1[\bar{P}x_1 \vee Py_1] \vee \exists x_2 \forall y_2[\bar{P}x_2 \vee Py_2]}\right]}}{\exists x_1 \forall y_1[\bar{P}x_1 \vee Py_1] \vee \exists x_2 \forall y_2[\bar{P}x_2 \vee Py_2]}}{\exists x \forall y[\bar{P}x \vee Py]}$$

Fig. 2. A Herbrand proof of the drinking principle

Proof. [7, Theorem 4.2]

3.4 KSh2 and Herbrand Normal Form

To aid the translation between open deduction proofs and expansion proofs, we introduce a slightly different proof system to KSh1. It involves two new rules.

Definition 10. *We define the rule* $\mathsf{h}\!\downarrow$*, which we call a* Herbrand expander *and the rule* $\exists\mathsf{w}\!\downarrow$*, which we call* existential weakening*:*

$$\mathsf{h}\!\downarrow \frac{\exists x A \vee A\{x \Leftarrow t\}}{\exists x A} \qquad \exists\mathsf{w}\!\downarrow \frac{\mathsf{f}}{\exists x A}$$

For technical reasons again, we insist that $A\{x \Leftarrow t\}$ is in fact $A'\{x \Leftarrow t\}$, where A' is an α-equivalent formula to A with fresh variables for all quantifiers, but for simplicity we will usually denote it A.

Definition 11

$$\mathsf{KSh2} = \mathsf{KS} + \begin{array}{c}\boxed{\begin{array}{cc} \mathsf{r1}\!\downarrow \dfrac{\forall x[A \vee B]}{[\forall x A \vee B]} & \mathsf{h}\!\downarrow \dfrac{\exists x A \vee A\{x \Leftarrow t\}}{\exists x A} \\ \mathsf{r2}\!\downarrow \dfrac{\forall x(A \wedge B)}{(\forall x A \wedge B)} & \exists\mathsf{w}\!\downarrow \dfrac{\mathsf{f}}{\exists x A} \end{array}} \\ + \\ \boxed{\begin{array}{rclrcl} \forall x A & = & \forall z A\{x \Leftarrow z\} & \exists z A & = & \exists z A\{x \Leftarrow z\} \\ \forall x \forall y A & = & \forall y \forall x A & \exists x \exists y A & = & \exists y \exists x A \\ \forall x \mathsf{t} & = & \mathsf{t} = \exists x \mathsf{t} & \forall x \mathsf{f} & = & \mathsf{f} = \exists x \mathsf{f} \end{array}} \end{array}$$

Where z does not occur in A for the top two equalities.

Remark 4. The $\exists\mathsf{w}\!\downarrow$ rule is derivable for $\mathsf{KSh2} \backslash \{\exists\mathsf{w}\!\downarrow\}$, but we explicitly include it so that we can restrict weakening instances in certain parts of proofs.

Definition 12. *We say that a proof in* $\mathsf{KSh2}$ *is* regular *if there are no α-substitutions in the proof, and no variable is used in two different quantifiers.*

Definition 13. *If ϕ is a closed* $\mathsf{KSh2}$ *proof in the following form, where $\forall \boldsymbol{x}$ is a list of universal quantifiers with distinct variables, and $Lo(\phi)$ is regular and in sequential form, we say ϕ is in* Herbrand Normal Form *(HNF):*

$$\begin{array}{c} {\scriptstyle Up(\phi)}\prod {\scriptstyle \mathsf{KS}} \\ \forall \boldsymbol{x} H_\phi(A) \\ \| {\scriptstyle \{\exists\mathsf{w}\downarrow\}} \\ \forall \boldsymbol{x} H_\phi^+(A) \\ {\scriptstyle Lo(\phi)} \| {\scriptstyle \{\mathsf{r1}\downarrow, \mathsf{r2}\downarrow, \mathsf{h}\downarrow\}} \\ A \end{array}$$

$H_\phi(A)$, the Herbrand disjunction of A according to ϕ*, or just the* Herbrand disjunction of A*, contains no quantifiers, whereas $H_\phi^+(A)$, the* expansive Herbrand disjunction of A according to ϕ*, may contain quantifiers. $Up(\phi)$ is called the* upper part *of ϕ, and $Lo(\phi)$ the* lower part *of ϕ.*

For an example of a proof in HNF, see (Fig. 3).

$$\mathsf{r1}{\downarrow}\dfrac{=\dfrac{=\dfrac{\mathsf{t}}{\forall y_1 \forall y_2 \left[\mathsf{ai}{\downarrow}\dfrac{\mathsf{t}}{Py_1 \vee \bar{P}y_1} \vee \left[\mathsf{aw}{\downarrow}\dfrac{\mathsf{f}}{\bar{P}c} \vee \mathsf{aw}{\downarrow}\dfrac{\mathsf{f}}{Py_2} \right] \right]}}{\forall y_1 \left[\mathsf{r1}{\downarrow}\dfrac{\forall y_2 \left[\left[\left[\exists\mathsf{w}{\downarrow}\dfrac{\mathsf{f}}{\exists x \forall y [\bar{P}x \vee Py]} \vee [\bar{P}y_1 \vee Py_2] \right] \vee [\bar{P}c \vee Py_1] \right] \right]}{\mathsf{h}{\downarrow}\dfrac{\mathsf{r1}{\downarrow}\dfrac{\forall y_2 [\exists x \forall y [\bar{P}x \vee Py] \vee [\bar{P}y_1 \vee Py_2]]}{\exists x \forall y [\bar{P}x \vee Py] \vee \forall y_2 [\bar{P}y_1 \vee Py_2]}}{\exists x \forall y [\bar{P}x \vee Py]} \vee [\bar{P}c \vee Py_1]} \right]}}{\mathsf{h}{\downarrow}\dfrac{\exists x \forall y [\bar{P}x \vee Py] \vee \forall y_1 [\bar{P}c \vee Py_1]}{\exists x \forall y [\bar{P}x \vee Py]}}$$

Fig. 3. A proof of the drinking principle in HNF

Proposition 4. *A formula A has a proof in HNF iff. It has a Herbrand proof.*

Proof. Let ϕ be a proof of A in HNF. As $H_\phi(A)$ is the Herbrand expansion of A, it is straightforward to construct a Herbrand proof for A: one can infer the necessary $\mathsf{n}{\downarrow}$ and $\mathsf{qc}{\downarrow}$ rules by comparing $H_\phi(A)$ and A. Now let ϕ be a Herbrand Proof. The order of the quantifiers in $Q\{\ \}$ (as in Definition 9) is used to build the HNF proof. Thus, we proceed by induction on the number of quantifiers in $Q\{\ \}$. If there are none, it is obviously trivial. We split the inductive step into two cases.

First, consider ϕ_1 of the form shown, where P is a quantifier-free context and $Q\{\ \} = \forall z Q'\{\ \}$. Clearly ϕ_2 is also a Herbrand proof, so by the IH the proof ϕ_3 in HNF is constructible, from which we can construct ϕ_4.

$$\begin{array}{c} \Vert\, \mathsf{KS} \\ \forall z \forall \boldsymbol{x} B\{\boldsymbol{y} \Leftarrow \boldsymbol{t}\} \\ \Vert\, \{\mathsf{n}{\downarrow}\} \\ \forall z Q'\{B\} \\ \Vert\, \{\mathsf{r1}{\downarrow},\mathsf{r2}{\downarrow},\mathsf{r3}{\downarrow},\mathsf{r4}{\downarrow}\} \\ P\{\forall z C'\} \\ \Vert\, \{\mathsf{qc}{\downarrow}\} \\ P\{\forall z C\} \\ \\ \phi_1 \end{array} \qquad \begin{array}{c} \Vert\, \mathsf{KS} \\ \forall \boldsymbol{x} B\{\boldsymbol{y} \Leftarrow \boldsymbol{t}\} \\ \Vert\, \{\mathsf{n}{\downarrow}\} \\ Q'\{B\} \\ \Vert\, \{\mathsf{r1}{\downarrow},\mathsf{r2}{\downarrow},\mathsf{r3}{\downarrow},\mathsf{r4}{\downarrow}\} \\ P\{C'\} \\ \Vert\, \{\mathsf{qc}{\downarrow}\} \\ P\{C\} \\ \\ \phi_2 \end{array} \qquad \begin{array}{c} Up\phi_3 \Vert\, \mathsf{KS} \\ \forall \boldsymbol{x} H_{\phi_3} P\{C\} \\ \Vert\, \{\exists\mathsf{w}{\downarrow}\} \\ \forall \boldsymbol{x} H^+_{\phi_3} P\{C\} \\ Lo(\phi_3) \Vert\, \{\mathsf{r1}{\downarrow},\mathsf{r2}{\downarrow},\mathsf{h}{\downarrow}\} \\ P\{C\} \\ \\ \phi_3 \end{array} \qquad \begin{array}{c} \forall z\, Up\phi_3 \Vert\, \mathsf{KS} \\ \forall z \forall \boldsymbol{x} H_{\phi_3} P\{C\} \\ \Vert\, \{\exists\mathsf{w}{\downarrow}\} \\ \forall z \forall \boldsymbol{x} H^+_{\phi_3} P\{C\} \\ \forall z Lo\phi_3 \Vert\, \{\mathsf{r1}{\downarrow},\mathsf{r2}{\downarrow},\mathsf{h}{\downarrow}\} \\ \forall z P\{C\} \\ \Vert\, \{\mathsf{r1}{\downarrow},\mathsf{r2}{\downarrow}\} \\ P\{\forall z C\} \\ \\ \phi_4 \end{array}$$

In the same way, we consider the case where $Q\{\ \} = \exists z Q'\{\ \}$. Below we only show the case where there is no contraction acting on $\exists z C$, but the case with such a contraction is similar.

$$
\begin{array}{cccc}
\begin{array}{c}
\Vert\,\mathsf{KS} \\
\forall \boldsymbol{x} B\{\boldsymbol{y} \Leftarrow \boldsymbol{t}\}\{z \Leftarrow t\} \\
\Vert\,\{\mathsf{n}\!\downarrow\} \\
\exists z Q'\{B\} \\
\Vert\,\{\mathsf{r1}\!\downarrow,\mathsf{r2}\!\downarrow,\mathsf{r3}\!\downarrow,\mathsf{r4}\!\downarrow\} \\
P\{\exists z C'\} \\
\Vert\,\{\mathsf{qc}\!\downarrow\} \\
P\{\exists z C\} \\
\\
\phi_1
\end{array}
&
\begin{array}{c}
\Vert\,\mathsf{KS} \\
\forall \boldsymbol{x} B\{\boldsymbol{y} \Leftarrow \boldsymbol{t}\}\{z \Leftarrow t\} \\
\Vert\,\{\mathsf{n}\!\downarrow\} \\
Q'\{B\}\{z \Leftarrow t\} \\
\Vert\,\{\mathsf{r1}\!\downarrow,\mathsf{r2}\!\downarrow,\mathsf{r3}\!\downarrow,\mathsf{r4}\!\downarrow\} \\
P\{C'\{z \Leftarrow t\}\} \\
\Vert\,\{\mathsf{qc}\!\downarrow\} \\
P\{C\{z \Leftarrow t\}\} \\
\\
\phi_2
\end{array}
&
\begin{array}{c}
Up(\phi_3)\,\Vert\,\mathsf{KS} \\
\forall \boldsymbol{x} P\{D\{z \Leftarrow t\}\} \\
\Vert\,\{\exists\mathsf{w}\!\downarrow\} \\
\forall \boldsymbol{x} P\{D^+\{z \Leftarrow t\}\} \\
Lo(\phi_3)\,\Vert\,\{\mathsf{r1}\!\downarrow,\mathsf{r2}\!\downarrow,\mathsf{h}\!\downarrow\} \\
P\{C\{z \Leftarrow t\}\} \\
\\
\phi_3
\end{array}
&
\begin{array}{c}
Up(\phi_3)\,\Vert\,\mathsf{KS} \\
\forall \boldsymbol{x} P\{D\{z \Leftarrow t\}\} \\
\Vert\,\{\exists\mathsf{w}\!\downarrow\} \\
\forall \boldsymbol{x} P\{\exists z C \vee D^+\{z \Leftarrow t\}\} \\
Lo(\phi_3)\,\Vert\,\{\mathsf{r1}\!\downarrow,\mathsf{r2}\!\downarrow,\mathsf{h}\!\downarrow\} \\
P\left\{\mathsf{h}\!\downarrow\dfrac{\exists z C \vee C\{z \Leftarrow t\}}{\exists z C}\right\} \\
\\
\phi_4
\end{array}
\end{array}
$$

where
$P\{D\{z \Leftarrow t\}\} = H_{\phi_3}(P\{C\{z \Leftarrow t\}\})$ and $P\{D^+\{z \Leftarrow t\}\} = H^+_{\phi_3}(P\{C\{z \Leftarrow t\}\})$.

4 Translations Between KSh2 and Expansion Proofs

Above, we gave translations between Herbrand proofs in KSh1 and KSh2 proofs in HNF. We will now give a translations between KSh2 proofs in HNF and expansion proofs, thus giving us a link between deep inference Herbrand proofs and expansion proofs.

Remark 5. We extend the notion and syntax of contexts from derivations to expansion trees. For the notion to make sense, a context can only take expansion trees with the same shallow formula.

4.1 KSh2 to Expansion Proofs

Before stating and proving the main theorem, we will define the map π_1 from KS proofs to expansion proofs, and then prove some lemmas to help prove that the dependency relation in all expansion proofs in the range of π_1 is acyclic.

Definition 14. *We define a map π_1' from the lower part of* KSh2 *proofs in HNF to expansion trees in the following way, working from the bottom*

On the conclusion of ϕ, we define π_1' as follows:

- $\pi_1'(B \star C) = \pi_1'(B) \star \pi_1'(C)$
- $\pi_1'(\forall x B) = \forall x B +^x \pi_1'(B)$
- $\pi_1'(\exists x B) = \exists x B$

The r1↓ *and* r2↓ *rules are ignored by expansion trees and each* h↓ *rule adds a branch to a* $\exists$*-node:*

- *If* $\phi = \begin{array}{c} K\left\{\mathsf{r1}\!\downarrow\dfrac{\forall x[B \vee C]}{\forall x B \vee C}\right\} \\ \phi' \,\Vert \\ A \end{array}$ *then* $\pi_1'(\phi) = \pi_1'\left(\begin{array}{c} K\{\forall x B \vee C\} \\ \phi' \,\Vert \\ A \end{array}\right)$.
- *If* $\phi = \begin{array}{c} K\left\{\mathsf{r2}\!\downarrow\dfrac{\forall x(B \wedge C)}{(\forall x B \wedge C)}\right\} \\ \phi' \,\Vert \\ A \end{array}$ *then* $\pi_1'(\phi) = \pi_1'\left(\begin{array}{c} K\{\forall x B \wedge C\} \\ \phi' \,\Vert \\ A \end{array}\right)$.

- *If* $\pi'_1\left(\begin{array}{c} K\{\exists xB\} \\ \phi\,\| \\ A \end{array}\right) = K_{\pi_1}(\exists xB +^{\tau_1} E_1 + \cdots +^{\tau_n} E_n)$*, then:*

$$\pi'_1\left(\begin{array}{c} K\left\{\mathsf{h}\!\downarrow\dfrac{\exists xB \vee B\{x \Leftarrow \tau_{n+1}\}}{\exists xB}\right\} \\ \phi\,\| \\ A \end{array}\right) = K_{\pi_1}(\exists xB +^{\tau_1} E_1 + \cdots +^{\tau_{n+1}} E_{n+1})$$

where $E_{n+1} = \pi'_1(B\{x \Leftarrow \tau_{n+1}\})$*.*

We then define the map π_1 *from* KSh2 *proofs in HNF to expansion trees as* $\pi_1(\phi) = \pi'_1(Lo(\phi))$*.*

To show that $\pi_1(\phi)$ is an expansion proof, we need to prove that $\forall \boldsymbol{x} H_\phi(A)$ is a tautology and $<_E$ is acyclic. As $\forall \boldsymbol{x} H_\phi(A)$ has a proof in KS it is a tautology. Thus all that is needed is the acyclicity of $<_E$. To do so, we define the following partial order on variables in the lower part of KSh2 proofs in HNF.

Definition 15. *Let* ϕ *be a proof in HNF. Define the partial order* $<_\phi$ *on the variables of occurring in* $Lo(\phi)$ *to be the minimal partial order such that* $y <_\phi x$ *if* $K_1\{Q_1 x K_2\{Q_2 y B\}\}$ *is a section of* $Lo(\phi)$*.*

Proposition 5. $<_\phi$ *is well-defined for all* KSh2 *proofs in HNF.*

Proof. Let ϕ be a proof of A in HNF, as in Definition 13. As $Lo(\phi)$ only contains $\mathsf{h}\!\downarrow, \mathsf{r1}\!\downarrow$ and $\mathsf{r2}\!\downarrow$ rules and no α-substitution, if a variable v occurs in $Lo(\phi)$ then v occurs in $\forall x H^+_\phi(A)$. Notice also that none of $\mathsf{h}\!\downarrow, \mathsf{r1}\!\downarrow$ and $\mathsf{r2}\!\downarrow$ can play the role of ρ in the following scheme:

$$\rho\,\frac{K\{Q_1 v_1 A_1\}\{Q_2 v_2 A_2\}}{K'\{Q_1 v_1\{K'' Q_2 v_2 B\}\}}\,.$$

Therefore, we observe that if $K_1\{Q_1 x K_2\{Q_2 y B\}\}$ is a section of $Lo(\phi)$, then $\forall x H^+_\phi(A)$ is of the form $L_1\{Q_1 x L_2\{Q_2 y C\}\}$, i.e. no dependencies can be introduced below $\forall x H^+_\phi(A)$. Thus $x <_\phi y$ iff. $\forall x H^+_\phi(A)$ can be written $L_1\{Q_1 x L_2\{Q_2 y C\}\}$ for some $L_1\{\ \}, L_2\{\ \}$ and C and is therefore a well-defined partial order.

Lemma 1. *Let* ϕ *be an* KSh2 *proof in HNF and* e' *an* $\forall$*-edge in* $\pi_1(\phi)$ *that points to the* $\exists$*-edge* e*. If* $Lab(e') = y$ *and the* $\exists$*-node below* e *is* $\exists xA$*, then* $x <_\phi y$*.*

Proof. Since we have an $\exists$-node $\exists xA$ in $\pi_1(\phi)$ with an edge labelled t below it, there must be the following $\mathsf{h}\!\downarrow$ rule in ϕ:

$$K\left\{\mathsf{h}\!\downarrow\frac{\exists xA \vee A\{x \Leftarrow t\}}{\exists xA}\right\}$$

Since e points to e', y must occur freely in t. As ϕ is closed, y cannot be a free variable in $K\{\exists xA \vee A\{x \Leftarrow t\}\}$. Thus $K\{\ \}$ must be of the form $K_1\{\forall y K_2\{\ \}\}$. Therefore $x <_\phi y$.

Lemma 2. *Let ϕ be an* $\mathsf{KSh2}$ *proof in HNF, e a $\forall$-edge of $\pi_1(\phi)$ labelled by x and e' an $\exists$-edge above an $\exists$-node $\exists yA$. If e is a descendant of e' then $x <_\phi y$.*

Proof. $Sh(\pi_1(\phi)) = K_1\{\exists y K_2 \forall x\{B\}\}$ (for some $K_1\{\ \}, K_2\{\ \}$, and B) is the conclusion of ϕ, so $x <_\phi y$.

Lemma 3. *Let ϕ be an* $\mathsf{KSh2}$ *proof in HNF, $E_\phi = \pi_1(\phi)$ and e and e' be edges in E_ϕ s.t. $e <_{E_\phi} e'$, $Lab(e) = x$ and $Lab(e') = x'$. Then $x <_\phi x'$.*

Proof. As $e <_{E_\phi} e'$, there must be a chain

$$e_{q_0} <^{-}_{E_\phi} \cdots <^{-}_{E_\phi} e_{p_1} <^{-}_{E_\phi} e_{q_1} <^{-}_{E_\phi} \cdots <^{-}_{E_\phi} e_{p_m} <^{-}_{E_\phi} e_{q_m} <^{-}_{E_\phi} \cdots <^{-}_{E_\phi} e_{p_n}$$

where $e_{q_0} = e$ and $e_{p_n} = e'$, e_{q_i} points to e_{p_i}, and e_{q_i} is a descendant of $e_{p_{i+1}}$ in the expansion tree. By Lemma 1, we know that if $\exists x_{p_i}$ is the node above p_i and $Lab(e_{q_i}) = x_{q_i}$, then $x_{p_i} <_\phi x_{q_i}$. By Lemma 2, since e_{q_i} is a descendant of $e_{p_{i+1}}$ in the expansion tree, $x_{q_i} <_\phi x_{p_{i+1}}$. Therefore $x <_\phi x'$.

Theorem 3. *If ϕ is an* $\mathsf{KSh2}$ *proof of A in HNF, then we can construct an expansion proof $E_\phi = \pi_1(\phi)$, with $Sh(E_\phi) = A$, and $Dp(E_\phi) = H_\phi(A)$.*

Proof. As described above, we only need to show that the dependency relation of E_ϕ is acyclic. Assume there were a cycle in $<_{E_\phi}$. Clearly, it could not be generated by just by travelling up the expansion tree. Thus, there is some e and e' such that e points to e' and $e <_{E_\phi} e' <_{E_\phi} e$. But then, if $Lab(e) = x$, by Lemma 3, $x <_\phi x$. But this contradicts Proposition 5. Therefore $<_{E_\phi}$ is acyclic.

Expansion Proofs to $\mathsf{KSh2}$: For the translation from expansion proofs to $\mathsf{KSh2}$ proofs in HNF, we show that we can always construct a total order on the edges in an expansion proof that guides the construction of the lower part of a proof in HNF. Unlike the previous translation, there is not necessarily a unique proof corresponding to each expansion proof, but the choice of a total order determines the proof that will be created.

Definition 16. *A* weak *expansion tree is defined in the same way as in Definition 1 except that the first condition is weakened to allow any formula to be a leaf of the tree. A weak expansion tree with an acyclic dependency relation is correct regardless of whether its deep formula is a tautology.*

Definition 17. *We define the* expansive deep formula *$Dp^+(E)$ for (weak) expansion trees, which is defined in the same way as the usual deep formula except that:*

$$Dp^+(\exists xA +^{t_1} E_1 +^{t_2} \cdots +^{t_n} E_n) := \exists xA \vee Dp^+(E_1) \vee \ldots \vee Dp^+(E_n)$$

Definition 18. *A* minimal edge *of a (weak) expansion tree E is an edge that is minimal w.r.t. to $<_E$.*

Definition 19. *Let E be a (weak) expansion proof. Let $<_E^+$ be a total order extending $<_E$ such that the following condition holds: if $\star$ is a node with edges e and e' below it, then e and e' are consecutive elements in the total order. We say $<_E^+$ is a* sequentialisation *of E.*

Lemma 4. *Every (weak) expansion proof has a sequentialisation, often many.*

Proof. We proceed by induction on the number of nodes in a weak expansion proof. The base case is trivial. For the inductive step, we will show that every weak expansion tree has either a minimal edge e below an existential or universal node, or that there are two minimal edges e_1 and e_2 below a $\star$-node. As the rest of the weak expansion proof has a sequentialisation by the inductive hypothesis, we can extend it with the minimal element e or the two minimal elements e_1 and e_2 for a sequentialisation for the full weak expansion proof.

Assume E is a weak expansion proof with no minimal edges below existential or universal nodes. As $<_E$ is a partial order, there must be at least one minimal edge e_0, and by the assumption it must be below a node $\star_0$. Let e_0' be the other edge below $\star_0$. If e_0' is minimal, we are done. If not, pick some minimal edge $e_1 < e_0'$, which again, with $e_1' < e_0'$, must be below some $\star_1$. For each e_i' that is not minimal, we can find $e_{i+1}' < e_i'$. As E is finite, this sequence cannot continue indefinitely, so eventually we will find two minimal edges e_n and e_n' below $\star_n$. Note that e_n and e_n' need not be unique and thus the sequentialisation is not unique.

Proposition 6. *Let $E = K_E\{\forall xA +^x A\}$, with $Dp^+(E) = K\{A\}$, be a correct weak expansion tree with the $\forall$-edge labelled by x (which we will call e) minimal w.r.t. $<_E$. Then there is a derivation* $\begin{array}{c}\forall x K\{A\}\\ \| \{\mathsf{r1}\downarrow,\mathsf{r2}\downarrow\}\\ K\{\forall x A\}\end{array}$.

Proof. We proceed by induction on the height of the node $\forall xA$ in E. If $\forall xA$ is the bottom node, then $K\{A\} = A$ and we are done. Let E be an expansion tree where $\forall x$ is not the bottom node. There are three possible cases to consider. In each case, $E_1 = K_{E_1}\{\forall xA +^x A\}$ is an expansion tree with $Dp^+(E_1) = K_1\{A\}$ and, by the inductive hypothesis, we have a derivation $\begin{array}{c}\forall x K_1\{A\}\\ \| \{\mathsf{r1}\downarrow,\mathsf{r2}\downarrow\}\\ K_1\{\forall x A\}\end{array}$.

1. $E = (E_1 \star E_2)$, with $Dp^+(E) = [K_1\{A\} \star Dp^+(E_2)]$. As e is minimal, it cannot point to any edge in E_2. Therefore $B := Dp^+(E_2)$ is free for x. Therefore we can construct the derivations:

$$\begin{array}{c}\mathsf{r1}\downarrow\dfrac{\forall x[K_1\{A\} \vee B]}{\forall x K_1\{A\}}\\ \| \{\mathsf{r1}\downarrow,\mathsf{r2}\downarrow\} \vee B\\ K_1\{\forall x A\}\end{array} \quad \textit{and} \quad \begin{array}{c}\mathsf{r2}\downarrow\dfrac{\forall x(K_1\{A\} \wedge B)}{\forall x K_1\{A\}}\\ \| \{\mathsf{r1}\downarrow,\mathsf{r2}\downarrow\} \wedge B\\ K_1\{\forall x A\}\end{array}$$

2. $E = \forall y(Sh(E_1)) +^y E_1$. As $Dp^+(E) = Dp^+(E_1)$, we are already done.

3. $E = \exists y K_0\{A_0\} +^{t_1} E_1 \cdots +^{t_n} E_n$, with $Dp^+(E_i) = B_i := [K_0\{A_0\}]\{y \Leftarrow t_i\}$ and in particular $B_1 = K_1\{A\}$. Thus $Dp^+(E) = \exists y B_0 \vee K_1\{A\} \vee B_2 \vee \ldots \vee B_n$. Again, e cannot point to any edge in any of the E'_i, so we can construct:

$$\mathsf{r1}{\downarrow}\,\dfrac{\forall x[\exists yB_0 \vee K_1\{A\} \vee B_2 \vee \ldots \vee B_n]}{\mathsf{r1}{\downarrow}\,\dfrac{\forall x[\exists yB_0 \vee K_1\{A\}]}{\left[\exists yB_0 \vee \begin{array}{c}\forall xK_1\{A\}\\ \Vert\,\{\mathsf{r1}{\downarrow},\mathsf{r2}{\downarrow}\}\\ K_1\{\forall xA\}\end{array}\right]} \vee [B_2 \vee \ldots \vee B_n]}$$

Definition 20. *We define the map* π_2^{Lo} *that takes an expansion tree* E *and a sequentialisation* $<_E^+$ *to a derivation:*

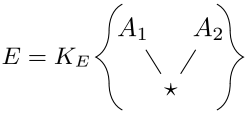
$$\pi_2^{Lo}(E, <_E^+) = \begin{array}{c}\forall \boldsymbol{x} Dp^+(E)\\ \Vert\,\{\mathsf{h}{\downarrow},\mathsf{r1}{\downarrow},\mathsf{r2}{\downarrow}\}\\ Sh(E)\end{array}$$

In each case $<_{E'}^+$ *is* $<_E^+$ *restricted to* E'*.*

- *If* E *is just a leaf* A, $\pi_2^{Lo}(E, <_E^+) = A$.
- *If* $E = K_E\{A_1 \star_E A_2\}$ *is* e_1*, and the minimal edge w.r.t* $<_E^+$ *is between* $\star_E$ *and* A_1*, then by Definition 19 the next-but-minimal edge is between* $\star_E$ *and* A_2*. Then,* $E' = K_E\{A_1 \star_F A_2\}$ *is a correct weak expansion tree and we can define:*

$$\pi_2^{Lo}(E, <_E^+) = \pi_2^{Lo}(E', <_{E'}^+)$$

Pictorially:

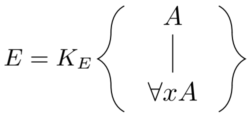


$$E' = K_E\{A_1 \star A_2\}$$

- *If* $E = K_E\{\forall xA +^x A\}$ *and the minimal edge w.r.t.* $<_E^+$ *is between* $\forall xA$ *and* A*, then, by Proposition 6,* $E' = K_E\{\forall xA\}$ *is a correct weak expansion tree and we can define:*

$$\pi_2^{Lo}(E, <_E^+) = \begin{array}{c}\forall x Dp^+(E)\\ \Vert\,\{\mathsf{r1}{\downarrow},\mathsf{r2}{\downarrow}\}\\ = \dfrac{Dp^+(E')}{\pi_2^{Lo}(E', <_{E'}^+)}\end{array}$$

Pictorially:

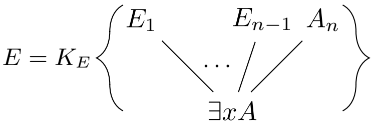


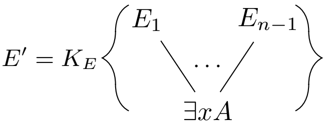
$$E' = K_E\{\forall x A\}$$

- *If the minimal edge of* $E = K_E\{\exists xA +^{t_1} E_1 \cdots +^{t_n} A_n\}$, *with* $Dp^+(E) = K\{\exists xA \vee A_1 \vee \ldots \vee A_n\}$, *is between* $\exists xA$ *and* A_n, *then* $E' = K_E\{\exists xA +^{t_1} E_1 \cdots +^{t_{n-1}} E_{n-1}\}$ *is a correct weak expansion tree with* $Dp^+(E') = K\{A_1 \vee \ldots \vee A_{n-1}\}$ *and we can define:*

$$\pi_2^{Lo}(E, <_E^+) = \quad {=}\,\dfrac{K\left\{ {=}\,\dfrac{\exists xA \vee A_1 \vee \ldots \vee A_n}{\mathsf{h}\!\downarrow \dfrac{\exists xA \vee A_n}{\exists xA} \vee A_1 \vee \ldots \vee A_{n-1}} \right\}}{\pi_2^{Lo}(E', <_{E'}^+)}$$

Pictorially:





Theorem 4. *If* E *is an expansion proof with* $Sh(E) = A$, *then we can construct an* $\mathsf{KSh2}$ *proof* ϕ *of* A *in HNF, where* $H_\phi(A) = Dp(E)$.

Proof. As $Dp(E)$ is a tautology, there is a proof $\begin{array}{c} \pi_2^{Up}(E)\,\Big\Vert\,\mathsf{KS} \\ \forall \boldsymbol{x} Dp(E) \end{array}$ and clearly there is a proof $\begin{array}{c} Dp(E) \\ \Vert\,\{\exists \mathsf{w}\!\downarrow\} \\ Dp^+(E) \end{array}$. Thus, choosing an arbitrary sequentialisation $<_E^+$ of E, we can define π_2 from expansion proofs to $\mathsf{KSh2}$ proofs in HNF as:

$$\pi_2(E) = \quad \begin{array}{c} \pi_2^{Up}(E)\,\Big\Vert\,\mathsf{KS} \\ \forall \boldsymbol{x} Dp(E) \\ \Vert\,\{\exists \mathsf{w}\!\downarrow\} \\ \forall \boldsymbol{x} Dp^+(E) \\ \pi_2^{Lo}(E, <_E^+)\,\Big\Vert\,\{\mathsf{r1}\!\downarrow, \mathsf{r2}\!\downarrow, \mathsf{h}\!\downarrow\} \\ Sh(E) \end{array}$$

Remark 6. For all expansion proofs E we have $\pi_2^{Up}(E) = Up(\pi_2(E))$ and $\pi_2^{Lo}(E) = Lo(\pi_2(E))$.

5 Further Work

The translations between deep inference proofs and expansion proofs should be seen as a springboard for further investigations. One obvious next step is to extend KSh2 with cut, and prove cut elimination, so that completeness does not depend on the translation into KSh1 and Brünnler's result. Having done so, we can then make a proper comparison with the cut elimination procedures for expansion proofs described in [1,12,15]. Additionally, it would be interesting to try and situate this work in the context of recent work by Aler Tubella and Guglielmi [2,3], in which they provide a general theory of normalisation for various different propositional logics. In their terminology, a Herbrand proof is close to the notion of a *decomposed* proof, which has two phases: the first contraction-free and the second consisting only of contractions. Extending the procedure, described in [4], to remove identity-cut cycles from SKS proofs to first-order systems is likely to be an important aspect of this research.

References

1. Alcolei, A., Clairambault, P., Hyland, M., Winskel, G.: The true concurrency of Herbrand's theorem (2017, submitted)
2. Aler Tubella, A.: A study of normalisation through subatomic logic. University of Bath (2016)
3. Aler Tubella, A., Guglielmi, A.: Subatomic proof systems: splittable systems. arXiv preprint arXiv:1703.10258 (2017)
4. Aler Tubella, A., Guglielmi, A., Ralph, B.: Removing cycles from proofs. In: Goranko, V., Dam, M. (eds.) 26th EACSL Annual Conference on Computer Science Logic (CSL 2017), Stockholm, Sweden 2017. Leibniz International Proceedings in Informatics (LIPIcs), pp. 9:1–9:17. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2017)
5. Brünnler, K.: Deep inference and symmetry in classical proofs. Logos Verlag (2003)
6. Brünnler, K.: Two restrictions on contraction. Logic J. IGPL **11**(5), 525–529 (2003)
7. Brünnler, K.: Cut elimination inside a deep inference system for classical predicate logic. Stud. Logica. **82**(1), 51–71 (2006)
8. Buss, S.R.: On Herbrand's theorem. In: Leivant, D. (ed.) LCC 1994. LNCS, vol. 960, pp. 195–209. Springer, Heidelberg (1995). https://doi.org/10.1007/3-540-60178-3_85
9. Chaudhuri, K., Hetzl, S., Miller, D.: A multi-focused proof system isomorphic to expansion proofs. J. Logic Comput. **26**(2), 577–603 (2016). https://doi.org/10.1093/logcom/exu030
10. Guglielmi, A., Gundersen, T., Parigot, M.: A proof calculus which reduces syntactic bureaucracy. In: Proceedings of the 21st International Conference on Rewriting Techniques and Applications (Rta 2010), vol. 6, pp. 135–150 (2010). https://doi.org/10.4230/LIPIcs.RTA.2010.135
11. Gundersen, T.E.: A general view of normalisation through atomic flows. University of Bath (2009)
12. Heijltjes, W.: Classical proof forestry. Ann. Pure Appl. Logic **161**(11), 1346–1366 (2010). https://doi.org/10.1016/j.apal.2010.04.006

13. Herbrand, J., Goldfarb, W.D.: Logical Writings. Springer, Dordrecht (1971). https://doi.org/10.1007/978-94-010-3072-4
14. Hetzl, S., Weller, D.: Expansion trees with cut. arXiv preprint arXiv:1308.0428 (2013)
15. McKinley, R.: A sequent calculus demonstration of Herbrand's theorem. arXiv preprint arXiv:1007.3414 (2010)
16. McKinley, R.: Proof nets for Herbrand's theorem. ACM Trans. Comput. Logic (TOCL) **14**(1), 1–31 (2013). https://doi.org/10.1145/2422085.2422090
17. Miller, D.A.: A compact representation of proofs. Stud. Logica. **46**(4), 347–370 (1987). https://doi.org/10.1007/BF00370646
18. Straßburger, L.: Some observations on the proof theory of second order propositional multiplicative linear logic. In: Curien, P.-L. (ed.) TLCA 2009. LNCS, vol. 5608, pp. 309–324. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02273-9_23
19. Straßburger, L.: Deep inference, expansion trees, and proof graphs for second order propositional multiplicative linear logic, vol. RR-9071, p. 38. Inria Saclay Ile de France (2017)

Metastability and Higher-Order Computability

Sam Sanders(✉)

Center for Advanced Studies and Munich Center for Mathematical Philosophy,
LMU Munich, Munich, Germany
sasander@me.com

Abstract. Tao's notion of *metastability* is classically equivalent to the well-known 'epsilon delta' definition of Cauchy sequence, but the former has much nicer *computational* properties than the latter. In particular, results from the *proof mining* program suggest that metastability gives rise to *highly uniform* and *computable* results under *very general* conditions, in contrast to the non-uniform and/or non-computable rates of convergence emerging from basic convergence theorems involving the definition of Cauchy sequence. As a trade-off for these high levels of uniformity, metastability only provides information about a finite *but arbitrarily large* domain. In this paper, we apply this 'metastability trade-off' to basic theorems of mathematics, i.e. we introduce in the latter finite domains in exchange for high(er) levels of uniformity. In contrast to the aforementioned *effective* results from proof mining, we obtain functionals of *extreme* computational hardness. Thus, our results place a hard limit on the generality of the metastability trade-off and show that metastability does not provide a 'king's road' to computable mathematics. Perhaps surprisingly, we shall make use of *Nonstandard Analysis* (NSA) to establish the aforementioned results (which *do not* involve NSA).

Keywords: Metastability · Higher-order computability
Nonstandard analysis

1 Introduction

Tao's notion[1] of *metastability*, as discussed in e.g. [1, Sect. 1], [12, Sect. 2.29], or [28, Sect. 2.3.1], is a notion of convergence, namely a classically[2] equivalent formulation of the (epsilon-delta) definition of Cauchy sequence. Interestingly, metastability gives rise to quite elegant *computable* (or 'effective') results, in contrast to the definition of Cauchy sequence, as rates of convergence can be non-computable even in the case of basic convergence theorems. In particular, *the trade-off* involved in metastability is that only computable information

S. Sanders—This research was supported by the Alexander von Humboldt Foundation and LMU Munich (via the Excellence Initiative).

[1] Note that the notion of metastability was known in mathematical logic (See [12, Remark 2.29]) under a different name well before Tao discussed this notion in [28].

[2] Unless explicitly stated otherwise, we work in *classical* mathematics.

S. Artemov and A. Nerode (Eds.): LFCS 2018, LNCS 10703, pp. 309–330, 2018.
https://doi.org/10.1007/978-3-319-72056-2_19

about convergence on a finite -but arbitrarily large- domain is obtained, but this information is 'highly uniform' in that it only depends on a small numbers of parameters (relative to the theorem at hand). This trade-off has been observed in *proof mining*:

> Whereas in general noneffective proofs of convergence statements [...] might not provide a (uniform) computable rate of convergence, highly uniform rates of metastability can under very general conditions always be extracted using tools from mathematical logic [...]. ([11, p. 1090])

Following the previous quote, the term 'metastability trade-off' shall refer to the phenomenon that introducing a finite *but arbitrarily large* domain results in highly uniform and computable results. We discuss the *textbook example* of the metastability trade-off in Sect. 2.2 in detail.

In light of the widespread interest in metastability, it is a natural question *how general* the metastability trade-off is? For instance: is this trade-off limited to convergence, or does it apply to other notions? Perhaps the metastability trade-off even constitutes a recipe for 'automatically' obtaining computable mathematics? The hope is that, when faced with non-computable objects, introducing a finite (but arbitrarily large) domain *always* results in highly uniform and computable results. To answer these questions, we shall apply the metastability trade-off to basic mathematical theorems, i.e. we introduce in the latter finite domains in exchange for high(er) levels of uniformity. However, in doing so, we obtain functionals of *extreme* computational hardness, in particular the *special fan functional* Θ from [21]. Intuitively speaking, Θ computes the finite sub-cover stated to exist by the Heine-Borel theorem (for uncountable covers), but Θ *cannot* be computed *by any type two functional* (See Sect. 2.3).

As to the structure of this paper, metastability is discussed in Sect. 2.2, while we introduce the functional Θ in Sect. 2.3 and list its basic properties. We obtain our main results in Sect. 3. In particular, we apply the metastability trade-off to the following theorems: the intermediate and extreme value theorems, $\mathsf{BD\text{-}N}$, and the existence of the Riemann integral. We shall make use of Nonstandard Analysis (NSA) *as a tool* to obtain the aforementioned computability theoretic results (whose formulation does not involve NSA). We provide a brief introduction to the computational content of NSA in Sect. 2.4. A detailed and elementary introduction to this topic may be found in [24], while advanced results are available in [19]. Finally, in Sect. 3.4, we provide a short and elegant derivation of Θ in full second-order arithmetic, also using NSA. This significantly improves the associated result from [19].

In conclusion, the results in this paper place a hard limit on the generality of the metastability trade-off and shows that metastability does not provide a 'king's road' towards computable mathematics. Along the way, we establish an intimate connection between higher-order computability theory and NSA, which we hope might spur the reader to become acquinted with the latter.

2 Background and Preliminaries

2.1 Introduction

First of all, we make our notion of computability precise.

(I) We adopt ZFC set theory as the official metatheory, unless stated otherwise.
(II) We adopt Kleene's notion of *higher-order computation* as given by his nine clauses S1-S9 (See [16, Definition 5.1.1, p. 170]) as our notion of 'computable'.

For reference and to promote understanding, we sometimes connect our results to well-known systems from the *Reverse Mathematics* program (See [26]).

Secondly, one does not require much familiarity with computability theory as in item (II) for this paper; we only assume familiarity with Gödel's system T of (higher-order) primitive recursion, and the associated formal system $\mathsf{E\text{-}PA}^{\omega*}$ from [3], i.e. Peano arithmetic with all finite types. We do point out one notational peculiarity: $\mathsf{E\text{-}PA}^{\omega*}$ has, for every finite type σ, the type σ^* of finite sequences of objects of type σ (See [3, Sect. 2]). In this way, '$x^\sigma \in y^{\sigma^*}$' is shorthand for $(\exists i^0 < |y|)(x =_\sigma y(i))$ where $|y|$ is the *length* of y, i.e. $|y| = k$ if $y = \langle x_0^\sigma, \dots, x_{k-1}^\sigma \rangle$, and the empty sequence (of any type) has length zero. A set of natural numbers X^1 is represented by its characteristic function, i.e. a binary sequence.

Thirdly, as to the structure of this section, we discuss basic results in and related to computability theory, namely Tao's notion of *metastability* and the associated trade-off in Sect. 2.2. As will be established in Sect. 3, applying the aforementioned trade-off to basic mathematical theorems will give rise to the *special fan functional* Θ, introduced in Sect. 2.3.

Finally, the functional Θ actually originates from Nonstandard Analysis (NSA), and we provide a brief introduction to the computational content of NSA in Sect. 2.4. The results in Sect. 3 are established by proving suitable theorems in NSA, and then translating these results to computability theory (no longer involving NSA) using Sect. 2.4. A detailed and elementary introduction to this topic is [24]; advanced results are available in [19].

2.2 The Textbook Example of Metastability

We study the 'textbook' example of metastability from [1, Sect. 1] and [28, Sect. 2.3.2]. We make use of basic type theoretic notation and the associated representation of real numbers (using fast-converging Cauchy sequences) from [13, Sect. 2].

Example 2.1 (Metastability). The monotone convergence theorem MCT states that *any non-decreasing sequence in* $[0,1]$ *converges to a limit.* Now, a convergent sequence is a *Cauchy sequence*, and the epsilon-delta definition of the latter (for a fixed sequence of rationals $a_{(\cdot)}$) is:

$$(\forall \varepsilon >_{\mathbb{R}} 0)\underline{(\exists N^0)(\forall n^0, m^0 \geq N)}(|a_n - a_m| \leq \varepsilon). \tag{2.1}$$

Using classical logic, (2.1) is equivalent to the definition of *metastable sequence*:

$$(\forall \varepsilon >_{\mathbb{R}} 0, F^1)\underline{(\exists M^0)}(\forall n^0, m^0 \in [M, F(M)])(|a_n - a_m| \leq \varepsilon). \tag{2.2}$$

While (2.1) and (2.2) are *classically* equivalent, we now show that their computational behaviour is quite different (a well-known phenomenon in general).

First of all, a *rate of convergence* for the sequence $a_{(\cdot)}$ is a function which provides an upper bound to N^0 from k^0 in (2.1). There are (Turing) *computable* monotone sequences of rationals in the unit interval for which there is no (Turing) computable rate of convergence (See e.g. [26, I.8.4] or [12, Sect. 13.3]). Furthermore, MCT is equivalent to ACA_0 in Reverse Mathematics, and the latter system implies the existence of a solution to the Halting problem ([26, III]).

Secondly, a *rate of metastability* for $a_{(\cdot)}$ provides an upper bound for M^0 from ε, F as in (2.2). As it happens, such a rate is given by the *elementary* function $F^{\lceil \frac{1}{\varepsilon} \rceil + 1}(0)$, i.e. the result of iterating F for $\lceil \frac{1}{\varepsilon} \rceil + 1$-many times and then evaluating at 0. In particular, M^0 as in (2.2) is exactly one of the values in the finite sequence $\langle 0, F(0), F(F(0)), \dots, F^{\lceil \frac{1}{\varepsilon} \rceil + 1}(0)\rangle$. We obtain:

$$\begin{aligned}(\exists \theta^{(0\times 1)\to 0^*})(\forall k^0, F, a_{(\cdot)} \in \mathsf{mon}([0,1]))(\exists M \in \theta(k, F)) \\ (\forall n, m \in [M, F(M)])(|a_n - a_m| \leq \tfrac{1}{k})\end{aligned} \tag{2.3}$$

where '$a_{(\cdot)} \in \mathsf{mon}([0,1])$' means that $a_{(\cdot)}$ is a non-decreasing sequence in $[0,1]$. In conclusion, while rates of convergence can be non-computable (in the sense of Turing), rates of metastability (in the case at hand) are *elementary* computable.

Example 2.1 illustrates a considerable advantage of metastability (2.2) over (2.1): By introducing F as in (2.2), we reduce the underlined pair of quantifiers of the form '$\exists$ $\forall$' in (2.1) to the underlined '$\exists$' quantifier in (2.2) as the universal quantifier $(\forall n, m)$ in (2.2) is bound by $F(M)$, and hence may be neglected. Thanks to this reduction in quantifier complexity (in favour of higher types), there is a finite sequence $\theta(k, F)$ which is *independent of the sequence* $a_{(\cdot)}$ as in (2.3), i.e. we in fact obtain *highly uniform* and computable information. Hence, we observe the following version of the aforementioned *metastability trade-off*:

> By introducing the interval $[M, F(M)]$ as in (2.2), one only obtains 'bounded' information (in contrast to the unbounded quantifier $(\forall n, m)$ in (2.1)) about the convergence of $a_{(\cdot)}$, but this information is *computable* and *highly uniform*, i.e. independent of the choice of sequence $a_{(\cdot)}$.

Now, there is nothing inherently special about convergence or metastability: Whenever we encounter a quantifier pair '$\exists$ $\forall$', we can formulate a classically equivalent 'metastable' version in which the universal quantifier is bounded, namely by introducing a functional similar to F as in (2.2). It is then a natural question whether, as suggested by the metastability trade-off and the quote in Sect. 1, we also obtain *computable* and *uniform* results. In Sect. 3, we shall study a number of theorems from basic mathematics from this point of view, i.e. we introduce finite domains in exchange for high(er) levels of uniformity.

Perhaps surprisingly, we obtain functionals of *extreme* computational hardness in each case; in particular, the special fan functional, introduced in the next section, shall come to the fore.

2.3 The Special Fan Functional

In this section, we introduce the special fan functional Θ, discuss its computational properties, and sketch its connection to NSA. The functional Θ emerges naturally from *Cousin's lemma* (published in 1895 in [5]), which is the Heine-Borel theorem *in the general*[3], i.e. the statement that any (possibly uncountable) open cover of the unit interval has a finite sub-cover. In particular, any Ψ^2 gives rise a 'canonical' open cover $\cup_{x\in[0,1]}(x-\frac{1}{\Psi(x)+1}, x+\frac{1}{\Psi(x)+1})$ of $[0,1]$. Hence, Cousin's lemma immediately implies:

$$(\forall \Psi^2)(\exists w^{1^*})(\forall x\in[0,1])(\exists y\in w)(|x-y|<\tfrac{1}{\Psi(y)}). \qquad (\mathsf{HBU})$$

The special fan functional computes such a finite sub-cover (given by w) from any Ψ^2 by Corollary 3.29. To avoid using representations of the real numbers, the special fan functional is defined on Cantor space. We usually talk about 'the' special fan functional Θ, though the latter is *not unique*, but given by the following specification. We reserve the variable 'T^1' for trees and denote by '$T^1\leq_1 1$' that T is a binary tree. We usually simplify the type of Θ to '3'.

Definition 2.2 (Special fan functional). For $\Theta^{(2\to(0\times 1^*))}$, $\mathsf{SCF}(\Theta)$ is as follows:

$$(\forall g^2, T\leq_1 1)\big[(\forall\alpha\in\Theta(g)(2))(\overline{\alpha}g(\alpha)\not\in T)\to(\forall\beta\leq_1 1)(\exists i\leq\Theta(g)(1))(\overline{\beta}i\not\in T)\big].$$

Any functional Θ satisfying $\mathsf{SCF}(\Theta)$ is referred to as a *special fan functional*.

From a computability theoretic perspective, the main property of Θ is the selection of $\Theta(g)(2)$ as a finite sequence of binary functions $\langle f_0,\dots,f_n\rangle$ such that the neighbourhoods defined from $\overline{f_i}g(f_i)$ for $i\leq n$ form a cover of Cantor space; almost as a by-product, $\Theta(g)(1)$ can then be chosen to be the maximal value of $g(f_i)+1$ for $i\leq n$. As it happens, Θ arises from the *nonstandard compactness of Cantor space* as in *Robinson's theorem* from NSA (See [6, p. 42]), as discussed in Sect. 2.4. In particular, Θ was first introduced in [21] in the study of the Gandy-Hyland functional using NSA.

Finally, Θ appears similar in name and behaviour to Tait's 'classical' fan functional. However, Θ behaves quite differently in that it *cannot* be computed by *any* type two functional, as proved in [19, Sect. 3]. As also proved in the latter, Θ can be computed from the 'second-order arithmetic' functional ξ^3 given by the following specification $\mathsf{SO}(\xi)$; one writes '$(\exists^3)$' for $(\exists\xi)\mathsf{SO}(\xi)$.

$$(\forall Y^2)\big[(\exists f^1)(Y(f)=0)\leftrightarrow\xi(Y)=0\big]. \qquad (\mathsf{SO}(\xi))$$

However, the intuitionistic fan functional MUC (See [13, Sect. 3]) does compute Θ (See [21, Sect. 3]), i.e. most of our results are part of intuitionistic mathematics.

[3] Heine-Borel theorem in Reverse Mathematics deals with *countable covers* ([26, IV.1]).

2.4 Nonstandard Analysis and Its Computational Content

We introduce Nelson's *internal set theory* IST and the fragment P based on *Peano Arithmetic* from [3]. We discuss the computational content of P, as this content is essential to the results in Sect. 3.

Internal Set Theory. In Nelson's *syntactic* (or 'axiomatic') approach to NSA ([17]), a new predicate 'st(x)', read as 'x is standard' is added to the language of ZFC, the usual foundation of mathematics[4]. The notations $(\forall^{st}x)$ and $(\exists^{st}y)$ are short for $(\forall x)(st(x) \to \dots)$ and $(\exists y)(st(y) \wedge \dots)$. A formula is called *internal* if it does not involve 'st', and *external* otherwise.

The external axioms *Idealisation*, *Standardisation*, and *Transfer* govern the new predicate 'st'; we state these axioms[5] as follows:

(I) $(\forall^{st\ \mathrm{fin}}x)(\exists y)(\forall z \in x)\varphi(z,y) \to (\exists y)(\forall^{st}x)\varphi(x,y)$, for any internal φ.
(S) $(\forall^{st}x)(\exists^{st}y)(\forall^{st}z)\big((z \in x \wedge \varphi(z)) \leftrightarrow z \in y\big)$, for any φ.
(T) $(\forall^{st}t)\big[(\forall^{st}x)\varphi(x,t) \to (\forall x)\varphi(x,t)\big]$, for internal φ with all variables shown.

The system IST is the *internal* system ZFC extended with the aforementioned *external* axioms; Internal set theory IST is a conservative extension of ZFC for the internal language ([17, Sect. 8]), i.e. these systems prove the same *internal* sentences. It goes without saying that the step from ZFC to IST can be done for a large spectrum of logical systems weaker than ZFC. In Sect. 2.4, we study this extension for *Peano Arithmetic with all finite types.*

The classical system P. In this section, we introduce the system P, a conservative extension of $\mathsf{E\text{-}PA}^{\omega*}$ with fragments of Nelson's IST.

We first introduce the system $\mathsf{E\text{-}PA}^{\omega*}_{st}$. We use the same definition as [3, Definition 6.1], where $\mathsf{E\text{-}PA}^{\omega*}$ is the definitional extension of $\mathsf{E\text{-}PA}^{\omega}$ with types for finite sequences as in [3, Sect. 2] and Sect. 2.1. The language of $\mathsf{E\text{-}PA}^{\omega*}_{st}$ (and P) is the language of $\mathsf{E\text{-}PA}^{\omega*}$ extended with a new symbol 'st_ρ' for any finite type ρ in the language of $\mathsf{E\text{-}PA}^{\omega*}$; the typing of 'st' is omitted.

Definition 2.3. The set $\mathcal{T}^*$ is the collection of all the terms in the language of $\mathsf{E\text{-}PA}^{\omega*}$. The system $\mathsf{E\text{-}PA}^{\omega*}_{\mathrm{st}}$ is $\mathsf{E\text{-}PA}^{\omega*} + \mathcal{T}^*_{\mathrm{st}} + \mathsf{IA}^{\mathrm{st}}$, where $\mathcal{T}^*_{\mathrm{st}}$ consists of:

(i) The schema[6] $\mathrm{st}(x) \wedge x = y \to \mathrm{st}(y)$,
(ii) The schema providing for each closed[7] term $t \in \mathcal{T}^*$ the axiom $st(t)$.

[4] The acronym ZFC stands for *Zermelo-Fraenkel set theory with the axiom of choice.*

[5] The superscript 'fin' in (I) means that x is finite, i.e. its number of elements is bounded by a natural number.

[6] The language of $\mathsf{E\text{-}PA}^{\omega*}_{st}$ contains a symbol st_σ for each finite type σ, but the subscript is essentially always omitted. Hence $\mathcal{T}^*_{st}$ is an *axiom schema* and not an axiom..

[7] A term is called *closed* in [3] if all variables are bound via lambda abstraction. Thus, if $\underline{x}, \underline{y}$ are the only variables occurring in the term t, the term $(\lambda \underline{x})(\lambda \underline{y})t(\underline{x}, \underline{y})$ is closed while $(\lambda \underline{x})t(\underline{x}, \underline{y})$ is not. The second axiom in Definition 2.3 thus expresses that $st_\tau\big((\lambda \underline{x})(\lambda \underline{y})t(\underline{x}, \underline{y})\big)$ if $(\lambda \underline{x})(\lambda \underline{y})t(\underline{x}, \underline{y})$ is type τ. We shall omit lambda abstraction.

(iii) The schema $\mathrm{st}(f) \wedge \mathrm{st}(x) \rightarrow \mathrm{st}(f(x))$.
(iv) The external induction axiom IA^{st}: For any external formula Φ:
$\Phi(0) \wedge (\forall^{st} n^0)(\Phi(n) \rightarrow \Phi(n+1)) \rightarrow (\forall^{st} n^0)\Phi(n)$.

Secondly, we introduce some essential fragments of IST studied in [3].

Definition 2.4 (External axioms of P)

1. $\mathsf{HAC}_{\mathsf{int}}$: For any internal formula φ, we have

$$(\forall^{st} x^\rho)(\exists^{st} y^\tau)\varphi(x,y) \rightarrow \big(\exists^{st} F^{\rho\rightarrow\tau^*}\big)(\forall^{st} x^\rho)(\exists y^\tau \in F(x))\varphi(x,y), \tag{2.4}$$

2. I: For any internal formula φ, we have

$$(\forall^{st} x^{\sigma^*})(\exists y^\tau)(\forall z^\sigma \in x)\varphi(z,y) \rightarrow (\exists y^\tau)(\forall^{st} x^\sigma)\varphi(x,y),$$

The system P is then defined as $\mathsf{E\text{-}PA}^{\omega*}_{\mathsf{st}} + \mathsf{I} + \mathsf{HAC}_{\mathsf{int}}$.

Note that I and $\mathsf{HAC}_{\mathsf{int}}$ are fragments of Nelson's axioms *Idealisation* and *Standard part*. By definition, F in (2.4) only provides a *finite sequence* of witnesses to $(\exists^{st} y)$, explaining its name *Herbrandized Axiom of Choice*.

Now, P is connected to $\mathsf{E\text{-}PA}^\omega$ by Theorem 2.5, which is essential to this paper. Indeed, Theorem 2.5 expresses that we may obtain effective results as in (2.6) from any theorem of NSA which has the same form as (2.5). By results in [21–25], the scope of Theorem 2.5 is huge.

Theorem 2.5 (Term Extraction). *For Δ_{int} a collection of internal formulas and ψ internal, if*

$$\mathsf{P} + \Delta_{\mathsf{int}} \vdash (\forall^{\mathrm{st}} \underline{x})(\exists^{\mathrm{st}} \underline{y})\psi(\underline{x}, \underline{y}, \underline{a}), \tag{2.5}$$

then one can extract from the proof a sequence of closed terms t in $\mathcal{T}^*$ such that

$$\mathsf{E\text{-}PA}^{\omega*} + \Delta_{\mathsf{int}} \vdash (\forall \underline{x})(\exists \underline{y} \in t(\underline{x}))\psi(\underline{x}, \underline{y}, \underline{a}). \tag{2.6}$$

Proof. See [21, Sect. 2], [23, Sect. 2], or [24, Appendix]. □

Note that the *conclusion* of the theorem, namely (2.6), *does not involve* NSA *anymore*. The term t from the previous theorem is part of Gödel's system T, i.e. essentially a computer program. For the rest of this paper, the notion 'normal form' shall refer to a formula as in (2.5), i.e. of the form $(\forall^{\mathrm{st}} x)(\exists^{\mathrm{st}} y)\varphi(x,y)$ for φ internal. For an internal formula φ, the (possibly external) φ^{st} is obtained by appending 'st' to all quantifiers, except bounded numerical ones.

Next, we show that Theorem 2.5 works for weaker systems.

1. Let $\mathsf{E\text{-}PRA}^\omega$ be the system defined in [13, Sect. 2] and let $\mathsf{E\text{-}PRA}^{\omega*}$ be its definitional extension with types for finite sequences as in [3, Sect. 2].
2. ($\mathsf{QF\text{-}AC}^{\rho,\tau}$) For every quantifier-free internal formula $\varphi(x,y)$, we have

$$(\forall x^\rho)(\exists y^\tau)\varphi(x,y) \rightarrow (\exists F^{\rho\rightarrow\tau})(\forall x^\rho)\varphi(x,F(x)) \tag{2.7}$$

The system RCA_0^ω is $\mathsf{E\text{-}PRA}^\omega + \mathsf{QF\text{-}AC}^{1,0}$, which is the 'base theory of higher-order Reverse Mathematics' as introduced in [13, Sect. 2]. We permit ourselves a slight abuse of notation by also referring to $\mathsf{E\text{-}PRA}^{\omega*} + \mathsf{QF\text{-}AC}^{1,0}$ as RCA_0^ω.

Corollary 2.6. *Theorem 2.5 goes through for* P *and* $\mathsf{E\text{-}PA}^{\omega*}$ *replaced by* $\mathsf{P}_0 \equiv \mathsf{E\text{-}PRA}^{\omega*} + \mathcal{T}_{\mathrm{st}}^* + \mathsf{HAC}_{\mathsf{int}} + \mathsf{I} + \mathsf{QF\text{-}AC}^{1,0}$ *and* RCA_0^ω*.*

Proof. The proof of [3, Theorem 7.7] goes through for any fragment of $\mathsf{E\text{-}PA}^{\omega*}$ which includes EFA, sometimes also called $\mathsf{I}\Delta_0 + \mathsf{EXP}$. In particular, the exponential function is (all what is) required to 'easily' manipulate finite sequences. □

Most of our results will pertain to P_0 and RCA_0^ω. We refer to [13, Sect. 3] for the usual definition (involving fast-converging Cauchy sequences) of real number, the associated equality '$=_{\mathbb{R}}$', and related notions inside RCA_0^ω. We finish with the following remark on how $\mathsf{HAC}_{\mathsf{int}}$ and I are used.

Remark 2.7 (Using $\mathsf{HAC}_{\mathsf{int}}$ and I). By definition, $\mathsf{HAC}_{\mathsf{int}}$ produces the functional $F^{\sigma\rightarrow\tau^*}$ which outputs a *finite sequence* of witnesses. However, $\mathsf{HAC}_{\mathsf{int}}$ provides an actual *witnessing functional* assuming (i) $\tau = 0$ in $\mathsf{HAC}_{\mathsf{int}}$ and (ii) the formula φ from $\mathsf{HAC}_{\mathsf{int}}$ is 'sufficiently monotone' as in: $(\forall^{st}x^\sigma, n^0, m^0)\big([n \leq_0 m \wedge \varphi(x,n)] \rightarrow \varphi(x,m)\big)$. Indeed, in this case one simply defines $G^{\sigma+1}$ by $G(x^\sigma) := \max_{i<|F(x)|} F(x)(i)$ which satisfies $(\forall^{st}x^\sigma)\varphi(x, G(x))$. To save space in proofs, we may skip the step involving the maximum of finite sequences. We assume the same convention for terms obtained from Theorem 2.5, and applications of the contraposition of idealisation I.

Nonstandard Compactness and Related Notions. We discuss some results from NSA. We will observe that the special fan functional Θ emerges from the *nonstandard compactness of* $2^{\mathbb{N}}$. The fragment of *Transfer* for Π_1^0-formulas is:

$$(\forall^{\mathrm{st}} f^1)\big[(\forall^{\mathrm{st}} n) f(n) \neq 0 \rightarrow (\forall m) f(m) \neq 0\big] \qquad (\Pi_1^0\mathsf{-TRANS})$$

is the nonstandard counterpart of arithmetical comprehension as in ACA_0. Similar to how one 'bootstraps' Π_1^0-comprehension to the latter, the system $\mathsf{P}_0 + \Pi_1^0\mathsf{-TRANS}$ proves $\varphi \leftrightarrow \varphi^{st}$ for any internal arithmetical formula (only involving standard parameters). The following fragment of *Standard Part* is the nonstandard counterpart of weak König's lemma ([10]):

$$(\forall \alpha^1 \leq_1 1)(\exists^{st} \beta^1 \leq_1 1)(\alpha \approx_1 \beta), \qquad (\mathsf{STP})$$

where $\alpha \approx_1 \beta$ is $(\forall^{st} n)(\alpha(n) =_0 \beta(n))$. Note that STP expresses the *nonstandard compactness of Cantor space* as in *Robinson's theorem* ([6, p. 42]). There is no deep meaning in the words 'nonstandard counterpart': This is just what STP and $\Pi_1^0\mathsf{-TRANS}$ are called in [10,23,27].

Secondly, the following theorem provides a normal form for STP and establishes a connection to Θ. In particular, the latter emerges from STP when applying Theorem 2.5. Note that $\mathsf{STP}_{\mathbb{R}}$ is the statement $(\forall x \in [0,1])(\exists^{st} y \in [0,1])(x \approx y)$, i.e. the nonstandard compactness of the unit interval, where infinitesimal proximity '$x \approx y$' is just $[x =_{\mathbb{R}} y]^{\mathrm{st}}$.

Theorem 2.8. *In* P, STP *is equivalent to* $\mathsf{STP}_{\mathbb{R}}$ *and to the following:*

$$(\forall^{st} g^2)(\exists^{st} w^{1^*} \leq_{1^*} 1, k^0)(\forall T^1 \leq_1 1)\big[(\forall \alpha^1 \in w)(\overline{\alpha} g(\alpha) \not\in T) \\ \rightarrow (\forall \beta \leq_1 1)(\exists i \leq k)(\overline{\beta} i \not\in T)\big], \tag{2.8}$$

and is equivalent to

$$(\forall T \leq_1 1)\big[(\forall^{st} n)(\exists \beta^0)(|\beta| = n \wedge \beta \in T) \rightarrow (\exists^{st} \alpha \leq_1 1)(\forall^{st} n^0)(\overline{\alpha} n \in T)\big]. \tag{2.9}$$

Furthermore, P_0 *proves* $(\exists^{\text{st}} \Theta)\mathsf{SCF}(\Theta) \rightarrow \mathsf{STP}$.

Proof. See e.g. [24, Appendix] or [21, Sect. 3]. □

By the theorem, STP is just WKL^{st} with the leading 'st' dropped; this observation explains why STP deserves the monicker 'nonstandard counterpart of WKL'.

3 Metastability and Computability

We apply the metastability trade-off from Sects. 1 and 2.2 to basic mathematical theorems, i.e. we introduce in the latter finite domains in exchange for high(er) levels of uniformity. In contrast to the effective results from proof mining (pertaining to metastability) mentioned in Sect. 1, we obtain the special fan functional Θ from Sect. 2.3. As the latter functional cannot be computed by any type two functional, we obtain a hard limit on the generality of the metastability trade-off. We shall study the *intermediate value theorem*, variations of the *extreme value theorem*, a theorem pertaining to *Riemann integration*, and BD-N.

3.1 The Intermediate Value Theorem

We apply the metastability trade-off to the *intermediate value theorem* (IVT), which states that a continuous function $f : I \rightarrow \mathbb{R}$ on $I = [0, 1]$ such that $f(0)f(1) <_{\mathbb{R}} 0$ has a zero, i.e. $(\exists x \in I)(f(x) =_{\mathbb{R}} 0)$. Unless **explicitly** stated, notions like continuity, convergence, etc. have their usual 'epsilon-delta' definitions, *not* involving NSA. The **nonstandard continuity** of $f : I \rightarrow \mathbb{R}$ is :

$$(\forall^{\text{st}} x \in I)(\forall y \in I)(x \approx y \rightarrow f(x) \approx f(y)). \tag{3.1}$$

As to its history, IVT is an example of a *non-constructive* theorem, as it implies a fragment of the law of excluded middle (See [2, I.7]). There are a number of *constructive* versions of IVT, e.g. dealing with *approximate* intermediate values like $(\forall k^0)(\exists q^0 \in I)(|f(q)| < \frac{1}{k})$ as in [4, II.4.8]. Furthermore, Kohlenbach introduces 'uniform IVT' in [13] where $\Phi^{(1\rightarrow 1)\rightarrow 1}$ outputs a zero on input a function $f^{1\rightarrow 1}$ as in IVT. Over RCA_0^{ω}, such a functional Φ computes (via a term of Gödel's T) the following 'arithmetical comprehension functional' by [13, Proposition 3.14]:

$$(\exists \varphi^2)(\forall f^1)\big[(\exists n^0)(f(n) = 0) \leftrightarrow \varphi(f) = 0\big]. \tag{$\exists^2$}$$

Hence, there is no way to *compute* intermediate values in general, and we shall therefore consider the following equivalence inspired by metastability:

$$(\exists x \in I)(f(x) =_{\mathbb{R}} 0) \leftrightarrow (\forall G^2)(\exists x \in I)(|f(x)| < \frac{1}{G(x)}). \tag{3.2}$$

The left-hand side of (3.2) expresses that f has a zero, and has the form '$\exists\ \forall$' due to the presence of '$=_{\mathbb{R}}$'. The right-hand side of (3.2) expresses that f has a 'metastable zero', and has the form '$\forall\ \exists$'. Furthermore, the following holds (even constructively) for continuous $f : I \to \mathbb{R}$, namely $(\exists x \in I)(f(x) =_{\mathbb{R}} 0)$ implies

$$(\forall G^2)(\exists x \in I)(|f(x)| < \tfrac{1}{G(x)}) \to (\forall k^0)(\exists q^0 \in I)(|f(q)| < \tfrac{1}{k}).$$

In light of the previous, we expect that computing a metastable zero is at most as hard as computing $(\exists^2)$ and at best computable. The question however remains *which inputs* we shall use to compute a metastable zero, besides G^2 as in (3.2). We provide the following answer:

First of all, an *indispensable* part of an 'epsilon-delta' definition in constructive/computable mathematics is the *modulus*, which computes the delta from the epsilon (and other data). Moreover, in the case of Riemann integration on I, a modulus of uniform continuity suffices to obtain a modulus of Riemann integration (See [4, p. 51]). Hence, to obtain highly uniform results *as suggested by the metastability trade-off*, it makes sense (in general) to keep the modulus of continuity as input and omit the function as input.

Secondly, we mentioned approximate versions of IVT above; it is fairly easy to show that there is a term t from Gödel's T such that for k^0 and $f : [0,1] \to \mathbb{R}$ with modulus of continuity H and $f(0)f(1) <_{\mathbb{R}} 0$, we have $(\exists q^0 \in t(H))(|f(q)| < \frac{1}{k})$. This fact will be proved below in Theorem 3.7 and we stress that the term t only depends on the modulus of continuity H. Hence, to obtain highly uniform results *as suggested by the metastability trade-off*, it makes sense (in the case of IVT) to keep the modulus of continuity as input and omit f as input.

The previous leads us to the following 'metastable' version of IVT in which a metastable zero is computed (only) from a modulus of continuity of the function at hand, i.e. a uniform result. Note that Ψ provides a finite sequence of witnesses to '$(\exists x \in I)$', just as θ in (2.3).

Principle 3.1 ($\mathsf{IVT}_{\mathsf{meta}}$). *There is* $\Psi^{2\to 1^*}$ *such that for* $f : [0,1] \to \mathbb{R}$ *with modulus of cont.* H^2 *and* $f(0)f(1) < 0$, $(\forall G^2)(\exists x \in \Psi(G,H))\big(x \in I \wedge |f(x)| < \frac{1}{G(x)}\big)$.

We could require that H in $\mathsf{IVT}_{\mathsf{meta}}$ be continuous (in the usual epsilon-delta sense), but that would not really change any of the below results. As noted above, we expect that computing a metastable zero is at most as hard as computing $(\exists^2)$. Such 'great expectations' turn out to be spectacularly wrong: Ψ as in $\mathsf{IVT}_{\mathsf{meta}}$ is surprisingly hard to compute by the following theorem, where $\mathsf{IVT}_{\mathsf{meta}}(\Psi)$ is just $\mathsf{IVT}_{\mathsf{meta}}$ with the leading existential quantifier dropped.

Theorem 3.2 (ZFC). *A functional* Ψ *satisfying* $\mathsf{IVT}_{\mathsf{meta}}(\Psi)$ *can be computed from* ξ^3 *as in* $(\exists^3)$. *No type two functional can compute such a functional* Ψ.

As noted above, we shall make use of NSA to prove Theorem 3.2. To this end, we introduce the following somewhat subtle[8] nonstandard version of IVT.

Definition 3.3 ($\mathsf{IVT}_{\mathsf{ns}}$). For $f : [0,1] \to \mathbb{R}$ with *standard* modulus of continuity and $f(0)f(1) <_{\mathbb{R}} 0$, we have $(\exists^{st} x \in [0,1])(f(x) \approx 0)$.

Theorem 3.4. *The system P_0 proves $\mathsf{STP} \leftrightarrow \mathsf{IVT}_{\mathsf{ns}}$.*

Proof. For the forward direction, we imitate the usual proof of IVT as follows: If for f as in $\mathsf{IVT}_{\mathsf{ns}}$, there is standard and rational $q^0 \in [0,1]$ such that $f(q) \approx 0$, we are done. Otherwise, we have $(\forall^{st} q^0 \in I)(\exists^{st} k^0)(|f(q)| > \frac{1}{k})$. Applying $\mathsf{HAC}_{\mathsf{int}}$, there is standard $\Phi^{0\to 0^*}$ such that $(\forall^{st} q^0 \in I)(\exists k^0 \in \Phi(q))(|f(q)| > \frac{1}{k})$. Now define g^1 as follows: $g(q) := \max_{i<|\Phi(q)|} \Phi(q)(i)$. Hence, we have $(\forall^{st} q^0 \in I)(|f(q)| > \frac{1}{g(q)})$. Approximating the function values of f up to precision $2^{g(q)}$, we can decide whether $f(q) \gg 0$ or $f(q) \ll 0$ for any standard $q^0 \in [0,1]$. Now use the usual interval halving technique (See [26, II.6.6]) relative to 'st' to define a real x_0 such that $f(x_0) \approx 0$. By STP, there is *standard* $x_1 \approx x_0$. Now, f as in $\mathsf{IVT}_{\mathsf{ns}}$ is also *nonstandard* continuous as in (3.1), as its modulus of continuity is standard, implying $f(x_1) \approx f(x_0) \approx 0$. For the reverse direction, assume $\mathsf{IVT}_{\mathsf{ns}}$ and fix $x_0 \in [0,1]$. Now consider $f : [0,1] \to \mathbb{R}$ defined as $f(x) := x - x_0$, which has a *standard* modulus of continuity, namely H^2 defined as $H(x^1, k^0) := k^0$. Clearly, we have $|f(x)| \gg 0$ for $x \not\approx x_0$, and since $\mathsf{IVT}_{\mathsf{ns}}$ implies there is *standard* $x_1 \in [0,1]$ such that $f(x_1) \approx 0$, we must have $x_0 \approx x_1$, and STP follows. □

Here, $\mathsf{SCF}(\Theta)$ is the specification of the special fan functional from Sect. 2.3.

Corollary 3.5. *There are terms s, t of Gödel's T such that RCA_0^ω proves:*

$$(\forall \Theta^3)\big[\mathsf{SCF}(\Theta) \to \mathsf{IVT}_{\mathsf{meta}}(s(\Theta))\big] \wedge (\forall \Psi)\big[\mathsf{IVT}_{\mathsf{meta}}(\Psi) \to \mathsf{SCF}(t(\Psi))\big], \tag{3.3}$$

Proof. We shall prove the second conjunction of (3.3) in detail, and leave the first one to the reader. By Theorem 2.8, STP has a normal form as in (2.8), which we abbreviate by $(\forall^{st} g^2)(\exists^{st} w^{1^*} \leq_{1^*} 1, k^0)\varphi(g, w, k)$. We now obtain a normal form for $\mathsf{IVT}_{\mathsf{ns}}$; the latter yields the following, where '$\mathsf{MPC}(f, H)$' expresses that H is a modulus of continuity for f, and '$f \in D$' that $f : [0,1] \to \mathbb{R} \wedge f(0)f(1) <_{\mathbb{R}} 0$:

$$(\forall^{st} H^2)(\forall f \in D)\big[\mathsf{MPC}(f, H) \to (\forall^{st} G^2)(\exists^{st} x \in [0,1])(|f(x)| < \tfrac{1}{G(x)})\big].$$

Pulling outside all standard quantifiers as far as possible, we obtain the following:

$$(\forall^{st} H^2, G^2)(\forall f \in D)(\exists^{st} x \in [0,1])\big[\mathsf{MPC}(f, H) \to (|f(x)| < \tfrac{1}{G(x)})\big].$$

[8] Note that $\mathsf{IVT}_{\mathsf{ns}}$ looks fairly close to $\mathsf{IVT}^{\mathrm{st}}$, and the latter is provable in P_0. Nonetheless, $\mathsf{IVT}_{\mathsf{ns}}$ turns out to be equivalent to STP from Sect. 2.4. Furthermore, an obvious mistake is to apply *Transfer* to the conclusion of $\mathsf{IVT}_{\mathsf{ns}}$ and conclude there is a *standard* intermediate value. Indeed, the function f from $\mathsf{IVT}_{\mathsf{ns}}$ need not be standard, making application of the *Transfer* axiom *illegal*, following Nelson [17].

Let $B(f,G,H)$ be the internal formula in square brackets in the previous formula; applying *Idealisation* to the latter yields:

$$(\forall^{st} H^2, G^2)(\exists^{st} y^{1^*})(\forall f \in D)(\exists x \in y)B(f,G,H), \tag{3.4}$$

which is a normal form, abbreviated $(\forall^{st} H^2, G^2)(\exists^{st} y^{1^*})\psi(H,G,y)$. Hence, the implication $\mathsf{IVT}_{\mathsf{ns}} \to \mathsf{STP}$ yields

$$(\forall^{st} H^2, G^2)(\exists^{st} y^{1^*})\psi(H,G,y) \to (\forall^{st} g^2)(\exists^{st} w^{1^*} \leq_{1^*} 1, k^0)\varphi(g,w,k), \tag{3.5}$$

and it is straightforward to obtain a normal form from (3.5) (See e.g. [24, Remark 4.8]). To further understanding, we will spell out how to obtain a normal from (3.5) *this once*: Since standard functionals produce standard outputs from standard inputs, (3.5) implies that for all standard $\varPsi^{2\to 1^*}$

$$(\forall^{st} H^2, G^2)\psi(H,G,\varPsi(G,H)) \to (\forall^{st} g^2)(\exists^{st} w^{1^*} \leq_{1^*} 1, k^0)\varphi(g,w,k).$$

We may trivially drop the remaining 'st' in the antecedent of the previous:

$$(\forall^{st} \varPsi^{2\to 1^*})\big[(\forall H^2, G^2)\psi(H,G,\varPsi(G,H)) \to (\forall^{st} g^2)(\exists^{st} w^{1^*} \leq_{1^*} 1, k^0)\varphi(g,w,k)\big],$$

and bringing outside all standard quantifiers, we obtain a normal form:

$$(\forall^{st} \varPsi^{2\to 1^*}, g^2)(\exists^{st} w \leq 1, k^0)\big[(\forall H^2, G^2)\psi(H,G,\varPsi(G,H)) \to \varphi(g,w,k)\big], \tag{3.6}$$

Applying Theorem 2.5 to '$\mathsf{P}_0 \vdash$ (3.6)' yields a term t of Gödel's T such that

$$(\forall \varPsi, g^2)(\exists w \leq 1, k \in t(\varPsi, g))\big[(\forall H^2, G^2)\psi(H,G,\varPsi(G,H)) \to \varphi(g,w,k)\big] \tag{3.7}$$

is provable in RCA_0^ω. Now, (3.7) implies the second conjunct of (3.3) by bringing the existential quantifiers into the consequent. Indeed, the antecedent of (3.7) is exactly $\mathsf{IVT}_{\mathsf{meta}}(\varPsi)$, while for such $\varPsi$, (3.7) yields $(\forall g^2)(\exists w^{1^*} \leq_{1^*} 1, k^0 \in t(\varPsi, g))\varphi(g,w,k)$, which readily implies the specification of $\varTheta$. □

Theorem 3.2 is now immediate from Corollary 3.5 and the following ([19, Sect. 3]).

Theorem 3.6 (ZFC). *A functional $\varTheta$ satisfying $\mathsf{SCF}(\varTheta)$ can be computed from $\exists^3$. No type two functional can compute such a functional $\varTheta$.*

As an aside, the 'hardness' of $\mathsf{IVT}_{\mathsf{meta}}$ is **not** due to the use of higher types: The 'lower type' version (3.8) of $\mathsf{IVT}_{\mathsf{meta}}$ cannot be proved in $\mathsf{RCA}_0^\omega + \mathsf{QF\text{-}AC} + (S^2)$, a $\varPi^1_3$-conservative extension of $\varPi^1_1$-CA_0. This result is however beyond the scope of this paper and requires results from [20].

$$(\forall G^2, H^2)(\exists w^{1^*})(\forall f \in D)\big(\mathsf{MPC}(f,H) \to (\exists x \in w)\big(|f(x)| < \frac{1}{G(x)}\big). \tag{3.8}$$

Finally, what is left is to prove the following theorem.

Theorem 3.7. *There is a term t from Gödel's T such that RCA_0^ω proves that for k^0 and $f \in D$ with modulus of continuity H, we have $(\exists q^0 \in t(H))(|f(q)| < \frac{1}{k})$.*

Proof. The theorem is established by modifying the proof of Corollary 3.5 to apply to '$\mathsf{P}_0 \vdash \mathsf{IVT}'_{\mathsf{ns}}$' (rather than $\mathsf{P}_0 \vdash [\mathsf{IVT}_{\mathsf{ns}} \to \mathsf{STP}]$); $\mathsf{IVT}'_{\mathsf{ns}}$ is defined as:

Definition 3.8 ($\mathsf{IVT}'_{\mathsf{ns}}$). For $f : [0,1] \to \mathbb{R}$ with *standard* modulus of continuity and $f(0)f(1) <_{\mathbb{R}} 0$, we have $(\forall^{st} k^0)(\exists^{st} q^0 \in [0,1])(|f(q)| < \frac{1}{k})$.

Note that $\mathsf{IVT}'_{\mathsf{ns}}$ has a normal form similar to (3.4), i.e. applying Theorem 2.5 is straightforward. To prove $\mathsf{IVT}'_{\mathsf{ns}}$ inside P_0, consider the proof of the forward implication in Theorem 3.4. Note that STP is only needed to convert x_0 to a standard real x_1, i.e. we can decide *inside* P_0 whether $f(q) \gg 0$ or $f(q) \ll 0$ for $q^0 \in [0,1]$ (if there is no standard rational $r^0 \in I$ such that $f(r) \approx 0$). Use the interval halving technique ([26, II.6.6]) relative to 'st' to define (for any standard k) q_k such that $|f(q_k)| \leq \frac{1}{2^k}$. By definition, q_k is standard for standard k. □

The final step of the proof involves a hidden subtlety: while q_k is a standard rational for every standard k^0, the (type one) *function* $\lambda k.q_k$ is *not* a standard function (since f may be nonstandard). This situation is actually familiar: every *binary* sequence α^1 is such that $\alpha(k)$ is standard (for all k^0), while α^1 itself may not be a standard sequence.

3.2 On Maxima, Suprema, and Integrals

It goes without saying that the results regarding IVT from Sect. 3.1 can be obtained for similar theorems. In this section, we discuss such results, but also uncover some non-obvious pitfalls. Our main result is that applying the metastability trade-off to theorems of *constructive mathematics* yields Θ. Thus, our results are not a 'defect' of classical logic, but also come to the fore constructively.

The Extreme Value Theorem. We apply the metastability trade-off to the *extreme value theorem* (EVT), which is the statement that a continuous function $f : I \to \mathbb{R}$ attains its maximum, i.e. $(\exists y \in I)(\forall x \in I)(f(x) \leq f(y))$. The treatment of EVT is similar to that of IVT.

As to its history, EVT is equivalent to WKL in Reverse Mathematics by [26, IV.2.3]. As for IVT, Kohlenbach introduces 'uniform EVT' in [13] where $\Phi^{1\to 1}$ outputs $y \in I$ such that $(\forall x \in I)(f(x) \leq_{\mathbb{R}} f(y))$ on input a function f as in EVT. In the presence of the axiom of extensionality, such a Φ also computes $(\exists^2)$ by [13, Proposition 3.14]. Hence, there is no way to (extensionally) compute extreme values in general, and we shall therefore consider the notion of 'metastable maximum', similar to 'metastable zero' as in (3.2). In particular, the following holds (even constructively) for continuous $f : I \to \mathbb{R}$, while the reversal holds classically:

$$\begin{aligned}(\exists y \in I)&(\forall x \in I)(f(x) \leq_{\mathbb{R}} f(y))\\ &\to (\forall G^{1\to 1^*})(\exists y \in I)(\forall x \in G(y) \cap I)(f(x) <_{\mathbb{R}} f(y) + \tfrac{1}{|G(y)|+1}).\end{aligned}$$

We say that 'f has a metastable maximum' if it satisfies the consequent. Inspired by $\mathsf{IVT}_{\mathsf{meta}}$, we formulate the following 'metastable' version of EVT in which a metastable maximum is computed (only) from a modulus of continuity.

Principle 3.9 ($\mathsf{EVT}_{\mathsf{meta}}$). *There is $\Psi^{2\rightarrow 1^*}$ such that for $f:[0,1]\rightarrow\mathbb{R}$ with modulus of continuity H^2, we have $(\forall G^{1\rightarrow 1^*})(\exists y\in\Psi(G,H)\cap I)(\forall x\in G(y)\cap I)(f(x)<_{\mathbb{R}} f(y)+\frac{1}{|G(y)|+1})$.*

We could require that f in $\mathsf{EVT}_{\mathsf{meta}}$ be *uniformly* continuous (in the usual epsilon-delta sense), but that would not really change any of the below results. We now show that Ψ as in $\mathsf{EVT}_{\mathsf{meta}}$ is surprisingly hard to compute by the following theorem, where $\mathsf{EVT}_{\mathsf{meta}}(\Psi)$ is just $\mathsf{EVT}_{\mathsf{meta}}$ without the leading existential quantifier.

Theorem 3.10 (ZFC). *A functional Ψ satisfying $\mathsf{EVT}_{\mathsf{meta}}(\Psi)$ can be computed $(\exists^3)$. No type two functional can compute such a functional Ψ.*

As above, we shall make use of NSA to prove Theorem 3.10. To this end, we introduce the following nonstandard version of EVT, similar to $\mathsf{IVT}_{\mathsf{ns}}$

Definition 3.11 ($\mathsf{EVT}_{\mathsf{ns}}$). For $f:[0,1]\rightarrow\mathbb{R}$ with *standard* modulus of continuity, we have $(\exists^{st}x\in[0,1])(\forall^{st}y\in[0,1])(f(y)\lessapprox f(x))$.

The use of WKL in the following proof is avoidable: we could require that H in $\mathsf{EVT}_{\mathsf{meta}}$ and $\mathsf{EVT}_{\mathsf{ns}}$ be continuous (in the usual sense), and the only change would be that we may remove WKL from the following theorem and corollary.

Theorem 3.12. *The system $\mathsf{P}_0+\mathsf{WKL}$ proves $\mathsf{STP}\leftrightarrow\mathsf{EVT}_{\mathsf{ns}}$.*

Proof. For the forward direction, WKL implies the usual extreme value theorem by combining [26, IV.2.3] and [14, Theorem 4.10]. Hence, f as in $\mathsf{EVT}_{\mathsf{ns}}$ satisfies $(\exists x_0\in[0,1])(\forall y\in[0,1])(f(y)\leq_{\mathbb{R}} f(x_0))$. Now let x_0 be as in the latter and use STP to obtain standard $x_1\approx x_0$. Since f as in $\mathsf{EVT}_{\mathsf{ns}}$ is also nonstandard continuous, we have $(\forall^{st}y\in[0,1])(f(y)\lessapprox f(x_1))$. For the reverse direction, assume $\mathsf{EVT}_{\mathsf{ns}}$ and fix $x_0\in[0,1]$ such that $0\not\approx x_0\not\approx 1$. Now consider $f:[0,1]\rightarrow\mathbb{R}$ defined as x/x_0 if $x\leq x_0$ and $\frac{-x}{3x_0}+\frac{4}{3}$ otherwise; the function f has a *standard* modulus of continuity, which is readily defined using any *standard* a,b such that $0\ll a<_{\mathbb{R}} x_0<_{\mathbb{R}} b\ll 1$ (which exist by assumption). Now $f(x_0)=_{\mathbb{R}}1$ and $f(x)\ll 1$ when $x\not\approx x_0\wedge x\in I$, and since $\mathsf{EVT}_{\mathsf{ns}}$ implies there is *standard* $x_1\in[0,1]$ such that $(\forall^{st}x\in I)(f(x)\lessapprox f(x_1))$, we have $x_0\approx x_1$. □

Corollary 3.13. *There are terms s,t of Gödel's T such that $\mathsf{RCA}_0^\omega+\mathsf{WKL}$ proves:*

$$(\forall\Theta^3)\big[\mathsf{SCF}(\Theta)\rightarrow\mathsf{EVT}_{\mathsf{meta}}(s(\Theta))\big]\wedge(\forall\Psi)\big[\mathsf{EVT}_{\mathsf{meta}}(\Psi)\rightarrow\mathsf{SCF}(t(\Psi))\big],\tag{3.9}$$

Proof. Analogous to Corollary 3.5. Note that the conclusion of $\mathsf{EVT}_{\mathsf{ns}}$, namely the formula $(\exists^{st}x\in[0,1])(\forall^{st}y\in[0,1])(f(y)\lessapprox f(x))$, implies a normal form:

$$(\forall^{st}G^{2\rightarrow 1^*})(\exists^{st}x\in[0,1])(\forall y\in G(x)\cap I)(f(y)<_{\mathbb{R}} f(x)+\tfrac{1}{|G(x)|+1}),$$

as $G(x)$ is standard (thus with only standard elements) for standard x, and hence its length $|G(x)|$ is also standard (by the axioms in Definition 2.3). □

Theorem 3.10 is now immediate from Corollary 3.13 and Theorem 3.6.

Supremum of Continuous Functions. In this section, we apply the metastability trade-off to a version of EVT which states the existence of a *supremum*. We shall observe a non-obvious complication which does not occur for EVT itself. On the other hand, the notion of supremum enables us to study a *constructive* theorem, namely that a uniformly continuous (with a modulus) function on I has a supremum (See [4, p. 94]). Despite the latter theorem's constructive standing, applying the metastability trade-off results in functionals computing Θ.

First of all, a word regarding variations of the above argument for EVT: a function with a standard modulus of continuity can only have a *standard* supremum if (and only if) it additionally has a standard upper bound. Thus, any variation of $\mathsf{EVT}_{\mathsf{ns}}$ involving the supremum of a function (rather than the attainment of a maximum) *must* assume a *standard* upper bound on that function, while such a bound is absent from $\mathsf{EVT}_{\mathsf{ns}}$ itself. The same holds for $\mathsf{EVT}_{\mathsf{meta}}$, which however is quite complicated compared to e.g. $\mathsf{IVT}_{\mathsf{meta}}$.

Definition 3.14 ($\mathsf{SUP}_{\mathsf{ns}}$). For $f : [0,1] \to \mathbb{R}$ with *standard* modulus of uniform continuity and $(\exists^{st} N^0)(\forall x \in I)(f(x) \leq_{\mathbb{R}} N)$, there is standard y^1 such that

$$(\forall^{st} x \in [0,1])(f(x) \lessapprox y) \wedge (\forall^{st} k^0)(\exists^{st} z \in I)(f(z) > y - \tfrac{1}{k})\big]. \tag{3.10}$$

Principle 3.15 ($\mathsf{SUP}_{\mathsf{meta}}$). *There is $\Psi^{2\to 1^*}$ such that for $f : I \to \mathbb{R}$ with modulus of continuity H^2 and upper bound N^0 on I, and all $G^{1\to 1^*}$ and k^0 we have:*

$$\begin{aligned}(\exists y \in \Psi(G,H,N,k))(\forall x \in G(y) \cap I)(f(x) <_{\mathbb{R}} y + \tfrac{1}{|G(y)|})\\ \wedge (\exists z \in \Psi(G,H,N,k) \cap I)(f(z) > y - \tfrac{1}{k}).\end{aligned}$$

Theorem 3.16. *The system $\mathsf{P}_0 + \mathsf{WKL}$ proves $\mathsf{STP} \leftrightarrow \mathsf{SUP}_{\mathsf{ns}}$.*

Proof. The forward direction is immediate by Theorem 3.12 as we clearly have $\mathsf{EVT}_{\mathsf{ns}} \to \mathsf{SUP}_{\mathsf{ns}}$. The reverse direction follows in the same way as for Theorem 3.12 by fixing $x_0 \in I$ and considering the function $f(x) := -x^2 + 2x_0 x$ (with obvious standard modulus of uniform continuity). Clearly, $f(x_0) = x_0^2$ and if $x \not\approx x_0$, we have $f(x_0) \ll x_0^2$. By $\mathsf{SUP}_{\mathsf{ns}}$, f has a standard supremum y^1 as in (3.10), and we must have $x_0^2 \approx y$. Since $\sqrt{y}$ is also standard, $x_0 \approx \sqrt{y}$. □

Corollary 3.17. *There are terms s, t of Gödel's T such that $\mathsf{RCA}_0^\omega + \mathsf{WKL}$ proves:*

$$(\forall \Theta^3)\big[\mathsf{SCF}(\Theta) \to \mathsf{SUP}_{\mathsf{meta}}(s(\Theta))\big] \wedge (\forall \Psi)\big[\mathsf{SUP}_{\mathsf{meta}}(\Psi) \to \mathsf{SCF}(t(\Psi))\big], \tag{3.11}$$

Proof. Analogous to Corollary 3.5. □

In conclusion, applying the metastability trade-off to *constructive* theorems can also give rise to Θ. One however needs to be careful: the treatment of suprema and maxima differs in a non-obvious way. Finally, $\mathsf{SUP}_{\mathsf{meta}}$ is not that elegant; we remedy this in the next section.

Riemann Integration. We previously studied the *non-constructive* IVT and EVT, while the results from Sect. 3.2 based on constructive mathematics were not that elegant. Lest the reader gets the idea that theorems of constructive mathematics cannot (elegantly and naturally) give rise to the special fan functional, we now study the statement that a uniformly continuous function is Riemann integrable on I ([4, p. 51]). As above, applying the metastability trade-off to this *constructive* theorem, gives rise to the special fan functional.

Bishop proves in [4, 6.6.3] that for $f : I \to \mathbb{R}$ with modulus of *uniform* continuity g^1, this modulus is also a modulus of Riemann integration for f. The Riemann sum $S_n(f)$ is defined (essentially) as $\sum_{i=0}^{2^n} f(\frac{i}{2^n})\frac{1}{2^n}$, and for $n \to \infty$ the latter is shown to converge to the Riemann integral $\int_0^1 f(x)dx$. Constructively (and with a classical equivalence), the previous implies that

$$(\forall G^2)(\exists x^1)(\forall k^0, n^0 \leq G(x))(n \geq 2^{g(k)} \to |S_n(f) - \textstyle\int_0^1 f(x)dx| < \frac{1}{k}), \tag{3.12}$$

for f with modulus of uniform continuity g^1. The following principle states that we can *uniformly* compute a 'metastable integral' as in (3.12), assuming an upper and lower bound.

Principle 3.18 ($\mathsf{RIE}_{\mathsf{meta}}$). *There is $\Psi^{(1\to1)\to1^*}$ such that for $f : [0,1] \to \mathbb{R}$ with modulus of uniform continuity g^1 and bound N^0 such that $(\forall x \in I)(|f(x)| \leq N)$:*

$$(\forall G^2)(\exists x^1 \in \Psi(G,N))(\forall k^0, n^0 \leq G(x))(n \geq 2^{g(k)} \to |S_n(f) - x| < \tfrac{1}{k}). \tag{3.13}$$

As discussed in Sect. 3.1, to obtain highly uniform results *as suggested by the metastability trade-off*, it makes sense to keep the modulus of continuity as input and omit the function as input. Furthermore, (2.3) does not depend on the choice of sequence $a_{(\cdot)}$, while Riemann integration deals with the convergence of the sequence $S_n(f)$. Hence, the functional Ψ as in (3.13) from $\mathsf{RIE}_{\mathsf{meta}}$ should be independent of $S_n(f)$, and hence definitely be independent of f.

To establish the properties of $\mathsf{RIE}_{\mathsf{meta}}$, we need a nonstandard principle.

Definition 3.19 ($\mathsf{RIE}_{\mathsf{ns}}$). For $f : [0,1] \to \mathbb{R}$ *with* standard *modulus of uniform cont. g^1 and such that* $(\exists^{st} N^0)(\forall x \in I)(|f(x)| \leq_{\mathbb{R}} N)$, *there is standard y^1 with*

$$(\forall^{st} k^0, n^0)(n \geq 2^{g(k)} \to |S_n(f) - y| < \tfrac{1}{k}). \tag{3.14}$$

It is an easy exercise to see that the bound N is essential.

Theorem 3.20. *The system P_0 proves $\mathsf{STP} \leftrightarrow \mathsf{RIE}_{\mathsf{ns}}$.*

Proof. The forward direction is almost immediate as follows: for f as in $\mathsf{RIE}_{\mathsf{ns}}$, the Riemann integral exists by following the usual proof (not involving 'st') in constructive ([4, 6.6.3]) or computable ([26], IV.2.6) mathematics. For (standard) N as in $\mathsf{RIE}_{\mathsf{ns}}$, the Riemann integral is a real x in $[-N, N]$, and use STP to obtain a *standard* y such that $x \approx y$. The reverse direction follows by fixing $x_0 \in I$ and considering $f(x) := \frac{x_0}{e-1}e^x$ (with obvious standard modulus of uniform continuity and standard bounds). By $\mathsf{RIE}_{\mathsf{ns}}$, there is a standard y^1 such that (3.14) and apply *overspill* (See [3, Prop. 3.3]) to the latter to obtain nonstandard M^0 such that $S_M(f) \approx y$. Since $S_M(f) \approx \int_0^1 f(x)dx = x_0$, we obtain $\mathsf{STP}_{\mathbb{R}}$ from $y \approx x_0$. □

Corollary 3.21. *There are terms s, t of Gödel's T such that* RCA_0^ω *proves:*

$$(\forall \Theta^3)\big[\mathsf{SCF}(\Theta) \to \mathsf{RIE}_{\mathsf{meta}}(s(\Theta))\big] \wedge (\forall \Psi)\big[\mathsf{RIE}_{\mathsf{meta}}(\Psi) \to \mathsf{SCF}(t(\Psi))\big], \tag{3.15}$$

Proof. Analogous to Corollary 3.5. □

3.3 The Principle BD-N

In this section, we apply the metastability trade-off to the 'boundedness' principle BD-N, and obtain Θ in the process. The BD-N axiom deals exclusively with sets of natural numbers, i.e. BD-N is 'more basic' than e.g. IVT in the sense that it involves only objects of types zero and one. Another interesting aspect of BD-N is that it yields a number of variations, all of which however give rise to Θ, i.e. our results exhibit some robustness.

First of all, as to its provenance, BD-N was introduced by Ishihara ([8,9]) and is an example of a statement which can be proved in classical, recursive, and intuitionistic mathematics, but cannot be proved in (systems generally considered providing formalisations of) Bishop's constructive analysis ([7,15]). Now, BD-N is the statement that a countable inhabited *pseudo-bounded* subset of $\mathbb{N}$, is also bounded; a set $X \subseteq \mathbb{N}$ is *pseudo-bounded* if

$$(\forall s^1_{(\cdot)})\big[(\forall n^0)(s_n \in X) \to (\exists k^0)(\forall n \geq k)(s_n < n)\big]. \tag{3.16}$$

Secondly, Normann shows in [18, Example 2.6] that BD-N fails in Kleene's second model K_2. In particular, the upper bound claimed to exist by BD-N cannot be computed (in the sense of K_2) from the other data. Now, the notion of *upper bound* of a set $X \subset \mathbb{N}$ has the typical '$\exists$ $\forall$' structure which invites the modification involving $[M, F(M)]$ as in metastability (2.2):

$$(\exists k^0)(\forall n \geq k)(n \notin X) \leftrightarrow (\forall g^1)(\exists k^0)(\forall n \in [k, g(k)])(n \notin X). \tag{3.17}$$

We say that $X \subset \mathbb{N}$ is *metastable-bounded* if it satisfies the right-hand side of (3.17); a functional G^2 outputting an upper bound to k^0 from g^1 is called a *rate of metastable-boundedness.*

It is now a natural question *from what inputs* we should compute a rate of metastable-boundedness: We could use a realiser for (3.16), but we could equally well use a realiser for (3.18), which introduces the notion of *metastable-pseudo-bounded* for X as follows:

$$(\forall s^1_{(\cdot)})\big[(\forall n)(s_n \in X) \to (\forall h^1)(\exists k^0)(\forall n^0 \in [k, h(k)])(s_n < n)\big]. \tag{3.18}$$

As we will establish below, either version gives rise to the special fan functional when applying the metastability trade-off. Finally, since metastable-boundedness -for fixed g^1 in (3.17)- only provides *approximations* to actual upper bounds, it seems reasonable that for such fixed g^1, we can compute k^0 such that $[k, g(k)] \cap X = \emptyset$ by *only* making use of (3.16) or (3.18) for a finite number of sequences $s_{(\cdot)}$ (dependent on the choice of g and other data).

The previous considerations lead to two versions of 'metastable BD-N'.

Definition 3.22 **($\mathsf{BD\text{-}N_{meta}}$).** There is Ψ such that for all X^1, $m^0 \in X$, g^1, G^2:

$$\begin{aligned}(\forall s^1_{(\cdot)}, h^1 \in \Psi(m,g,G)(1))\big[(\forall n)(s_n \in X)\\ \to (\exists k^0 \leq G(s_{(\cdot)},h))(\forall n^0 \in [k,h(k)])(s_n < n)\big]\\ \to (\exists n \leq \Psi(m,g,G)(2))(\forall k \in [n,g(n)])(k \notin X).\end{aligned}$$

Definition 3.23 **($\mathsf{BD\text{-}N'_{meta}}$).** There is Ψ such that for all X^1, $m^0 \in X$, g^1:

$$\begin{aligned}(\forall s^1_{(\cdot)} \in \Psi(m,g)(1))\big[(\forall n)(s_n \in X) \to (\exists k^0)(\forall n \geq k)(s_n < n)\big].\\ \to (\exists n \leq \Psi(m,g)(2))(\forall k \in [n,g(n)])(k \notin X).\end{aligned}$$

Note that Ψ does not have access to the set X as an input, as suggested by the metastability trade-off. Furthermore, since $\mathsf{LPO} \to \mathsf{BD\text{-}N}$ in constructive mathematics (See [7]), $(\exists^2)$ yields a realiser for $\mathsf{BD\text{-}N}$. Nonetheless, we now show that Θ computes Ψ as in $\mathsf{BD\text{-}N_{meta}}$ via a term of Gödel's T, and vice versa. We shall use 'nonstandard $\mathsf{BD\text{-}N}$' as follows.

Definition 3.24 [$\mathsf{BD\text{-}N_{ns}}$]**.** For any set X^1 and any *standard* $m^0 \in X$:

$$(\forall^{st} s^1_{(\cdot)})\big[(\forall n)(s_n \in X) \to (\exists^{st} k)(\forall^{st} n \geq k)(s_n < n)\big] \to (\exists^{st} n)(\forall^{st} k \geq n)(k \notin X). \tag{3.19}$$

The absence of 'st' in '$(\forall n)(s_n \in X)$' in (3.19) (only) leads to a nicer normal form; the *standard* $m^0 \in X$ *is* however essential, as is clear from the following proof. Let $\mathsf{BD\text{-}N'_{ns}}$ be $\mathsf{BD\text{-}N_{ns}}$ with $(\exists k)(\forall n \geq k)(s_n < n)$ in the antecedent (which follows from $\mathsf{BD\text{-}N_{ns}}$ via $\Pi^0_1\text{-}\mathsf{TRANS}$).

Theorem 3.25. *The system* $\mathsf{P_0}$ *proves* $\mathsf{STP} \leftrightarrow$ *$\mathsf{BD\text{-}N_{ns}}$; $\mathsf{P_0} + \mathsf{TRANS}$ proves* $\mathsf{STP} \leftrightarrow \mathsf{BD\text{-}N'_{ns}}$.

Proof. For the first forward implication, fix X^1_0 and standard $m^0_0 \in X_0$ such that

$$(\forall^{st} s^1_{(\cdot)})\big[(\forall n)(s_n \in X_0) \to (\exists^{st} n)(\forall^{st} k \geq n)(s_k < k)\big] \tag{3.20}$$

Now fix standard $t^1_{(\cdot)}$ such that $(\forall^{st} n)(t_n \in X_0)$ and suppose $(\forall^{st} n)(\exists^{st} k \geq n)(t_k \geq k)$. Applying $\mathsf{HAC_{int}}$, there is standard f^1 such that $(\forall^{st} n)(f(n) \geq n \wedge t_{f(n)} \geq f(n))$. Now define the standard sequence $s_{(\cdot)}$ by $s_n := t_{f(n)}$ if $t_{f(n)} \geq n$, and m_0 otherwise. Clearly, $s_{(\cdot)}$ satisfies the antecedent of (3.20), and hence there is some standard n_0 such that $(\forall^{st} k \geq n_0)(s_k < k)$. By definition, we have $(\forall^{st} k \geq n_0)(k > s_k = t_{f(k)} \geq k)$, a contradiction. Thus, we have:

$$(\forall^{st} t^1_{(\cdot)})\big[(\forall^{st} n)(t_n \in X_0) \to (\exists^{st} n)(\forall^{st} k \geq n)(t_k < k)\big] \tag{3.21}$$

follows from (3.20), i.e. we may indeed drop the 'st' in the antecedent of $\mathsf{BD\text{-}N_{ns}}$, as claimed right after Definition 3.24. In light of the connection between (3.20) and (3.21), $\mathsf{P_0} + \mathsf{STP}$ proves $[\mathsf{BD\text{-}N}]^{st} \to \mathsf{BD\text{-}N_{ns}}$ as X only occurs as 'call by (standard) value'. To establish $[\mathsf{BD\text{-}N}]^{st}$ in $\mathsf{P_0}$, suppose X is standard and satisfies $(\forall^{st} n)(\exists^{st} k \geq n)(k \in X)$. Applying $\mathsf{HAC_{int}}$, there is standard

$\Phi^{0\to 0^*}$ such that $(\forall^{st} n)(\exists k \in \Phi(n))(k \geq n \wedge k \in X)$. Since X is actually a *standard* binary sequence, one readily uses Φ to define *standard* g^1 such that $(\forall^{st} n)(g(n) \geq n \wedge g(n) \in X)$. Clearly, X is not pseudo-bounded *relative to* 'st', due to the sequence g, and $[\mathsf{BD\text{-}N}]^{st}$ follows.

For the first reverse implication, we establish $\mathsf{BD\text{-}N_{ns}} \to$ (2.9) in P_0, and the theorem then follows from Theorem 2.8. Working in $\mathsf{P}_0 + \mathsf{BD\text{-}N_{ns}}$, fix $T \leq_1 1$ such that $(\forall^{st} n)(\exists \beta^0)(|\beta| = n \wedge \beta \in T)$. Applying overspill ([3, Proposition 3.3]) to the latter formula yields $\beta_0^{0^*}$ such that $\neg st(|\beta_0|)$ and $\beta_0 \in T$. Now define the set $X := \{\overline{\beta_0}1, \overline{\beta_0}2, \overline{\beta_0}3, \dots\} \subseteq \mathbb{N}$ using a (standard) sequence coding function π. Note that for *standard* n, the sequence $\overline{\beta_0}n$ is standard as it has standard length n and standard inputs (See [3, Corollary 2.19]). Hence, X satisfies $(\forall^{st} n)(\exists^{st} k \geq n)(k \in X)$ and by $\mathsf{BD\text{-}N_{ns}}$ there is a standard sequence $s^1_{(\cdot)}$ of elements of X such that $(\forall^{st} n)(\exists^{st} k \geq n)(s_k \geq k)$. Applying $\mathsf{HAC_{int}}$ to the latter, there is standard f^1 such that $(\forall^{st} n)(s_{f(n)} \geq f(n) \wedge f(n) \geq n)$. Let the (standard) function h^1 be such that $(\forall x^{0^*} \leq 1, n^0)(|x| < n \to \pi(x) < h(n))$. By definition, $s_{f(h(n))}$ codes a subsequence of β_0 of at least length n due to $s_{f(h(n))} \geq h(n)$ (for standard n). Modulo the (standard) decoding function associated to π, the *standard* function $\lambda n.s_{f(h(n))}$ thus defines a *standard* path of T, i.e. $(\forall^{st} n^0)(s_{f(h(n))} \in T)$. Hence, we obtain (2.9). The second equivalence is now immediate as $\Pi^0_1\text{−}\mathsf{TRANS}$ allows us to drop the second and third 'st' in the antecedent of (3.19). Indeed, the formula $(\exists^{st} k)(\forall^{st} n \geq k)(s_n < n)$ does not involve X (which is nonstandard), while the only other parameter is the *standard* $s_{(\cdot)}$. □

The proof of the first forward implication contains a subtlety: To define the *standard* g^1 from Φ, it is essential that X be *standard*. Thus, P_0 proves $\mathsf{BD\text{-}N}^{st}$, but to conclude $\mathsf{BD\text{-}N_{ns}}$, STP is essential, as is also clear from the theorem.

Corollary 3.26. *The functional Ψ from* $\mathsf{BD\text{-}N_{meta}}$ *computes Θ via a term of Gödel's T, and vice versa, provable in* RCA_0^ω.

Proof. Proceed as for Corollary 3.5 for $\mathsf{STP} \leftrightarrow \mathsf{BD\text{-}N_{ns}}$. A normal form for $\mathsf{BD\text{-}N_{ns}}$ is obtained using the 'metastability trick' of introducing $[M, F(M)]$ as in (2.2) on both the consequent and antecedent, and bringing outside the standard quantifiers (using *Idealisation* I). □

Applying the ECF-translation (See [13, Sect. 2] for details) to the previous proof, we observe that $\mathsf{WKL} \leftrightarrow [\mathsf{BD\text{-}N_{meta}}]_{\mathsf{ECF}}$, i.e. weak König's lemma turns out to be equivalent to computing a metastable bound for $\mathsf{BD\text{-}N}$ from continuous (as in Reverse Mathematics) realisers for pseudo-boundedness. This result should be compared to [18, Example 2.6]. Finally, we need Feferman's 'search functional' μ^2 which is defined by the following specification:

$$(\forall f^1)\big[(\exists n^0)(f(n) = 0) \to f(\mu(f)) = 0\big]. \qquad (\mathsf{MU}(\mu))$$

Corollary 3.27. *The functional Ψ from* $\mathsf{BD\text{-}N'_{meta}}$ *computes Θ via a term of Gödel's T with input Feferman's μ^2, and vice versa, provable in* $\mathsf{RCA}_0^\omega + (\mu^2)$.

Proof. Proceed as for Corollary 3.5 for $[\mathsf{STP} + \Pi_1^0\text{−}\mathsf{TRANS}] \rightarrow \mathsf{BD\text{-}N}'_{\mathsf{ns}}$ and $[\Pi_1^0\text{−}\mathsf{TRANS} + \mathsf{BD\text{-}N}'_{\mathsf{ns}}] \rightarrow \mathsf{STP}$. A normal form for $\mathsf{BD\text{-}N}'_{\mathsf{ns}}$ is readily obtained using the 'metastability trick' of introducing $[M, F(M)]$ as in (2.2) on the consequent (only); $\Pi_1^0\text{−}\mathsf{TRANS}$ is equivalent to:

$$(\forall^{st} f^1)(\exists^{st} n^0)\big[(\exists m^0)(f(m) = 0) \rightarrow f(n) = 0\big], \tag{3.22}$$

which is a normal form, and explains the presence of Feferman's μ^2. □

As an aside, similar to $\mathsf{IVT}_{\mathsf{meta}}$, the 'hardness' of $\mathsf{BD\text{-}N}_{\mathsf{meta}}$ is **not** due to the use of high types. There is a *second-order* version of $\mathsf{BD\text{-}N}_{\mathsf{meta}}$ which cannot be proved in $\mathsf{RCA}_0^\omega + \mathsf{QF\text{-}AC} + (S^2)$, a Π_3^1-conservative extension of $\Pi_1^1\text{-}\mathsf{CA}_0$. This result is however beyond the scope of this paper.

3.4 Computing the Special Fan Functional

In this section, we provide a short and elegant derivation of Θ in second-order arithmetic. This significantly improves the associated result from [19]. We need the following instance of *Tranfer*:

$$(\forall^{st} Y^2)\big[(\exists f^1)(Y(f) = 0) \rightarrow (\exists^{st} f^1)(Y(f) = 0)\big], \tag{SOT}$$

and recall HBU introduced in Sect. 2.3, which follows from *Cousin's lemma* ([5]); the latter is essentially the Heine-Borel theorem *in the general case*, i.e. the statement that any (possibly uncountable) open cover of I has a finite sub-cover. We have the following theorem.

Theorem 3.28. *The system* $\mathsf{P}_0 + \mathsf{HBU}$ *proves* $\mathsf{SOT} \rightarrow \mathsf{STP}$.

Proof. First of all, SOT implies $\Pi_1^0\text{−}\mathsf{TRANS}$ and hence (3.22). Apply $\mathsf{HAC}_{\mathsf{int}}$ to the latter, bearing in mind Remark 2.7, and obtain standard μ^2 such that

$$(\forall^{st} f^1)(\exists n^0 \leq \mu(f))\big[(\exists m^0)(f(m) = 0) \rightarrow f(n) = 0\big]. \tag{3.23}$$

Applying SOT to (3.23) yields $(\exists^{st}\mu^2)\mathsf{MU}(\mu)$. Now, HBU trivially implies:

$$(\forall \Psi^2)(\exists w^{1^*})\underline{(\forall q^0 \in I)(\exists y \in w)(|q - y| < \tfrac{1}{\Psi(y)+1})}. \tag{3.24}$$

Thanks to $(\exists^{st}\mu^2)\mathsf{MU}(\mu)$, the underlined formula in (3.24) is equivalent to a formula $Y_0(\Psi, w) = 0$ for any Ψ, w, and where Y_0^3 is standard. For standard Ψ^2, apply SOT to $(\exists w^{1^*})Y_0(\Psi, w) = 0$, yielding:

$$(\forall^{st}\Psi^2)(\exists^{st} w^{1^*})(\forall q^0 \in I)(\exists y \in w)(|q - y| < \tfrac{1}{\Psi(y)+1}), \tag{3.25}$$

and the latter readily yields STP. Indeed, (3.25) trivially implies:

$$(\forall^{st}\Psi^2)(\forall q^0 \in I)(\exists^{st} y)(|q - y| < \tfrac{1}{\Psi(y)+1}), \tag{3.26}$$

as a standard sequence w^{1^*} only has standard elements (and standard length) by Definition 2.3. But (3.26) also trivially yields (by switching quantifiers):

$$(\forall q^0 \in I)\underline{(\forall^{st}\Psi^2)(\exists^{st} y \in I)(|q-y| < \tfrac{1}{\Psi(y)+1})}, \tag{3.27}$$

and the underlined formula in (3.27) is equivalent (using the usual 'metastability' trick and $\mathsf{HAC_{int}}$) to $(\exists^{st} y \in I)(\forall^{st} k^0)(|q-y| < \frac{1}{k})$. Now, for nonstandard N^0 and $x^1 \in I$, we have $x \approx [x](N)$ where $[x](N)$ is the N-th approximation of x. Hence, we have obtained $(\forall x^1 \in I)(\exists^{st} y \in I)(x \approx y)$, which is equivalent to STP by Theorem 2.8. □

Consider the following specification, similar to $\mathsf{SO}(\xi)$ from Sect. 2.3, as follows:

$$(\forall Y^2)\big[(\exists f^1)(Y(f)=0) \to Y(\xi(Y))=0\big]. \tag{$\mathsf{SOC}(\xi)$}$$

Corollary 3.29. *There is t in Gödel's T such that $\mathsf{RCA}_0^\omega + \mathsf{HBU}$ proves that* $(\forall \xi)\big[\mathsf{SOC}(\xi) \to \mathsf{SCF}(t(\xi))\big]$.

Proof. Apply term extraction as in Theorem 2.5 to the proof of the theorem. □

The previous corollary significantly improves the associated result from [19, Sect. 3]:

1. The base theory in Corollary 3.29 is conservative[9] over WKL_0, which is weak compared to the results in [19, Sect. 3].
2. The extracted term in Corollary 3.29 seems simpler than the one in [19, Sect. 3].

Finally, the proof of the theorem can be adapted to show that HBU does not follow from the existence of the Suslin functional (akin to the results claimed for (3.8) and $\mathsf{BD\text{-}N_{meta}}$). However, we are running out of space *and* we would require the (unpublished) results from [20].

References

1. Avigad, J., Dean, E.T., Rute, J.: A metastable dominated convergence theorem. J. Log. Anal. **4**, 1–19 (2012)
2. Beeson, M.J.: Foundations of Constructive Mathematics: Metamathematical Studies. Ergebnisse der Mathematik und ihrer Grenzgebiete, vol. 6. Springer, Heidelberg (1985)
3. van den Berg, B., Briseid, E., Safarik, P.: A functional interpretation for nonstandard arithmetic. Ann. Pure Appl. Logic **163**, 1962–1994 (2012)
4. Bishop, E., Bridges, D.S.: Constructive Analysis, Grundlehren der Mathematischen Wissenschaften, vol. 279. Springer, Berlin (1985)

[9] It is an easy exercise to show that (3.24) is provable in $\mathsf{RCA}_0^\omega + \mathsf{MUC}$, where the latter states the existence of the intuitionistic fan functional. The latter system is conservative over WKL_0 (See [13, Sect. 3]).

5. Cousin, P.: Sur les fonctions de n variables complexes. Acta Math. **19**(1), 1–61 (1895)
6. Hurd, A.E., Loeb, P.A.: An Introduction to Nonstandard Real Analysis, Pure and Applied Mathematics, vol. 118. Academic Press Inc., Orlando (1985)
7. Ishihara, H.: Reverse mathematics in Bishop's constructive mathematics. Philosophia Scientiae (Cahier Spécial) **6**, 43–59 (2006)
8. Ishihara, H.: Continuity properties in constructive mathematics. JSL **57**, 557–565 (1992)
9. Ishihara, H.: Continuity and nondiscontinuity in constructive mathematics. JSL **56**, 1349–1354 (1991)
10. Keisler, H.J.: Nonstandard arithmetic and reverse mathematics. Bull. Symb. Log. **12**, 100–125 (2006)
11. Kohlenbach, U., Koutsoukou-Argyraki, A.: Rates of convergence and metastability for abstract Cauchy problems generated by accretive operators. J. Math. Anal. Appl. **423**(2), 1089–1112 (2015)
12. Kohlenbach, U.: Applied Proof Theory: Proof Interpretations and Their Use in Mathematics. Springer Monographs in Mathematics. Springer, Berlin (2008)
13. Kohlenbach, U.: Higher order reverse mathematics. In: Reverse mathematics 2001. Lecture Notes in Logistics, vol. 21, pp. 281–295. ASL (2005)
14. Kohlenbach, U.: Foundational and mathematical uses of higher types. Lecture Notes in Logistics, vol. 15, pp. 92–116. ASL (2002)
15. Lietz, P., Streicher, T.: Realizability models refuting Ishihara's bound- edness principle. Ann. Pure Appl. Log. **163**(12), 1803–1807 (2012)
16. Longley, J., Normann, D.: Higher-Order Computability. Theory and Applications of Computability. Springer, Heidelberg (2015)
17. Nelson, E.: Internal set theory: a new approach to nonstandard analysis. Bull. Am. Math. Soc. **83**(6), 1165–1198 (1977)
18. Normann, D.: The extensional realizability model of continuous functionals and three weakly non-constructive classical theorems. Log. Methods Comput. Sci. **11**, 1–27 (2015)
19. Normann, D., Sanders, S.: Nonstandard analysis, computability theory, and their connections (2017, submitted). https://arxiv.org/abs/1702.06556
20. Normann, D.: Nonstandard analysis, computability theory, and metastability (2017)
21. Sanders, S.: The Gandy-Hyland functional and a hitherto unknown computational aspect of Nonstandard Analysis. Computability, arXiv: http://arxiv.org/abs/1502.03622 (2015)
22. Sanders, S.: Grilliot's trick in Nonstandard Analysis. Logical Methods in Computer Science, Special Issue for CCC15 (2017)
23. Sanders, S.: The unreasonable effectiveness of Nonstandard Analysis (2015). http://arxiv.org/abs/1508.07434
24. Sanders, S.: To be or not to be constructive, Indagationes Mathematicae, p. 69, arXiv: https://arxiv.org/abs/1704.00462 2017
25. Sanders, S.: The refining of the taming of the Reverse Mathematics zoo. Notre Dame J. Formal Log. (2016). http://arxiv.org/abs/1602.02270
26. Stephen, G.: Simpson, Subsystems of Second Order Arithmetic. Perspectives in Logic, 2nd edn. CUP, Cambridge (2009)
27. Simpson, S.G., Yokoyama, K.: A nonstandard counterpart of WWKL. Notre Dame J. Form. Log. **52**(3), 229–243 (2011)
28. Tao, T.: Structure and randomness, pp. xii+298. American Mathematical Society, Providence, RI (2008)

The Completeness of BCD for an Operational Semantics

Rick Statman(✉)

Department of Mathematical Sciences, Carnegie Mellon University,
Pittsburgh, PA 15213, USA
statman@cs.cmu.edu

Keywords: Lambda calculus · Intersection types
Completeness theorem · Operational semantics

1 Introduction

The theorem of Coppo et al. ([3,6]) states that an untyped lambda term is strongly normalizable if and only if it provably has an intersection type. Here we consider which terms have which types.

We define an operational semantics for the collection of intersection types which assigns to every intersection type A a set of strongly normalizable terms [[A]]. We show that the theory of intersection types BCD (Barendregt, Coppo, Dezani) [2], pp. 582–583, proves X : A for an untyped term X if and only if $X : [[A]]$ for all interpretations of $[[,]]$ in the operational semantics. Here we shall use the notation ':' for both the formal statement that X has type A, and also set theoretic membership.

Our view of what operational semantics should be begins with Tait style proofs ([8]) of strong normalization. These proofs consider a complete lattice of sets S of strongly normalizable untyped terms ([2], 9.3). Not all such sets are considered but the lattice operations are union and intersection. We require that S is closed under reduction, and possibly some other conditions, such as head expansion with strong normalizable arguments, depending on the variant. The operation → is then introduced

$$S \rightarrow T = \{X | \text{for all } Y : S \Rightarrow (XY) : T\}.$$

This is certainly familiar from the theory of logical relations ([2], 3.3) for the simple typed case, positive recursive types, and our principal concern in this note; intersection types. Given an intersection type A, if the atoms (atomic types) of A are evaluated among the sets S then A has a value among the sets S. This will be the interpretation $[[A]]$.

2 Beth Models

SN is the set of strongly normalizable untyped terms. Here, we do not distinguish beta from beta-eta strong normalizability since they are equivalent. A Beth

S. Artemov and A. Nerode (Eds.): LFCS 2018, LNCS 10703, pp. 331–336, 2018.
https://doi.org/10.1007/978-3-319-72056-2_20

model consists of a pair (O, E) where O is a poset with partial order [, and E is a monotone map from O x Atoms into the subsets of SN closed under beta reduction and we shall assume that $E(p, a)$ is non empty except possibly when p is the [smallest element of O, should this exist.

For λ terms X we define the "forcing relation" $\models$ by

$p \models X : a$ if for all $q]p$ there exists $r]q$ *s.t.* $X : E(r, a)$
$p \models X : A/\backslash B$ if $p \models X : Ap \models X : B$
$p \models X : A \rightarrow B$ if whenever $q]p$ and $q \models U : A$ there exists $r]q$
such that $r \models (XU) : B$

and we assume that E satisfies the generalized monotonicity property if $[Y/x]X : E(p, a), q]p,$ and $q \models Y : A$ then there exists $r]q$ such that $(\backslash xXY) : E(r, a)$ where $[Y/x]$ is the the substitution operation (the term Y for the variable x).

Definition 1. An O chain (linearly ordered subset) W is generic if

(i) for any X and atom a there exists $p : W$ such that either $X : E(p, a)$ or there is no $q]p$ such that $q \models X : a$, and
(ii) for each $A \rightarrow B$ there exists $p : W$ such that either $p \models X : A \rightarrow B$ or there exists U and $q : W$ such that $q]p$ and $q \models U : A$ but there is no $r]q$ such that $r \models (XU) : B$. We could just as easily use directed subsets of O instead of chains but chains suffice.

For what follows the reader should consult the definition of BCD in [2] which appears on pages 582–583, but without the top element (U_{top}). When we wish to include the top element, U_{top}, together with its axiom ([2], p. 583), we will write $BCD{+}U_{top}$. Especially, the reader should look at the definitions of equality and the ordering of types on page 582. These are reproduced in the appendix.

Facts

1. if $p \models X : A$ and $q]p$ then $q \models X : A$
2. $p \models X : A$ iff for each $q]p$ there exists $r]q$ *s.t.* $r \models X : A$
3. if $p \models X : A$ and

$$A < B \text{ or } A = B \text{ in } BCD \text{ (page 582)}$$

then $p \models X : B$
4. if W is generic then for any X and atom a there exists $p : W$ such that for all $q]p$ we have $q \models X : a$ or there is no $q]p$ *s.t.* $q \models X : a$
5. if W is generic then for any X and $A/\backslash B$ there exists a $p : W$ such that $p \models X : A/\backslash B$ or there is no $q]p$ such that $q \models X : A/\backslash B$
6. if W is an O chain with a maximal element then there exists a generic O chain extending W.

Proposition 1. Let W be a generic O chain and set $R(A) = \{X|$ there exists $p : W$ such that $p \models X : A\}$. Let $X : SN$ then

(i) $X : R(a)$ iff there exists $p : W$ $s.t.$ $X : E(p, a)$
(ii) $X : R(A \to B)$ iff for each
$U : R(A)$ we have $(XU) :$ R (B)
(iii) $X : R(A/\backslash B)$ iff $X : R(A)$ & $X : R(B)$

Proof by induction on A. The basis case (i) is by definition. Induction step; Case (ii) $\Rightarrow$. Suppose that we have a $p : W$ such that $p \models X : A \Rightarrow B$ and we have $U : R(A)$. Thus there exists $q : W$ such that $q \models U : A$. By fact (1) we may assume that $q]p$. Now for any $r]q$ there exists $t]r$ such that $t \models (XU) : B$ but W is generic so there must be an $r : W$ such that $r \models (XU) : B$. That is $(XU) : R(B)$. $\Leftarrow$. Suppose that for each $U : R(A)$ we have $(XU) : R(B)$. Now if there is no $p : W$ such that $p \models X : A \to B$, since W is generic, there exists $p : W$ and aU such that $p \models U : A$ but there is no $q]p$ such that $q \models (XU) : B$. But by fact (1) this contradicts the hypothesis. Case (iii) similar to case (ii). End of proof.

The proposition clearly states that if the atoms a are evaluated $\{X|$ for some $p : W$ we have $X : E(p, a)\}$ then the value of the type A mentioned in the introduction is $R(A)$.

Example 1 (finite sets). In this case we let O be the collection of finite sets of SN terms closed under beta-eta reduction and ordered by inclusion. We set $E(p, a) = p$. Suppose that $A = A(1) \to (...(A(t) \to)...)$ and $\sim (p \models X : A)$. Then there exists $q]p$ and $Y(1), ..., Y(t)$ s, t $q \models Y(i) : A(i)$ for $i = 1, ..., t$ but there is no $r]q$ with $XY(1)...Y(t) : r$. But this can only be the case if $XY(1)...Y(t)$ is not SN. Thus we can find a generic W such that $X : R(A)$ or there exists $Y(1), ..., Y(t)$ $s.t.$ $Y(i) : R(A(i))$ for $i = 1, ..., t$ and $XY(1)...Y(t)$ is not SN.

Example 2. In this case we consider sets $S : O$ of closed beta-eta normal terms for which there exists an integer n such that $X : S$ iff every path in the Bohm tree of X has at most n lambdas and every node in the Bohm tree ([1], p. 212) of X has at most n descendants. Then for any partial recursive function f which is total on S and maps S to S there exists $M : R(S \to S)$ such that for any $N : S$ we have $MN = f(N)$ modulo beta-eta conversion.

Proposition 2. Suppose that O has a smallest element 0. Then,
$0 \models X : A$ iff for every generic W we have
$X : R(A)$.

Proof. Immediate by facts (1)–(6). End of proof.

We next consider the theory BCD with its provability relation $\vdash$ as described in [2] and reproduced in the appendix. A basis F is a map from a finite set of lambda calculus variables, $dom(F)$, to the set of types. Below we shall often conflate F with the finite set

$$\{x : F(x)|x : dom(F)\}.$$

Let O, E be as above and W generic.

Proposition 3 (soundness). Suppose that @ is a substitution and F is a base such that for all $x : dom(F), @(x) : R(F(x))$. Then if in BCD

$$F \vdash X : A$$

we have $@(X) : R(A)$

Now let O be the set of bases partially ordered by $F[G$ iff $dom(F)$ is contained in $dom(G)$ and for each $x : dom(F)$ we have $G(x)$ and $F(x)$ are equal types in BCD. Now define $E(a)$ by $X : E(F, a)$ if $FV(X)$ is contained in $dom(F)$ and $F \vdash X : a$. Clearly E is [monotone. In addition, $E(F, a)$ is closed under beta-eta reduction. However, generally $E(F, a)$ is not closed under beta head expansion for reasons similar to the case of BCD. In particular this happens when $(\backslash uUV)$ reduces to X, u is not free in U and there is an $x : FV(V)/\backslash FV(U)$ such that the basis entry $x : F(x)$ prevents V from having a BCD type. Thus we have to verify the generalized monotonicity property to insure soundness.First, we observe that there is no difference between E and $\models$ at atomic types.

Fact 7. $F \models X : a$ iff $X : E(F, a)$

Proof. If $FV(X)$ is contained in $dom(F)$ then the equivalence follows from the monotonicity of E and the weakening rule of BCD ([2], p. 585). Otherwise suppose that $F \models X : a$. For each $x : FV(X) - dom(F)$ add a new atom $a(x)$ and extend F to G by $G(x) = a(x)$. Then $G \models X : a$ so by the previous argument $G \vdash X : a$ in BCD. But we may substitute U_{top} for each $a(x)$ and $(\backslash x.xx)(\backslash x.xx)$ for each x. So in $BCD + U_{top}$ we have

$$F \vdash [..., (\backslash x.xx)(\backslash x.xx)/x, ...]X : a$$

and this contradicts the fact that if a term has a BCD type (U_{top} free) in $BCD + U_{top}$ then it is strongly normalizable (Theorem 17.2.15 (i) [2]).

Lemma 1. Suppose that $FV(X)$ is contained in $dom(F)$.

$$F \models X : A \text{ iff } F \vdash X : A \text{ in } BCD$$

Proof this is proved by induction on A. The basis case is by fact 7. For the induction step the case $A = B/\backslash C$ is obvious. We consider the case $A = B \to C$. $\Rightarrow$. Suppose that $F \models X : B \to C$. Let z be a new variable and extend F by G by with $G(z) = B$. Since $G \models z : B$ by induction hypothesis $G \models z : B$ thus there exists $H]G$ such that $H \models Xz : C$. Again by induction hypothesis $H \vdash Xz : C$ in BCD. Reasoning in BCD, $H - \{z : B\} \vdash \backslash z(Xz) : B \to C$. Now by hypothesis $FV(X)$ is contained in $dom(F)$, so by weakening, $F \vdash \backslash z(Xz) : B \to C$. Hence by subject reduction for eta ([2], p. 621)

$$F \vdash X : B \to C.$$

Conversely, suppose that $F \vdash X : A$. Let $G]F$ and $G \models U : B$. By induction hypothesis $G \vdash U : B$ in BCD Thus by induction hypothesis $G \models (XU) : C$. Hence

$$F \models X : B \to C.$$

End of proof.

Corollary 1. Generalized monotonicity holds and we have a Beth model.
From the lemma we get the completeness theorem.

Theorem. Let M be closed. Then $BCD \vdash M : A$ iff for every Beth model (O, E) and generic $W, M : R(A)$.

Proof. $\Rightarrow$. This is the soundness proposition. $\Leftarrow$. Consider the Beth model defined by the conditions above. By Proposition 2 $0 \models M : A$. Hence by the lemma $\vdash M : A$ in BCD. End of proof.

Appendix

(1) terms and types
variables $x, y, z, ...$ are terms
if X and Y are terms then so are (XY) and $\backslash xX$
atoms $a, b, c, ...$ are types
if A and B are types then so are $A/\backslash B$ and $A \to B$

(2) (quasi) order on types
A less than or equal A
$A/\backslash B$ less than or equal A
$A/\backslash B$ less than or equal B
$(A \to B)/\backslash(A \to C)$ less than or equal $A \to (B/\backslash C)$
if C less than or equal A and C less than or equal B
then C less than or equal $A/\backslash B$
if C less than or equal B and B less than or equal A
then C less than or equal A
if A less than or equal C and D less than or equal B
then $C \to D$ less than or equal $A \to B$
A equals B if A less than or equal B and
B less than or equal A

(3) axioms and rules of BCD
$F \vdash x : A$ if $(x : A)$ belongs to F
if $F, x : A \vdash X : B$ then $F \vdash \backslash xX : A \to B$
if $F \vdash X : A \to B$ and $F \vdash Y : A$ then $F \vdash (XY) : B$
if $F \vdash X : A$ and $F \vdash X : B$ then $F \vdash X : A/\backslash B$
if $F \vdash X : A$ and A less then or equal B in BCD
then $F \vdash X : B$

References

1. Barendregt, H.P.: The Lambda Calculus. North Holland, New York (1981)
2. Barendregt, H., Dekkers, W., Statman, R.: Lambda Calculus with Types. Cambridge University Press, New York (2013)
3. Coppo, M., Dezani, M.: A new type assignment for lambda terms. Archiv fur Math. Logik **19**, 139–156 (1978)
4. van Dalen, D.: Intuitionistic logic. In: Gobble, L. (ed.) The Blackwell Guide to Philosophical Logic. Blackwell, Oxford (2001)
5. Plotkin, G.D.: Lambda definability in the full type hierarchy. In: Hindley, J.R., Seldin, J.P. (eds.) To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism, pp. 363–373. Academic Press, London (1980)
6. Pottinger, G.: A type assignment for the strongly normalizable lambda terms. In: Hindley, J., Seldin, J. (eds.) To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism, pp. 561–577. Academic Press, London (1980)
7. Statman, R.: Logical relations and the typed lambda calculus. Inf. Contr. **165**(2/3), 85–97 (1985)
8. Tait, W.W.: Constructive reasoning. In: Van Rootselaar, B., Staal, J.F. (eds.) Studies in Logic and the Foundations of Mathematics, vol. 52, pp. 185–199. Elsevier, NorthHolland (1968)

A Tableau System for Instantial Neighborhood Logic

Junhua Yu(✉)

Department of Philosophy, Tsinghua University, Beijing 100084, China
junhua.yu.5036@outlook.com

Abstract. Extending classical propositional logic, instantial neighborhood logic (INL) employs formulas like $\Box(\alpha_1, ..., \alpha_j; \alpha_0)$. The intended meaning of such a formula is: there is a neighborhood (of the current point) in which α_0 universally holds and none of $\alpha_1, ..., \alpha_j$ universally fails. This paper offers to INL a tableau system that supports mechanical proof/counter-model search.

Keywords: Neighborhood logic · Tableau

1 Instantial Neighborhood Logic

Instantial neighborhood logic (INL) [14] is a recent development in neighborhood logic [4,12], a field dedicated to describe neighborhood models using methods of logic. In the most general setting, *neighborhood models* are defined as triples of the form (W, N, V), where W is a non-empty set of points, $N :: W \mapsto \mathcal{P}(\mathcal{P}(W))$ is a *neighborhood function* that maps each point to a set of point-sets (called *neighborhoods*), and V is a propositional valuation. The notion of neighborhood model is related to both relational model (also known as Kripke model) [3,4] and topological model [2], as they each can be seen as a neighborhood model with additional properties.

Given a model $\mathfrak{M} = (W, N, V)$ and $u \in W$, the *generated sub-model* of $\mathfrak{M}$ by u is $\mathfrak{M}_u := (W_u, N \restriction W_u, V \restriction W_u)$ where W_u is the collection of all points in W that can be reached from u via a chain of $N\!-\!\ni$ steps.[1] As expected, at point u models $\mathfrak{M}$ and $\mathfrak{M}_u$ agree on all INL-formulas [14].

Most of existing works on neighborhood logic use modal language, that is, propositional language extended by unary operator $\Box$. There are two popular ways to interpret a formula like $\Box\alpha$: (A) the current point has a neighborhood in which α universally holds; and (B) the set of all points satisfying α is a neighborhood of the current point. It is obvious that (B) implies (A), and additional

J. Yu—Supported by Tsinghua University Initiative Scientific Research Program 20151080426.

[1] By $v_1(N\!-\!\ni)v_2$ we mean that $v_2 \in S \in N(v_1)$ for some $S \subseteq W$, i.e., v_2 is a point in a neighborhood of v_1.

S. Artemov and A. Nerode (Eds.): LFCS 2018, LNCS 10703, pp. 337–353, 2018.
https://doi.org/10.1007/978-3-319-72056-2_21

conditions like monotonicity[2] are required for the other direction to hold. These two interpretations are equally good, but only on (A) INL naturally extends. In this paper, the neighborhood logic that uses modal language with (A) is called *basic neighborhood logic* (BNL).

Language of INL is classical propositional language enriched by a two-sorted operator (also written as $\Box$). For each natural number j, if $\alpha_0, \alpha_1, ..., \alpha_j$ are all formulas, then so is $\Box(\alpha_1, ..., \alpha_j; \alpha_0)$. In a neighborhood model, $\Box(\alpha_1, ..., \alpha_j; \alpha_0)$ means "the current point has a neighborhood in which α_0 universally holds, and none of $\alpha_1, ..., \alpha_j$ universally fails". That is to say, in order to justify $\Box(\alpha_1, ..., \alpha_j; \alpha_0)$, a neighborhood should not only support α_0 everywhere, but also include 'instantial' points that respectively support $\alpha_1, ..., \alpha_j$. In a $\Box$-prefixed formula like $\Box(\alpha_1, ..., \alpha_j; \alpha_0)$, sub-formulas $\alpha_1, ..., \alpha_j$ are called *instances*. By a *literal*, we mean $\top$, $\neg\top$, $\bot$, $\neg\bot$, a propositional atom or its negation. The size of finite set S is denoted by $|S|$.

As shown in [14], INL has a strictly stronger expressive power on neighborhood models than BNL, and hence its axiomatization cannot be achieved using reduction axioms. In [14], a Hilbert-style axiomatization of INL is provided, and we denote it by Hinl.

Definition 1 (Calculus Hinl). As a Hilbert-style axiomatization, Hinl has schematic axioms:

(Prop) Propositional tautologies,
(R-Mon) $\Box(\alpha_1, ..., \alpha_j; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j; \alpha_0 \vee \beta)$,
(L-Mon) $\Box(\alpha_1, ..., \alpha_{j-1}, \alpha_j; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_{j-1}, \alpha_j \vee \beta; \alpha_0)$,
(Inst) $\Box(\alpha_1, ..., \alpha_{j-1}, \alpha_j; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_{j-1}, \alpha_j \wedge \alpha_0; \alpha_0)$,
(Norm) $\neg\Box(\bot; \alpha_0)$,
(Case) $\Box(\alpha_1, ..., \alpha_j; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j, \delta; \alpha_0) \vee \Box(\alpha_1, ..., \alpha_j; \alpha_0 \wedge \neg\delta)$,
(Weak) $\Box(\alpha_1, ..., \alpha_{i-1}, \alpha_i, \alpha_{i+1}, ..., \alpha_j; \alpha_0)$
$\rightarrow \Box(\alpha_1, ..., \alpha_{i-1}, \alpha_{i+1}, ..., \alpha_j; \alpha_0)$,
(Dupl) $\Box(\alpha_1, ..., \alpha_j; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_i, \beta, \alpha_{i+1}, ..., \alpha_j; \alpha_0)$
where $\beta \in \{\alpha_1, ..., \alpha_j\}$;

and rules:

$$\frac{\alpha \rightarrow \beta \quad \alpha}{\beta}\text{(MP)},$$

$$\frac{\alpha \rightarrow \beta \quad \beta \rightarrow \alpha \quad \phi^q_\alpha}{\phi^q_\beta}\text{(RE)},$$

where ϕ^q_ψ is the result of uniformly substituting all occurrences of propositional atom q in ϕ by ψ.

As usual, $\vdash_{\mathsf{Hinl}} \phi$ means that ϕ is provable in Hinl.

Most axioms in Hinl are straight-forward, and we explain (Case) here a bit. Once a neighborhood justifying $\Box(\alpha_1, ..., \alpha_j; \alpha_0)$ is given, we can check it with

[2] That is, if $S_1 \in N(w)$ and $S_1 \subseteq S_2$, then $S_2 \in N(w)$.

a formula δ, to see whether or not that neighborhood accommodates δ somewhere. If it does, then we safely take δ as an extra instance, and have disjunct $\Box(\alpha_1, ..., \alpha_j, \delta; \alpha_0)$; otherwise, δ universally fails and hence we have disjunct $\Box(\alpha_1, ..., \alpha_j; \alpha_0 \wedge \neg\delta)$.

Shown in [14], Hinl is sound and complete for INL:

Theorem 1 (Sound-and-completeness of Hinl). *For every* INL*-formula ϕ, it holds that:* $\vdash_{\mathsf{Hinl}} \phi$ *iff* ϕ *is valid.*

This paper offers to INL a tableau system called Tinl. In the following Sect. 2, we recall a tableau system for classical propositional logic, and define Tinl as an extension of it. In Sect. 3, we show that proof search in Tinl always terminates. Then in Sects. 4 and 5 soundness and completeness of Tinl are respectively verified. The final Sect. 6 concludes this paper.

In the author's other paper in a parallel process [15], a sequent calculus G3inl for INL is introduced, and based on it the Lyndon interpolation theorem of INL is constructively proved. Actually, the discovery of G3inl was inspired by Tinl, and a hyper-sequent version of the former amounts to a dual of the latter. Although it is possible to establish the equivalence between these two, the author intentionally use them to explore different aspects of INL. As presented in [15], G3inl is purely proof-theoretical, and whose completeness is shown via a syntactical cut-elimination. In this paper, we emphasize Tinl's close relation to semantics of INL, and completeness of Tinl will be shown via an extraction of counter-models out of failed attempts of Tinl-proofs.

2 Tableau System Tinl

Invented independently by Beth [1] and Hintikka [10] and later refined independently by Lis [11] and Smullyan [13], tableau is a popular framework of formal proofs [8]. Compared to other styles of formulation used in proof theory, the most distinguishing feature of tableau is that it closely connects to semantics, which earns it another well-known name: "semantic tableau". A good tableau system of a (decidable) logic should support proof/counter-model search in the sense that, once a formula ϕ is given, by mechanically trying rules in the tableau system, one can decide whether or not ϕ is valid, construct either a (tableau) proof (when ϕ is valid) or a counter-model of ϕ (when ϕ is not valid). This paper will offer to INL a tableau system Tinl that is good in the above sense.

Definition 2 (Basic tableau). A *basic tableau* is a tree of formulas that can be created from its root by consecutively applying specific tree extension rules in a way that, each node triggers at most one rule-application on each branch it belongs to.[3] A basic tableau is said to be *exhausted*, if each of its non-literal nodes triggers a rule-application on each branch it belongs to.

[3] In some tableau systems, like that for intuitionistic logic [7], it is sometimes necessary to trigger rule-applications multiple times at a same node. Since all systems we consider in this paper are free of such a need, we simply exclude it at this beginning definition.

For classical propositional logic, we have the following seven rules $(\neg\neg)$, $(\wedge)$, $(\neg\vee)$, $(\neg\rightarrow)$, $(\neg\wedge)$, $(\vee)$, and $(\rightarrow)$ (in the order displayed below) [9].

$$\begin{tikzcd}
\neg\neg\phi \arrow[d] & \phi\wedge\psi \arrow[d] & \neg(\phi\vee\psi) \arrow[d] & \neg(\phi\rightarrow\psi) \arrow[d] \\
\phi & \phi & \neg\phi & \phi \\
 & \psi & \neg\psi & \neg\psi
\end{tikzcd}$$

$$\begin{tikzcd}
 & \neg(\phi\wedge\psi) \arrow[dl] \arrow[dr] & & \phi\vee\psi \arrow[dl] \arrow[dr] & & \phi\rightarrow\psi \arrow[dl] \arrow[dr] & \\
\neg\phi & & \neg\psi \quad \phi & & \psi \quad \neg\phi & & \psi
\end{tikzcd}$$

These seven rules each offers an option to extend an existing branch in a tree. Among them, the four in the first line are *non-branching rules*. For instance, if an existing branch has node $\phi \wedge \psi$ (not necessarily at the end of that branch), then that node can *trigger* an application of $(\wedge)$ that extends the branch by adding two new nodes ϕ and ψ one after another to the end. To the contrary, the three in the second line are *branching rules*. For instance, if an existing branch has node $\phi \rightarrow \psi$ (not necessarily at the end of that branch), then that node can trigger an application of $(\rightarrow)$ that adds two distinct children $\neg\phi$ and ψ to the end of that branch, thereby extending it while splitting it into two.

We are ready to define the tableau system Tcpc for classical propositional logic.

Definition 3 (Tableau system Tcpc). The system Tcpc has all the seven tree extension rules presented above, and a Tcpc-*tableau* is a basic tableau as defined in Definition 2 w.r.t. these seven rules. A branch in a Tcpc-tableau is said to be *closed*, if it has node $\bot$, node $\neg\top$, or both node ϕ and node $\neg\phi$ for a formula ϕ. A Tcpc-tableau is said to be *closed*, if all its branches are closed. A branch (resp. Tcpc-tableau) is *open* if it is not closed. A *proof* of formula ϕ in Tcpc is a closed Tcpc-tableau with root $\neg\phi$.

By $\vdash_{\mathsf{Tcpc}} \phi$, we mean that formula ϕ has a Tcpc-proof. It is well-known that Tcpc is sound and complete for classical propositional logic [9].

Theorem 2 (Soundness and completeness of Tcpc). *For every propositional formula ϕ, it holds that: ϕ is a tautology iff $\vdash_{\mathsf{Tcpc}} \phi$.*

It is obvious that, in these seven Tcpc-rules, each formula introduced is strictly shorter compared to the formula by which the rule-application is triggered (actually, only its proper sub-formulas and their negations are permitted). A rule with such a property is said to be *analytic*, and a tableau system is said to be *analytic* if all of its rules are. Since all rules of Tcpc are finitely-branching (i.e., can introduce only finitely many children) and analytic, as well as the fact that no literals can trigger any rule-application, we see that all Tcpc-tableaux are finite in size. This facilitates a straight-forward proof/counter-model search in Tcpc. Starting from the initial root and consecutively extending the Tcpc-tableau by any rule that applies, one finally ends by getting either a closed Tcpc-tableau (a proof) or an exhausted open Tcpc-tableau. If the latter happens, then in the Tcpc-tableau there must be an open branch, and a propositional valuation that satisfies exactly propositional atoms in that branch is a counter-model of the root. Details of what sketched above can be found in [5].

We now turn to a tableau system for INL called Tinl. Objects of Tinl are neighborhood tableaux instead of basic tableaux.

Definition 4 (Neighborhood tableau). In a tree of formulas and finite formula multi-sets:

- a node is *regular* if it is a formula, and is *irregular* if it is a formula multi-set;
- a regular node that is not a child of any regular node is called an *initial node*, and a regular node that has no regular child is called an *end node*;
- a *branch* is a path from an initial node to an end node that passes through *only* regular nodes;
- a regular node that is neither a literal nor a $\neg\Box$-prefixed formula is said to be *active*.

A *neighborhood tableau* is a tree of formulas and finite formula multi-sets in which:

- the root is regular;
- children of nodes are created by applying specific tree extension rules;
- each node triggers at most one rule-application on each branch it belongs to.

A branch is *propositionally exhausted*, if each of its active nodes that are not $\Box$-prefixed (i.e., nodes with $\wedge$, $\neg\wedge$, $\vee$, $\neg\vee$, $\rightarrow$, or $\neg\rightarrow$ as main connective(s)) triggers a rule-application on it. A branch is *fully exhausted*, if each of its active nodes triggers a rule-application on it. A neighborhood tableau is said to be *exhausted*, if all branches in it are fully exhausted, and each of its non-empty irregular nodes triggers an application of rule (MS).

For INL, we employ not only propositional rules but also two extra rules.

Definition 5 (Rule (MS)). Rule (MS) (where "MS" stands for "multi-set")

$$\begin{tikzcd}
 & & \{\phi_1, \phi_2, ..., \phi_m\} \arrow[dll] \arrow[dl] \arrow[dr] & \\
\phi_1 & \phi_2 & \cdots & \phi_m
\end{tikzcd}$$

can be triggered by any non-empty irregular node. It introduces exactly elements of the irregular node as its children, which are all regular.

Definition 6 (Rule $(\Box)$). Rule $(\Box)$[4] is a branching rule that introduces irregular children to the end node of a propositionally exhausted branch, where the

[4] While "$\Box$" is an operator, "$(\Box)$" is the name of a rule.

number of these children depends on $\neg\Box$-prefixed formulas in that branch. The rule has the following form:

$$\frac{\begin{array}{c}\Box(\alpha_1, ..., \alpha_j; \alpha_0)\\ \neg\Box(\beta^1_1, ..., \beta^1_{j_1}; \beta^1_0)\\ \vdots \\ \neg\Box(\beta^k_1, ..., \beta^k_{j_k}; \beta^k_0)\end{array}}{\text{irregular child } \{\alpha_0 \wedge \sigma \wedge (\bigwedge^{I(i)\neq 0}_{i\in\{1,...,k\}} \neg\beta^i_{I(i)}) \mid \sigma \in A \cup B^I\} \text{ for each } I \in \prod^k_{l=1}\{0, 1, ..., j_l\}}$$

where $A = \{\alpha_1, ..., \alpha_j\}$ and $B^I = \{\neg\beta^i_0 \mid i \in \{1, ..., k\}, I(i) = 0\}$.

Certainly this rule needs explanations. Only if a propositionally exhausted branch has a $\Box$-prefixed node (formula) $\Box(\alpha_1, ..., \alpha_j; \alpha_0)$ with j-many instances, as well as *exactly* k-many $\neg\Box$-prefixed nodes (formulas) $\neg\Box(\beta^1_1, ..., \beta^1_{j_1}; \beta^1_0)$, ..., $\neg\Box(\beta^k_1, ..., \beta^k_{j_k}; \beta^k_0)$ that respectively have $j_1, ..., j_k$-many instances, then an application of rule ($\Box$) can be triggered by the node $\Box(\alpha_1, ..., \alpha_j; \alpha_0)$, and it creates a new *phase* (separated by the horizontal line) in which $\prod^k_{l=1}(j_l + 1)$-many irregular nodes are attached to the end of the existing branch as children. Each of these irregular nodes is indexed by an $I \in \prod^k_{l=1}\{0, 1, ..., j_l\}$ and has $|A \cup B^I|$-many elements that are identical to each other except for the conjunct σ. The following points deserve special attentions:

- We have emphasized that the existing branch should have exactly k-many $\neg\Box$-prefixed nodes. In other words, rule ($\Box$) always takes *all* $\neg\Box$-prefixed nodes on the branch as inputs. In the special case $k = 0$, only one irregular node indexed the by empty sequence is introduced.
- In the special case that $j = 0$ (hence $A = \varnothing$) and none of $j_1, ..., j_k$ is 0, there can be some $I \in \prod^k_{l=1}\{0, 1, ..., j_l\}$ s.t. $A \cup B^I = \varnothing$. Hence, the irregular node with index I is empty.[5]
- Although an application of rule ($\Box$) takes (k+1)-many nodes as inputs, only the $\Box$-prefixed one is said to *trigger* that application. Rule ($\Box$) always takes exactly one $\Box$-prefixed node as input, even though there may be more on the branch.
- Rule ($\Box$) is *destructive*, as all information on the existing branch cannot be referred later in the new phase created. While nodes taken as inputs jointly determine irregular nodes introduced, all other nodes simply expire (in the created phase), including nodes with $\Box$-prefixed or $\neg\Box$-prefixed sub-formulas like $\top \wedge \neg\Box(\phi; \psi)$. In order to avoid unnecessary loss of information, rule ($\Box$) is restricted to apply *only on propositionally exhausted branches.*

[5] Since rule (MS) can only be triggered by non-empty irregular nodes, the irregular node with index I has no child. It will not be hard to see from Definition 7 and Theorem 5 (both to be presented later) that empty irregular nodes are always open and indicate empty neighborhoods of the current point.

Having presented all rules that will be employed, we are now ready to define Tinl. In contrast to Tcpc in which we define openness and closure of branches, here in Tinl it is more convenient to talk about also that of nodes.

Definition 7 (Tableau system Tinl). System Tinl inherits all the seven rules of Tcpc (now in the language of INL), and also employs rules (MS) and $(\Box)$. A Tinl-*tableau* is a neighborhood tableau as defined in Definition 4 w.r.t. to all these nine rules. A branch in a Tinl-tableau is said to be *closed*, if it has (regular) node $\bot$, node $\neg\top$, or both node ϕ and node $\neg\phi$ for a formula ϕ. A node in a Tinl-tableau is said to be *closed*, if one of the following holds:

- it is a regular node with no child, and the unique branch it belongs to is closed;
- it is a regular node with regular children, and *all of* its children are closed;
- it is a regular node with irregular children, and there *exists* a phase in which *all* its irregular children are closed;
- it is an irregular node, and *one of* its children is closed.

A Tinl-tableau is said to be *closed*, if so is its root. A branch (resp. node, or Tinl-tableau) is *open* if it is not closed. A *proof* of formula ϕ in Tinl is a closed Tinl-tableau with root $\neg\phi$.

As usual, by $\vdash_{\mathsf{Tinl}} \phi$, we mean formula ϕ has a Tinl-proof.

We have some remarks here:

- according to rules of Tinl, a regular node may have only regular children or only irregular children, but not a mixture of both.
- rule (MS) is the only rule that an irregular node can trigger, and hence an irregular node can only have regular children (viz. its elements).
- by definition, an empty irregular node is always open.

Machineries in a Tinl-tableau have their semantic counterparts. Intuitively speaking, initial nodes correspond to points in the model, branches starting from initial nodes correspond to possibilities of points, irregular nodes correspond to neighborhoods in the model, and regular children of an irregular node correspond to points inside the neighborhood. These correspondences will be made more formal in a model construction employed later in the proof of Theorem 5.

Example 1. A Tinl-proof of $\Box(p \vee q; r) \to \Box(p; r) \vee \Box(q; r)$ is shown in Fig. 1. For the only application of rule $(\Box)$, in notations of Definition 6, we have $k = 2$ and $j = j_1 = j_2 = 1$, hence that application introduces four irregular nodes, each indexed by some $I \in \prod_{l=1}^{2}\{0, 1, ..., j_l\}$. These four irregular nodes are listed in the index-order of $(0, 0), (0, 1), (1, 0), (1, 1)$, and indexes also determine elements of these irregular nodes. Take $I = (1, 0)$ as an instance, since each of "A" and "$B^{(1,0)}$" in Definition 6 is a singleton, the irregular node with index $(1, 0)$ has two elements as shown. Among them the first is indexed by $p \vee q \in A$, and the second is indexed by $\neg r \in B^{(1,0)}$.

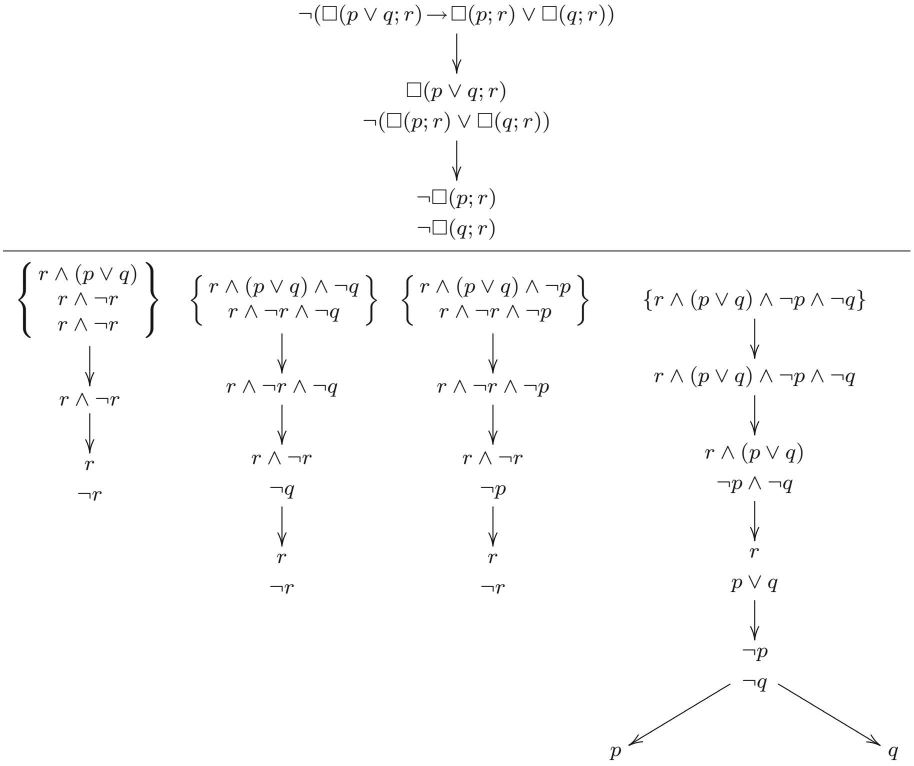

$\neg(\Box(p \vee q; r) \to \Box(p; r) \vee \Box(q; r))$
$\Box(p \vee q; r)$
$\neg(\Box(p; r) \vee \Box(q; r))$
$\neg\Box(p; r)$
$\neg\Box(q; r)$
$\left\{\begin{matrix} r \wedge (p \vee q) \\ r \wedge \neg r \\ r \wedge \neg r \end{matrix}\right\}$
$r \wedge \neg r$
r
$\neg r$
$\left\{\begin{matrix} r \wedge (p \vee q) \wedge \neg q \\ r \wedge \neg r \wedge \neg q \end{matrix}\right\}$
$r \wedge \neg r \wedge \neg q$
$r \wedge \neg r$
$\neg q$
r
$\neg r$
$\left\{\begin{matrix} r \wedge (p \vee q) \wedge \neg p \\ r \wedge \neg r \wedge \neg p \end{matrix}\right\}$
$r \wedge \neg r \wedge \neg p$
$r \wedge \neg r$
$\neg p$
r
$\neg r$
$\{r \wedge (p \vee q) \wedge \neg p \wedge \neg q\}$
$r \wedge (p \vee q) \wedge \neg p \wedge \neg q$
$r \wedge (p \vee q)$
$\neg p \wedge \neg q$
r
$p \vee q$
$\neg p$
$\neg q$
p
q

Fig. 1. A Tinl-proof of $\Box(p \vee q; r) \to \Box(p; r) \vee \Box(q; r)$.

In order for Tinl to be indeed a good tableau system for INL, its provability should be both decidable and correct. Here "decidable" means that, there is a mechanical method to determine whether or not any given formula is Tinl-provable, which amounts to termination of proof search in Tinl (Sect. 3); and "correct" means that, Tinl proves exactly all of valid INL-formulas, which amounts to its soundness (Sect. 4) and completeness (Sect. 5). Since Tinl is an extension of Tcpc, each of these three meta-theorems can be proved by extending a proof of the corresponding meta-theorem for Tcpc. As all these three are well-known to hold for Tcpc with proofs available [5], we omit details of propositional base and concentrate on extensions typical for Tinl.

3 Termination of Proof-Search in Tinl

In this section we will show that every proof-search in Tinl always terminates. What follows is a definition of degrees that takes only operator $\Box$ into account, which is a convenient simplification, since we will concentrate on only factors of Tinl that are absent in Tcpc.

Definition 8 (Degrees). The *degree of an* INL*-formula* ϕ, denoted by $dg(\phi)$, is the maximal number of nested $\Box$'s in ϕ, that is:

$$dg(\phi) := \begin{cases} 0 & \text{if } \Box \text{ does not occur in } \phi \\ dg(\phi_1) & \text{if } \phi \text{ is } \neg\phi_1 \\ \max\{dg(\phi_1), dg(\phi_2)\} & \text{if } \phi \text{ is } \phi_1 \vartriangle \phi_2 \text{ where } \vartriangle \in \{\wedge, \vee, \rightarrow\} \\ 1+\max\{dg(\phi_i) \mid i \in \{0, 1, ..., j\}\} & \text{if } \phi \text{ is } \Box(\phi_1, ..., \phi_j; \phi_0) \end{cases}$$

Since regular nodes in Tinl-tableaus are merely formulas, what above also defines *degree of regular nodes.* The *degree of a branch* is the maximal degree enjoyed by nodes on the branch.

Theorem 3 (Termination theorem). *There is a mechanical procedure of proof-search in* Tinl *that terminates on every input.*

Proof. The mechanical procedure can be described by the following instruction:

- **if** the tableau is closed,
  then exit with `success`.
- **else if** there is a regular node that can, but has not, triggered a propositional rule-application on a branch,
  then perform that application, and start over again;
- **else if** there is a regular node that can, but has not, triggered an application of rule ($\Box$) on a branch,
  then perform that application, immediately apply rule (MS) on each irregular node introduced that is non-empty, and then start over again;
- **else** exit with `failure`.

In each round, the procedure distinguishes four cases, among them two lead to exits, and two lead to start-overs. The procedure is designed so that applications of propositional rules have priority over that of rule ($\Box$). It is known (like in Tcpc) that, in finitely many rounds since any snapshot, the procedure will either exit or make all branches propositionally exhausted. Hence, the procedure cannot loop forever through only the first **else if**.

If the second **else if** is called, then a $\Box$-prefixed node on a propositionally exhausted branch is about to trigger an application of rule ($\Box$). By the design of the procedure, a same application will not happen again on this branch. Introduced by this application are finitely many irregular children, each of which has finitely many regular children created by subsequent applications of rule (MS). In total, only finitely many regular nodes are introduced, and these regular nodes are themselves new branches that the procedure will take into account in the next round. Recall that an application of rule ($\Box$) takes as inputs one $\Box$-prefixed node and all $\neg\Box$-prefixed nodes on the existing branch, and the maximal degree enjoyed by these input formulas cannot exceed the degree of that branch. By an easy observation of the rule ($\Box$) (and rule (MS)), we see that aforementioned newly introduced regular nodes are Boolean combinations of proper sub-formulas from the scope of the outermost $\Box$ of these input formulas. That is to say, aforementioned newly introduced branches each has a degree strictly smaller than that of the existing branch where the application is triggered. Since on the existing branch there are only finitely many $\Box$-prefixed nodes, fully exhausting a branch amounts to reducing it into finitely many branches with strictly smaller degrees. Note that, applications of propositional rules do not increase degrees of branches, and hence in summary, the procedure cannot go through the second **else if** infinitely many times. □

It may be helpful to emphasize that, the procedure we just offered may (meaninglessly) extend branches that are already closed. But doing so can waste only finitely many rounds (before exhausting everything in the occasion), and hence has no effect in terms of termination. Optimization will be welcomed, although satisfiability problem of INL is known to be PSPACE-complete [14].

4 Soundness of Tinl

This section addresses soundness of the tableau system Tinl. With the help of Hinl's soundness (Theorem 1), it is sufficient to show that Tinl-provability entails Hinl-provability.

Observe that in a Tinl-tableau, each end node *determines* an unique branch. A branch is said to be *inconsistent*, if $\vdash_{\mathsf{Hinl}} (\bigwedge S) \to \bot$ for S being the set of regular nodes (formulas) on that branch. A regular node (formula) ϕ is said to be *inconsistent*, if it holds that $\vdash_{\mathsf{Hinl}} \phi \to \bot$. Soundness of Tinl can be proved as a corollary of the following theorem.

Theorem 4. *In a* Tinl*-tableau, every closed initial node is inconsistent.*

Corollary 1 (Soundness of Tinl). *For every* INL*-formula ϕ, if $\vdash_{\mathsf{Tinl}} \phi$, then ϕ is valid.*

Proof. Assume that $\vdash_{\mathsf{Tinl}} \phi$, then there exists a closed Tinl-tableau with root $\neg\phi$. Since root $\neg\phi$ is initial, by Theorem 4, the root is inconsistent, and hence $\vdash_{\mathsf{Hinl}} \neg\phi \to \bot$, which implies $\vdash_{\mathsf{Hinl}} \phi$. By soundness of Hinl, we know that ϕ is valid. □

It remains to verify Theorem 4.

Proof (of Theorem 4). By an induction on the given Tinl-tableau.

For the base case. Observe that minimal initial nodes in a Tinl-tableau are such initial nodes that all branches from them terminate at end nodes with no irregular children. Assume that a minimal initial node is closed, then by definition of open/close-ness in Definition 7, we see that all branches from it terminate at closed end nodes with no child, and hence all these branches are closed. Since all these branches were extended from the initial node by only propositional rules, we can directly apply existing verifications for Tcpc's soundness, and conclude that the initial node itself is inconsistent.

For the induction step. Assume that an initial node is closed, by Definition 7, all branches from it terminate at closed end nodes that may have irregular children. We **claim** that: *If an end node is closed, then the branch it determines is* *inconsistent.* With this claim, we can conclude as in the base case that the initial node itself is inconsistent. It is therefore sufficient to verify the claim as follows.

If the closed end node has no child, then by Definition 7, the branch it determines is closed. Also by that definition, that branch has node $\bot$, node $\neg\top$, or nodes χ and $\neg\chi$ for some formula χ, and hence is inconsistent.

If the closed end node (denote it by d) has irregular children, then by definition, there exists a phase in which all its irregular children are closed. Note that an irregular node can only have regular children, and is closed only if at least one of these regular children are closed. Thus, there exists a phase in which each irregular child S of d has a closed regular child e (which is an element of S). Observe that, e cannot be a child of any regular node, and hence is initial, which by induction hypothesis implies the inconsistency of e. In summary, there exists a phase in which each irregular child of d has an inconsistent element.

By the design of Tinl, we know that irregular children of d in that phase are introduced by an application of rule ($\Box$), triggered by a $\Box$-prefixed node on the branch determined by d. Without loss of generality, assume that the $\Box$-prefixed node is formula $\Box(\alpha_1, ..., \alpha_j; \alpha_0)$, and there are k-many $\neg\Box$-prefixed nodes $\neg\Box(\beta^1_1, ..., \beta^1_{j_1}; \beta^1_0)$, ... , $\neg\Box(\beta^k_1, ..., \beta^k_{j_k}; \beta^k_0)$ on the branch determined by d. It is therefore sufficient to show inconsistency of formula set

$$D = \{\Box(\alpha_1, ..., \alpha_j; \alpha_0), \neg\Box(\beta^1_1, ..., \beta^1_{j_1}; \beta^1_0), ..., \neg\Box(\beta^k_1, ..., \beta^k_{j_k}; \beta^k_0)\}.$$

Observe the form of rule ($\Box$), we see that in the phase created by the above mentioned application, each irregular node introduced as a child of d is indexed by an $I \in \prod_{l=1}^{k}\{0, 1, ..., j_l\}$, while the irregular node with index I has regular children $\alpha_0 \wedge \sigma \wedge (\bigwedge_{i\in\{1,...,k\}}^{I(i)\neq 0} \neg\beta^i_{I(i)})$ for each $\sigma \in \{\alpha_1, ..., \alpha_j\} \cup \{\neg\beta^i_0 \mid i \in \{1, ..., k\}, I(i) = 0\}$. Since each irregular child of d has an inconsistent element, we know that for each $I \in \prod_{l=1}^{k}\{0, 1, ..., j_l\}$, at least one of the following holds:

$$\begin{cases} \text{there exists } x \in \{1, ..., j\} \text{ s.t. } \vdash_{\mathsf{Hinl}} \alpha_0 \wedge \alpha_x \wedge (\bigwedge\limits_{i\in\{1,...,k\}}^{I(i)\neq 0} \neg\beta^i_{I(i)}) \to \bot, \quad \text{or} \\ \text{there exists } y \in \{1, ..., k\} \text{ s.t. } I(y) = 0 \text{ and } \vdash_{\mathsf{Hinl}} \alpha_0 \wedge \neg\beta^y_0 \wedge (\bigwedge\limits_{i\in\{1,...,k\}}^{I(i)\neq 0} \neg\beta^i_{I(i)}) \to \bot. \end{cases}$$

Entailment from this to

$$\vdash_{\mathsf{Hinl}} \Box(\alpha_1, ..., \alpha_j; \alpha_0) \to \bigvee_{l\in\{1,...,k\}} \Box(\beta^l_1, ..., \beta^l_{j_l}; \beta^l_0)$$

(and hence to the desired inconsistency of D) was verified by the author in the proof of Theorem 2.6 in [15].[6] □

[6] That proof is devoted to establish soundness of $\mathsf{G3inl}$, a sequent calculus for INL. Due to the length of that proof and the limit of space here, we have to omit further details and refer to [15] for a full presentation.

5 Completeness of Tinl

In this section, we prove completeness of Tinl by extracting counter-models from failed attempts of Tinl-proofs.

Theorem 5 (Completeness of Tinl). $\vdash_{\mathsf{Tinl}} \phi$ *holds for every* INL*-formula* ϕ *that is valid.*

Proof. We prove the contrapositive of the theorem.

With the assumption that $\nvdash_{\mathsf{Tinl}} \phi$, the procedure presented in Theorem 3, when being performed on the Tinl-tableau with only one node $\neg\phi$, terminates with `failure`. That is to say, the result of the procedure is an exhausted open Tinl-tableau $\mathcal{T}$ with root $\neg\phi$.

We define an INL-model $\mathfrak{M} = (W, N, V)$ as follows:

- The point-set W is the collection of all open initial regular nodes in $\mathcal{T}$. For each open initial regular node d in tableau $\mathcal{T}$, by Definition 7, there must be a branch O_d that starts from d, travels through only open regular nodes, and ends at open end regular node e_d.
- Let $V(d)$, the propositional valuation on d, be the one that assigns exactly propositional atoms on branch O_d true.
- Let $N(d)$, the collection of d's neighborhoods, be exactly the collection of e_d's open irregular children in all phases. Each of these neighborhoods is an open irregular node, and by Definition 7, all of its elements (viz. its children) are open initial regular nodes, and hence are also points in W.[7]

It remains to show that $\mathfrak{M}$ satisfies $\neg\phi$ at the root, and hence ϕ cannot be valid.

Temporally in this proof, call a formula non-compound, if it is a propositional atom, a $\Box$-prefixed formula, or a negation of one of that. As in Tcpc, *if a model satisfies all non-compound formulas (regular nodes) in a branch, then it satisfies all formulas on that branch including the initial node.* We **claim** that:

> *for each open initial regular node d in $\mathcal{T}$, the generated sub-model $\mathfrak{M}_d$ satisfies d (as a formula) at d (as a point in $\mathfrak{M}_d$).*

With that claim, we know that $\mathfrak{M}$, as the generated sub-model of itself by the root, satisfies the root $\neg\phi$ at the root, and hence ϕ is not valid. Therefore, it is sufficient to verify the claim by an induction on points in $\mathfrak{M}$ as follows.

For the base case. If $d \in W$ and $N(d) = \varnothing$, then d is a minimal open initial regular node. Without loss of generality, assume among all possibilities, the one represented by branch O_d is taken in the construction of $\mathfrak{M}$. So O_d starts from d, travels through only open nodes, and ends at some open end node e_d that has

[7] This vacuously covers the special case that the irregular node is empty and hence has no elements.

no child.[8] By the design of $\mathfrak{M}$, we know that $V(d)$ assigns exactly propositional atoms in branch O_d true, and $\mathfrak{M}_d$ has d as its only point.

By the design of V, the collection of propositional atoms and their negations are satisfied by $\mathfrak{M}_d$ at d. Since O_d is fully exhausted but yet e_d has no irregular child, we know that O_d has no $\Box$-prefixed formulas. Since d has no neighborhood in $\mathfrak{M}$, all $\neg\Box$-prefixed formulas in O_d are automatically satisfied. In summary, $\mathfrak{M}_d$ satisfies all non-compound formulas in O_d at point d, and hence it also satisfies the initial node d at point d.

For the induction step. Assume $d \in W$ and $N(d) \neq \varnothing$. Without loss of generality, assume among all possibilities, the one represented by branch O_d is taken in the construction of $\mathfrak{M}$. So O_d starts from d, travels through only open nodes, and ends at some open end node e_d whose open irregular children in all phases are exactly neighborhoods of d. As in the base case, the design of V guarantees the satisfaction of all propositional atoms and their negations in O_d by $\mathfrak{M}_d$ at d.

For each $\Box$-prefixed formula $\Box(\alpha_1, ..., \alpha_j; \alpha_0)$ (abbreviated as α) in O_d, since O_d is fully exhausted, α has triggered an application of rule ($\Box$) that created a phase with irregular nodes. Since e_d is open, by Definition 7, in that phase there exists an open irregular child S. By the design of $\mathfrak{M}$, we see that $S \in N(d)$. Since S is open, so are all its elements (viz. all its regular children). Thus, as desired, elements of S are open initial regular nodes and hence are points in $\mathfrak{M}_d$. As a formula, each regular node $s \in S$ has a form of $\alpha_0 \wedge \sigma \wedge (\bigwedge_{i\in\{1,...,k\}}^{I(i)\neq 0} \neg\beta^i_{I(i)})$ where $\sigma \in \{\alpha_1, ..., \alpha_j\} \cup \{\neg\beta^i_0 \mid i \in \{1, ..., k\}, I(i) = 0\}$. Since s is an open initial regular node, by induction hypothesis, $\mathfrak{M}_s$ satisfies $\alpha_0 \wedge \sigma \wedge (\bigwedge_{i\in\{1,...,k\}}^{I(i)\neq 0} \neg\beta^i_{I(i)})$ at s, and hence $\mathfrak{M}_d$ satisfies α_0 at s. By arbitrariness of s, we know that α_0 universally holds in S. For each $x \in \{1, ..., j\}$, let s_x be the element of S whose index σ is α_x. By induction hypothesis, $\mathfrak{M}_{s_x}$ satisfies $\alpha_0 \wedge \alpha_x \wedge (\bigwedge_{i\in\{1,...,k\}}^{I(i)\neq 0} \neg\beta^i_{I(i)})$ at s_x, and hence $\mathfrak{M}_d$ satisfies α_x at s_x. By arbitrariness of x, we know that none of $\alpha_1, ..., \alpha_j$ universally fails in S. In summary, $\mathfrak{M}_d$ satisfies α at d.[9]

For each $\neg\Box$-prefixed formula μ in O_d, we need to show that $\mathfrak{M}_d$ satisfies μ on d. Let $S \in N(d)$ be an arbitrary neighborhood of d in $\mathfrak{M}$ (also in $\mathfrak{M}_d$). By design of $\mathfrak{M}$ and that of Tinl, neighborhood S is an open irregular child of e_d that is introduced by an application of rule ($\Box$) with one $\Box$-prefixed formula and

[8] Suppose that e_d has children. Since e_d is an end node, all children it has are irregular. Since $N(d) = \varnothing$, all these irregular children are closed. As e_d's irregular children must be introduced in phases by applications of rule ($\Box$), by Definition 7, e_d is closed, a contradiction.

[9] Note that our proof works in a vacuous way for the special case that $S = \varnothing$. In that case, the set $A \cup B^I$ in Definition 6 is be empty, and hence it must be the case that $j = 0$.

all $\neg\Box$-prefixed formulas in O_d (including μ) as inputs. Assume that in O_d there are k-many $\neg\Box$-prefixed formulas $\mu^1, ..., \mu^k$, where μ^y is $\neg\Box(\beta_1^y, ..., \beta_{j_y}^y; \beta_0^y)$ for each $y \in \{1, ..., k\}$. Without loss of generality, let μ be μ^k. Observe that as an irregular node introduced by an application of rule ($\Box$), S is indexed by some $I \in \prod_{l=1}^{k}\{0, 1, ..., j_l\}$. [Case (i)] $I(k)$ (the k-th element of sequence I) is 0, then $\neg\beta_0^k$ is in set "B^I" of Definition 6, hence there is $s_k \in S$ that (as a formula) is $\alpha_0 \wedge \neg\beta_0^k \wedge (\bigwedge_{i\in\{1,...,k\}}^{I(i)\neq 0} \neg\beta_{I(i)}^i)$. As a node, s_k is also a regular child of S, and since S is open, so is s_k. That is to say, s_k is an open initial regular node, and hence by induction hypothesis, $\mathfrak{M}_{s_k}$ satisfies $\alpha_0 \wedge \neg\beta_0^k \wedge (\bigwedge_{i\in\{1,...,k\}}^{I(i)\neq 0} \neg\beta_{I(i)}^i)$ at s_k. As a consequence, $\mathfrak{M}_d$ falsifies β_0^k at $s_k \in S$, and hence β_0^k is not universally true in S. [Case (ii)] $I(k) \neq 0$. As a formula, an arbitrary $s \in S$ should be $\alpha_0 \wedge \sigma \wedge (\bigwedge_{i\in\{1,...,k\}}^{I(i)\neq 0} \neg\beta_{I(i)}^i)$ for some formula σ. Since S is open, so is s, and hence s is an open initial regular node. By induction hypothesis, $\mathfrak{M}_s$ satisfies $\alpha_0 \wedge \sigma \wedge (\bigwedge_{i\in\{1,...,k\}}^{I(i)\neq 0} \neg\beta_{I(i)}^i)$ at s, and hence, $\mathfrak{M}_d$ falsifies $\beta_{I(k)}^k$ at s. By arbitrariness of s, we see that $\beta_{I(k)}^k$ universally fails in S. [Summary] In both Case (i) and Case (ii), S fails to be a neighborhood that justifies $\Box(\beta_1^k, ..., \beta_{j_k}^k; \beta_0^k)$. By arbitrariness of S, we see that point d has no neighborhoods that justify $\Box(\beta_1^k, ..., \beta_{j_k}^k; \beta_0^k)$, and hence $\mathfrak{M}_d$ satisfies μ at d.

We have shown that $\mathfrak{M}_d$ satisfies all non-compound formulas in branch O_d at d, and hence it also satisfies the initial node d at point d. This finishes the induction as well as our proof of the claim. □

In the proof above, we have actually given a method to extract counter-models from failed attempts of proof-search. By termination theorem (Theorem 3), a proof-search in Tinl always terminates. If it terminates with a success, then by definition we have a proof of the goal; otherwise it terminates with a failure, and employing the method from the proof of Theorem 5, we get a counter-model of the goal.

We finish this section with an illustration of counter-model construction.

Example 2. We perform for formula

$$\Box(; q) \wedge \Box(p \vee q; \neg\Box(; q)) \rightarrow \Box(p; \top)$$

(abbreviate it by η) a proof/counter-model search in Tinl. Start with a tableau with $\neg\eta$ as root and the only node, using propositional rules, we get the following tableau whose only branch is propositionally exhausted.

$$\begin{tikzcd}
\neg(\Box(;q) \wedge \Box(p \vee q; \neg\Box(;q)) \rightarrow \Box(p;\top)) \arrow[d] \\
\begin{array}{c} \Box(;q) \wedge \Box(p \vee q; \neg\Box(;q)) \\ \neg\Box(p;\top) \end{array} \arrow[d] \\
\begin{array}{c} \Box(;q) \\ \Box(p \vee q; \neg\Box(;q)) \end{array}
\end{tikzcd}$$

The only option to continue is to apply the rule ($\Box$), which by definition takes all $\neg\Box$-prefixed formulas (here $\neg\Box(p;\top)$ is the only one of that type) and one $\Box$-prefixed formula on the branch as inputs. There are two $\Box$-prefixed formulas, $\Box(;q)$ and $\Box(p \vee q; \neg\Box(;q))$, on the branch, so rule ($\Box$) can be applied twice, thereby create two phases. Since there is exactly one instance in the only $\neg\Box$-prefixed formula, we know that in each of these two phases two irregular children respectively indexed by (0) and (1) are introduced. By Definition 7, in order to close the end node as well as the tableau, it is sufficient to find one phase in which all irregular children introduced are closed.

Phase $\Box(;q)$. By the design of rule ($\Box$) (cf. Definition 6), irregular node with index (0) is $\{q \wedge \neg\top\}$, and that with index (1) is $\varnothing$. While the former can be closed, the latter cannot. By Definition 5, on irregular node $\varnothing$ rule (MS) is not applicable, so $\varnothing$ has no closed regular child and hence is open. Therefore, this phase does not close the tableau.

Phase $\Box(p \vee q; \neg\Box(;q))$. By the design of rule ($\Box$), irregular node with index (0) is $\{\neg\Box(;q) \wedge (p \vee q), \neg\Box(;q) \wedge \neg\top\}$, and that with index (1) is $\{\neg\Box(;q) \wedge (p \vee q) \wedge \neg p\}$. While the former can be closed (left to the reader), the latter cannot. Try all applicable to continue on the latter, we have:

$$\begin{tikzcd}
 & \{\neg\Box(;q) \wedge (p \vee q) \wedge \neg p\} \arrow[d] & \\
 & \neg\Box(;q) \wedge (p \vee q) \wedge \neg p \arrow[d] & \\
 & \begin{array}{c} \neg\Box(;q) \wedge (p \vee q) \\ \neg p \end{array} \arrow[d] & \\
 & \begin{array}{c} \neg\Box(;q) \\ p \vee q \end{array} \arrow[dl] \arrow[dr] & \\
p & & q
\end{tikzcd}$$

in which the right branch is fully exhausted[10] yet open. Therefore, this phase does not close the tableau either.

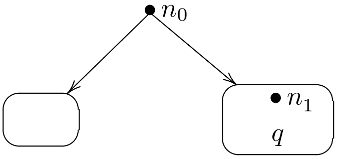
In conclusion, we result in an exhausted open tableau (denote it by $\mathcal{T}$) with root $\neg\eta$. We now construct a counter-model (W, N, V) of η following the instruction presented in the proof of Theorem 5.

Above applications of rule $(\Box)$, there is only one open initial regular node (i.e., the root) and only one branch in $\mathcal{T}$. Let us call the root n_0, and that branch O_0. Since there are no literals in O_0, we see that $V(n_0)$ should make no propositional atom true.

In phase $\Box(;q)$, the irregular node with index (0) is closed and that with index (1) is $\varnothing$ (hence open by definition). As no open initial regular node exists in this phase, no new point is added to the model. However, as $\varnothing$ is an open irregular child, we have $\varnothing \in N(n_0)$.

In phase $\Box(p \vee q; \neg\Box(;q))$, the irregular node with index (0) is closed and that with index (1) is open. Denote the irregular node with index (1) by S, we should have $S \in N(n_0)$. In $\mathcal{T}$, irregular node S has only one child (its only element), which is initial and open, and hence is a point in the model. Let us call that node n_1. Starting from n_1, there is an open branch that ends at end node q, and q is the only propositional atom on that branch. Therefore, $V(n_1)$ should make q the only true propositional atom.

In summary, the model constructed can be displayed as



and we leave it to the reader to check that the constructed model falsifies η at n_0.

6 Conclusions and Discussions

In this paper we offered to INL a sound-and-complete tableau system Tinl that supports mechanical proof/counter-model search.

While Tinl naturally connects to semantics of the logic, it also connects to other formal proof systems. It is well-known that tableau system and sequent calculus for classical propositional logic each can be seen as the dual of the other. Due to our employment of irregular nodes, the dual of Tinl is a hypersequent calculus. In the author's other parallel work [15], we offer to INL a sequent calculus G3inl that amounts to the result of breaking the only essentially-"hyper" rule scheme in the dual of Tinl into infinitely-many (normal) sequent rule schemes with sophisticated parameters. As a normal sequent calculus, G3inl admits cut, enjoys the sub-formula property, and has a splitting version that supports a construction of Lyndon interpolants [15].

[10] Recall here that unlike $\Box$-prefixed formulas, $\neg\Box$-prefixed formulas are not active.

One may say that INL is simple, as with primary formulations like normal sequent calculus we already had a nice treatment. As instanced by modal logics, compared to a calculus for a logic (e.g., K), that for an extension (e.g., S5) can be essentially harder. There are meaningful extensions of INL with neighborhood function N in models restricted, and for which adding rule schemes to G3inl may not directly work. When difficulties are encountered, one coping strategy is to consult formulations closer to semantics, like Tinl.

Acknowledgements. The tableau system introduced in this paper was invented by the author during his January 2015 visit to University of Amsterdam, where he was led by Johan van Benthem and Nick Bezhanishvili to the field of neighborhood logic. Four anonymous referees have offered helpful suggestions to the initial submission of this paper.

References

1. Beth, E.W.: Semantic Entailment and Formal Derivability. Mededelingen van de Koninklijke Nederlandske Akademie van Wetenschappen, vol. 18, no. 13, pp. 309–342 (1955)
2. ten Cate, B., Gabelaia, D., Sustretov, D.: Modal languages for topology: expressivity and definability. Ann. Pure Appl. Log. **159**(1), 146–170 (2006)
3. Chagrov, A., Zakharyashche, M.: Modal Logic. Oxford Science Publications, Oxford (1997)
4. Chellas, B.F.: Modal Logic - An Introduction. Cambridge University Press, Cambridge (1980)
5. D'Agostino, M.: Tableau methods for classical propositional logic. In: [6]
6. D'Agostino, M., Gabbay, D.M., Hähnle, R., Posegga, J. (eds.): Handbook of Tableau Methods. Kluwer Academic Publisher, Dordrecht (1999)
7. Fitting, M.: Proof Methods for Modal and Intuitionistic Logics. D. Reidel, Dordrecht (1983)
8. Fitting, M.: Introduction. In: [6]
9. Fitting, M.: Types, Tableaus, and Gödel's God. Kluwer Academic Publisher, Dordrecht (2002)
10. Hintikka, J.: Two papers on symbolic logic: form and content in quanlification thoery. Acta Philos. Fenn. **8**, 8–55 (1955)
11. Lis, Z.: Wynikanie semantyczne a wynikanie fonnalne. Stud. Log. **10**, 39–60 (1960). (in Polish)
12. Pacuit, E.: Neighborhood semantics for modal logic - An introduction. ESSLLI 2007 course notes (2007)
13. Smullyan, R.: First-Order Logic. Springer, Heidelberg (1968)
14. van Benthem, J., Bezhanishvili, N., Enqvist, S., Yu, J.: Instantial neighbourhood logic. Rev. Symb. Log. **10**(1), 116–144 (2017)
15. Yu, J.: Lyndon interpolation theorem of instantial neighborhood logic - constructively via a sequent calculus (submitted)

Interpretations of Presburger Arithmetic in Itself

Alexander Zapryagaev(✉) and Fedor Pakhomov

Steklov Mathematical Institute, Russian Academy of Sciences,
8, Gubkina Street, Moscow 119991, Russian Federation
rudetection@gmail.com

Abstract. Presburger arithmetic **PrA** is the true theory of natural numbers with addition. We study interpretations of **PrA** in itself. We prove that all one-dimensional self-interpretations are definably isomorphic to the identity self-interpretation. In order to prove the results we show that all linear orders that are interpretable in $(\mathbb{N}, +)$ are scattered orders with the finite Hausdorff rank and that the ranks are bounded in terms of the dimension of the respective interpretations. From our result about self-interpretations of **PrA** it follows that **PrA** isn't one-dimensionally interpretable in any of its finite subtheories. We note that the latter was conjectured by A. Visser.

Keywords: Presburger Arithmetic · Interpretations
Scattered linear orders

1 Introduction

Presburger Arithmetic **PrA** is the first-order theory of natural numbers with addition. It was introduced by Presburger in 1929 [13]. Presburger Arithmetic is complete, recursively-axiomatizable, and decidable.

The method of interpretations is a standard tool in model theory and in the study of decidability of first-order theories [8,12]. An interpretation of a theory **T** in a theory **U** essentially is a uniform first-order definition of models of **T** in models of **U** (we present a detailed definition in Sect. 3). In the paper we study certain questions about interpretability for Presburger Arithmetic that were well-studied in the case of stronger theories like Peano Arithmetic **PA**. Although, from technical point of view the study of interpretability for Presburger Arithmetic uses completely different methods than the study of interpretability for **PA** (see for example [18]), we show that from interpretation-theoretic point of view, **PrA** has certain similarities to strong theories that prove all the instances of mathematical induction in their own language, i.e. **PA**, Zermelo-Fraenkel set theory **ZF**, etc.

A *reflexive* arithmetical theory ([18, p. 13]) is a theory that can prove the consistency of all its finitely axiomatizable subtheories. Peano Arithmetic **PA**

This work is supported by the Russian Science Foundation under grant 16-11-10252.

S. Artemov and A. Nerode (Eds.): LFCS 2018, LNCS 10703, pp. 354–367, 2018.
https://doi.org/10.1007/978-3-319-72056-2_22

and Zermelo-Fraenkel set theory **ZF** are among well-known reflexive theories. In fact, all sequential theories (very general class of theories similar to **PA**, see [5, III.1(b)]) that prove all instances of induction scheme in their language are reflexive. For sequential theories reflexivity implies that the theory cannot be interpreted in any of its finite subtheories. A. Visser have conjectured that this purely interpretational-theoretic property holds for **PrA** as well. Note that **PrA** satisfies full-induction scheme in its own language but cannot formalize the statements about consistency of formal theories.

The conjecture was studied by Zoethout [19]. Note that Presburger Arithmetic, unlike sequential theories, cannot encode tuples of natural numbers by single natural numbers. And hence for interpretations in Presburger Arithmetic it is important whether individual objects are interpreted by individual objects (one-dimensional interpretations) or by tuples of objects of some fixed length m (m-dimensional interpretations). Zoethout considered only the case of one-dimensional interpretations and proved that if any one-dimensional interpretation of **PrA** in $(\mathbb{N}, +)$ gives a model that is definably isomorphic to $(\mathbb{N}, +)$ then Visser's conjecture holds for one-dimensional interpretations, i.e. there are no one-dimensional interpretations of **PrA** in its finite subtheories. In the present paper we show that the following theorem holds and thus prove Visser's conjecture for one-dimensional interpretations:

Theorem 1.1. *For any model $\mathfrak{A}$ of* **PrA** *that is one-dimensionally interpreted in the model $(\mathbb{N}, +)$, (a) $\mathfrak{A}$ is isomorphic to $(\mathbb{N}, +)$; (b) the isomorphism is definable in $(\mathbb{N}, +)$.*

Note that Theorem 1.1(a) was established by Zoethout in [19].

We also study whether the generalization of Theorem 1.1 to multi-dimensional interpretations holds. We prove:

Theorem 1.2. *For any m and model $\mathfrak{A}$ of* **PrA** *that is m-dimensionally interpreted in $(\mathbb{N}, +)$, the model $\mathfrak{A}$ is isomorphic to $(\mathbb{N}, +)$.*

We don't know whether the isomorphism is always definable in $(\mathbb{N}, +)$.

In order to prove Theorem 1.2, we show that for every m each linear order that is m-dimensionally interpretable in $(\mathbb{N}, +)$ is *scattered*, i.e. it doesn't contain a dense suborder. Moreover, our construction gives an estimation for Cantor-Bendixson ranks of the orders (a notion of Cantor-Bendixson rank for scattered linear orders goes back to Hausdorff [7] in order to give more precise estimation we use slightly different notion of VD_*-rank from [10]):

Theorem 1.3. *All linear orders m-dimensionally interpretable in $(\mathbb{N}, +)$ have the VD_*-rank at most m.*

Note that since every structure interpretable in $(\mathbb{N}, +)$ is automatic, the fact that both the VD_* and Hausdorff ranks of any scattered linear order interpretable in $(\mathbb{N}, +)$ is finite follows from the results on automatic linear orders by Khoussainov et al. [10].

The work is organized as follows. Section 2 introduces the basic notions. In Sect. 3 we give the definitions of non-parametric interpretations and definable isomorphism of interpretations. In Sect. 4 we define the dimension of Presburger sets and prove Theorem 1.3. In Sect. 5 we prove Theorem 1.1 and explain how it implies the impossibility to interpret **PrA** in its finite subtheories. In Sect. 6 we discuss the approach for the multi-dimensional case.

2 Presburger Arithmetic and Definable Sets

In the section we give some results about Presburger Arithmetic and definable sets in $(\mathbb{N}, +)$ from the literature that will be relevant for our paper.

Definition 2.1. Presburger Arithmetic (**PrA**) *is the elementary theory of the model* $(\mathbb{N}, +)$ *of natural numbers with addition.*

It is easy to see that every number $n \in \mathbb{N}$, the relations $<$ and $\leq$, modulo comparison relations $\equiv_n$, for natural $n \geq 1$, and the functions $x \longmapsto nx$ of multiplication by a natural number n are definable in the model $(\mathbb{N}, +)$. We fix some definitions for these constants, relations, and functions. This gives us a translation from the first-order language $\mathcal{L}$ of the signature $\langle =, \{n \mid n \in \mathbb{N}\}, +, < \;, \{\equiv_n \mid n \geq 1\}, \{x \longmapsto nx \mid n \in \mathbb{N}\}\rangle$ to the first-order language $\mathcal{L}^-$ of the signature $\langle =, +\rangle$. Since **PrA** is the elementary theory of $(\mathbb{N}, +)$, regardless of the choice of the definitions, the translation is uniquely determined up to **PrA**-provable equivalence. Thus we could freely switch between $\mathcal{L}$-formulas and equivalent $\mathcal{L}^-$-formulas. Note that **PrA** admits the quantifier elimination in the extended language $\mathcal{L}$ [13].

The well-known fact about order types of nonstandard models of PA also holds for models of Presburger arithmetic:

Theorem 2.1. *Any nonstandard model* $\mathfrak{A} \models$ **PrA** *has the order type* $\mathbb{N} + \mathbb{Z} \cdot A$, *where* $\langle A, <_A\rangle$ *is some dense linear order without endpoints. Thus, in particular, any countable model of* **PrA** *either has the order type* $\mathbb{N}$ *or* $\mathbb{N} + \mathbb{Z} \cdot \mathbb{Q}$.

For vectors $\overline{c}, \overline{p_1}, \ldots, \overline{p_n} \in \mathbb{Z}^m$ we call the set $\{\overline{c} + \sum k_i \overline{p_i} \mid k_i \in \mathbb{N}\}$ a *lattice* with the *generating* vectors $\overline{p_1}, \ldots, \overline{p_n}$ and the *initial* vector $\overline{c}$. If $\overline{p_1}, \ldots, \overline{p_n}$ are linearly independent ($n \leq m$) we call the set an *n-dimensional fundamental lattice.*

Ito [9] have proved that any union of finitely many (possibly, intersecting) lattices in $\mathbb{N}^m$ is a disjoint union of finitely many fundamental lattices. Ginsburg and Spanier [4, Theorem 1.3] have shown that the subsets of $\mathbb{N}^k$ definable in $(\mathbb{N}, +)$ are exactly the subsets of $\mathbb{N}^k$ that are unions of finitely many (possibly, intersecting) lattices; note that the sets from the latter class are known as *semilinear* sets. Combining these two results we obtain

Theorem 2.2. *All subsets of* $\mathbb{N}^k$ *definable in* $(\mathbb{N}, +)$ *are exactly the subsets of* $\mathbb{N}^k$ *that are disjoint unions of finitely many fundamental lattices.*

Let us now consider the extension of the first-order predicate language with an additional quantifier $\exists^{=y}x$, called a *counting quantifier* (notion introduced in [2]), used as follows: if $f(\overline{x}, z)$ is an $\mathcal{L}$-formula with the free variables $\overline{x}, z$, then $F = \exists^{=y}z\, G(\overline{x}, z)$ is also a formula with the free variables $\overline{x}, y$.

We extend the standard assignment of truth values to first-order formulas in the model $(\mathbb{N}, +)$ to formulas with counting quantifiers. For a formula $F(\overline{x}, y)$ of the form $\exists^{=y}z\, G(\overline{x}, z)$, a vector of natural numbers $\overline{a}$, and a natural number n we say that $F(\overline{a}, n)$ is true iff there are exactly n distinct natural numbers b such that $G(\overline{a}, b)$ is true. Apelt [1] and Schweikardt [15] have discovered that such an extension does not extend the expressive power of **PrA**:

Theorem 2.3 *([15, Corollary 5.10]). Every $\mathcal{L}$-formula $F(\overline{x})$ that uses counting quantifiers is equivalent in $(\mathbb{N}, +)$ to a quantifier-free $\mathcal{L}$-formula.*

3 Interpretations

Definition 3.1. *Suppose we have two first-order signatures Ω_1 and Ω_2. An m-dimensional* translation *ι of a first order language of the signature Ω_1 to the first-order language of the signature Ω_2 consists of*

1. *a first-order formula $Dom_\iota(\overline{y})$ of the signature Ω_2, where $\overline{x}$ is a vector of variables of the length m, with the intended meaning of the definition of the domain of translation;*
2. *first-order formulas $Pred_{\iota,P}(\overline{y}_1, \dots, \overline{y}_n)$ of the signature Ω_2, where each $\overline{y}_i$ is a vector of variables of the length m, for each predicate $P(x_1, \dots, x_n)$ from Ω_1 (including $x_1 = x_2$);*
3. *first-order formulas $Fun_{\iota,f}(\overline{y}_0, \overline{y}_1, \dots, \overline{y}_n,)$ of the signature Ω_2, where each $\overline{y}_i$ is a vector of variables of the length m, for each function $f(x_1, \dots, x_n)$ from Ω_1.*

Translation ι is an interpretation *of a model $\mathfrak{A}$ of the signature Ω_1 with the domain A in a model $\mathbf{B}$ of the signature Ω_2 with the domain B if*

1. *$Dom_\iota(\overline{y})$ defines a non-empty subset $D \subseteq B^m$;*
2. *$Pred_{\iota,=}(\overline{y}_1, \overline{y}_2)$ defines an equivalence relation $\sim$ on the set D;*
3. *there is a bijection $h\colon D/{\sim} \to A$ such that for each predicate $P(x_1, \dots, x_n)$ from Ω_1 and $\overline{b}_1, \dots, \overline{b}_n \in D$ we have*

$$\mathfrak{A} \models P(h([\overline{b}_1]_\sim), \dots, h([\overline{b}_n]_\sim)) \iff \mathfrak{B} \models Pred_{\iota,P}(\overline{b}_1, \dots, \overline{b}_n)$$

and for each function $f(x_1, \dots, x_n)$ from Ω_1 and $\overline{b}_0, \overline{b}_1, \dots, \overline{b}_n \in D$ we have

$$\mathfrak{A} \models h([\overline{b}_0]_\sim) = f(h([\overline{b}_1]_\sim), \dots, h([\overline{b}_n]_\sim)) \iff \mathfrak{B} \models Fun_{\iota,f}(\overline{b}_0, \overline{b}_1, \dots, \overline{b}_n).$$

Translation ι is an interpretation of a theory $\mathbf{T}$ of the signature Ω_1 in a model $\mathfrak{B}$ of the signature Ω_2 if it is an interpretation of some model of $\mathbf{T}$ in $\mathfrak{B}$. ι is an interpretation of a theory $\mathbf{T}$ of the signature Ω_1 in a theory $\mathbf{U}$ of the signature Ω_2 if it is an interpretation of $\mathbf{T}$ in every model $\mathfrak{B}$ of $\mathbf{U}$.

Translation ι is called non-relative *if the formula $Dom_\iota(\overline{y}) \equiv \top$, where $\overline{y}$ is $(y_1, \dots, y_m)$. We say that translation ι has* absolute equality *if the formula $Pred_{\iota,=}(\overline{y}, \overline{z})$ is $y_1 = z_1 \wedge \dots \wedge y_m = z_m$, where $\overline{y}$ is $(y_1, \dots, y_m)$ and $\overline{z}$ is $(z_1, \dots, z_m)$.*

Note that naturally for each translation ι of a signature Ω_1 to a signature Ω_2, we could define a map $F(x_1, \dots, x_n) \longmapsto F^\iota(\overline{y}_1, \dots, \overline{y}_m)$ from formulas of the signature Ω_1 to formulas of the signature Ω_2 such that if ι is an interpretation of a model $\mathfrak{A}$ in a model $\mathfrak{B}$ then for each $\overline{b}_1, \dots, \overline{b}_n \in D$ we have

$$\mathfrak{A} \models F(h([\overline{b}_1]_\sim), \dots, h([\overline{b}_n]_\sim)) \iff \mathfrak{B} \models F^\iota(\overline{b}_1, \dots, \overline{b}_n),$$

where m, D, and h are as in the definition above.

Also we note that if ι is an interpretation of a theory $\mathbf{T}$ in a model $\mathfrak{B}$ then there is a unique up to isomorphism model $\mathfrak{A}$ of $\mathbf{T}$ such that ι is an interpretation of $\mathfrak{B}$ in $\mathfrak{A}$.

Definition 3.2. *Suppose ι_1 and ι_2 are respectively an m_1-dimensional and m_2-dimensional translations from a signature Ω_1 to a signature Ω_2. And suppose that $I(\overline{y}, \overline{z})$ is a first-order formula of the signature Ω_2, where $\overline{y}$ consists of m_1 variables and $\overline{z}$ consists of m_2 variables.*

Now assume ι_1 and ι_2 are interpretations of the same model $\mathfrak{A}$ of the signature Ω_1 with the domain A in a model $\mathfrak{B}$ of the signature Ω_2 with the domain B. As in Definition 3.1 translations ι_1 and ι_2 give us respectively sets $D_1 \subseteq B^{m_1}$, $D_2 \subseteq B^{m_2}$ and equivalence relations $\sim_1$ on D_1 and $\sim_2$ on D_2. Under this assumption we say that $I(\overline{y}, \overline{z})$ is a definition of an isomorphism *of ι_1 and ι_2 if we could choose bijections $h_1 \colon D_1 \to A$ and $h_2 \colon D_2 \to A$ (satisfying properties of h from Definition 3.1, for respective ι_i) such that for each $\overline{b} \in D_1$ and $\overline{c} \in D_2$ we have*

$$h_1([\overline{b}]_{\sim_1}) = h_2([\overline{c}]_{\sim_1}) \iff \mathfrak{B} \models I(\overline{b}, \overline{c}).$$

If ι_1 and ι_2 are interpretations of the theory $\mathbf{T}$ in a theory $\mathbf{U}$ and for each model $\mathfrak{B}$ of $\mathbf{U}$ the formula $I(\overline{y}, \overline{z})$ is a definition of an isomorphism between ι_1 and ι_2 as interpretations in $\mathfrak{B}m$ then we say that $I(\overline{y}, \overline{z})$ is a definition of an isomorphism *between ι_1 and ι_2 as interpretations of $\mathbf{T}$ in $\mathbf{U}$.*

If ι_1 and ι_2 are interpretations of a theory $\mathbf{T}$ in a theory $\mathbf{U}$ (a model $\mathfrak{A}$) and there is a definition of an isomorphism then we say that ι_1 and ι_2 as interpretations of a theory $\mathbf{T}$ in a theory $\mathbf{U}$ (a model $\mathfrak{A}$) are definably isomorphic*.*

Since the theory $\mathbf{PrA}$ that we study is an elementary theory of some model ($\mathbf{PrA} = \mathbf{Th}(\mathbb{N}, +)$), actually there is not much difference between interpretations in the standard model and in the theory. A translation ι is an interpretation of some theory $\mathbf{T}$ in $\mathbf{PrA}$ iff ι is an interpretation of $\mathbf{T}$ in $(\mathbb{N}, +)$. A formula I is a definition of an isomorphism between interpretations ι_1 and ι_2 of some theory $\mathbf{T}$ in $\mathbf{PrA}$ iff I is a definition of an isomorphism between ι_1 and ι_2 as interpretations of $\mathbf{T}$ in $(\mathbb{N}, +)$.

4 Linear Orders Interpretable in $(\mathbb{N}, +)$

4.1 Functions Definable in Presburger Arithmetic

Definition 4.1. *Suppose $A \subseteq \mathbb{N}^n$ is a definable set. We call a function $f\colon A \to \mathbb{N}$* piecewise polynomial *of a degree $\leq m$ if there is a decomposition of A into finitely many fundamental lattices $C_1, \ldots, C_k$ such that the restriction of f on each C_i is a polynomial with rational coefficients of a degree $\leq m$*[1].

In particular, a *piecewise linear* function is a piecewise polynomial function of a degree ≤ 1.

Theorem 4.1. *All definable in $(\mathbb{N}, +)$ functions $f\colon \mathbb{N}^n \to \mathbb{N}$ are exactly piecewise linear.*

Proof. The definability of all piecewise linear functions in Presburger Arithmetic is obvious. A function $f\colon \mathbb{N}^n \to \mathbb{N}$ is definable iff its graph

$$G = \{(f(a_1, \ldots, a_n), a_1, \ldots, a_n) \mid (a_1, \ldots, a_n) \in \mathbb{N}^n\}$$

is definable. According to Theorem 2.2, G is a finite union of fundamental lattices $J_1 \sqcup \ldots \sqcup J_k$. For $1 \leq i \leq k$ we denote by J_i' the projections of J_i along the first coordinate, $J_i' = \{(a_1, \ldots, a_n) \mid \exists a_0((a_0, a_1, \ldots, a_n) \in J_i)\}$. Clearly, all J_i' are fundamental lattices. And the restriction of the function f on each of J_i' is linear.

Corollary 4.1. *All definable in $(\mathbb{N}, +)$ functions $f\colon \mathbb{N} \to \mathbb{N}$ can be bounded from above by a linear function with a rational slope. Conversely, if $h_1(x) < f(x) < h_2(x)$ for all x, where $h_1(x)$ and $h_2(x)$ are linear functions of the same irrational slope, then $f(x)$ is not definable.*

4.2 Dimension

Here we give the definition for the notion of dimension of Presburger-definable sets.

Definition 4.2. *The* dimension $\dim(A)$ *of a Presburger-definable set $A \subseteq \mathbb{N}^m$ is defined as follows.*

- $\dim(A) = 0$ *iff A is empty or finite;*
- $\dim(A) = k \geq 1$ *iff there is a definable bijection between A and $\mathbb{N}^k$.*

The following theorem shows that the definition indeed gives the unique dimension for each **PrA**-definable set.

Theorem 4.2. *Suppose M is an infinite Presburger definable subset of $\mathbb{N}^k$, $k \geq 1$. Then there is a unique natural number $l \in \mathbb{N}$ such that there is a Presburger definable bijection between M and $\mathbb{N}^l$, $1 \leq l \leq k$.*

[1] In our work, we use the word ‘piecewise’ only in the sense defined here.

Proof. First let us show that there is some l with the property. According to Theorem 2.2, all definable in $(\mathbb{N}, +)$ sets are disjoint unions of fundamental lattices $L_1, \dots, L_n$ of the dimensions $s_1, \dots, s_n$, respectively. It is easy to see that for each L_i there is a linear bijection with $\mathbb{N}^{s_i}$, which is obviously definable. Let us put l to be the maximum of s_i's. Now we just need to notice that for each sequence of natural number $r_1, \dots, r_m$ and $u = \max(r_1, \dots, r_m)$ if $u \geq 1$ then we could split a set $\mathbb{N}^u$ into sets $A_1, \dots, A_m$ for which we have definable bijections with $\mathbb{N}^{r_1}, \dots, \mathbb{N}^{r_m}$, respectively. We prove the latter by induction on m.

Now let us show that there is no other l with this property. Assume the contrary. Then clearly, for some $l_1 > l_2$ there is a mapping $f \colon \mathbb{N}^{l_1} \to \mathbb{N}^{l_2}$. Let us consider a sequence of expanding cubes, $I_n^{l_1} \overset{\text{def}}{=} \{(x_1, \dots, x_k) \mid 0 \leq x_1, \dots, x_k \leq n\}$. We define function $g \colon \mathbb{N} \to \mathbb{N}$ to be the function which maps a natural number n to the least m such that $f(I_n^{l_1}) \subseteq I_m^{l_2}$. Clearly, g is a Presburger-definable function. Then there should be some linear function $h \colon \mathbb{N} \to \mathbb{N}$ such that $g(n) \leq h(n)$, for all n. But since for each $n \in \mathbb{N}$ and $m < n^{l_1/l_2}$ the cube $I_n^{l_1}$ contains more points than the cube $I_m^{l_2}$, from the definition of g we see that $g(n) \geq n^{l_1/l_2}$. This contradicts the linearity of the function h. □

From the proof above we see that the following corollary holds:

Corollary 4.2. *The dimension of a set $M \subseteq \mathbb{N}^k$ is equal to the maximal l such that there exists an exactly l-dimensional fundamental lattice which is a subset of M.*

4.3 Presburger-Definable Linear Orders

Lemma 4.1. *Let $\overline{x} = (x_1, \dots, x_n)$ and $\overline{y} = (y_1, \dots, y_k)$ be vectors of free variables, where $\overline{y}$ will be treated as a vector of parameters. Let $F(\overline{x}, \overline{y})$ be an $\mathcal{L}^-$-formula such that for an infinite set of parameter vectors $B = \{\overline{b}_1, \overline{b}_2, \dots\}$ the sets defined by $F(\overline{x}, \overline{b}_i)$ are disjoint in $\mathbb{N}^n$. Then only a finite number of those definable sets can be exactly n-dimensional.*

Proof. Let us consider the set $A \subseteq \mathbb{N}^{n+k}$ defined by the formula $F(\overline{x}, \overline{y})$. For each vector $\overline{b} = (b_1, \dots, b_k) \in \mathbb{N}^k$ and set $S \subseteq \mathbb{N}^{n+k}$ we consider section $S \restriction \overline{b} = \{(a_1, \dots, a_n, b_1, \dots, b_k) \mid (a_1, \dots, a_n, b_1, \dots, b_k) \in S\}$. Clearly in this terms in order to prove the lemma, we need to show that there are only finitely many distinct $\overline{b} \in B$ such that the section $A \restriction \overline{b}$ is an n-dimensional set. By Theorem 2.2, the set A is a disjoint union of finitely many of fundamental lattices $J_i \subseteq \mathbb{N}^{n+k}$. It is easy to see that if some section $A \restriction \overline{b}$ were an n-dimensional set then at least for one J_i, the section $J_i \restriction \overline{b}$ were an n-dimensional set. Thus it is enough to show that for each J_i there are only finitely many vectors $\overline{b} \in B$ for which the section $J_i \restriction \overline{b}$ is an n-dimensional set.

Let us now assume for a contradiction that for some J_i there are infinitely many $J_i \restriction \overline{b}_0$, for $\overline{b}_0 \in B$, that are n-dimensional sets. Let us consider some parameter vector $\overline{b} \in \mathbb{N}^k$ such that the section $J \restriction \overline{b}$ is an n-dimensional set. Then by Corollary 4.2 there exists an n-dimensional fundamental lattice $K \subseteq J_i \restriction \overline{b}_0$. Suppose the generating vectors of K are $\overline{v}_1, \dots, \overline{v}_n$ and initial vector of K is

$\overline{u}$. It is easy to see that each vector $\overline{v}_j$ is a non-negative linear combination of generating vectors of J, since otherwise for large enough $h \in \mathbb{N}$ we would have $\overline{c}+h\overline{v}_j \notin J$. Now notice that for any $\overline{b} \in B$ and $\overline{a} \in J \restriction \overline{b}$ the n-dimensional lattice with generating vectors $\overline{v}_1, \ldots, \overline{v}_n$ and initial vector $\overline{a}$ is a subset of $\overline{a} \in J \restriction \overline{b}$.

Thus infinitely many of the sets defined by $F(\overline{x}, \overline{b})$, for $\overline{b} \in B$ contain the shifts of the same n-dimensional fundamental lattice. It is easy to see that the latter contradicts the assumption that all the sets are disjoint. □

Definition 4.3. *We call a linear ordering* $(L, <)$ scattered *if it does not have an infinite dense suborder.*

Definition 4.4. *Let* $(L, \prec)$ *be a linear ordering. We define a family of equivalence relations* $\simeq_\alpha$*, for ordinals* $\alpha \in \mathbf{Ord}$ *by transfinite recursion:*

- $\simeq_0$ *is just equality;*
- $\simeq_\lambda = \bigcup\limits_{\beta<\lambda} \simeq_\alpha$*, for limit ordinals* λ*;*
- $a \simeq_{\alpha+1} b \overset{\text{def}}{\iff} |\{c \in L \mid (a \prec c \prec b) \text{ or } (b \prec c \prec a)\}/{\simeq_\alpha}| < \aleph_0$.

Let us define VD_*-rank[2] $\text{rk}(L, \prec) \in \mathbf{Ord} \cup \{\infty\}$ *of the order* $(L, \prec)$*. The* VD_**-rank* $\text{rk}(L, \prec)$ *is the least* α *such that* $L/{\simeq_\alpha}$ *is finite. And if for all* $\alpha \in \mathbf{Ord}$ *the factor-set* $L/{\simeq_\alpha}$ *is infinite then we put* $\text{rk}(L, \prec) = \infty$*.*

By definition we put $\alpha < \infty$*, for all* $\alpha \in \mathbf{Ord}$*.*

Remark 4.1. Linear orders $(L, \prec)$ such that $\text{rk}(L, \prec) < \infty$ are exactly the scattered linear orders.

Example 4.1. The orders with the VD_*-rank equal to 0 are exactly finite orders, and the orders with VD_*-rank ≤ 1 are exactly the order sums of finitely many copies of $\mathbb{N}$, $-\mathbb{N}$ and 1 (one element linear order).

Theorem 4.3 (Restatement of Theorem 1.3). *For every natural* $m \geq 1$*, linear orders which are* m*-dimensionally interpretable in* $(\mathbb{N}, +)$ *have* VD_**-rank* m *or below.*

Proof. We prove the theorem by induction on m.

Suppose we have an m-dimensional interpretation of a linear order $(L, \prec)$ in $(\mathbb{N}, +)$, i.e. there is an $\mathcal{L}^-$ formula $D(\overline{x})$ giving the domain of the interpretation and $\mathcal{L}^-$ formula $\prec_*$ $(\overline{x}, \overline{y})$ giving interpretation of the order relation, where both $\overline{x}$ and $\overline{y}$ consist of m variables. Without loss of generality we may assume that $L = \{\overline{a} \in \mathbb{N}^m \mid (\mathbb{N}, +) \models D(\overline{a})\}$ and $\prec$ is defined by the formula $\prec_*$.

Now assume for a contradiction that $\text{rk}(L, \prec) > m$. By the definition of VD_*-rank, there are infinitely many distinct $\simeq_m$-equivalence classes in L. Hence there is an infinite chain $\overline{a}_0 \prec \overline{a}_1 \prec \ldots$ of elements of L such that $\overline{a}_i \not\simeq_m \overline{a}_{i+1}$, for each i. Let us consider intervals $L_i = \{\overline{b} \in L \mid \overline{a}_i < \overline{b} < \overline{a}_{i+1}\}$. Since $\overline{a}_i \not\simeq_m \overline{a}_{i+1}$, the set $L_i/{\simeq_{m-1}}$ is infinite and $\text{rk}(L_i, \prec) > m - 1$.

[2] VD stand for *very discrete*; see [14, pp. 84–89].

Clearly, all L_i are Presburger definable sets. Let us show that $\dim(L_i) \geq m$, for each i. If $m = 1$ then it follows from the fact that L_i is infinite. If $m > 1$ then we assume for a contradiction that $\dim(L_i) < m$. And notice that in this case $(L_i, \prec)$ would be $m-1$-dimensionally interpretable in $(N, +)$ which contradict induction hypothesis and the fact that $\mathrm{rk}(L_i, \prec) > m - 1$. Since $L_i \subseteq \mathbb{N}^m$, we conclude that $\dim(L_i) = m$, for all i.

Now consider the parametric family of subsets of $\mathbb{N}^m$ given by the formula $\overline{y}_1 \prec_* \overline{x} \prec_* \overline{y}_2$, where we treat variables $\overline{y}_1$ and $\overline{y}_2$ as parameters. We consider sets given by pairs of parameters $\overline{y}_1 = \overline{a}_i$ and $\overline{y}_2 = \overline{a}_{i+1}$, for $i \in \mathbb{N}$. Clearly the sets are exactly L_i's. Thus we have infinitely many disjoint sets of the dimension m in the family and hence we have contradiction with Lemma 4.1.

Remark 4.2. Each scattered linear order of VD_*-rank 1 is 1-dimensionally interpretable in $(\mathbb{N}, +)$. There are scattered linear orders of VD_*-rank 2 that are not interpretable in $(\mathbb{N}, +)$.

Proof. The interpretability of linear orders with rank 0 and rank 1 follows from Example 4.1.

Since there are uncountably many non-isomorphic scattered linear orders of VD_*-rank 2 and only countably many linear orders interpretable in $(\mathbb{N}, +)$, there is some scattered linear order of VD_*-rank 2 that is not interpretable in $(\mathbb{N}, +)$. □

5 One-Dimensional Self-interpretations and Visser's Conjecture

The following theorem is a generalization of [19, pp. 27–28, Lemmas 3.2.2–3.2.3].

Theorem 5.1. *Let* $\mathbf{U}$ *be a theory and* ι *be an* m*-dimensional interpretation of* $\mathbf{U}$ *in* $(\mathbb{N}, +)$*. Then for some* $m' \leq m$ *there is an* m'*-dimensional non-relative interpretation with absolute equality* κ *of* $\mathbf{U}$ *in* $(\mathbb{N}, +)$ *which is definably isomorphic to* ι*.*

Proof. First let us find κ with absolute equality. Indeed there is a definable in $(\mathbb{N}, +)$ well-ordering $\prec$ of $\mathbb{N}^m$:

$$(a_0, \ldots, a_{m-1}) \prec (b_0, \ldots, b_{m-1}) \overset{\text{def}}{\iff} \exists i < m (\forall j < i\ (a_j = b_j) \wedge a_i < b_i).$$

Now we could define κ by taking the definition of $+$ from ι, taking the trivial interpretation of equality, and taking the domain of interpretation to be the part of the domain of ι that consists of the $\prec$-least elements of equivalence classes with respect to ι-interpretation of equality. It is easy to see that this κ is definably isomorphic to ι.

Now assume that we already have ι with absolute equality. We find the desired non-relative interpretation κ by using Theorem 4.2 and bijectively mapping the domain of ι to $\mathbb{N}^{m'}$, where m' is the dimension of the domain of the interpretation ι. □

Combining Theorems 2.1 and 4.3, we obtain

Theorem 5.2 (Restatement of Theorem 1.1). *For any model $\mathfrak{A}$ of* **PrA** *that is one-dimensionally interpreted in the model* $(\mathbb{N}, +)$*, (a) $\mathfrak{A}$ is isomorphic to* $(\mathbb{N}, +)$*; (b) the isomorphism is definable in* $(\mathbb{N}, +)$*.*

Proof. Let us denote by $<_*$ the order relation given by the **PrA** definition of $<$ within $\mathfrak{A}$. Clearly $<_*$ is definable in $(\mathbb{N}, +)$. Thus we have an interpretation of the order type of $\mathfrak{A}$ in **PrA**. Hence by Theorem 4.3 the order type of $\mathfrak{A}$ is scattered. But from Theorem 2.1 we know that the only case when the order type of a model of **PrA** is scattered is the case when it is exactly $\mathbb{N}$. Thus $\mathfrak{A}$ is isomorphic to $(\mathbb{N}, +)$. From Theorem 5.1 it follows that it is enough to show the definability of the isomorphism only in the case when the interpretation that gives us $\mathfrak{A}$ is a non-relative interpretation with absolute equality.

It is easy to see that, the isomorphism f from $\mathfrak{A}$ to $(\mathbb{N}, +)$ is the function $f : x \longmapsto |\{y \in \mathbb{N} \mid y <_* x\}|$. Now we use counting quantifier to express the function:

$$f(a) = b \iff (\mathbb{N}, +) \models \exists^{=b} z \; (z <_* a) \qquad (1)$$

Now apply Theorem 2.3 and see that f is definable in $(\mathbb{N}, +)$.

Theorem 5.3. *Theory* **PrA** *is not one-dimensionally interpretable in any of its finitely axiomatizable subtheories.*

Proof. Assume ι is an one-dimensional interpretation of **PrA** in some finitely axiomatizable subtheory T of **PrA**. In the standard model $(\mathbb{N}, +)$ the interpretation ι will give us a model $\mathfrak{A}$ for which there is a definable isomorphism f with $(\mathbb{N}, +)$. Now let us consider theory T$'$ that consists of T and the statement that the definition of f gives an isomorphism between (internal) natural numbers and the structure given by ι. Clearly T$'$ is finitely axiomatizable and true in $(\mathbb{N}, +)$, and hence is subtheory of **PrA**. But now note that T$'$ proves that if something was true in the internal structure given by ι, it is true. And since T$'$ proved any axiom of **PrA** in the internal structure given by ι, the theory T$'$ proves every axiom of **PrA**. Thus T$'$ coincides with **PrA**. But it is known that **PrA** is not finitely axiomatizable, contradiction.

6 Multi-dimensional Self-interpretations

We already know that the only linear orders that it is possible to interpret in $(\mathbb{N}, +)$ (even by multi-dimensional interpretations) are scattered linear orders. And we could use this to prove the analogue of Theorem 1.1(a) for multi-dimensional interpretations by the same reasoning as we have used for Theorem 1.1(a).

However, the only way any interpretation can be isomorphic to trivial in a multi-dimensional case is by having a one-dimensional set as its domain and from Theorem 1.1 it follows that all interpretations of **PrA** in $(\mathbb{N}, +)$ that have one-dimensional domain are definably isomorphic to $(\mathbb{N}, +)$. Thus in order to prove the analogue of Theorem 1.1(b) for multi-dimensional interpretations one should in fact show that the domain of any interpretation of **PrA** in $(\mathbb{N},+)$ should be one-dimensional set.

In the section we will give some partial results about multi-dimensional self-interpretations of **PrA**.

Cantor polynomials are quadratic polynomials that define a bijection between $\mathbb{N}^2$ and $\mathbb{N}$:

$$C_1(x,y) = C_2(y,x) = \frac{1}{2}(x+y)^2 + \frac{1}{2}(x+3y). \tag{2}$$

The bijections C_1 and C_2 are the isomorphism of $(\mathbb{N}^2, \prec_1)$ and $(\mathbb{N}, <)$ and the isomorphism of $(\mathbb{N}^2, \prec_2)$ and $(\mathbb{N}, <)$, where

$$(a_1,a_2) \prec_1 (b_1,b_2) \overset{\text{def}}{\iff} (a_2 < b_2 \wedge a_1 + a_2 = b_1 + b_2) \vee (a_1 + a_2 < b_1 + b_2),$$
$$(a_1,a_2) \prec_2 (b_1,b_2) \overset{\text{def}}{\iff} (a_2 > b_2 \wedge a_1 + a_2 = b_1 + b_2) \vee (a_1 + a_2 < b_1 + b_2).$$

Note that both $\prec_1$ and $\prec_2$ are definable in $(\mathbb{N},+)$. The following theorem show that this interpretations of $(\mathbb{N},<)$ could not be extended to interpretations of $(\mathbb{N}, x \mapsto sx)$, for some s and thus shows that this interpretations could not be extended to interpretations of $(\mathbb{N},+)$.

Theorem 6.1. *Let s be a natural number that is not a square and i be either* 1 *or* 2. *Let us denote by $f\colon \mathbb{N}^2 \to \mathbb{N}^2$ the function $f(\overline{a}) = C_i^{-1}(s \cdot C_i(\overline{a}))$, i.e. the preimage of the function $x \mapsto s \cdot x$ under the bijection $C_i\colon \mathbb{N}^2 \to \mathbb{N}$. Then the function f is not definable in* $(\mathbb{N},+)$.

Proof. Since the cases of $i = 1$ and $i = 2$ are essentially the same, let us consider just the case of $i = 1$. Suppose the contrary: there is an $\mathcal{L}^-$-formula $F(x_1,x_2,y_1,y_2)$ which defines the graph of f:

$$(\mathbb{N},+) \models F(a_1,a_2,b_1,b_2) \iff f(a_1,a_2) = (b_1,b_2), \text{for all } a_1,a_2,b_1,b_2 \in \mathbb{N}.$$

Then the following function $h(x) : \mathbb{N} \to \mathbb{N}$ is also definable:

$$h(a) = b \overset{\text{def}}{\iff} \exists c,d(f(a,0) = (c,d) \wedge b = c + d). \tag{3}$$

Now it is easy to see that the following inequalities holds for all $a \in \mathbb{N}$:

$$\begin{gathered} C_1(h(a),0) \leq s \cdot C_1(a,0) < C_1(h(a)+1,0) \\ \Rightarrow \frac{h(a)(h(a)+1)}{2} \leq \frac{sa(a+1)}{2} < \frac{(h(a)+1)(h(a)+2)}{2} \\ \Rightarrow y^2 < S(x+1)^2 \text{ and } Sx^{<}(y+2)^2 \\ \Rightarrow \sqrt{S}x - 2 < y < \sqrt{S}x + \sqrt{S}. \end{gathered}$$

We conclude that a Presburger-definable function $h(x)$ is bounded both from above and below with linear functions of the same irrational slope. Contradiction with Corollary 4.1. □

We conjecture the following general fact holds:

Conjecture 6.1. *For any (multi-dimensional) interpretation ι of* **PrA** *in the model* $(\mathbb{N},+)$ *there is a definable isomorphism with the trivial interpretation of* $(\mathbb{N},+)$ *in* $(\mathbb{N},+)$.

The following theorem is a slight modification of the theorem by Blakley [3].

Theorem 6.2. *Let A be a $d\times n$ matrix of integer numbers, function $\varphi_A\colon \mathbb{Z}^d \to \mathbb{N}\cup\{\aleph_0\}$ is defined as follows:*

$$\varphi_A(u) \overset{\text{def}}{=} |\{\overline{\lambda}=(\lambda_1,\ldots,\lambda_n)\in\mathbb{N}^n \mid A\lambda = u\}|.$$

Then if the values of φ_A are always finite, the function φ_A is a piecewise polynomial function of a degree $\le n-\mathrm{rk}(A)$.

Proof. The existence of the fundamental lattices $C_1,\ldots,C_l$ on which φ_A is polynomial follows from [17, p. 302]. Now we prove that the $n-\mathrm{rk}(A)$ bound on the degree holds.

Let us consider any fundamental lattice L with the initial vector $\overline{v}$ and generating vectors $\overline{s}_1,\ldots,\overline{s}_m$ such that the restriction of φ_A to L is a polynomial. Now it is easy to see that we could find a polynomial $P(x_1,\ldots,x_m)$ such that $\varphi_A(\overline{v}+\eta_1\overline{s}_1+\ldots+\eta_m\overline{s}_m)=P(\eta_1,\ldots,\eta_m)$, for all $\eta_1,\ldots,\eta_m\in\mathbb{N}$. Since the choice of L was arbitrary, we could finish the proof of the theorem by showing that P is of the degree $\le n-\mathrm{rk}(A)$. Let us assume for a contradiction that the degree of P is $> n-\mathrm{rk}(A)$. Clearly, then there are $\theta_1,\ldots,\theta_m\in\mathbb{N}$ such that the polynomial $Q(y)=P(\theta_1 y,\ldots,\theta_m y)$ is of the degree $k>n-\mathrm{rk}(A)$. Now we consider the vector $\overline{d}=\eta_1\overline{s}_1+\ldots+\eta_m\overline{s}_m$ and the vectors $\overline{e}_l=\overline{v}+l\overline{d}$, for $l\in\mathbb{N}$. We have $\varphi_A(\overline{e}_l)=Q(l)$.

Let us now estimate the values of $\varphi_A(\overline{e}_l)$. The value $\varphi_A(\overline{e}_l)$ is the number of integer points in the polyhedron $H_l=\{(\lambda_1,\ldots,\lambda_n)=\overline{\lambda}\in\mathbb{R}^n \mid A\overline{\lambda}=\overline{e}_l \text{ and } \lambda_1,\ldots,\lambda_n\ge 0\}$. And now it is easy to see that $\varphi_A(\overline{e}_l)\le h_l/o$, where o is the volume of $(n-\mathrm{rk}(A))$-dimensional sphere of the radius $1/2$ and h_l is the $(n-\mathrm{rk}(A))$-dimensional volume of (at most) $(n-\mathrm{rk}(A))$-dimensional polyhedron $H'_l=\{(\lambda_1,\ldots,\lambda_n)=\overline{\lambda}\in\mathbb{R}^n \mid A\overline{\lambda}=\overline{e}_l \text{ and } \lambda_1,\ldots,\lambda_n\ge -1\}$. Now we just need to notice that the linear dimensions of the polyhedra H'_l are bounded by a linear function of l and hence the volumes h_l are bounded by some polynomial of the degree $n-\mathrm{rk}(A)$, contradiction with the fact that the polynomial $Q(y)$ were of the degree $k>n-\mathrm{rk}(A)$. □

Recall that a semilinear set is a finite union of lattices and that by result of [9] any semilinear set is a disjoint union of fundamental lattices. It is easy to see that the following lemma holds:

Lemma 6.1. *1. If $f,g\colon A\to\mathbb{Z}$ are piecewise polynomial functions of a degree $\le m$ then the function $h\colon A\to\mathbb{Z}$, $h(\overline{v})=f(\overline{v})+g(\overline{v})$, is a piecewise polynomial function of a degree $\le m$;*

2. *if $A \subseteq \mathbb{Z}^n$ is a semilinear set, $f\colon A \to \mathbb{Z}$ is a piecewise polynomial function of a degree $\leq m$, and $B \subseteq A$ is* **PrA***-definable set then the restriction of f to B is a piecewise polynomial function of a degree $\leq m$;*
3. *if $A \subseteq \mathbb{Z}^n$ is a semilinear set, $f\colon A \to \mathbb{Z}$ is a piecewise polynomial function of a degree $\leq m$, and $F\colon \mathbb{Z}^n \to \mathbb{Z}^k$ is a linear operator, then the function $h\colon F(A) \to \mathbb{Z}^k$ is a piecewise polynomial function of a degree $\leq m$.*

We prove the lemma that generalizes the one-dimensional construction of the cardinality of sections.

Lemma 6.2. *Let $S \subseteq \mathbb{N}^{n+m}$ be a definable set in $(\mathbb{N}, +)$. For each vector $\overline{b} = (b_1, \dots, b_m) \in \mathbb{N}^m$ we define section $A \restriction \overline{b}$ to be the set of all elements of S of the form $(a_1, \dots, a_n, b_1, \dots, b_m)$. Suppose all sets $S \restriction \overline{b}$ are finite. For each vector $\overline{a} \in \mathbb{N}^n$. Consider the section cardinality function $f_S\colon \mathbb{N}^m \to \mathbb{N}$, $f_S\colon \overline{a} \mapsto |S \restriction \overline{b}|$. Then f_S is a piecewise polynomial function of a degree $\leq n$.*

Proof. Let us first prove the theorem for the case when S is a fundamental lattice with the initial vector $\overline{c}$ and the generating vectors $\overline{v}_1, \dots, \overline{v}_s \in \mathbb{N}^{n+m}$. We consider the vectors $\overline{c}', \overline{v}'_1, \dots, \overline{v}'_s \in \mathbb{N}^m$ that consist of the last m components of vectors $\overline{c}, \overline{v}_1, \dots, \overline{v}_s$, respectively. Clearly, for each $\overline{b} \in \mathbb{N}^m$, the value $f_S(\overline{b}) = |A \restriction \overline{b}|$ is equal to the number of different $\overline{\lambda} = (\lambda_1, \dots, \lambda_s) \in \mathbb{N}^s$ such that $\lambda_1 \overline{v}'_1 + \dots + \lambda_s \overline{v}'_s = \overline{b} - \overline{c}'$. Now we compose a matrix A from the vectors $\overline{v}'_1, \dots, \overline{v}'_s$ and see that $f_S(\overline{b}) = |\{\overline{\lambda} \in \mathbb{N}^m \mid A\overline{\lambda} = \overline{b} - \overline{c}'\}| = \varphi_A(\overline{b} - \overline{c})$. Note that since S was a fundamental lattice, $s - \mathrm{rk}(A) \leq n$. Now we apply Theorem 6.2 and see that φ_A is a piecewise polynomial of a degree $\leq n$. Now from Lemmas 6.1(2) and 6.1(3) it follows that f is piecewise polynomial of a degree $\leq n$ too.

In the case of arbitrary definable A, we apply Theorem 2.2 and find fundamental lattices $J_1, \dots, J_s$ such that $A = J_1 \sqcup J_2 \sqcup \dots \sqcup J_s$. Now we see that for each $\overline{b} \in \mathbb{N}^m$, we have $f_A(a) = f_{J_1}(a) + \dots + f_{J_s}(a)$ and since we already know that all f_{J_i} are piecewise polynomial of a degree $\leq n$, by Lemma 6.1(1) the function f_A is piecewise polynomial of a degree $\leq n$. □

Theorem 6.3. *Suppose a definable in $(\mathbb{N}, +)$ binary relation $\prec$ on $\mathbb{N}^n$ has the order type $\mathbb{N}$. Then the order isomorphism between $(\mathbb{N}^m, \prec)$ and $(\mathbb{N}, <)$ is a piecewise polynomial function of a degree $\leq n$.*

Proof. We see that the order isomorphism is the function $f\colon \mathbb{N}^m \to \mathbb{N}$ given by

$$f(a_1, \dots, a_n) = |\{(b_1, \dots, b_n, a_1, \dots, a_n) \mid (b_1, \dots, b_n)\ R\ (a_1, \dots, a_n)\}|.$$

By Lemma 6.2 we see that f is a piecewise polynomial function. □

Fueter-Pólya theorem [6,11] states that every quadratic polynomial that maps $\mathbb{N}^2$ onto $\mathbb{N}$ is one of two Cantor polynomials (2). If one would want to prove Conjecture 6.1 one of the possible approaches would be to give a classification of all piecewise polynomial bijections and then use the classification and a generalization of Theorem 6.1 in order to show that no two-dimensional nonrelative interpretation of $(\mathbb{N}, <)$ in $(\mathbb{N}, +)$ could be extended to an interpretation of $(\mathbb{N}, +)$.

Acknowledgments. The authors wish to thank Lev Beklemishev for suggesting to study Visser's conjecture, number of discussions of the subject, and his useful comments on the paper.

References

1. Apelt, H.: Axiomatische Untersuchungen über einige mit der Presburgerschen Arithmetik verwandte Systeme. MLQ Math. Log. Q. **12**(1), 131–168 (1966)
2. Barrington, D., Immerman, N., Straubing, H.: On uniformity within NC1. J. Comput. System Sci. **41**(3), 274–306 (1990)
3. Blakley, G.R.: Combinatorial remarks on partitions of a multipartite number. Duke Math. J. **31**(2), 335–340 (1964)
4. Ginsburg, S., Spanier, E.: Semigroups, Presburger formulas, and languages. Pacific J. Math. **16**(2), 285–296 (1966)
5. Hájek, P., Pudlák, P.: Metamathematics of First-Order Arithmatic. Springer, New York (1993)
6. Fueter, R., Pólya, G.: Rationale Abzhlung der Gitterpunkte, Vierteljschr. Naturforsch. Ges. Zrich **58**, 280–386 (1923)
7. Hausdorff, F.: Grundzüge einer Theorie der geordneten Mengen. Math. Ann. **65**(4), 435–505 (1908)
8. Hodges, W.: Model Theory, vol. 42. Cambridge University Press, Cambridge (1993)
9. Ito, R.: Every semilinear set is a finite union of disjoint linear sets. J. Comput. Syst. Sci. **3**(2), 221–231 (1969)
10. Khoussainov, B., Rubin, S., Stephan, F.: Automatic linear orders and trees. ACM Trans. Comput. Log. **6**(4), 675–700 (2005)
11. Nathanson, M.B.: Cantor polynomials and the Fueter-Pólya theorem. Am. Math. Monthly **123**(10), 1001–1012 (2016)
12. Tarski, A., Mostowski, A., Robinson, R.M.: Undecidable Theories. Studies in Logic and the Foundations of Mathematics. North-Holland, Amsterdam (1953)
13. Presburger, M.: Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt. Comptes Rendus du I congrès de Mathématiciens des Pays Slaves 92101 (1929). English translation in [16]
14. Rosenstein, J.: Linear Orderings, vol. 98. Academic Press, New York (1982)
15. Schweikardt, N.: Arithmetic, first-order logic, and counting quantifiers. ACM Trans. Comput. Log. **6**(3), 634–671 (2005)
16. Stansifer, R.: Presburger's Article on Integer Arithmetic: Remarks and Translation (Technical report). Cornell University (1984)
17. Sturmfels, B.: On vector partition functions. J. Combin. Theory Ser. A **72**(2), 302–309 (1995)
18. Visser, A.: An overview of interpretability logic. In: Kracht, M., de Rijke, M., Wansing, H., Zakharyaschev, M. (eds.) Advances in Modal Logic. CSLI Lecture Notes, vol. 87, pp. 307–359 (1998)
19. Zoethout, J.: Interpretations in Presburger Arithmetic. BS thesis. Utrecht University (2015)

Author Index